SIEMENS

SIMATIC

BRAUMAT/SISTAR Process Control System V7.5

Function Manual

Valid as of V7.5

05/2018

A5E44319256-AA

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E44319256-AA
℗ 05/2018 Subject to change

Preface

Purpose of the Manual

This manual describes the basic properties of the system.

This manual is intended for those responsible for configuring, commissioning, and servicing automation systems.

Where is this Manual valid?

This manual is valid for the **BRAUMAT/SISTAR V7.5** process control system.

The offered electronic manual is most largely identical with the contents of the on-line help. Due to a technically necessary editorial deadline for the generation of electronic manuals occasionally smaller deviations can give up opposite the on-line helps. The statements in the on-line helps are primary to those of the manual.

Note

Complementary to the BRAUMAT control system for control and monitoring of brewing processes, the scope of delivery starting from Version V7.5 additionally contains an industry-neutral variant of the control system named SISTAR. By running the corresponding setup program from the data storage medium, SISTAR can be installed alternatively to BRAUMAT.

This system documentation is generally valid for both systems. In this regard, the product will be referred to generally as „BRAUMAT/SISTAR“. Isolated uses of the individual terms BRAUMAT or SISTAR also describe both systems.

Further Support

If you have any technical questions, please get in touch with your Siemens representative or agent responsible.

You will find your contact person at:

http://www.siemens.com/automation/partner (http://www.siemens.com/automation/partner)

You will find a guide to the technical documentation offered for the individual SIMATIC products and systems at: http://www.siemens.com/simatic-tech-doku-portal (http://www.siemens.com/simatic-tech-doku-portal)

The online catalog and order system is found under:

http://mall.industry.siemens.com/ (http://mall.industry.siemens.com/)

Training Centers

Siemens offers a number of training courses to familiarize you with the SIMATIC S7 automation system. Please contact your regional training center or our central training center in D 90327 Nuremberg, Germany for details:

Internet: http://www.sitrain.com (http://www.sitrain.com)

Technical Support

You can reach the Technical Support for all Industry Automation and Drive Technology products via the Web formula for the Support Request

http://www.siemens.com/automation/support-request (http://www.siemens.com/automation/support-request)

Additional information about our Technical Support can be found on the Internet pages

http://www.siemens.com/automation/service (http://www.siemens.de/automation/service)

Service & Support on the Internet

In addition to our documentation, we offer our Know-how online on the internet at:

http://www.siemens.com/automation/service&support (http://www.siemens.com/automation/service&support)

where you will find the following:

- The newsletter, which constantly provides you with up-to-date information on your products.
- The right documents via our Search function in Service & Support.
- A forum, where users and experts from all over the world exchange their experiences.
- Your local representative for Industry Automation and Drive Technology.
- Information on field service, repairs, spare parts and consulting.

Table of contents

Safety notes 1

1.1 Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines, and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.

Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.

Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit:

http://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under

http://www.siemens.com/industrialsecurity.

1.2 Introduction

Introduction

In the modern office and production world, information and communication technology is playing an ever more important role. In the same way as the further developments in information technology and its processing in process control technology, the requirements for IT security have become ever more complex and extensive. Protecting the IT landscapes is therefore becoming increasingly important.

Growing vulnerability and the risk of severe financial damage as a result of IT risks increase the pressure to act, to prevent damages through active IT security management and to minimize the residual risk.

Security objectives

There are three main basic values of IT security: Availability, confidentiality and integrity.

- Availability: Services, functions of an IT system or even information are available to the user at the required time
- Confidentiality: Confidential information must be protected against unauthorized disclosure.
- Integrity: The data is complete and unchanged. Loss of integrity of the information can mean that it has been changed without authorization or information has been falsified by the author, or the time of creation has been manipulated.

Note

In process automation, the security objective "Availability" has the highest priority. The requirements for IT security here differ significantly from the office world, where "Confidentiality" has the highest priority.

Motivation

In the following sections, the IT security aspects relevant for an overall system are described and recommendations are made. The entire lifecycle of a system should be covered by this:

- Installation
- Configuration
- Engineering
- Commissioning and operation, including delta engineering, updates and enhancements

Most of these topics are already covered in comprehensive Siemens documents within the framework of the SIMATIC products for process automation. For this reason, these documents will be referenced in the following section for the aspects applicable to BRAUMAT/SISTAR, in a suitable form for fast retrieval. Only product-specific topics are described in their own sections.

1.3 Recommendations and cross-references

PCS 7 & WinCC security concept

With regard to the utilized Windows operating systems and the communication architecture and networking, the BRAUMAT/SISTAR and PCS 7 process control systems are very similar. Consequently, the documentation already available concerning IT security, specifically in production plants and process control systems, is referred to here.

Complete documentation of BRAUMAT/SISTAR and PCS 7 is available for free and in multiple languages on the "Simatic Product Support" Internet page.

The manual "SIMATIC Process Control System PCS 7 Security concept PCS 7 & WinCC - Basic document (http://support.automation.siemens.com/WW/view/en/60119725/0/en)" supports administrators in the planning and configuration of the PC networks:

- It will give you an overview of network, computer and Windows user management, as well as the management of user and access rights.
- It will provide you with important information about integrating networks (process bus, terminal bus) into the Windows management system.
- You can find information about how to establish patch management and secure communication paths between the PC stations in a system.

Note

This manual builds on the possibilities of the operating systems used in both systems and assumes the reader has proficiency in administering these operating systems.

The following terms/functions are realized differently in BRAUMAT/SISTAR or are not relevant:

- The management of users and their rights is NOT via SIMATIC Logon, but rather with separate user / operator levels.
- OS and ES together on one PC station
- S7-400H, OS-Webclient/OS-Webserver are not relevant
- Windows XP, Server 2003 are not relevant
- SIMATIC BATCH, SIMATIC Route Control are not relevant

SIMATIC process control system PCS 7 Compendium Part A to Part F

This manual serves as a compendium in addition to the comprehensive product documentation for SIMATIC PCS 7. The basic work steps of project creation and parameter assignment are described in the form of instructions with numerous figures.

"SIMATIC process control system PCS 7 Compendium Part F - Industrial Security (https://support.industry.siemens.com/cs/ww/en/view/109742220)", in particular, describes many strategies, recommendations and instructions that are also valid for and applicable to BRAUMAT/SISTAR in identical or slightly modified form.

Topics in the above-mentioned document that are also applicable here are analyzed in the following sections (organized by chapter of that document) and presented in tabular form in which the specific particularities are indicated.

Preface

Subsection	Topics	BRAUMAT/SISTAR
Subject matter of this manual	Security objectives, prioritization, possible consequences	No restriction
Validity, further support, Service &Support, Training	Versions, hyperlinks	Valid as of V7

Security strategies

Subsection	Topics	BRAUMAT/SISTAR
"Defense-in-Depth" concept	Security measures, security cells, access points	No restriction
Configuration model	Configuration model without security cells (Negative example)	OS and ES together Not S7-400H no OS-Webclient/OS-Webserver

Network security

Subsection	Topics	BRAUMAT/SISTAR
Automation and security cells	Description of a configuration model from separate security cells	OS and ES together Not S7-400H no OS-Webclient/OS-Webserver
Addressing and segmenting	Configuration model, IP addresses, IP sub-networks, Windows settings	No restriction
Name resolution	Computer names, change, NetBIOS, DNS, Windows settings	Computer name can also be changed after installation no WinCC
Administering networks and network services	Central administration (Windows working groups) Local administration (Domain/Active Directory)	No restriction
Access points to security cells	Automation firewall Access rules for the configuration model Web publishing	Firewall products from PCS 7 AddOn Catalog can be used Web server is not supported
Secure communication between security cells	AS - AS data exchange with VPN connections (IPSec log) via SCALANCE S modules Quarantine station (File server)	No restriction Not required
Configuration of the SCALANCE X network components	Security settings for the Ethernet switches	No restriction

System hardening

Subsection	Topics	BRAUMAT/SISTAR
Overview	General	No restriction
Installation of the operating system.	Uninstall Windows components, restrict Windows services	Valid as of V7
Security Controller	PCS 7 automatic security settings	Valid as of V7

Subsection	Topics	BRAUMAT/SISTAR
Windows Firewall	Firewall settings with regard to subnetworks in security cells (configuration model)	No limitation, especially important for 'SQL DB host' option in the other subnet
BIOS settings	Security settings in the BIOS	No restriction
Use of mobile data carriers	Information, references, Windows settings	No restriction (Windows XP, 2003 not relevant)
Whitelisting	McAfee Application Control	No restriction
Web-Server for SIMATIC S7-CPUs	Deactivate Web Server in CPU	No restriction

User administration and operator authorization

Subsection	Topics	BRAUMAT/SISTAR
Overview	Separating user and operator authorization	No restriction (Windows user / BRAUMAT/SISTAR user)
Administering computers and users	Windows user accounts, "Autologon"	OS and ES together SIMATIC WinCC not relevant SIMATIC BATCH not relevant SIMATIC Route Control not relevant
Password Policies	Information, references "secure passwords"	No restriction
Operator authorizations - Management of user rights	Job-based operator authorizations (role-based access) SIMATIC Logon Changes to the change or ES log	Own user / operating level concept (see system documentation). Replacement by "Global Logging"
Protection level concept	Protection levels in automation device, CPU key switch, CPU password protection	Not S7-400H Can only be used for STEP 7

Patch management

Subsection	Topics	BRAUMAT/SISTAR
Overview	Microsoft Windows update distribution • automatic (WSUS) • manual • Links to Microsoft and Simatic documentation	No restriction
Windows Server Update Service (WSUS)	Configuration model with security cells	OS and ES together Not S7-400H no OS-Webclient/OS-Webserver
Manual update	Procedure	No restriction

Protection against malware using virus scanners

Subsection	Topics	BRAUMAT/SISTAR
Overview	Security objectives, types of malware, distribution of virus signature files (virus patterns), update sources, Firewall rules	OS and ES together Not S7-400H no OS-Webclient/OS-Webserver
What to do in case of virus infection	Introduction, procedure	No restriction

Backing up and restoring data

Subsection	Topics	BRAUMAT/SISTAR
Overview	Types of backup (Project backup, system backup)	No restriction
Backup strategy	Extent of project / system backup, backup interval, retention and archiving	No restriction (WinCC, Windows XP, 2003 not relevant)

Remote access

Subsection	Topics	BRAUMAT/SISTAR
Secure remote maintenance based on the Siemens Remote Service Platform	SIEMENS Remote Service (SRS) Platform	No restriction
Creation of a remote service concept	Questions, security aspects	No restriction (WinCC not relevant)
Connection possibilities to the Siemens Remote Service Platform	Concepts, internet access ADSL/SDSL, VPN connection	No restriction

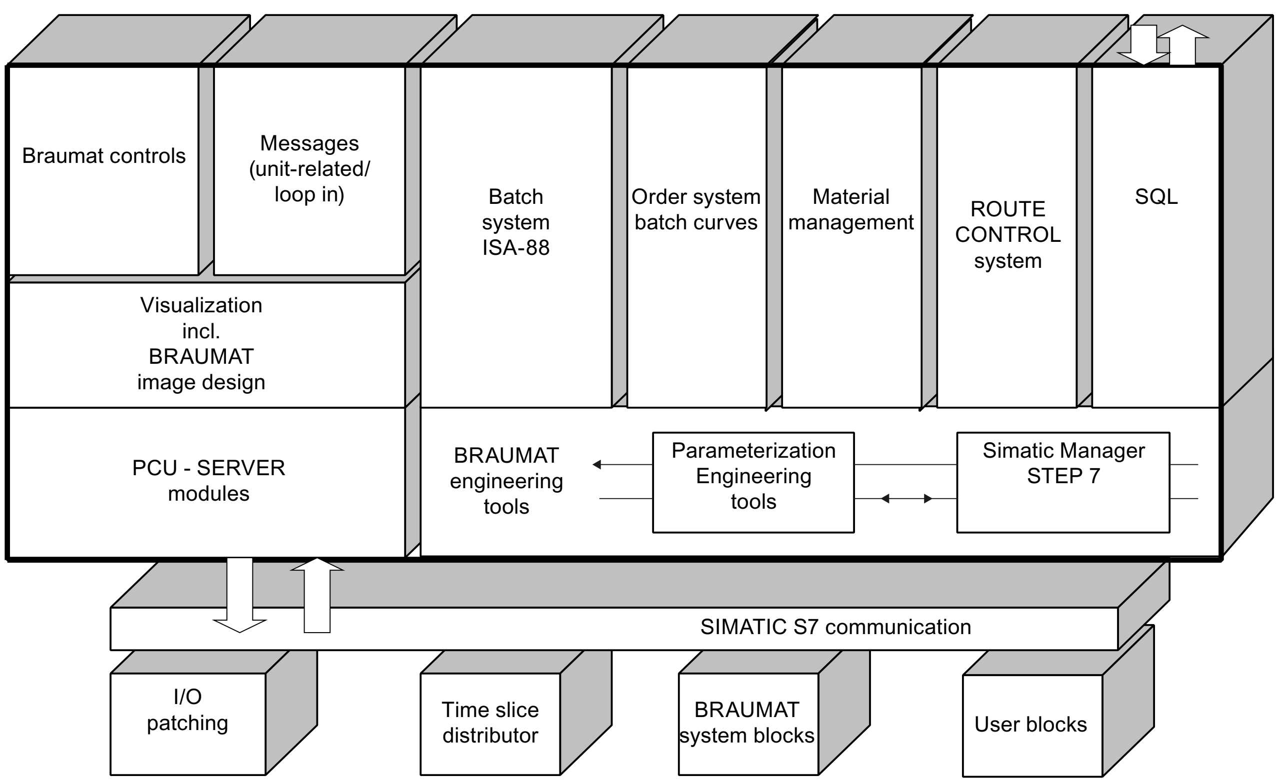
2 Description of the system

2.1 General

2.1.1 What is BRAUMAT/SISTAR

The system consists primarily of modules and components:

- Separate visualization system
- Parameter assignment of the system objects
- Graphical recipe system
- Job management
- Route control (option)
- SQL adapter (option)
- Framework software SIMATIC



The system also includes the runtime environment in the SIMATIC S7 controller, the technological functions, the visualization component and the messages.

The SIMATIC Runtime system is available on the installation medium as a separate delivered STEP 7 project for each supported CPU generation.

It must be adapted to the actual HW configuration (number of CPUs, communication, I/O) during project engineering and supplemented with the user software (units, equipment operations and interlocks).

The user-provided SIMATIC S7 programming software tools are needed for this:

- CPU type S7-400: STEP 7 V5.6
- CPU type S7-1500: STEP 7 TIA Portal V15

In other sections, the user program consists of configuring of blocks using the BRAUMAT engineering tools.

Note on CPU type SIMATIC S7-1500:

With the current system version, the system also supports the SIMATIC S7-1500 controller family.

The release is initially subject to the following limitations in this stage.

- The "Route Control System" option is not supported (modern solution for future version planned)
- AS-AS cross coupling is not supported (modern solution for future version planned)
- STATUS.exe is not supported (status display only possible via BLR objects)
- GF weigher is not supported (modern solution for future version planned)
- TF weigher with SIWAREX M is not supported (outdated technology)
- SILO-TANK data is not supported (modern solution for future version planned)

2.1.2 Definitions

Basic principle

The table below shows terms used in these manuals.

procedural control

Control in which device-oriented actions are controlled in an ordered fashion to allow a process-oriented task to be performed.

procedural element

A building block for procedural control that is defined by the procedural control model.

process cell

A logical grouping of equipment that includes the equipment required for production of one or more batches. It defines the span of logical control of one set of process equipment within an area.

NOTE: This term applies to both the physical equipment and the equipment entity.

area

A component of a batch manufacturing site that is identified by physical, geographical or logical segmentation within the site.

NOTE: An area may contain process cells, units, equipment modules, and control modules.

equipment procedure

A procedure that is part of equipment control.

exception handling

Those functions that deal with plant or process contingencies and other events which occur outside the normal or desired behavior of batch control.

basic control

Control that is dedicated to establishing and maintaining a specific state of the equipment or process condition.

mode

The manner in which the transition of sequential functions is carried out within a procedural element or the accessibility for manipulating the states of equipment entities manually or by other types of control.

state

The condition of an equipment entity or of a procedural element at a given time.

NOTE: The number of possible states and their names vary for equipment and procedural elements.

arbitration

A form of coordination control that determines how a resource should be allocated when there are more requests for the resource than can be accommodated at one time.

batch

The material that is being produced or that has been produced by a single execution of a batch process.

An entity that represents the production of a material at any point in the process.

NOTE: Batch means both the material made by and during the process and also an entity that represents the production of that material. Batch is used as an abstract contraction of the words “the production of a batch”.

batch control

Control activities and control functions that provide a means to process finite quantities of input materials by subjecting them to an ordered set of processing activities over a finite period of time using one or more pieces of equipment.

batch process

A process that leads to the production of finite quantities of material by subjecting quantities of input materials to an ordered set of processing activities over a finite period of time using one or more pieces of equipment.

batch schedule

A list of batches to be produced in a specific process cell.

NOTE: The batch schedule typically contains such information as what is to be produced, how much is to be produced, when or in what order the batches are to be produced, and what equipment is to be used.

control module

The lowest level grouping of equipment in the physical model that can carry out basic control.

NOTE: This term applies to both the physical equipment and the equipment entity.

equipment control

The equipment-specific functionality that provides the actual control capability for an equipment entity, including procedural, basic, and coordination control, and that is not part of the recipe.

equipment entity

A collection of physical processing and control equipment and equipment control grouped together to perform a certain control function or set of control functions.

exclusive-use resource

A common resource that only one user can use at any given time.

Shared equipment

A resource that can provide services to more than one requester.

NOTE: Common resources are identified as either exclusive-use resources or shared-use resources.

Basic recipe

A recipe that accounts for equipment capabilities and may include process cell-specific information.

ID

An ID is a unique identifier for objects of the physical model and the batch objects. These can be: units, equipment operations, batches, raw materials, recipes, etc.

coordination control

A type of control that directs, initiates, and/or modifies the execution of procedural control and the utilization of equipment entities.

operation

A procedural element defining an independent processing activity consisting of the algorithm necessary for the initiation, organization, and control of phases.

shared-use resource

A common resource that can be used by more than one user at a time.

lot

A unique amount of material having a set of common traits.

NOTE: Some examples for common traits are material source, the master recipe used to produce the material, and distinct physical properties.

line/train

See definition for “train”.

procedure

The strategy for carrying out a process.

NOTE: In general, it refers to the strategy for making a batch within a process cell. It may also refer to a process that does not result in the production of a product, such as a clean-in-place procedure.

process

A sequence of chemical, physical or biological activities for the conversion, transport or storage of material or energy.

process action

Minor processing activities that are combined to make up a process operation.

NOTE: Process actions are the lowest level of processing activity within the process model.

process control

The control activity that includes the control functions needed to provide sequential, regulatory, and discrete control and to gather and display data.

process input

The identification and quantity of a raw material or other resource required to make a product.

process management

The control activity that includes the control functions needed to manage batch production within a process cell.

process operation

A major process activity that usually results in a chemical or physical change in the material being processed and that is defined without consideration of the actual target equipment configuration.

process output

An identification and quantity of material or energy expected to result from one execution of a control recipe.

process parameter

Information that is needed to manufacture a material but does not fall into the classification of process input or process output.

NOTE: Examples of process parameter information are temperature, pressure, and time.

process stage

A part of a process that usually operates independently of other process stages and that usually results in a planned sequence of chemical and physical changes in the material being processed.

recipe header

Information about the purpose, source, and version of the recipe such as the recipe and product identification, creator, and issue date.

recipe management

The control activity that contains the functions for creating, storing, and maintaining general, site, master, and control recipes.

recipe operation

An operation that is part of a recipe procedure in a master or control recipe.

recipe phase

A function, that is part of a recipe procedure in a basic or control recipe.

recipe procedure

The part of a recipe which defines the strategy for producing a batch.

personnel and environmental protection

The control activity that prevents events from occurring that would cause the process to react in a manner that would jeopardize personnel safety and / or harm the environment

and / or

takes additional measures, such as starting standby equipment, to prevent an abnormal condition from proceeding to a more undesirable state that would jeopardize personnel safety and/or harm the environment.

formula

A category of recipe information that includes process input, process parameters, and process outputs.

Stream

A collection of one or more units and associated lower level technical equipment groupings that has the ability to be used to make a batch of material.

control recipe

A type of recipe which, through its execution, defines the manufacture of a single batch or a specific product.

equipment module

A functional group of equipment that can carry out a finite number of specific minor processing activities.

NOTE:

An equipment module is typically centered around a piece of process equipment (a weigh tank, a process heater, a scrubber, etc.). This term applies to both the physical equipment and the equipment entity.

Two examples of minor process activities are dosing and weighing.

equipment operation

An operation that is part of equipment control.

equipment phase

A phase that is part of equipment control.

unit

A collection of associated control modules and/or equipment modules and other process equipment in which one or more major processing activities can be conducted.

NOTE:

Units are presumed to operate on only one batch at a time. Units operate relatively independently of one another.

This term applies to both the physical equipment and the equipment entity.

Examples of major processing activities are react, crystallize, and make a solution.

unit procedure

A strategy for carrying out a contiguous process within a unit. It consists of contiguous operations and the algorithm necessary for the initiation, organization, and control of those operations.

unit recipe

The part of a control recipe that uniquely defines the contiguous production requirements for a unit.

NOTE: The unit recipe contains the unit procedure and its related formula, header, equipment requirements, and other information.

recipe unit procedure

A unit procedure that is part of a recipe procedure in a master or control recipe.

enterprise

An organization that coordinates the operation of one or more sites.

general recipe

A type of recipe that expresses equipment and site-independent processing requirements.

site

A component of a batch manufacturing enterprise that is identified by physical, geographical or logical segmentation within the enterprise.

NOTE: A site can contain areas, process cells, units, equipment modules, and control modules.

site recipe

A type of recipe that is site-specific.

NOTE: Site recipes may be derived from general recipes recognizing local constraints, such as language or available raw materials.

2.2 System Overview

2.2.1 General

Although the process control system is predestined for the automation of batch tasks and was developed for that reason, also continuous processes can be handled by the system as the successful adaptation in several "continuous" plants shows.

Aims of the automation are:

- A constant product quality
- Increase of the safety of operation
- Improvement of the transparency of the operating actions
- Reproducibility of well-proved recipes
- best usage of raw materials
- Reducing of repetitive work of the personnel
- Engineering according to ISA-88 (IEC 61512-1) rules
- Route control

The systems software matched the technological requirements and contains standard functions for control modules, monitoring and control, recipe management, alarming and logging, communication, test and diagnosis.

The system is so built up that it can be adapted to the system optimally with a comparably small project planning expenditure onto the requirements. Project planning and servicing of the system can be done by technologists or operating technicians.

2.2.2 Parameterization

In the dialog, the object-specific parameters are modified in tables for each object class and instance.

These include: names of individual control modules and types, limit values/ranges, control parameters, activations, etc.

As by the configuration the lists not only are created, but also are preset with default values, normally only a part of the parameters needs to be reedited by the user.

2.2.3 Argument addressing

In the system every element has its own name (property or attribute) and is grouped together with other elements into a parameter set (object or instance of a class). Several parameter sets are assembled to a parameter list (filed as *.pcu files in the system).

The parameter set of the component MESS (Measurement) contains, e.g., arguments for the following:

- Process value (XIST)
- lower scale and upper scale value (XANF, XEND)
- state bits

In addition it is assigned a system-specific name.

Parameter sets of the same structure (class) are numbered starting by 1. Therefore MESS, 25,XIST is the 25th measuring object.

The physical address of a parameter the automation device of the arguments stands is not important for the user and normally he doesn't want to know the address. He wants to select "an object" instead of a physical address. This way brings an advantage that with process specific names can be worked easier and the system performs calculating the address (DB data block number, DW data word number = address).

2.2.4 System blocks

The system has a lot of parameterizable blocks. Every block is related to a process function (e.g., analog output, unit sequencer, PID control). The number of instances per block that can be done at runtime must be configured. Detailed information is provided by the respective subchapter (see chapter System blocks and parameters (Page 214)).

2.2.5 Recipe management

Recipes contain the necessary specifications for the task, processed by the sequencer of a unit.

The executable control recipe is downloaded into the PLC

and is divided into recipe procedures for every unit. General information is:

- Technical operation to be processed
- Monitoring time and / or max. duration
- Set points ("how" it is supposed to be made)

The operating and editing functions contain input, correct, copy, and log mechanisms of recipes, for working offline as well as online. Correcting means modifying set point values, deleting and inserting whole parts of recipes.

2.2.6 Linking functions

For the linking of function modules with each other, e.g., measurement, measured value check, control module, analog output, control system, etc., a conventional model is used like modules wired together.

Function modules provide output parameters serving as sources for other modules input parameters.

E.g., a PID control receives its actual value (XIST) from a MESS object or the set point value (W) from the sequencer (TEILANL).

2.2.7 Diagnostic functions

For test purposes and for a fast fault location the system provides test routines for:

- states of individual control modules (ICMs)
- Interlocking ICMs
- Simulation of measured values
- Program status with symbolic names for interlocking and triggering operations
- Setting variables in the AS (for example I(n), O(ut), M(emory), ...)
- Object state via parameterization

2.2.8 Standard functions

A basic concept of the system is to offer a large amount of standard functions, so that the user has to do not as many complex and repetitive tasks.

To be flexible for automation, the system provides default attributes that can be modified by the user through parameterization. The systems software contains components for:

- monitoring individual control modules
- sequencer modules
- Analog value processing
- Production logs (e.g., brewing or cleaning logs)
- Event log files
- Recipe management
- order management system
- control routines
- Process visualization
- Data exchange and communication
- Test and diagnosis

2.2.9 Alarms and messages

Events during operations, both disturbances and also manual interactions of the automatic process, the beginning and ending of production are displayed on the screen in a message window and archived on the hard disk. What is recorded depends on configured criteria.

In order to reduce the time of engineering the appropriate standard functions are provided with an interface to a buffer and write messages into this buffer on occurrence, time and additional message flags are added.

On the OS a program receives each message from the buffer, converts the message and additional information into one line and outputs the message onto the screen.

The message texts contain the names of the control elements, units, measure or control modules extended with "INTERRUPT", "AUTOMATIC", "MANUAL", "ERROR" texts. The texts are to be parameterized by the user so that also foreign-language versions based on ASCII code are possible.

For user-specific messages an own alarm block is available which can be linked together with user signals. Per message a message text is available for incoming and outgoing messages. (Please refer to the chapter MSG.)

The message system, including display and recording, is described in chapter 'Message system (Page 1009)'.

2.2.10 Operating philosophy

Operator elements are screen, keyboard with function keys and mouse. The function keys are numbered; their respective meaning is explained in the menus. With the function keys:

- select an application
- trigger commands
- go forward or backward within data sets.

The system is handled by menu and additional dialogs which are language-dependent.

Detailed information about the screen layout and control elements is in the next chapter.

2.2.11 Monitoring and control

The system supports one to many IOS clients, depending on the system configuration, either area-based or as multiclient, on which one to many AS systems (Braumat PCUs) can be operated. Each workplace comprises one or more color screen(s), a keyboard and a mouse.

Ready made dialogs and using routines are standard.

See also

Process pictures (Page 409)

2.2.12 Logging

The system supports the printers installed in the Windows operating system (directly connected or network printers). Messages, logs, and hardcopies can be printed out here.

Printing logs are on request by the user.

Stepping and alarm logs are stored on hard disk.

2.2.13 Picture construction

The application 'screen construction' allows the system's visualization to be defined by creating executable process diagrams.

The dynamic process diagram parts are facilitated by the free placing and activating of the standard variables and control elements (controls) available in the system.

Existing static graphics (icons, parts pictures) in format "bmp", "emf", "jpg" and "png" can be added.

The background image is created with an external standard tool in format "bmp" or "jpg". The screen construction reads this screen and places the configured static graphic elements or dynamic picture elements over the top.

2.3 Processes and batches

2.3.1 Overview

A process is a sequence of chemical, physical or biological activities which transform, transport or store material or energy. Industrial production processes can be subdivided into continuous processes, into processes with single job production or into batch processes.

The classification of a process, when regarding the process output

- with a continuous product flow (continuous)
- with specific numbers of individual jobs of product (job production) or
- with specific amount of product (batch process).

Even if individual subjects of this description are valid for processes with job production or continuous processes these process types are not described more detailed.

2.3.2 Continuous processes

In a continuous process the material flows continuously through the processing equipment. If a stationary operating state is achieved, the production process is independent of the duration of the operation. The start up, transient, and shutdown steps are usually unimportant.

2.3.3 Single job production

A process with job production divides products into several production lots, that are based on common raw materials, production requirements, and production histories. Processes with job production transport a specific amount of a product from unit to unit whereas the specific properties are preserved.

2.3.4 Batch processes

In this chapter the modeling of a batch system, the procedures, and their recipes are handled. The models are the base of the system.

The models are:

- process model
- physical model
- technical procedural model

A batch process provides an amount of product as the output of the preceding step. which is called a batch. Within a batch process specific amounts are transformed on different units. A batch process is neither continuous nor discrete, but has, however, such features.

Example of a batch process:

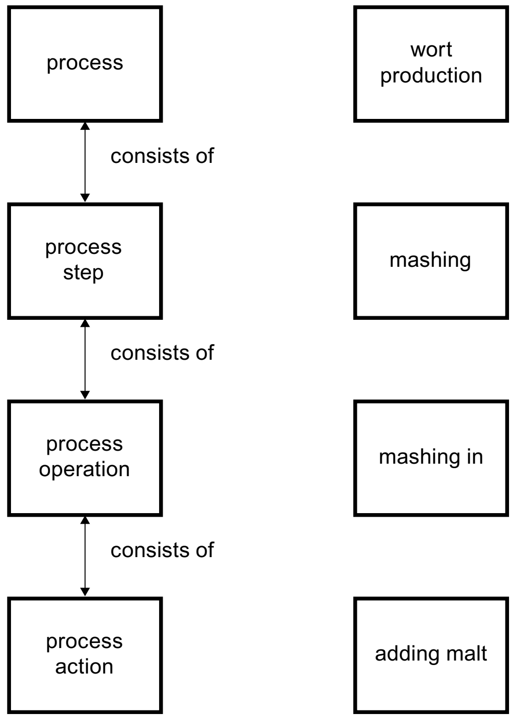


Figure 2-1 Process model and example of the Wort production

2.3.4.1 Process step

A process consists of one or several steps. These process-steps are processed together and can run sequentially or simultaneously. A process-step is part of a process that runs independently of other process-steps and creates a planned order of chemical or physical transformations of dealt materials.

For the example of the Wort production these are:

- Milling / Grinding
- Mashing
- Lautering
- Heating
- Casting wort
- Cooling

2.3.4.2 Process operation

Every one process step consists of process operations, which describe larger production sequences. A process function transforms the material either chemically or physically.

For the example of the Wort production / mashing these are:

- Providing water
- Mashing in
- Heating
- Saccharifying
- Mashing out

2.3.4.3 Process action

Every process function consists of process actions. The process actions are needed for the execution of the processing. These smaller parts of processing build a process operation.

For the example of the Wort production / mashing / Mashing in these are:

- Providing water
- Agitation <A[Agitator|Propeller]>activate
- Pre-mashing
- Draining grind<A[fuselage|hull]>

2.3.5 Physical model

In this part the physical model of a batch system is described.

The model consists of seven levels. At the top it starts with an enterprise, an area, and a site. It is used to show the relationship of the lower levels to the producing enterprise. These three levels are not described here in detail.

The lower four levels of this model refer to concrete types of equipment. The figure below shows a combination of procedural and technical controls to form a group. These four levels (process cell, units, equipment units, and single control units) are determined through engineering. The single control units are grouped on a lower level to get an element on a higher level. This handling simplifies the operation of this equipment. This created unit can not be split, except for through further engineering on this level.

For an easier understanding the entire model is shown, whereas the part is covered by the BRAUMAT/SISTAR system from "area" and below.

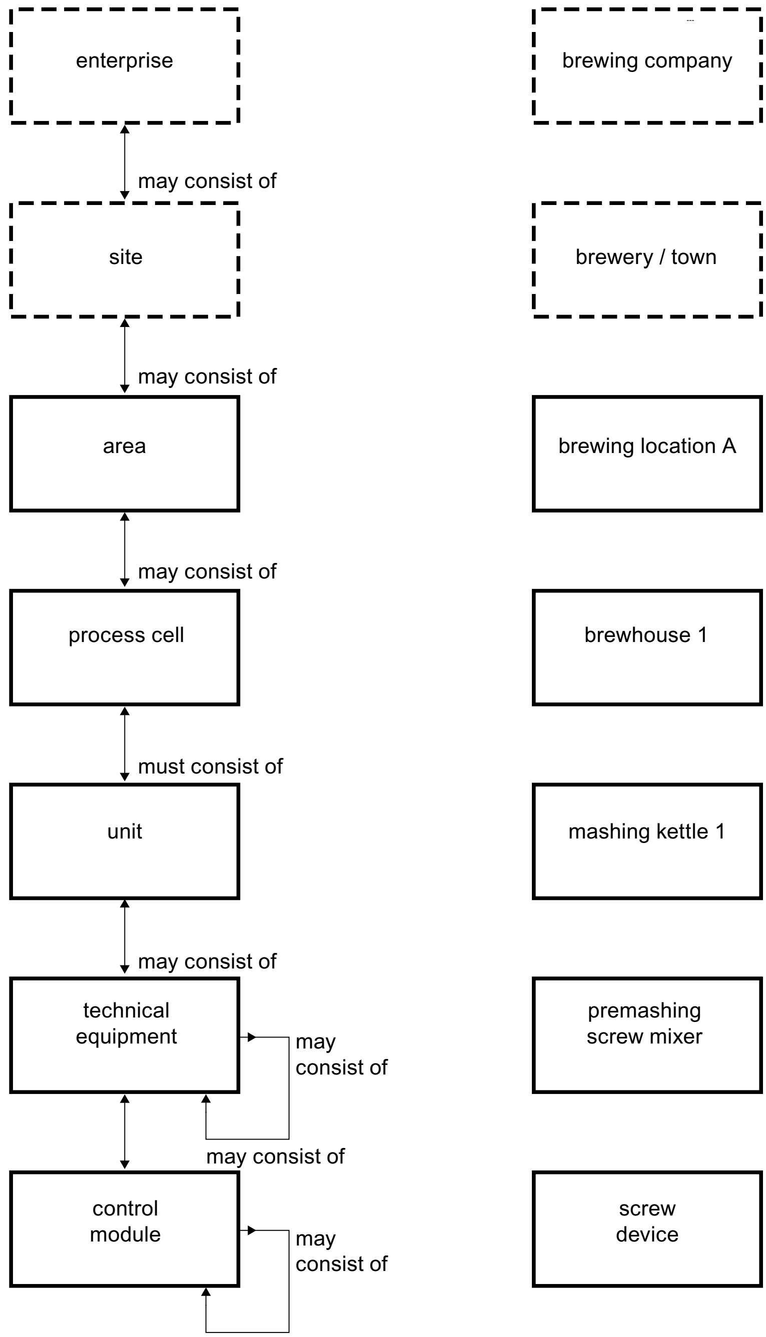


Figure 2-2 Physical model

2.3.5.1 Enterprise level

An enterprise consists of one or more areas. It includes sites, process cells, units, technical equipment, and single control units.

The enterprise plans where, which, and what products are produced.

Beside the charge oriented production of an enterprise, there are still many other facts that affect the limitation of the enterprise. This limitation is not described here.

2.3.5.2 Site level

A site is a structural, geographical or logical classification of an enterprise. It can contain areas, process cells, units, technical equipment, and single control units.

The classification of a factory is related to organizational and entrepreneurial criteria.

Beside the charge oriented production, there are still many other facts that affect the classification. This limitation is not described here.

2.3.5.3 Area

An area is a physical, geographical or logical group described by an enterprise. The Area can consist of process cells, units, technical equipment, and single control units.

Next to the charge oriented production, there are still many other factors that affect this limitation. This limitation is not described here.

2.3.5.4 Process cell

A process cell contains all units, technical equipment, and single control units to produce a batch.

The activities for the process control system match the requirements, where many different methods and technologies are necessary. Physical actions related to control are fixed through process-specific circumstances or administrative requirements.

A line is a part of the process cell which consists of all units and other equipment which depends on the batch to be produced. But a batch does not necessarily use all units within a line, whereas several batches and products can use a line simultaneously. The order in which the batch passes the units is named a path. An area can consist of several lines, which must, however, be part of the area.

All units which are used during the batch production are grouped logically within a system. Here it is determined which logical control options for the units within a process cell are available. A process cell provides the possibility of a disposition on process cell-level and the planning of further controlling strategies. This can be useful especially in case of emergency.

2.3.5.5 Unit

A unit consists of technical equipment and Single Control Units. Parts of the unit can be assigned either to the unit itself or be part of a resource pool and used for a specific time to perform tasks.

A unit is able to execute larger processing activities and connects the required procedural and control procedures in order to run it as an independent equipment group. A unit is related to bigger processing equipment, e.g., a mixing kettle. It contains the logical statements of the equipment for greater processing activities in order to occupy these or to run completely. The individual units work mostly independently of each other.

A batch or quantity of a batch is included in an individual unit or it is processed here at a specific time. It is expected, that the unit only runs one batch at a time.

2.3.5.6 Technical equipment

The technical Equipment consists of single control units and related technical equipment. It can be part of a unit or an independent group of elements of a system. As an independent resource element it can be used exclusively or in parallel.

Technical equipment runs specific, smaller processing activities, e.g., scaling or dosing. It contains the control and procedural equipment necessary for the execution. It is located mostly around a part of processing equipment, e.g., a filter. The frame of the technical equipment sets the number of steps which can be executed on equipment.

2.3.5.7 Single control unit

A single control unit is a group consisting of sensors, control elements, control modules, and the related equipment. Several single control units can be combined into another single control unit, e.g., several individual control modules (ICMs) are grouped into a dosing unit.

2.3.6 Classification of plants

Batch systems can be classified with respect to two criteria:

- The products produced on them (only 1 product can be produced on the system or different products can be produced on the system)
- The physical structure

This process control system allows the mapping of all the system types presented here.

2.3.6.1 Single line production

The term "Single path structure" comes from the English standard.

A batch uses a set of units. From a set of input materials, an end product is produced in multiple process stages (units).

Several batches can run simultaneously in the system. The batches pass through the units sequentially.

The mapping takes place using recipe procedures. The mapping of recipe procedure and units is fixed.

Mapping in the process control system

A simple brew house represents a single path structure. A single path structure can be mapped directly with the recipe system.

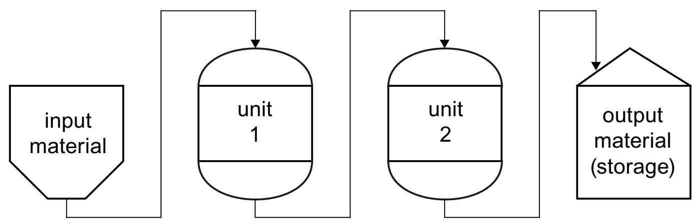


Figure 2-3 Single path structure

2.3.6.2 Multiple line production

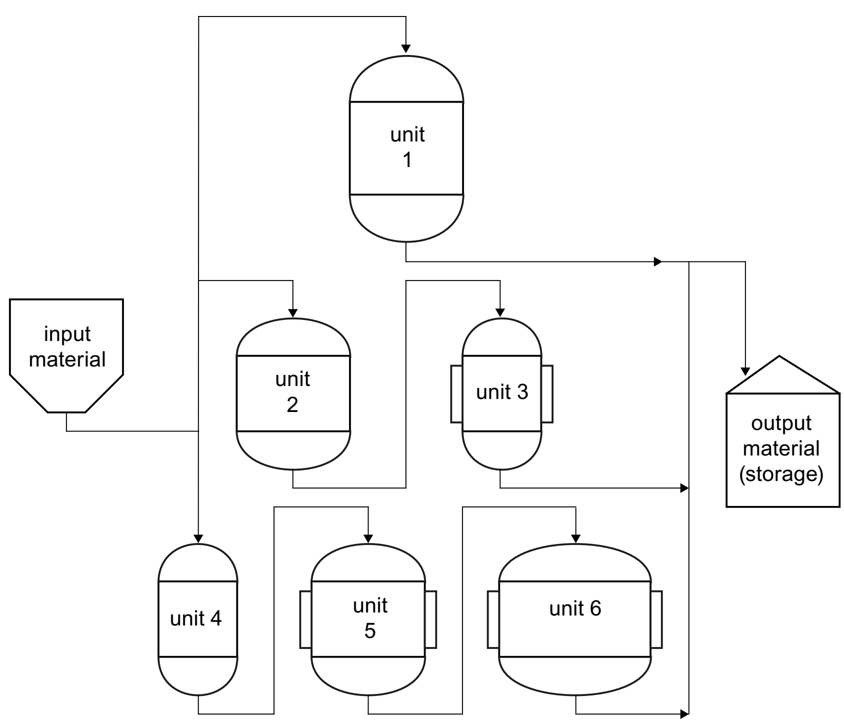
The term "multiple path structure" comes from the English standard.

A multiple path structure consists of multiple single path structures. The single path structures share raw materials and target containers. Multiple batches can run in parallel in the structure. The units of the paths can be physically identical or similar. However, they can also be completely different. A batch can pass through only one path during the process.

Mapping in the process control system

These structures are mapped using the line recipe function. An automatic implementation on very different units can also be achieved with suitable engineering.

The fermenting cellar and also the storage cellar of a brewery are examples of multiple path structures. Also brewhouses with several brewing lines are multiple path structures.



TA: Teilanlage

Figure 2-4 Multiple path structure

2.3.6.3 Network systems

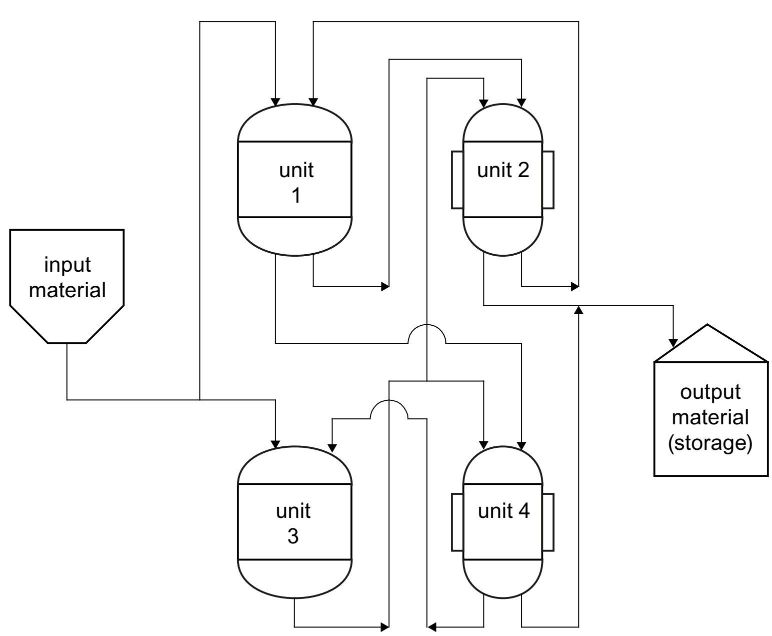
The term "network structure" comes from the English standard. The paths through the structure can be fixed or variable. Before the transition of the batch to the next process stage, the unit used for it can still be changed.

Mapping in the process control system

A network structure can be mapped using three mechanisms.

1. Line realization
   - All possible combinations of units have to be engineered in the recipe system.
   - It is not possible to change the unit at runtime.
2. Dynamic lines:
   - Possible combinations of units are created,
   - where at runtime the user can and must decide whether to change a unit or not.
   - The system checks whether switching is allowed.
   - The operator must actively perform switching.
3. Free path selection:
   - For the recipe unit procedures (RUP) of the recipe procedure more than one unit can be provided, they are called "candidates".
   - The RUP is assigned automatically a candidate,
   - whereas a default path can be assigned.
   - The switching can be done by the operator or through the "Late binding" functionality.

A complex brew house with connections between the brew lines is an example of a network structure.



TA: Teilanlage

Figure 2-5 Network structure

2.4 Batch controlled operating

2.4.1 Base automation

The automation of a system is done in several levels. The bottom level represents the base automation

that includes:

- Control modules
- Interlock mechanisms
- Monitoring programs
- Exception handling
- Manual or automatic control

The base automation of a batch system is not different to a continuous production system.

2.4.2 Procedure control

Procedural functions using base level elements provide an automated controlling of an area The procedural elements are and are divided into the following groups depending on the complexity:

Designation DE	Designation EN	Abbreviation
Recipe procedure	Recipe Procedure	RP
Unit procedure	Recipe Unit Procedure	RUP
Recipe operation	Recipe Operation	ROP
Recipe phase *1)	Recipe Phase	RPH

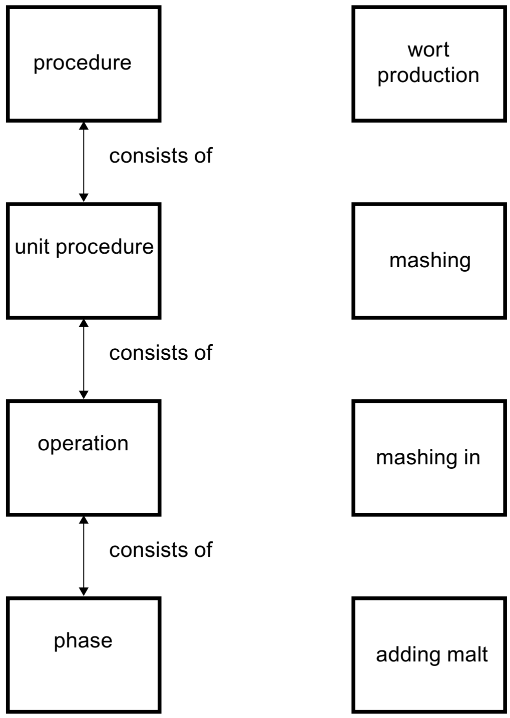


Figure 2-6 Procedural model of a batch system

Note

1. The phases of the higher-level operation are mapped together in the STL or SCL code of this operation. For a better overview, the phase actions can also be placed in different networks

2.4.2.1 Procedure

The procedure is a pattern for the system to run and create a batch. A procedure consists of several recipe unit procedures. "Wort" is an example of a procedure of a product, but there are also production for cleaning units (CIP) and other sequences in the unit.

2.4.2.2 Recipe unit procedure

The recipe unit procedure consists of a sequential order of phases (operations), where at a time only one phase is active. All operations of a recipe unit procedure run on the same unit. The recipe unit procedure can also contain the following elements:

- synchronization lines
- labels
- jumps
- alternatives

2.4.2.3 Operations

Basic principle

and modify a charge in a

- chemical (Saccharification)
- biological (Fermentation) or
- physical (Fill one container, transportation) manner

Transients of operations represent mostly a secure state within a procedure at which delays or breaks are possible without effects.

Representation of the phases within the operation

The smallest procedural elements in a batch controlled system are the phases. A phase can execute a series of single actions.

Note

A phase in the sense of ISA-88 does not exist in this process control system.

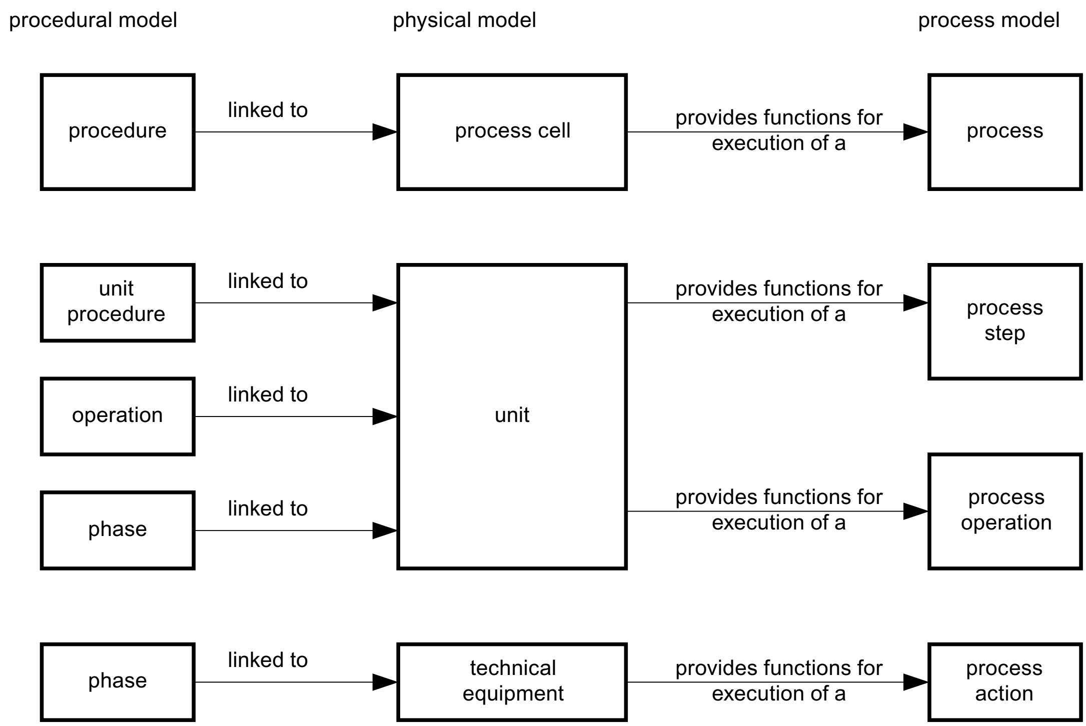
The phases of the higher-level operation are mapped together in the STL or SCL code of this operation. For a better overview, the phase actions can also be placed in different networks.

A phase can:

- lock and unlock control modules
- control and check limits
- provide set points
- read process values and calculate them

2.4.3 Connection between models

The representation of the three models is shown in the figure below. Basically each level of the process level is assigned to a level of the physical or procedural model.



2.4.3.1 Tasks of a process cell

The unit must include the following tasks:

- Processing activity and controlling of one or several batches
- Monitoring resources
- Control unit allocation
- Base automation of functions spread over several units
- Controlling procedures
- Coordination of the unit recipes

2.4.3.2 Tasks of a unit

The unit must include the following tasks:

- Base automation
- Controlling unit procedures
- Coordinated control of units

2.5 Recipes

2.5.1 Implementation of the models

2.5.1.1 Area

The area provides functions for several process cells.

Batches are input and started in the batch management. Batches of different recipes and systems can be created here. Starting a batch can be triggered by time, an event or load of the system.

2.5.1.2 Process cell

The process cell controls all functions necessary for one or several batches. The control orders are transferred down to the underlying objects. These are units, technical equipment elements, and single control units. According to the complexity of the system the batch can run on one or more units in parallel. The allocation of units is done by the control recipe. According to the system type (single line, multiple line or network structure) the allocation of units can vary dynamically.

Base automation of the system

Units are interlocked with one another in the basic automation of the system. Each unit provides interfaces in which system interlocks can be incorporated.

Procedure control of the system

The processing of a procedure is taken over by the recipe server. The recipe server starts and coordinates the sequence of the individual recipe unit procedures. Starting a procedure is done by starting a recipe unit procedure, all other recipe unit procedures are started via synchronizations.

Coordination of units

Several mechanisms are available for coordinating the individual recipe unit procedures:

- Synchronizations: Unit procedures are held until other unit procedures have reached a specific state (synch. line)
- Alternatives: Within a unit procedure one of n procedures can be started based on conditions
- Start call: Within a component (function block in SIMATIC) make a call to start other unit procedures

Units

Units coordinate the functions of the elements on lower levels, as for example technical equipment and single control units. The main purpose of the equipment control within a unit is to control the execution of a batch running on this unit at this time.

2.6 Structure of the System

2.6.1 Structure of a PLC (AS)

PCU systems (programmable control units) are used for the process control level. These consist of SIMATIC S7-400 or S7-1500 type automation devices and their associated technological program modules. They do control tasks, monitoring of the process execution, as well as measurement and processing. References to technical data and quantity schedules see Quantity schedule of functions and classes (Page 65).

2.6.1.1 Standard structure of data blocks

Standard structure

The data blocks of most of the technological functions have a standardized structure:

- Header data (structural description of the DB)
- Fill area
- Runtime copy of the dataset currently being processed
- Fill area
- Field of datasets

Header data

Data word	Function
DBW2	Offset to the start of the field for the datasets
DBW4	dataset length
DBW6	maximum number of datasets
DBW8	Used number of datasets
DBW10	offset to the runtime copy
DBW12	Current dataset number (used by the Scheduler)

Runtime copy (PCU type S7-400)

The "SCHEDULE" class provides a special working framework for technological function blocks using execution type "Tec FB, u-copy". The sequence is as follows:

- The current dataset is copied to the area of the runtime data
- The suitable function block is called
- The runtime data is copied back to the current dataset

Advantage: Programming in STL is considerably simplified because the technological function block only needs to work on the runtime data. Complicated indirect memory or register access to the field of data records can therefore be avoided along with the associated errors it causes.

Disadvantage: Cyclic copying of the datasets back and forth takes up CPU time.

Example in STL

```
DATA_BLOCK "SOME_USER_DB"
TITLE = Sample for a multiple instance data block

STRUCT
     iType      : INT        := 1;                        //Set this to 1
     iOffset    : INT        := 200;                      //Byte offset
to first record
     iLen       : INT        := 42;                       //
Sizeof(SOME_USER_UDT)
     iMax       : INT        := 128;                      //Max. number
of records
     iAct       : INT        := 128;                      //Active number
of records
     iRunOffset : INT        := 100;                      //Byte offset
to runtime copy
     iRecord    : INT;                                    //Current record ID
     abFill1    : ARRAY [1..88] OF BYTE;                  //Fill area
to address specified by "iRunOffset"
     u          : "SOME_USER_UDT";                        //Runtime copy
     abFill2    : ARRAY [1..58] OF BYTE;                  //Fill area
to address specified by "iOffset"
     au         : ARRAY [1..128] OF "SOME_USER_UDT";     //Record field
END_STRUCT

BEGIN

END_DATA_BLOCK
```

2.6.1.2 Description of block structures in *.pcu files

The structure of data blocks is used internally by the system mainly and preconfigured in description files. During startup of the system, the object manager (OM) reads these files and provides them to all clients.

For user-defined blocks the description can be extended. As this is only important mainly for system engineering of the S7 controls, this is described in the 'Engineering of the AS' manual.

2.6.2 Structure of a server (OS)

IOS systems are used for visualization and data archiving for the process control level. The systems are based on standard system components such as PC hardware and Windows operating systems, expanded to include the PCS software.

For to technical specifications and configuration limits, refer to the section Quantity schedule of functions and classes (Page 65).

2.6.2.1 Help functions

The system provides a lot of online-help. In every application you can start the online help via the menu.

2.6.2.2 Printing

The system supports one printer per IOS station. The system logs and hardcopies can be printed here.

Printing logs are on request by the user.

Stepping and alarm logs are stored on hard disk.

2.6.2.3 Hardcopy

A hardcopy from the system can be triggered on the IOS stations. Start clipprnt.exe which runs as a background process. Press the PRINT key if you want to make a hardcopy, a menu appears and you can decide on the page orientation (landscape or portrait).

2.7 Performance data

2.7.1 Quantity schedule of functions and classes

Core statement

The table below contains information about the configuration limits and the following preconditions apply:

- Number per function or instances per object class
- Per PCU or AS
- The specifications apply to the standard CPU type **CPU 416-3 (work memory 2x8 MB)** and **CPU 1518-4 (work memory 5MB/20MB)**

The following applies when using smaller CPU types:

Any restrictions depend on the project engineering requirements; in other words, the project engineer needs to calculate and assess whether the use of the relevant CPU is possible for the required BRAUMAT application and make sure that the available resources are adequate.

Configuration limits for standard functions

Object class/Function	PCU Version < V7	PCU Version V7
Actuators (ICM1-4)	4 x 255	4 x 256
Sequences (SEQUENCES)	64	128
Measured value recording (AIN)	256	512
Measured value check (MVC)	128	128
PID-controller (PID)	64	64
Three-point controller (THRESTEP)	96	96
Analog value outputs (AOUT)	256	256
Incremental encoder (INCO)	16	16
User messages (MSG)	1024	2048
Analog constants (ANA)	256	256
Start block sequence (SEQS)	96	96
Multifunctions (MULT)	128	128
Polygon adaptation (POLY)	32	32
Special values (SPEVAL)	511	511
Batch job - Start (SEQSTART)	64	128
Digital function modules DINT (DFM0-3/4)	4 x 255	5 x 255
Digital function modules REAL (DFM5-8)	(not available)	4 x 255
Trend reference values (CURVESCAN)	64	64
Maintenance data ICM (MAINT_ICM)	1023	1023

Object class/Function	PCU Version < V7	PCU Version V7
Maintenance data user (MAINT_USR)	1023	S7-400: 1023 S7-1500: -/-
Delay modules (TIMER1-2)	2 x 512	2 x 512
Values monitoring (VMON)	256	256
Binary connection objects (BLR)	(not available)	9 x 256
Control state (LINE)	(not available)	256
Tank state (TANK)	(not available)	128
Seat lifting (SLB)	(not available)	S7-400: 436 S7-1500: 342
Pulse generation (PULSE)	(not available)	S7-400: 96 S7-1500: 84
Recipe system		
Technical operations	999 per AS	1999 per AS
Recipe categories	255	
Master recipes per category	10000	
Process / job parameters per category	220	
Recipe procedures	32767	
Recipe unit procedures per recipe procedure	256	
Recipe operations per recipe unit procedure *)	255 guaranteed; more ROPs available, Max. number depends on number of setpoint values per operation and on the type of operation	
Set points per recipe operation	20 for PCUs with recipe system V5 13 for PCUs with recipe system V3	24 (recipe system V7)
Setpoints per sequence	24	
Total number of batches of all jobs	10000	
Process pictures		
Process pictures (OS)	Any	
Tags per process diagram	3700	
Number of controls + standard tags without tags (static texts/pictures, picture change, ExecPc, ExecFc) per picture	1024	
Number of standard tags with tags (Int, Hex, Bit, ICM, etc.) per picture	1200	
Measured value curves		
Analog measured values per curve server	1200	
Digital measured values per curve server	2400	

Object class/Function	PCU Version < V7	PCU Version V7
Number of curves per image	8 analog or 32 digital or 8 analog + 16 digital	
Number of curve groups:	240 (triggering of start/end of recording)	
System configuration		
Cross-coupling AS-AS **(S7-400 only!)**	31 connections (32 AS possible) 512 QK jobs	
OS–OS communication	Any number of connections Terminal bus/standard Ethernet LAN (TCP/IP)	
Total number of PCUs/PCU number range	A process cell can consist of a maximum of 64 PCUs. The PCU numbers may range from 1 to 239 and are unique throughout the process cell.	
Number of PCUs per server/area	16	
Number of clients per server	Sensible upper limit = 32	
Number of servers per area	1 server or 1 redundant server pair	
Number of areas per site	Max. 16 areas per process cell (site)	
"Route Control System" option (RCS / S7-400 only)		
Number of PCUs per area	16	
Number of simultaneous material transport operations (routes)	300	
Number of elements per calculated route	450 per PCU	
Control elements (CE)	1024 per PCU	
Sensor elements (SE)	1024 per PCU	
Parameter elements (PE)	1024 per PCU	
Link elements (LE)	1024 per PCU	
Locations/partial routes/process cells	Any	

*) Note on the number of operations per recipe unit procedure

The unit control recipes are saved in the PCU in TA data blocks. These are structured dynamically. Their size depends on the number of operations, the type of operation used and on the reference value for each operation. This is why there is no specified fixed maximum for the operations. The length of recipes is automatically verified when the recipe procedure is saved and when new operations are added.

Configuration limits for PA tags (operator control and monitoring)

Function	PCU type S7-400	PCU type S7-1500
Number of S7 connections per PCU	1	1
Number of data points	3700	5000
Number of data points per IOS server	16320	

Configuration limits for process data logging

Function	PCU type S7-400	PCU type S7-1500
Number of S7 connections per PCU	1 ... 4	1 ... 4
Number of data points per connection	3700 RT + 3700 ENG	5000 RT + 5000 ENG
Sum of data points per PCU	4 x 3700 RT + 4 x 3700 ENG = 29600	4 x 5000 RT + 4 x 5000 ENG = 40000
Number of data points per IOS server	255000	

2.7.2 Special characters in the object name

Core statement

The following characters must be avoided when assigning instance names in the system :

Special characters not allowed: , ; / ? * ~ # | @ > <

System installation and configuration 3

3.1 OS Installation

3.1.1 Installation of the operating system.

3.1.1.1 Workgroup and domain

Working environment

You have the option of operating the computers of your system in a workgroup or domain environment.

The following tables present a comparison of the advantages and disadvantages (from a technical perspective) as well as scenarios (examples) and their use in workgroups and domains.

Note

This information is provided as a basis for discussion to assist you in the selection of the appropriate environment for your computers. In addition to the technical criteria, also consider organizational criteria (for example, plant-specific IT, Microsoft expertise, resources, workload and maintenance, etc.).

Advantages and disadvantages

	Domain (AD; Active Directory)	Workgroup (WG)
General information (in relation to Microsoft Windows)	The user information is stored centrally.	Standard user settings are required at each PC station.
	Central safety database	Local safety database
	Suitable for very large networks (scalability)	Suitable for short distances in the plant and a limited number of PC stations.
	Permissions for devices and files are managed in the AD.	Permissions for devices and files are managed by individual PC stations.
Administrator	The administrator must have comprehensive knowledge (substantially higher qualifications).	The administrator must be familiar with the setup and management of PC stations in the workgroup network.
User accounts, group policies, resources, security	Central administration in the AD.	Distributed administration on each PC station.

	Domain (AD; Active Directory)	Workgroup (WG)
Planning	Comprehensive planning is required.	Less comprehensive planning is required.
Number of PC stations	Additional PC stations required to manage the AD. Windows server required to store central security information.	Windows server only required if this is used to set up the infrastructure, for example, as a DHCP server

DHCP server

Note

In BRAUMAT/SISTAR, the use of a DHCP server is only permitted with address reservation. All PC stations must work with permanent addresses.

DNS / WINS server

The recommended solution for name resolution is the use of a DNS / WINS server.

Scenarios

Scenarios (examples)	Domain (AD; Active Directory)	Workgroup (WG)
New user is added.	The new user is created centrally in the AD and is then available immediately on all PC stations.	The new user must be added on each PC station.
User password is changed.	The password is changed centrally in the AD.	The password must be changed on each PC station.
Add or replace computer	The settings are applied automatically in accordance with the group policies of the corresponding organizational unit.	The settings must be adapted locally on the PC station.
Modify security settings	The settings are changed centrally in the AD and distributed by means of group policies.	All settings must be changed locally.
Microsoft Updates are to be distributed.	The settings for the Microsoft Windows Software Update Service (WSUS) are made centrally in the domain.	The settings for the Microsoft Windows Software Update Service (WSUS) are made at each PC station.
Devices are added (e.g. printer, fax or scanner).	The devices are announced by AD.	Devices not announced. The information for connecting to the device must be known.
	The security settings can be made centrally for each device.	Each user that uses the device must be created locally at this PC station and log on with local user information.
One user uses multiple computers.	The user profile can be saved centrally. This applies the settings to all PC stations at logon.	The user has a local profile on each PC station.
	If the profile is not saved centrally, every user will have their own local profile.	

The lists above are not weighted in terms of importance; You must determine which advantage or disadvantage is the overriding factor for each system.

Domain example

Changes can be made centrally, fast and conveniently in the domain structure. However, the administration of a domain requires know-how, experience and additional work.

Workgroup example

In the workgroup, the majority of settings have to be made on the local computer. If a system is running with fixed settings, however, and few changes are expected, these settings only need to be made once. In this case, additional time for creating a domain and providing extra training for the operating personnel is not required.

Note

You have access to the following supplementary documentation via the PCS 7 website http://www.siemens.com/pcs7-documentation (http://www.siemens.com/pcs7-documentation):

- PCS 7 Security Concept; Recommendations and Notes
- SIMATIC Process Control System PCS 7; Security concept PCS 7 & WinCC - Basic document
- SIMATIC Process Control System PCS 7; Support and Remote Dialup
- SIMATIC Process Control System PCS 7; Patch management and security updates
- SIMATIC Process Control System PCS 7; Managing virus scanners

---

Integration in a domain

All preinstalled IOS Pcs are configured for a Windows workgroup by default.

If you want to include the preconfigured PC systems in a Windows domain, it must be ensured that the Windows group guidelines or domain restrictions are automatically replicated on the PC systems.

For this reason, this step must be agreed in advance with the domain administrator. In so doing, the domain policies must be checked and, if necessary, adapted to prevent disturbance of the system installation.

3.1.1.2 Notes on installing the operating system

Introduction

Only appropriately qualified personnel may install the operating system, particularly the servers, and configure networks.

Below you will information about how to make the settings.

Operating systems

You can find information on the operating systems in the "Readme" file for the current product version.

In the following you will find the most important information and settings for installation of the released operating systems.

Requirements

- All the necessary modules and devices have been installed in your PC.
- Check if you have a 32-bit or 64-bit version of the operating system and whether the PC station is suitable for the operating system.
- Check whether the operating system is suitable for the software that you want to install on the PC.
- Check that you have the product key. Example
  PRODUCT KEY: XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
  The product key is available in the following places:
  - On the cover of the installation data medium
  - When you purchase software online: In a confirmation e-mail
- Ensure that you know the name of the computer to be configured (name of the PC station in the plant).
- Back up the files of the PC station if you may need them later.
- The installation data is available on a bootable medium.
  The description below assumes that the installation is performed from a SETUP DVD.

Integration in networks

Observe the following information when integrating PC stations in a network.

Note

Install the PC stations locally or in a workgroup. Using this procedure prevents group policies or domain restrictions from hindering the installation.

The PC stations (IOS client, IOS server) themselves must not be used for managing domains (e.g. as DHCP server or DNS server), because the additional operating system services can interfere with the process control.

NOTICE
Software products of other manufacturers Only install software products of BRAUMAT/SISTAR named in this documentation on a PC station. If you install applications that are not released by Siemens and activate these applications in parallel, the system behavior can be negatively influenced. The user bears sole responsibility when third-party products are used. If you use third-party products, you must install them before the BRAUMAT/SISTAR installation. BRAUMAT/SISTAR and these third-party products must not be active simultaneously.

Reinstalling and updating

Note the following general information in regard to updating software.

> **NOTICE**
>
> **Important information**
>
> Please note the following:
>
> - If the suitable operating system has not yet been installed on the PC station yet, reinstall the PC.
> - Check the suitability of your hardware before performing a new installation. You can find additional information on this in the *"Readme"* file.
> - Back up your licenses (authorizations and license keys).
> - Save your unchanged project and associated data prior to reinstallation (in an image, for example).
> - Back up the data required for the software update.
> - Format the hard disks and create new partitions.

Operating system languages

We recommend the following Windows MUI operating systems (Multilingual User Interface) with the following languages and corresponding regional settings:

- German
- English
- Spanish
- Chinese

For use of BRAUMAT/SISTAR, it is necessary to set the desired target language and region consistently at all points in the language settings. This affects all the settings available under “Regional and Language Options”.

If you are using the BRAUMAT/SISTAR in Chinese, make the following settings in the Windows Region and Language options:

- For the "Language version of the non-Unicode programs“ select "Chinese (PRC)"
- For the "Language for menus and dialogs", select English if you have set the language of the system user interfaces to English.
- For the "Language for menus and dialogs", select "English" or "Chinese (simplified)" if you have set the language of the system user interfaces to Chinese.
- For the "Standards and formats", select "Chinese (PRC)".

If you change the "Language used in menus and dialogs", the character set may not be changed automatically in all cases. We therefore recommend that you go to Control Panel and select the "Windows Classic" style under "Display", "Display Properties", "Appearance".

Note

Of the operating system languages, BRAUMAT/SISTAR supports only the primary languages in each cases, for example, English (US), but not English (GB).

Specifying the name of the IOS station

Note

Change the computer name in Windows System Control.

Note that the computer name and the name of the IOS station must be identical in the configuration of an IOS.

For the computer name, select short and self-explanatory names, which already indicate the function of the PC station in the overall system.

Specifications:

- The computer name starts with a letter.
- The computer name only contains letters and numbers.
- The computer name has a maximum of 15 characters (limited by the operating system).

Preconfigured IOS station

The preconfigured PCs are supplied with a multilingual operating system.

The default language is English; the multilingual packages for German, Spanish, and Chinese are installed.

If you install a PC yourself, we recommend the following when selecting the operating system language:

- If the PC users only use one language, select the operating system language used for configuration/process control.
- If the users use multiple languages on the PC, select a multilingual operating system (English/USA by default) and then install the additional Multilingual User Interface Packs.

3.1.1.3 Hard disk partitioning

Partitioning the hard disks

If you are not using a preconfigured PC, but instead a different type, divide the hard disks of the PC into the following partitions to facilitate data backup.

- For the operating system and the BRAUMAT/SISTAR installation: At least 50 GB
- For project data: At least 100 GB, possibly several partitions
- For backups: Back up the projects and archives to media that will not be affected by a failure of the PC or the hard disk with the original data, for example, a DVD or on the network.

Additional information

You can find information about the minimum equipment of the basic hardware in the *"Readme*" document.

3.1.1.4 PC configuration security settings

Introduction

Current production methods increasingly require process control systems and the IT environment to come together. The need to connect company networks to public ones brings with it increased hazard potential. Information system and network security has therefore become an important factor in operating process control systems. Accurate knowledge of the sources of risks and the implementation of measures to prevent these risks from occurring is of utmost importance.

If you want to integrate PC stations of the process control system into a PC network, proficiency in the administration of Microsoft Windows networks is required. You must isolate the network via switches, routers or gateways to ensure that no disturbances reach the production plant via office networks, for example.

Important information about recommended security concepts and their realization in security solutions can be found in the introductory chapter 'Safety notes (Page 27)' at the start of this system manual.

Virus scanners and updates

- Approved virus scanners and updates can be installed later. You can find information about this in the *"Readme"* file.
- Security patches, hotfixes and service packs are software packages, which serve to correct security gaps, unwanted properties in software and similar.
- Siemens checks the compatibility of Microsoft Security Patches for the Microsoft operating system, the SQL server and Internet Explorer. You can find information about this in the "*Readme"* file.

Note

Please note that some updates will require you to restart your PC and cannot, therefore, be installed in process mode.

Additional information

- Section "Notes on installing the operating system (Page 71)"
- *"Readme"* file

3.1.1.5 Installing an operating system

Configuring the operating system

The operating system setup is largely self-explanatory. Below, you will information about the settings which you need to adjust.

- **Security**
  - Assign a secure password for the local administrator.
  - You should only make connections to secure networks when connecting to wireless networks. Make the system-specific settings to protect the computer.
- Specify and format **hard drives/partitions**
  You can find additional information on this in the "Hard disk partitioning (Page 75)" section. If you want to change the partitions, you may need to delete partitions and create new ones. Windows Setup shows the existing partitions.
- **Computer name**
  You can find additional information on this in the "Notes on installing the operating system (Page 71)" section.
- **Licensing the operating system**
  Make sure you have the correct number of licenses. Each server requires a server license and a system-specific number of client licenses (CALs).
  - Additional CALs are required, depending on the number of clients.
  - The CAL are not registered or recorded in the operating system.

Installation of specific language components for a multilingual installation

1. Open the **Control Panel**.
2. Select the "Change display language" option in the "Clock, Language, and Regional Options" area. The "Regional and Language Options" dialog box opens.
3. Open the "Keyboards and Languages" tab.
4. Click "Install/remove languages...".
5. Select the option "Browse computer or network".

6. Follow the Setup instructions for the installation.
7. Reboot the PC.

Settings made after installation of the Windows multilingual user interface pack

The selected language is English.

1. Open the **Control Panel**.
2. Select the "Change display language" option in the "Clock, Language, and Regional Options" area.
3. Open the "Keyboards and Languages" tab.
4. If a language pack is installed, you will see a "Choose a display language" drop-down list in the "Display language" area.
5. Select the desired language from the drop-down list.
6. Reboot the PC.

Windows swap file

The operating system requires additional hard drive space for the swap file (virtual working memory. This is created as standard on the operating system partition.

The online help for the operating system will explain how to open the dialog so to configure the virtual working memory

Rules for setting the size of the swap file

The size of the swap file depends on the PC's memory configuration. We recommend the following settings:

- For the memory space for the swap file, enter the following:
  Value: 1.5 to 2-times the working memory (RAM) installed in the PC.
- Enter the same value for the initial size and the maximum size of the swap file.

Windows Service Packs

If the operating system has been installed without a service pack, you have to install this subsequently.

Note

Refer to the *"Readme*" file to find out which version is required.

You should only install service packs if it is possible to restart the PC station. A service pack may automatically lead to a restart of the PC station or make a restart necessary for the amended functions to be used.

Recommendations for Windows options

Make the following changes in the Windows system settings:

- Deactivate games
- Disable "Automatic System Restore"

Other system settings

1. Set up the local users and groups via system control. You can find additional information on this topic in section "Creating user groups and users (Page 78)".
2. We recommend the following screen settings:
   – Select one of the following settings:
     - Use of standard monitors: Resolution at least 1280 x 1024
     - use of Widescreen monitors: Resolution at least 1680 x 1050
   – You can use the representation possibilities better with the higher resolution, e.g. for online operation of the picture blocks from the libraries. Ensure that you have installed the appropriate drivers and are using appropriate monitors.
   – Screen savers and energy savers should be deactivated on the PC stations which are being used for process operations.
   – For a Windows server, activate the background services optimization. To optimize the background services, in **System control > System > Advanced system settings** select the tab "Advanced".
3. Deactivate the energy saving option for network cards.
4. In the event display, check the that all services and drivers are functioning correctly

See also

http://www.siemens.com/automation/service&support (http://www.siemens.com/automation/service&support)

3.1.1.6 Creating user groups and users

Basics

The unique identification of persons and the allocation of rights to these persons are the most important security measures of any company, the logging-on of a person to a computer is an everyday, recurring task. The following describes the particularities of automation with regard to the management of role-based operator authorizations and their integration in the security concept of BRAUMAT/SISTAR.

Distinction between users and operators

When logging on to a process control computer, a distinction is made between a login to the operating system by a user and the login to the process control system by an operator. The user is authorized to use or administer certain process control applications; within these applications (e.g. "process diagrams" LzSys.exe) the operator has operator authorizations to make operator inputs to the automation system, thereby controlling the process. Rights and authorizations should only ever be granted where absolutely necessary. User and operators are only ever granted those specific rights that they require to fulfill their tasks.

Automatic user login

With process visualization systems, the permanent graphic representation of the process operation and the possible operation through it are in the foreground. After using the system, an operator logs out of an application of the process control software and normally does not end this application. This is in contrast to the logout of a user, who started this process control software. To close the graphic display and also to prevent the unauthorized launch of applications by operating personnel, automatic user login can be configured. The user used for this is not a real person, but instead represents the process control software of such a system and is therefore a sort of "Process control system account" (not to be confused with the system account of the operating system).

User rights administration

The administration of user rights in the Windows operating system is realized through the strategies recommended by Microsoft, in which individual users with the same tasks are combined into groups. The setup program creates a "BRAUMAT Group" or "SISTAR Group" group during installation, whose members (e.g. the above-mentioned "process control system account") can start and use the system applications.

It is advisable to use "non-personalized", device-specific user names the for user account or user accounts of the permanently used operator control and monitoring stations. Accounts are possible here, which can create a reference to the respective computer (e.g. 'IOSClient_5'). This account must be used when using "Auto login" to log in to the operating system. Person-based user accounts for each user/engineer are available for an engineering station, which is not permanently in operation but which is used by different users / engineers for configuring.

Operator rights administration

The operator administration of the process control system and its role assignment is configured using the "User administration" system application. The operator level range of the operator that has just logged in is compared with the function levels stored in the applications. The desired operation or entry is only possible when these are included in the operating level range.

For further details, see "Access authorization, Password (Page 154)".

Windows - User groups and users

For preparation of a secure network communication between the PC stations, the special Windows group "BRAUMAT Group" or "SISTAR Group" is created by the setup. The new users set up subsequently by the administrator must be added as members of at least this local Windows group if they want to use the system applications. This can also be done with the WINDOWS System settings (Page 90).

When installing SIMATIC NET on the Server PC stations, the following local user group is added to the user and group administration:

- SIMATIC NET

All users that work with BRAUMAT/SISTAR must be a member of this group.

Additional membership in the normal Windows user group is also necessary for local operation.

NOTICE
The following guidelines should be urgently observed as the basis of a secure operating environment: • Each active user account facilitates access to the system and is therefore a potential risk. • Reduction of configured / activated user accounts to the minimum actually required • Use of secure access data for the existing accounts • Regular check, in particular, of the locally configured user accounts • Deactivate the active user account "Guest" or "Anonymous user" (if applicable) after an operating system installation.

3.1.1.7 User accounts control

Basic principles

The Windows "User Account Control" (UAC) prevents potentially damaging programs from making changes to your computer by querying the user's permission, in a dialog field, to accept changes and continue or to cancel.

With Windows 7/Server 2008, the way, in which 'User Account Control' works can be set via a configuration dialog.

- In order to open the settings dialog for UAC, click on 'Start', enter the character string 'uac' in the text field Search programs and files.
- The "Settings ..." window is opened by clicking on the text found.
- For BRAUMAT, the **Standard** settings are recommended; these offer the best mix of protection and user friendliness:
  "Only inform if changes are made to the computer by programs"

3.1.1.8 Setting the user language (MUI)

Introduction

In BRAUMAT/SISTAR you can choose between the following languages for engineering and for the user interface language, depending on the languages installed:

- Chinese
- German
- English
- Spanish

Requirement

An operating system with a multilanguage user interface is installed.

Settings of the region and language options

- Regional settings
  Make the settings in accordance with the region your system is assigned to:
- Language for menus and dialog boxes
  Select the required language.

  Note

  The drop-down list only contains languages which are installed.

- Advanced settings for standard users (all new users):
  Recommendations:
  - Language for programs that do not support Unicode:
    The main language in which you want to operate the system.
  - Code page conversion tables
    The main language in which you want to operate the system.
  - Select the default settings for user accounts so that all settings are applied to the current user account and the standard user profile.

Restart

Reboot the PC station if you are prompted by Windows to do so after you have changed the language.

3.1.2 Installation of the product software

3.1.2.1 Setup program

Introduction

The *Process Control System; BRAUMAT/SISTAR V7.x.y* DVD contains the complete software for all the applications (IOS PC station, SQL DB host station).

Only certain applications have to be installed depending on the PC station. A general setup standardized for SIMATIC products will help you here.

Note

Complementary to the BRAUMAT control system for control and monitoring of brewing processes, the scope of delivery starting from Version V7.5 additionally contains an industry-neutral variant of the control system named SISTAR. By running the corresponding setup program in the respective subdirectory of the data storage medium, SISTAR can be installed alternatively to BRAUMAT.

Note

Paths for system and project data

- By default, the setup program uses the Windows 'Programs' directory as the system directory:
  - BRAUMAT → `'c:\Program Files (x86)\Siemens\Braumat'`
  - SISTAR → `'c:\Program Files (x86)\Siemens\Sistar'`
- The project can be located in any drive/directory that is independent of the system (preferably drive d:).
- The user must specify the active project with assignment of drive/directory using the "BM_Config.exe" configuration tool, which is available after installation (see Project folders and active project (Page 88)).

Licensing

You can transfer the license keys needed for system operation before or after the installation.

You can find a description of the license concept in the section "Licensing of software (Page 94)".

Preparing for installation

NOTICE
Installation requirements
For requirements and preparatory activities to be observed or performed before the installation, refer to the current Readme document. This is located in the base directory of the installation data storage medium.

Checking the installation requirements

The setup program automatically checks whether the software components needed for the installation are already installed on the PC.
Missing software components are indicated with a message. The setup program is terminated after the message is confirmed . You must install the missing software components. Then, restart the setup program.

Calling the setup program

1. Insert the *Process Control System; BRAUMAT/SISTAR V7.x.y* DVD into the DVD drive.
2. Double-click on "Setup.exe" in the product-specific subdirectory "BRAUMAT" or "SISTAR" to start the setup.
3. If applicable, an administrator login is queried in a dialog window of the User Account Control (UAC) in order to carry out the changes.

Settings in the setup

Below, you will find information about the settings made in the setup dialog boxes, listed in the order in which they are processed:

1. Setup language
   - For example "English" and "Next"
2. Welcome
   - ... "Next"
3. Product information
   ... Read this if needed and click "Next"
4. License agreement
   - ... Select the check box "I accept the conditions of the above license agreement as well as the conditions of the Open Source license agreement. I confirm that I have read and understood the security information".
   - ... "Next"

5. Setup type
   - Radio button: Install → New installation or upgrade from an earlier version (V7.0 and higher)
   - Update → Update the existing subversion (e.g. Service Pack)
   - Uninstall → Uninstall the current version
6. User information
   - The current Windows user is displayed. The user name can be modified individually depending on the project requirements and does not refer to existing Windows users. The organization name can be entered or left blank as desired.
   - Enter the user information and "Next".
7. Installation type
   - Activate the "Package installation" check box to install the preselected software package.
   - or alternatively: Select the "Custom installation" check box in order to install individual programs.
   - ... "Next"
8. Program package
   - The program packages are shown.
   - Activate the corresponding check boxes.

Pack	Description
BRAUMAT/SISTAR IOS ... • **System package**	**Applies to all IOS PC stations (server, client, single station, etc.)**
SQL Server • SQL system package • MES API package	Should be installed on a separate PC station with Microsoft SQL server.

   *) You can find information on the functions to be enabled for each PC station in the Readme document, section "**Licenses and configuration limits**"
   - ... "Next"
9. Programs
   - A blue check mark denotes the existence of the corresponding program.
   - For an update or ore-installation, select the desired pack again.
   - **The "Automation License Manager Vx.y ..." option is currently required only in connection with the "IOS system package" and should not be selected when a "SQL Server" option package is selected.**
   - If a package has been selected, the readme document with the product information can be opened. In the "Target directory" box, it is also possible to set a different target folder with the "Browse" button if the package is not yet installed.
   - ... "Next"

10. System adjustments
    - The Windows system settings / security settings to be made by setup are displayed.
    - Activate the "I accept the changes to the system settings" check box in order to implement the displayed settings.
    - **If this does not happen, these settings have to be made manually by a system administrator in order to ensure seamless client-server communication.**
    - ..."Continue"
11. Setup
    - The installation procedure starts.
    - The software installed on the PC for this program selection is listed in the display window.
    - The installation progress can be followed from the program icons.
12. Setup
    - Completion message for the installation operation
    - ..."End"
13. Restart the PC if required to do so by setup.

Display of installed software

You can check which software packages are installed on the PC station.

In the Start menu, select Programs → Siemens Automation → **Inst. Software**.
You will find the installed software packages in the "Products" tab.

See also

WINDOWS System settings (Page 90)

3.1.2.2 Windows settings

Windows firewall settings

A Firewall is activated when the Windows operating system is installed. In its basic state, this prevents incoming and outgoing network traffic for non-registered programs.

Note

After user confirmation, the setup program makes the necessary Windows firewall settings for the dedicated network traffic (PCU server, Trend Manager, SQL adapter) in the "System settings" step. The user does not have to make any further settings to do this.

Project paths and shares for Windows network connections

The system generally requires Windows shares ('network shares') for the project path and archive path on each IOS.

- The directories can be freely selected and can be set up on an (internal) hard drive or in subfolders (preferably drive /Partition 'D:').
- The folder name is prescribed and is "BM_PROJ" or "BM_ARCH".
- Starting with Version V7, you can find the `"BM_Config.exe"` configuration tool for setting up the paths in the installation directory of the system `'<sys-path>\sys'`. This configuration tool is automatically called when the system is started without an activated project. It must be manually called for changing a project path or for activating another project (for additional information, see WINDOWS System settings (Page 90)).
- In the "SiteCfg.exe" configuration tool, the folder share is shown as a 'tooltip' of the respective IOS (\\<Computer name>\BM_PROJ).
- The correct project paths of the IOS stations of an AREA can be displayed on the PCU server and verified in the view 'Display→Path list'.

Example using 2 server / 1 client PC stations:

	Client IOS "IOS21"	Server IOS "IOS1"	Server IOS "IOS2"
Local project path	D:\ProjectData \BM_project_brew-house	D:\ProjectData \BM_proj_BH_1	D:\ProjectData \BM_proj_BH_2
Display in	`\\IOS21\BM_PROJ` `\\IOS21\BM_ARCH`	`\\IOS1\BM_PROJ` `\\IOS1\BM_ARCH`	`\\IOS2\BM_PROJ` `\\IOS2\BM_ARCH`

Security settings of the project folder for local access and network access

Note

Settings for the release and NTFS rights with the configuration tool "BM_Config.EXE"

- The "User group" section shows the user group set up by the setup program as well as the current users.
- The "..." button can be used to open Windows User Manager in order to check and manually expand the user group.
- The Windows release and NTFS rights for the project path created are set automatically by the configuration tool, no further user actions are required for this.
- The following sections therefore described the desired state for the release and NTFS rights. This must be checked by the network administrator if there are network access problems.

NTFS rights

The members of the "BRAUMAT Group" or "SISTAR Group" user group must have the following authorizations for local access to the project directory:

Group / Authorization	for 'BRAUMAT/SISTAR Group'	for 'Administrators'	for 'Users'
Master level	Enable	Enable	
Change	Enable	Enable	
Read, execute	Enable	Enable	Enable
Display folder content	Enable	Enable	Enable
Read	Enable	Enable	Enable
Write	Enable	Enable	
Special permissions			

Release rights

The fundamental functions of a process control system include client/server communication via Windows folder releases and the corresponding network accesses via the UNC path system. So that only the users/groups authorized for and logged on to the process control system receive network access, it is necessary to create specific release security settings for the project path.

- The user group must be added to the share authorizations.
- The standard release right "everyone" must be removed.
- The user group must receive the following authorizations:

Group / Authorization	for 'BRAUMAT/SISTAR Group'	other users/groups	
Master level	Enable	<blank>	
Change	Enable	<blank>	
Read	Enable	<blank>	

3.1.2.3 Uninstalling the product software

If you no longer need the system on the PC station, you can remove the product software by performing the steps described below.

Recommendation

- Backup the existing License Keys with the "Automation License Manager" via the Start menu or Desktop icon on the original license data carrier (USB stick).
- Back up any project data you still need, for example, the complete project folder.

Procedure

Programs are uninstalled in Windows Control Panel.

- Close all system applications.
- Open the "Programs / Uninstall a program" entry in the category view or ...
- Open the "Programs and Features" entry in the icon view.
- Select the program that is no longer needed on the PC station.
  - BRAUMAT/SISTAR System V7.x
  - BRAUMAT/SISTAR SQL Adapter V7.x
  - Siemens Automation License Manager Vx.y
- Then start the "Uninstall" function.
- If applicable, an administrator login is queried in a dialog window of the User Account Control (UAC) in order to carry out the changes.
- Follow the instructions for removing the software.

3.1.3 Starting the System

3.1.3.1 Project folders and active project

Basic principle

Starting in product version V7, a basic separation of the system (all SW resources of the product including online help system, manuals, language resources, etc.) from the project (all configuration, engineering and process data of an automation project) was introduced.

Note

WINDCS folder name during system update to Version V7

- The common system/project directory WINDCS (=WINDOWS Digital Control System) known from earlier product versions has a different meaning starting in Version V7, because it only still contains project data.
- Several project directories can be kept on the IOS hard disk, in which case only one of them can be declared and activated as the current project.
- The project path can be freely selected and can be created on an (internal) hard disk drive either directly or as a subdirectory in a "Projects folder" (preferably drive/partition 'D:').
- In the case of redundant IOS servers, it must be ensured before the system start that the same or related projects are activated on both IOS servers. The folder names are not checked.
- For the above reasons, it is urgently recommended that the previous WINDCS folder be renamed during system updates. However, this renaming is not mandatory.

Check of project path and folder share

- At the system start (Start menu → "Siemens Automation / SIMATIC / BRAUMAT/SISTAR / Application Center" or double-click on desktop icon "Application Center") a check is made to determine whether a valid project path has already been specified and whether a Windows folder share exists for this path.
- If these requirements are not met, a corresponding notice appears and you are prompted to open the Windows system settings. This assists the user or system administrator in the case of multiple administrative one-time actions that are required for activation of a project.

Project ID

If two redundant IOS servers are connected over the network and inadvertently configured with different projects, the project on both servers could already suffer damage when the first server starts as a result of the activated server synchronization. To prevent this, the project ID has been introduced:

- Each project is identified by a unique project ID (example: BM_160307_175533_240FEB1E)
- A check is made in the system settings (BM_Config.exe) as to whether a project ID has been set. A project ID is assigned automatically when you create a new project.
- The project ID is distributed to all stations by the system configuration (SiteCfg.exe).
- You can also use the system configuration to generate a new project ID (only necessary when a project template was copied, for example).
- The system checks in the background whether the project ID is the same on all stations of a site. If this is not the case, the following security mechanisms go into effect:
  - Path monitoring blocks access via a network share → **synchronization is not possible**
  - The PCU server prevents coupling by IP connection → **operation over foreign IOS is not possible**
  - In path monitoring and the path list of the PCU server, **blocked paths are identified with 'ProjectID!'**.
    → Easiest way to check whether or not the access paths are blocked!

Note

Setup and distribution of the project ID

Requirement for fault-free operation is that the same project is active on all IOS stations. You can check this with the system configuration and correct it, if necessary (Area / Factory settings / Project ID).

In case of an upgrade, it is important to note that this protection initially does not exist. To set it up, a project ID should first be set on an IOS server in the system settings application (BM_Config.exe) and be distributed to all other IOS stations immediately using the system configuration.

If a new project is to be generated from an existing project (as template), a new project ID must be assigned on all stations in the system configuration (SiteCfg) on an IOS after activation in the system settings (BM_Config) and distributed to all IOS stations.

3.1.3.2 WINDOWS System settings

Basics

The "BM_Config" configuration tool is called on the 'Application Center' "Administration / System settings" tab.

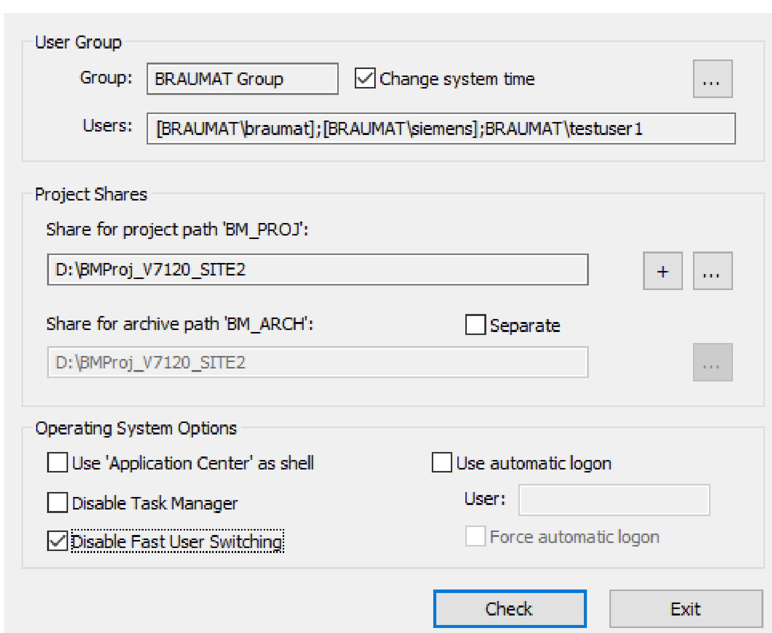
Directly after the new installation, it is called at the time 'Application Center' is started.

If applicable, an administrator login is queried in a dialog window of the User Account Control (UAC) in order to carry out the changes.

The tool enables the following administrative actions as preparation for the successful start of a project.

- Specification of the active project and archive folder
- Create Windows shares "BM_PROJ" and "BM_ARCH".
- Set up Windows file/folder rights (NTFS)
- Check the project folder and if necessary clean up system components from previous versions
- Setting of special Windows options

Dialog window:



User group and user

- The "User Group" field contains the Windows group "BRAUMAT Group" or "SISTAR Group" set up by the setup program. This cannot be changed.
- The "Users" field contains the associated user(s).

- Windows user administration can be opened via the button "..." . Users can be added or removed there.
- If the "change system time" check box is selected, the right to change the PC system time will be entered in the Windows group policies for the user group. This is required if the time synchronization between the IOS stations is to be carried out via system-internal mechanisms (see section System time synchronization (Page 118)).

NOTICE
Exiting the Windows user administration • The Windows user administration is an independent application, which has to be exited explicitly by the system administrator. • BM_Config does not control other Windows applications and therefore cannot close this automatically when exiting via the "Exit" button. • The system administrator has a particular responsibility here for the system security and must ensure the prompt closing of all administration applications and of the Windows session.

Project path and activation

- The current project and archive paths are shown in the "Shares" section.
- The "+" can be used to create a new project from a template project `'<sys-path>\!BM_Proj'` stored in the system directory with the installation.
  – This already contains a basic configuration with 1 x IOS1, 1 x PCU001.
  – The destination folder must be empty, as otherwise an error message is displayed.
  – All files and subfolders of the template project are copied to the destination folder.
- An existing project folder can be allocated via the button "...", which makes it the "active project". This also applies when upgrading projects from previous versions.

Archive path separated from the project path

- Archive data created during production (global logging, messages, step logs, trends) is accessed via the standard path "BM_ARCH".
- In the basic state, this is identical to the project folder addressed via "BM_PROJ".
- These two paths can be stored in separate folders, however, using the "Separate" selection box. In this case, use the button "..." to select the new folder.

BRAUMAT/SISTAR as shell, automatic logon activation, restrict operating system access

The following start options can be specified in the 'Operating System Options' section:

- Start Application Center as shell *1)
  - After Windows user login, the system starts directly, i.e., the Windows desktop and Start menu are no longer available.
  - As the sole access to the operating system, Windows Explorer can be opened using menu item "Program / Windows Explorer" by authorized users only (protected with password level "MENU_P").
  - The Windows user is logged out when Application Center is closed (protected with password level "MENU_E").
- Disable Task Manager *1)
  - Calling the task manager with "Ctrl+Alt+Delete" or the taskbar is blocked.
- Disable Fast User Switching
  - The fast change between Windows users is blocked. A user must first log off before the next user can log on. This option should always be active.
- Automatic logon activation
  If this option is enabled, the Windows desktop or the Application Center is started with the preset Windows user each time the PC is restarted.
  - Select the selection box "Use automatic logon"
  - following a further query, the Windows user account dialog is called
  - select the required auto-logon user and then disable the selection box:
    "Users must enter a username and password"
  - The corresponding password must now be entered twice
  - after a brief time the auto-logon activation becomes visible
  - Force automatic logon
    If this option is enabled in addition, the automatic new logon takes place immediately after logging off.

NOTICE
***1) Important requirement for enabling these options** • Before enabling these settings always ensure that the start operation of the computer, including the complete system start-up, has finished without errors (Application Center is available after startup) and that "BM_Config" can be called using the "System settings" application. • After making the change the Windows desktop and the Start menu can no longer be reached. • If you try to access the desktop in any other way, in the worst case scenario (no Windows Shell and no external network access) any access to the Windows registry may be blocked and incorrect start settings can no longer be corrected. The computer must be reinstalled in this case.

Check and adjust the active project folder

To work with the BRAUMAT/SISTAR applications, various subdirectories containing the respective configuration and runtime data are needed. From Version V7, only the project-specific configuration, engineering and process data are stored in the project folder.

Before starting, the subdirectories of the active project must meet certain requirements, for example, to eliminate system components from earlier versions and to avoid keeping duplicate data. Furthermore, a decision has to be made as to which of the two English language folders "texte.1" / "texte.10" apply or which text files are to be retained.

Procedure:

After specifying the active project path, the check is started via the button "Check" or when ending via "Exit". The process is performed in the following steps:

1. Clean up system files
   If system files (dll, exe, ocx, help, language resources, etc.) from a system version are found, you are notified of this and prompted to delete/clean up these files/subdirectories.
2. Specify user text "English"
   If there is an existing "texte.10" folder, another dialog window opens with the query "Replace all text files of language 1 ...", i.e. which of the two English language folders are to be transferred to the "texte.1" folder.
   - Yes→ Folder "texte.1" is replaced by the content from "texte.10" (English S88) "texte.10" is deleted
   - No → Folder "texte 1" is retained
3. Folder "texte.10" can be deleted via another dialog.
   If this is rejected, the configuration remains unchanged and inconsistent (when exiting from "BM_Config" with information box) and can be checked again subsequently.

The result of the cleanup process can be checked by opening the "Check" function again. If system files are repeatedly found, the process could not be completed because of file access problems.

In this case, check the following points:

- All system applications closed?
- No files opened in the project folder with other applications/editors (Notepad, Paint, etc.)?
- No folder in the project folder with file managers (Windows Explorer, xyzCommander, ...) opened?
- No folder/files opened via the network by other PC stations?
- Local file/folder rights (NTFS rights) of the entire folder tree sufficient for deletion?

Note

Cleanup of the project directory

- No relevant project files and, in general, no user files or subdirectories will be deleted.
- Risk-free cleanup is therefore possible when migrating from an earlier version to the current version.
- After the cleanup, the project directory can now only be used with the current system version.
- **Before performing the cleanup, the user himself must create a backup copy!**

Upgrade of earlier system versions

NOTICE

Important information

Due to new functions and properties, certain configuration settings and configurations must be checked and/or performed. The settings and configurations can be found in the latest readme document in the section **"Upgrading existing systems / IOS migration".** This document is located in the "_Manuals" directory of the installation data storage medium or – after installation – in system directory `'<sys-path>\help.nnn'`.

See also

System configuration 'SiteCfg' (Page 108)

3.1.4 Licensing

3.1.4.1 Licensing of software

Important terms

The following table lists terminology that is important for licensing:

Term	Description
License	A license provides the right to use products. This right is in the form of: • CoL (Certificate of License) • License key
CoL (**C**ertificate **o**f **L**icense)	The CoL certifies the license. The product may only be used by the license owner or authorized persons.
License key	The license key is the "technical representative" of the license (also called an "electronic license stamp").

License properties

Every license is made up of a basic license type and a license type.

The basic license type defines how many PC stations the associated software may be installed and used on.

Basic license type	Description
Single license	With this license, the software may be used on any single computer. The type of permissible usage is defined in the Certificate of License.
Floating license	With this license, the software may be used on any computer within a network. The software may be installed on several computers for this purpose.

The license type specifies a limit for the use of the associated software.

License types	Description
Unlimited license	With this license, the software may be used without restriction.
Count relevant license	With this license, software usage is restricted to: • The number of PCUs specified in the contract **Display:** The licensed number is displayed in the Automation License Manager under "Scope."
Upgrade license	An upgrade license can be used to convert an "old" version x to a version x +... license. **Note:** With the introduction of ALM licensing in BRAUMAT, upgrade licenses are only offered from version/subversion levels > V7.0.

ALM licenses Version V7 and higher

You will find the available ALM licenses and their properties in the current readme document. This is in the basic directory of the installation data medium or (after installation) in the BRAUMAT system directory.

Licenses prior to Version V7

With the introduction of ALM licensing, licenses (soft dongle, license diskettes) and project-based license files "Prod32.dll" from previous system versions are invalid. Any files from these previous versions still on the PC station after the SW update are ignored by the system.

Note

Backup of earlier licenses before update to Version V7

If existing IOS PC stations are to be updated, the following must be observed for a possible re-use or for requesting system update packs:

- Back up the soft dongle licenses of the previous version to the associated yellow original license diskettes using the "License management" application.
- Copy the license file `'...\windcs\sys\Prod32.dll'` to a data carrier of your choice

---

Missing license key / DEMO mode

When a software program requests a license key, but it cannot be located on the network in a "valid" format, it is described as a "missing license key".

You can use the system up to 4 hours in DEMO mode.

Note

- In the case of a missing or corrupt license key, the user can activate DEMO mode at system start by confirming a corresponding prompt.
- DEMO mode is displayed at the top of the application frame.
- In uninterrupted DEMO mode, system applications can no longer be started after 4 hours have elapsed.
- **The system is not permitted to be used in real process mode without a valid license, i.e. in DEMO mode.**

Automation License Manager

All license keys are managed in a central location using the *Automation License Manager* software that is installed by the system setup.

Virus-free

Note

Check that your PC is virus-free prior to each installation/uninstallation of a license key.

Viruses may pass between the hard disk and storage medium (license key USB stick, license key diskette) when using this method.

Write protection must not be activated when transferring license keys.

Additional information

- Manual *Automation License Manager*

3.1.4.2 How to transfer license keys

Introduction

The following license keys are transferred to your IOS PC station with the *Automation License Manager* software:

- License keys belonging to supplied licenses or licenses purchased at a later date.
- License keys stored at locations which cannot be accessed by a computer's applications

Possible license key storage locations

- License key USB stick
- License key diskettes
- Local storage media
- Storage media on connected computers
- Removable disks (for example USB stick, but not CDs or DVDs)

Note

Licenses for versions before V7 cannot be backed up on a license key USB stick or other USB stick.

Options for transferring license keys

The *Automation License Manager* gives you the following options for transferring license keys between the various possible storage locations:

- Drag-and-drop
- Cut and paste
- Offline transfer

Requirement

The *Automation License Manager* has been started up.

Procedure - drag-and-drop

1. Select the appropriate view using the menu command **View > Manage**.
2. On your own or the connected computer, open the storage location to where you wish to copy the license keys.
3. On your own or the connected computer, open the storage location from where you wish to remove the license keys.
4. Select the license keys, hold down the left mouse button and drag the license keys to the folder where you want to insert them.
   The license keys are transferred.

Procedure - Cut and paste

1. Select the appropriate view using the menu command **View > Manage**.
2. On your own or the connected computer, open the storage location from where you wish to cut the license keys.
3. Select the menu command **Edit > Cut**.

4. On your own or the connected computer, open the storage location to where you wish to copy the license keys.
5. Select the menu command **Edit > Paste**.
   The license keys are transferred.

Procedure – Offline transfer

The *Automation License Manager* online help describes how to perform an offline license key transfer.

Defective cluster on hard disk

Note

As part of the license key safety system, when license keys are transferred, clusters identified as "defective" are created on the target drive. You must not reconstruct these clusters, as this will destroy the license keys.

Backing up license keys

You can use the *Automation License Manager* to backup **all** license keys.

Read the information under "Possible license key storage locations" provided above.

3.1.5 Information about supplemental and help programs

3.1.5.1 Virus scanners

You can find information on virus scanners for the current system version in the *Readme* file (online version).

3.1.5.2 Burner software

Basic principles

In the interests of maintaining performance levels, data archiving devices should not be used on PC stations in process mode. We recommend that you only use data archiving devices (for example CD/DVD burners) on the IOS station (client IOS) envisaged for engineering .

3.1.5.3 Defragmentation programs

Using defragmenters

Defragmentation programs are used to optimize the time it takes to access a hard disk.

Note

Defragmentation programs that move fixed blocks can destroy license key files required for the authorization of software packages.

Recommendation:
Always exclude the license keys from editing before using the defragmentation program.

Excluding license keys from editing

You can exclude the license keys from editing as follows:

- Transfer all license keys to another storage medium, for example license key USB stick/ license key diskette.
- Transfer all license keys to a partition that is not defragmented, for example the security partition.

3.1.5.4 Remote diagnostics functions

Security requirements

If you want to perform remote diagnostics on a system, you must protect this system against unauthorized access.

Several steps have to be taken in order to implement a security concept. Only when taken together as a whole will these security measures offer the plant optimum protection.

Transmission paths

Data can be transmitted on the following paths:

- Telephone line (modem)
- Network connection (e.g. in-plant network TCP/IP connection)

Remote service and remote operation

For PC stations, we recommend the following tools for remote diagnostics and administrative access:

- **VNC**
  You can find information about the use of "RealVNC" in BRAUMAT systems in the Industry Online Support under Entry ID 55422236:
  Internet link http://support.automation.siemens.com/ (http://support.automation.siemens.com/WW/view/en/55422236)
- **RDP**
  Use of the Windows Remote Desktop Protocol (RDP) is permitted only for remote maintenance of IOS clients. In addition, no server services (e.g. OPC servers) may be active on these computers.
  The cause for this lies in the handling of the remote desktop sessions by the Microsoft operating system. The RealVNC software must be used for remote access to the other computers in a distributed system (e.g. IOS server).

Using the Remote Desktop Protocol (RDP)

- Access via RDP may only be gained by means of console takeover with the same user, or initial login ("/admin" or "/console" parameter when connecting)
- Administrator rights are required for initial login on a PC with a server operating system.
- The user must be a member of the "BRAUMAT Group"/"SISTAR Group" of the target computer. A **maximum of 1** user is permitted per target computer.
- Actions via RDP:
  – Hot restart/restart of the PC with corresponding logon
  – Setting up/configuration of computer

  Note

  Non-supported functions

  The following actions are not supported via RDP:
  - Multiple logons to the operating system
- Access to computers via RDP should only be temporary and not static.

Additional information

- White paper *PCS 7 Process Control System Security Concept PCS 7 and WinCC - Basic Document*
  (You can find information on availability in section "Safety information")
- Online help of the Windows operating system

See also

support.microsoft.com/kb/878451/en-us (http://support.microsoft.com/kb/878451/en)

Internet (http://support.industry.siemens.com/cs/ww/en/view/19292127)

3.1.5.5 Screen savers

Using a screen saver

Use of a screen saver on an IOS PC used to control production plants is **not** recommended for the following reasons:

- The screen saver uses CPU time and may therefore cause the system to be overloaded.
- The screen saver reduces the amount of RAM that can be used continuously. There are screen savers with which the amount of RAM used by the screen saver is not freed up.

If you use a screen saver, you should deactivate it during system operation

3.2 AS Installation

3.2.1 Creating an S7-400 user project

Core statement

The SIMATIC STEP 7 V5.x software package with the STL and SCL languages is used for configuring the application software for the S7-400 PCU type. Configuring in CFC (Continuous Function Chart) and SFC (Sequential Function Chart) analogous to PCS 7 is not intended.

Note

- If the AS blocks contain new functions (e.g. additional data record parameters of a particular class) after a system update and these new functions are to be used, it is not enough to install the associated OS system version (including new param.pcu file). In this case, the respective PCU SW version must also be completely upgraded to the OS SW version.

Creating a new user project

The following delivered AS projects for the S7-400 PCU type are available on the installation data storage medium:

- PCU Version V6 (system enhancements) → ...\S7Projects__Vx.y \BM_S7_400\BM_S7_400_V0600xxxx'
- PCU Version V7 → ...\S7Projects__Vx.y\BM_S7_400\BM_S7_400_V0705xxxx'

Open one of these STEP 7 projects with the SIMATIC Manager and save a copy under <MyProjectName>.

Adapting the HW configuration

- Delete all ASs that do not actually exist.
- Delete all PROFIBUS or PROFINET lines that do not actually exist.
- Next, adapt the hardware configuration in the hardware configuration manager so that it matches the actual configuration.

Note

Using other CPU types

- If another module (e.g. another CPU version, etc.) is actually going to be used, replacement should be performed, wherever possible, by dragging and dropping the new module from the hardware catalog to the default module slot. Do not delete the default module. As a result of the replacement operation, the hardware manager attempts to transfer all hardware parameterization data to the new module. The connection configuration between the AS stations (ASx ↔ ASy) in particular remains intact.
- The same applies when you need to move modules in the rack: Move, but do not delete. Unless the CP slot matches the actual situation, it is not possible to establish a PG connection via SOFTNET.

Copying blocks to the AS container:

- Copy all blocks from BM_SYS\Bausteine* (V7) or SIS_SYS\Bausteine* (V6) to all AS operated with this variant.
- Copy all blocks from BM_USR\Bausteine* (V7) or SIS_USR\Bausteine* (V6) to all AS operated with this variant.

Note

New project vs. update

Since BM_USR or SIS_USR always contain "empty" program/data templates to be configured by the project engineer, only one copy operation should be performed from this folder at the beginning of the configuration process (not e.g with an update). The same applies to all subsequent copy steps.

Copying the symbol table

- Copy <_DE/_EN>\Symbols_xxxxxx symbols into all ASs which are operated with this variant.

Route Control System option

The following libraries can also be installed as an option for RCS:

- Copy all blocks from BM_RCS_SYS\Bausteine* (V7) or SIS_RCS_SYS\Bausteine* (V6) to all AS's operated with this variant.
- Copy all blocks from BM_RCS_USR\Bausteine* (V7) or SIS_RCS_USR\Bausteine* (V6) to all AS's operated with this variant.

3.2.2 Creating an S7-1500 user project

Core statement

The SIMATIC STEP 7 TIA Portal software package is used for configuring the application software for the S7-1500 PCU type.

Note

- If the AS blocks contain new functions (e.g. additional data record parameters of a particular class) after a system update and these new functions are to be used, it is not enough to install the associated OS system version (including new param.pcu file). In this case, the respective PCU SW version must also be completely upgraded to the OS SW version.

Creating a new user project

The following delivered AS project is available on the installation data storage medium for the S7-1500 CPU type:

- PCU Version V7: → ...\S7Projects__Vx.y\BM_S7_1500\BM_S7_1500_Vxxxxxxxx'

Open this project in the TIA Portal and save a copy as <MyProjectName>.

Adapting the device configuration

- You can increase the "BM_S7_1500_Vxxx" CPUs to the required quantity in the device tree of the project tree using the "Copy" and "Paste" functions in the Edit menu. In so doing, all program blocks/PLC tags as well as the device configuration are also copied.
- Afterwards, the hardware configuration and the configuration of every CPU, especially CPU type and interface parameter assignment/TCP-IP addresses, must be adapted to the real hardware installed.

Note

Released CPU types

The following CPU types are released with this product version (for ordering data see Readme document):

- CPU 1518-4 PN/DP→ no configuration limits
- CPU 1516-3 PN/DP→ subject to configuration limits

Changeover to another CPU type in the user project

- The delivered project is supplied without configuration limits only for CPU type CPU 1518-4 PN/DP.
- If a different module (e.g. different CPU version, etc.) is used in the user project, it can be interchanged with the "Change device" function. When the module is interchanged, the Device Manager attempts to transfer the parameter assignment to the new module. In particular, the configured connection between the AS stations in "Devices and networks" is retained.
- In addition, it may be necessary to make several adaptations in the program blocks to reduce the configuration limits (see section further below).

Buffering in case of power failure

The retentive memory allocated to the complete system program of the delivered project is greater than the retentive memory volume provided by the CPU alone. To prevent data loss in the event of a power failure, use of the special **backup power supply "PS 60W 24/48/60VDC HF" is strongly recommended**. This is already taken into consideration in the device configuration of the delivered project.

Structure of the delivered project

A brief description of the contents of the delivered project is given below. Error-free operation is only guaranteed when all program resources are fully applied to the user project. **The System / User column provides an indication of which program resources may be changed by the user!**

Project tree/devices	Description	System / User
Device configuration	included in delivery state 1 x CPU 1518 1 x Backup power supply	User - Expansion and parameter assignment with respect to net-working, I/O devices and mod-ules
Program blocks \ BRAUMAT/ SISTAR		
...\!System\...	Group folder for runtime system	System - no user configuration
...\...\Communication	Communication and routing blocks	System - no user configuration
...\...\Configuration	System control data block	System - no user configuration
...\...\Control recipe data	Control recipe data blocks	System - no user configuration (Exception: deleting instances for reducing configuration limits)

Project tree/devices	Description	System / User
...\...\Miscellaneous	Diverse system functions	System - no user configuration
...\...\Runtime environment	Organization blocks (cyclic interrupts, error handling) and sequential control	System - no user configuration
...\...\Technology	Processing of technological system classes	System - no user configuration
...\...\Terminal blocks	System-internal routing data blocks	System - no user configuration
...\Data containers	Group folder for class instance data	
...\...\Technology	Engineering and runtime data of the technological system classes	The online blocks in the CPU work memory contain the engineering data of the OS configuration applications. Reading/writing of runtime data in the user program
...\Runtime interfaces	Group folder for runtime interfaces	
...\...\01_Environment	I/O signal routing, cyclic processing	User programs
...\...\02_Control modules	ICM interlocks	User programs
...\...\03_Miscellaneous	User interfaces: Unit calls, EOP block calls Batch order start in the AS Measured value conversions Setpoint curve DBs	User programs
...\...\Sample process cell 1	Group folder for units/EOP functions	
...\...\...\!EOP template	Program template for an "Equipment operation" function (max. 1999 EOPs possible)	User programs
...\...\...\Sequence xxx	Program templates for max. 128 unit functions	User programs
Program blocks \ system blocks	Standard blocks used by the system	System - no user configuration
Technology objects	Not used	
External sources	Not used	
PLC tags	Group folder	
...\BmInterfaces	Standard system flag interfaces	Use in the user program
PLC data types \ BRAUMAT/SISTAR **...\!System**	Type definition of technological system classes	System - no user configuration

Reduction of configuration limits for CPU 1516-3 PN/DP

By analyzing the memory utilization and testing the cycle time load, the CPU data work memory was identified as the primary limiting element.

Consequence:

When the number of units and tanks is limited to a maximum of 32, the requirements for memory usage and cycle time load can also be met with the smaller PCU type CPU 1516-3 PN/DP.

Note

Usable sequence / tank instances with reduced PCU configuration limits

As a result of the procedure described here, only the following system resources can be used for this PCU:

- Parameter assignment of sequence class (1 ... 32)
- Parameter assignment of tank class (1 ... 32)
- TIA Portal project / program blocks ...\Runtime interfaces\Sample process cell 1\Sequence xxx
  BmUsrSeq001FC [FC3001] ... BmUsrSeq032FC [FC3032]

Based on the delivered project (as described above), the following steps are required for this:

1. In the device configuration for the CPU, select the following type using the "Change device" function:
   CPU 1516-3 PN/DP 6ES7 516-3AN01-0AB0 (FW Version V2.5 or higher)
2. In the program blocks in the ...\!System\Control recipe data folder, delete the following data blocks
   BmControlRecipeSeq033 [DB533] ... BmControlRecipeSeq128 [DB1628]
3. In the program blocks in the ...\Data containers\Technology folder, change the SEQ [DB725] block as follows:
   Set static tag `s_uiRecordCountAct` start value = 32
4. In the program blocks in the ...\Data containers\Technology, folder, change the TANK [DB2100] block as follows:
   Set static tag `s_uiRecordCountAct` start value = 32

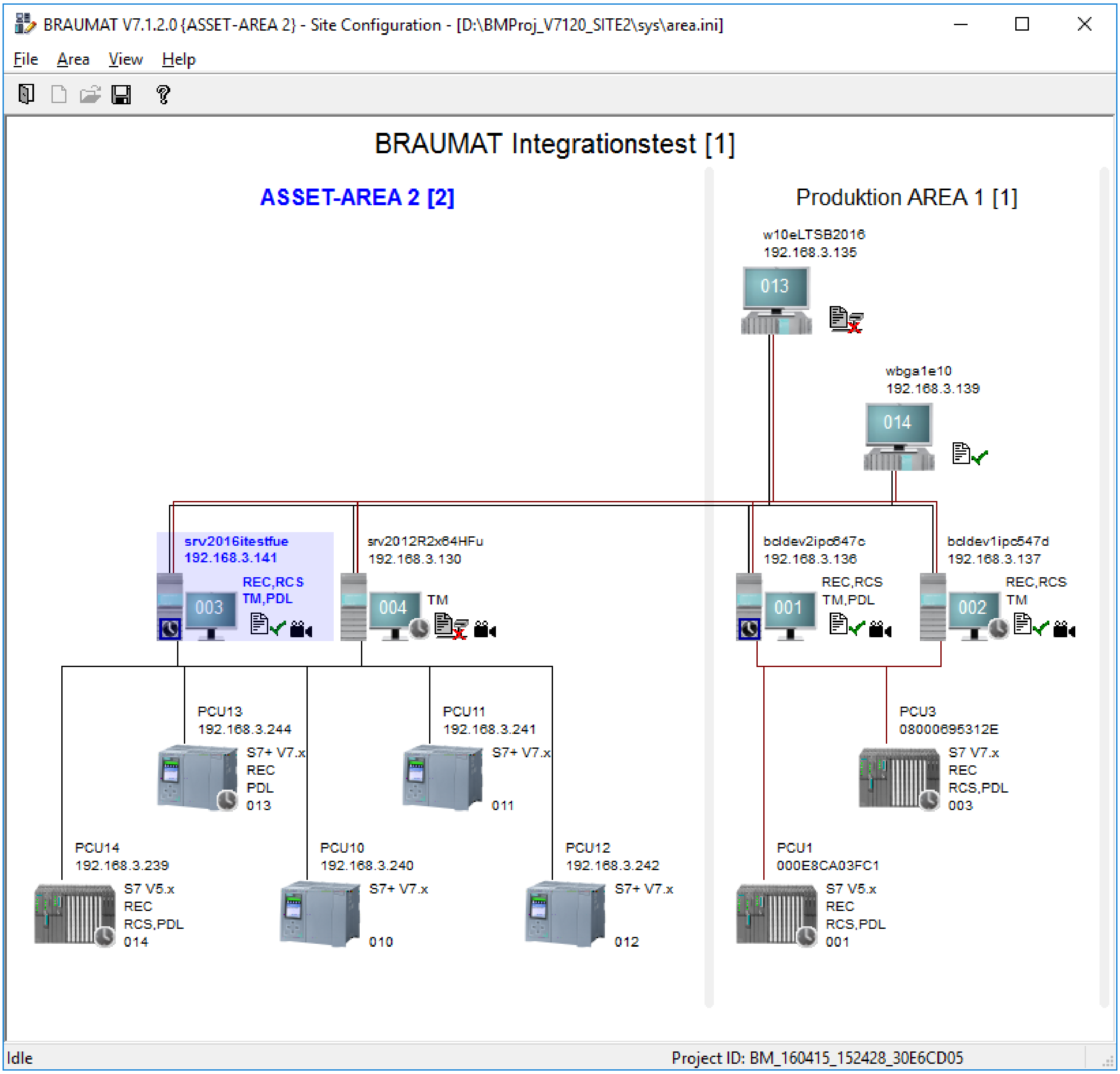
3.3 System configuration 'SiteCfg'

3.3.1 Introduction

The sections below describe the configuration tool "SiteCfg.exe", which supports users when creating, converting, modifying and distributing the area.ini.

- To create a new configuration use the menu item File →New (after first closing the current view).
  A basic structure for Area 1 is now created. This configuration must be enhanced with detailed parameters for the corresponding system.
- An existing configuration is opened with the menu item "File →Open" (after first closing the current view) and then by selecting the configuration file `'<proj-path>\sys\Area.ini'`.

The application interface is shown in the following screenshot:



3.3.2 Modifying Plant settings

Select Area>Plant settings to start the dialog for editing the factory settings.

Register Tab General:

Element	Meaning
No.	Factory number
Name	Factory/Plant name
Project ID	Unique identification of the project; used to check the file access paths between IOS stations of a project "Generate new project ID" button: To generate a new project ID by means of an additional query. The new project ID must be distributed to the other IOS stations in a second step.
Title bar:	Text displayed in the title bars or the dialog windows
Engineering languages:	• Default: "(System)" Applies to the use of a **fixed** engineering language regardless of the current system language. The engineering texts are stored in the directory "`<proj-path>\PCU.nnn\Texte\...`". • Default: <any language> Defines the default engineering language when using **multiple** engineering languages regardless of the current system language. The table shows the assignment of the engineering language to the particular system language. If this contains the entry "(Default)", the language from the "Default" box applies. The engineering texts are stored in the directories "`<proj-path>\PCU.nnn\Texte.x\...`".

Note

The project ID is mandatory!

It is mandatory to assign the unique identification by means of a project ID when upgrading. An empty project ID or one that does not match blocks the PCU server connections between the IOS stations.

Register Tab S7

Element	Meaning
Use CRC:	**This option must be selected when using BRAUMAT Lean on IPC427D Microbox RTX**. Otherwise, the option applies to existing systems in combination with S7-H machines for backward compatibility purposes only.
Use S7H:	This option applies to existing systems in combination with S7-H machines for backward compatibility purposes only. **The "Process data logging" option (replay mode) is not enabled when using S7-H machines.**
S7 mnemonic:	Language for the S7 mnemonic • (System): following runtime language • German • English

3.3.3 Area settings

Core statement

In BRAUMAT one area has the following properties:

- One area has one or two (then redundant) recipe and (or) Route Control Servers.
- Clients are assigned to one area.
- PCUs are assigned to one area.
- The servers of the area have direct communication connection to the PCUs of the area.
- A data server of a different area cannot be used as redundancy IOS of the area.

Adding an area

Using the "Area / Insert area" menu item, you can add a further area.

Deleting an area

You can select the "Delete" command from the shortcut menu of the area to delete the current area.

You can open the shortcut menu of the area by clicking in the area between the corresponding IOS and PCU objects.

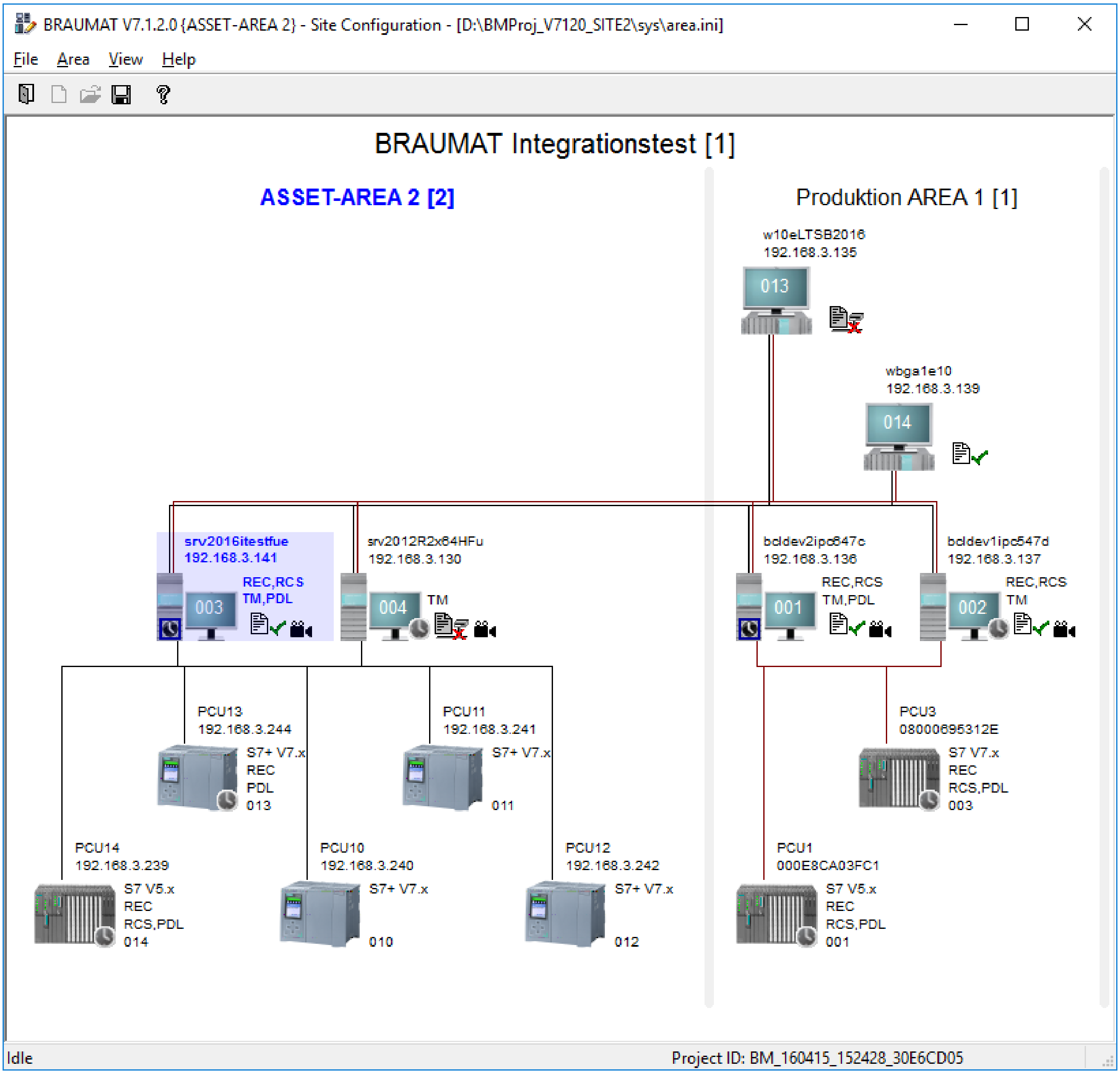
Changing the area parameters

With menu item "Area / Parameters" you can modify the parameters of the current area.

The dialog contains the section "Parameter used for batch creation" with the following entries:

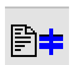
Dialog element	Function
No.	AREA Identifier: is assigned once when you create the AREA
Name	AREA Name: is assigned when you create the AREA and can be changed later.
Parameters for batch formation: "Starting time of calendar week"	Used for the automatic batch start.
Parameters for batch formation: "Starting day of calendar week"	Used for the automatic batch start.
Process data logging: "Storage period"	Default value = 90 days; this specifies how long the data archive ("`<arch-path>\LOGGING\DataLog\...`") for **"replay mode"** is kept. Older archive data is deleted on a daily basis automatically by the systerm!

3.3.4 Display of the connection configuration



- The local IOS (own PC station) and the associated AREA is viewed in blue font type
- The different area interconnections are shown in different colors.
- The most important configuration information is displayed above or beside the respective station

The state of the Area.ini (right beside the IOS) compared to the actual configuration is indicated as follows:

	saved Area.ini is the same as actual configuration
	No Area.ini found
	saved Area.ini is not the actual configuration
	Access to this IOS/folder denied

When saving the configuration the following options are available in a separate dialog:

- "Update all stations" → the local configuration is copied to all IOS stations, while for each IOS the network path is displayed and the overwrite has to be confirmed
- "Checking PCU subdirectories" → the PCU subdirectories of the local or all IOS servers are checked and created – if not existent – by copying the …\pcu.xxx\... folders
- "Reset local system" → after a user prompt and confirmation dialog, the BRAUMAT system on the current IOS is shut down completely. You can restart using the "Application Center" desktop icon or using the Windows Start menu.

3.3.5 IOS settings

3.3.5.1 Inserting an IOS

You can add another IOS station with the menu item "Insert new IOS" in the shortcut menu of the area or in the "Area" menu. Right clicking in the the free space of the area opens the shortcut menu.

- The dialog window "Add new IOS" opens.
- After entering a valid IOS no. (range 1 ... 255) and selecting an IOS type (client, server 1 or server 2), the OK button is enabled.
- The IOS configuration dialog opens after you have entered this information.

3.3.5.2 Modifying IOS settings

Basics

Using the "Settings" menu command in the shortcut menu of the IOS station, you can change the parameters of the current IOS.

The IOS settings dialog makes the following tabs available:

- General
- Server sequences tab
- Remote paths (server IOS only)

"IOS settings / General" dialog:

Dialog element	Function
TCP/IP-Adr:	Address for client/server connections.
Autostart:	When the system is started from the shell, all comma-separated applications in the list are also automatically started.
Enable ‘recipe control’:	This computer can operate as a recipe server.
Enable ‘route control’:	This computer can operate as a route control server.
Enable ‘change logging’:	This computer supports the change logging function
Enable ‘maintenance supervising’:	This computer supports maintenance supervising
Enable ‘pipe entity’:	This computer supports the ICM maintenance groups
Enable 'trendmanager'	The trend manager (`'trendman.exe'` application) is started along with the system. This dispenses, e.g. with: • Starting from "Autostart" • Starting from "Scheduler"
Enable 'SQL adapter'	The SQL adapter (`'sistar_adp.exe'` application) is started along with the system. The option can only be activated with a valid license.
Enable 'Time-of-day master'	This computer has the function "time master IOS". • Only a **maximum of 2 redundant IOS servers of an AREA** should enabled for this function. One of the two can also be configured as a time slave IOS (important if this one does not have its own DCF/Sat. hardware clock). • All other IOS servers should be configured as time slave IOS.
Release 'process data logging'	The communication port "S7 Logging Port" for data logging that is required for "**Replay mode**" is started along with the system.
Option: Replay mode	**"Replay mode"** is activated for this IOS.
Option: Retrieve system time from time master	This computer has the function "time slave IOS". • Function on IOS client: obtains the system time from its own preferred AREA server • Function on IOS server: obtains the system time from the time master IOS of the other AREA. There must be a multiclient relationship with this master. • Function with time master IOS: obtains the system time from the redundant AREA server
Option: Use S7 redundancy address:	If selected, this computer uses the addresses of the redundant network for PCU access.

"IOS settings / Server sequences" dialog:

Here you define server access for the current IOS. Each AREA for which the IOS can be a multiclient appears in the AREA list box. The following parameters can be defined for each connected AREA:

Default:	By selecting the list box, the standard sequence for connection establishment and file access to the relevant server of the selected area is specified. Default entry is the server sequence of the area.
RECIPE, RCS:	By enabling the list box, a separate access sequence can be defined for each path category, that overrides the default access settings.

"IOS Settings / Remote paths" dialog:

If the default access to the system path of the server IOS is not desirable in specific configurations you can define a separate remote path for path categories. Such an entry overrides the system path/server sequence.

The "Remote paths" tab is only visible for server IOSs.

Default:	For file access by means of server sequence, the default setting shows the specified system path of the server IOS.
GLOBLOG:	Remote path for change logs
MSG:	Remote path for messages
STEP:	Remote path for step logs
TREND:	Remote path for trends

See also

Basics (Page 513)

3.3.5.3 Assigning multiclient areas

You can select the "Multiclient Areas" command from the IOS shortcut menu to assign the current IOS to the areas you want to enable for multiclient applications.

The "Allocate areas" dialog is displayed.

Table 3-1 Description of dialog elements:

Assigned:	List of areas enabled for multiclient logon.
Not assigned:	List of areas locked for multiclient logon.

Double-click the area name or "⇔" to enable/disable the selected area.

Note:

The local IOS area always remains assigned!

3.3.5.4 Assigning a local area

You can select the "Assign local area" command from the IOS shortcut menu to assign the current IOS to another area group.

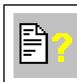
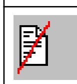
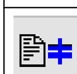
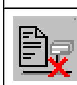
The "Allocate local area" dialog is displayed.

Double click an area name in the "Not allocated" field to select this area as another area (IOS with multi-client function).

3.3.5.5 Comparing the area.ini of the IOS

You can select the "Compare Area.ini " command from the IOS shortcut menu to verify that the IOS can be accessed on the network and to check the status of its local Area.ini .

The bitmaps below represent the status of the local Area.ini of the selected IOS, compared with the Area.ini last opened or saved in Configtool.

	Status not verified
	Local Area.ini is the last saved version
	No Area.ini found
	Local Area.ini is not the last saved version
	Access to this IOS/folder denied

3.3.5.6 Update area.ini

You can select the "Update Area.ini " command from the IOS shortcut menu to copy the last edited Area.ini into the system path of the selected IOS.

Note:

On the destination IOS the last Area.ini is saved as a *.bak file!

3.3.5.7 Changing the IOS ID

You can select the "Change ID" command from the IOS shortcut menu to change the ID of the current IOS.

3.3.5.8 Configuring the SQL adapter

'BRAUMAT SQL adapter IOS' option package

With correct licensing of the option, the menu command 'Area / IOS... / Configure SQL adapter' can be opened.

Note

The description of the sections and fields of this dialog is available in the "SQL-DB Host" manual, section "Installation and configuration / Enabling the functions".

3.3.5.9 Client/server configuration

You can select the "Configure as client/server" command from the IOS shortcut menu to configure server mode for a client or vice versa.

3.3.5.10 Deleting an IOS

You can select the "Delete" command from the IOS shortcut menu to delete the current IOS.

Note:

You must delete the complete area to delete its last server!

3.3.5.11 System time synchronization

Basics

In a process control system, it is an absolute necessity that the current date and time of day are synchronized on all IOS and PCU stations that exchange process data with each other. This is the only way to ensure precise telegram assignment. System-internal functions for time-of-day synchronization are used for the monitoring and time synchronization, which is controlled via the communication relations defined in the "SiteCfg" configuration tool and activated for the respective station.

Hierarchical concept

In terms of an entire system, the time-of-day synchronization is performed with several AREAs based on a hierarchical concept.

1. At the highest hierarchical level, one or two (redundant) IOS servers take over the time master function.
   - A single time master IOS **should** have a radio or satellite supported hardware clock although this is not absolutely necessary.
   - When using two time master IOS stations, these **must** be the two IOS servers of the same AREA.
   - The second time master IOS **must** be synchronized with another time master IOS if it does not have its own radio or satellite supported hardware clock.
2. At the next level, the other IOS servers of the system obtain their time of day from a time master IOS.
   - These IOS servers must have a multiclient relationship to the time master IOS station(s).
   - The same rules apply to prioritization as with other communications; in other words, the system selects the server defined as the "Preferred Server" in the "Server sequences" of the AREA of the time master IOS stations.
3. At the lowest level, IOS clients or PCU stations can obtain their time of day from their own AREA server. With redundant IOS servers, the system automatically selects the IOS server configured as the "Preferred Server" in the "Server sequences".

Activation and display

To activate the system time synchronization, for all IOS stations involved, the users brought together in the "BRAUMAT Group" user group must be authorized to change the PC clock. This authorization is granted by selecting "Change system time" in the "System settings" (see section WINDOWS System settings (Page 90)).

The "Configuration" application is used to activate the time synchronization in the respective IOS station.

- Modifying IOS settings (Page 114)
- Modifying PCU settings (Page 121)

In the graphic full overview of the system configuration, additional icons appear for each station configured for system time synchronization and these indicate the activated functions. This makes it both quick and simple to check that the system time synchronization of the entire system is complete.

Symbol	Function
	Time slave • Possible for IOS client, IOS server, PCU • Station obtains time of day from a time master IOS
	Time master • Possible with a maximum of 2 redundant IOS servers of the same AREA • One of the two time master IOS stations can **also** be configured as a time slave. This is necessary when it does not have its own DCF/Sat. supported hardware clock.

Note

Administrator user blocks the time-of-day synchronization.

Due to a Windows property, the setting of the PC clock via time-of-day synchronization fails when the logged in user has administrator rights. The users set up for normal system operation should therefore not be members of the administrator group.

Update rate of the system time synchronization

The system time monitoring and synchronization on the IOS stations is only active on a started system.

- At a time discrepancy of >= 15 sec, the time synchronization occurs immediately
- At a time discrepancy of < 15 sec, the time synchronization occurs once per hour

3.3.6 PCU settings

3.3.6.1 Insert new PCU

You can select the "Insert new PCU" command from the server IOS shortcut menu to add a further PCU to the relevant server IOS.

- This opens the "Insert new PCS" dialog box.
- The OK button is not enabled until you enter a valid PCU no. (range 1 ... 255).
- The dialog for assigning PCU parameters opens after you have entered this information.

3.3.6.2 Modifying PCU settings

Core statement

With the Settings menu command from the PCU shortcut menu you can edit the parameters of the current PCU.

The PCU settings dialog makes the following tabs available:

- General
- Coupling
- Software
- FIFOs

"PCU settings / General" dialog:

Table 3-2 Description of dialog elements

Dialog element	Function
Name:	PCU name
AS type	• S7-400 • S7-1500
Enable: Recipe control	This PCU is relevant to the recipe server.
Enable: Route control	This PCU must be taken into consideration by the route control server **(S7-400 PCU only)**
Option: Retrieve system time from server	This PCU obtains its system time from the preferred AREA server
Option: Process data logging	The data logging for the **Replay mode** via the "S7 Logging Port" of the PCU server is activated for this PCU.

"PCU settings / Coupling" dialog:

Table 3-3 Description of dialog elements

Dialog element	Function
Slot:	Slot number in the rack **(S7-400 PCU only)**
Coupling type:	Because of the interface parameterization selected in the Simatic NET configuration dialog 'Set PG/PC interface' (Windows system control) for the access point 'S7ONLINE', one of the following coupling types must be selected: • Ind. Ethernet **(S7-400 PCU only)** • MPI **(S7-400 PCU only)** • TCP/IP

Dialog element	Function
Address:	Depending on the selected coupling type, the associated MAC, MPI or TCP/IP address must be entered.
Redundancy:	Addresses used to configure a redundant network. Requirements here are: • 2 communication groups CP443 in the AS. • Only the servers configured for this mode ("General" tab in "IOS settings") evaluate these addresses.

"PCU settings / Software" tab dialog:

Table 3-4 Description of dialog elements

Dialog element	Function
Base:	PCU type S7-400: Selection of the installed PCU system version *) • S7 V4.x • S7 V4.6 • S7 V5.x • S7 V7.x • PCS7 → PCS7 option is valid only in existing systems for downward compatibility
Recipe control:	Selection of the recipe version for PCU type S7-400: • V3 is specified with S7 V4.x, S7 V4.6 • V5 is specified for S7 V5.x • V7 is specified for S7 V7.x
Cross-coupling:	Selection of the cross coupling for PCU type S7-400 • V1 is specified for S7 V4.x • V2 (extension for 32 S7-S7 connections) specified for S7 V5.x, S7 V7.x
Technology blocks:	• only possible with the earlier PCS 7 option (note see above)
Route control	• Only available for PCU type S7-400 • V5 specified, available only as of V4.6

Note

***) Compatibility with earlier PCU system versions**

The BRAUMAT IOS can be operated both with the current and with earlier PCU system versions of the "SIMATIC S7" AS.

This means that it is not necessary to upgrade the PCU however the new technology functions of the BRAUMAT Version V7. cannot be used.

"PCU settings" dialog box, "FIFOs" tab:

Table 3-5 Description of dialog elements

Dialog element	Function
Parameter assignment during server startup:	When activated, the transfer of the FIFO assignment configured here to the PCU takes place automatically when the PCU server is started: • If only the recipe control is enabled, ("General" tab), all FIFOs of this PCU are loaded from the recipe server • If the route control is also enabled, FIFOs 1 and 4 of this PCU are loaded from the RCS server **(Important: The PCU must already be configured in the RCS engineering)**
FIFO1 (RCS):	Telegram types via FIFO 1 to server1 *) **Default setting: 22 ("Tankinventory")**
FIFO2 (recipe):	Telegram types via FIFO 2 to server1 **Default setting: <empty>**
FIFO3 (general):	Telegram types via FIFO 3 to server1 **Default setting: 3, 5, 6, 18, 19**
FIFO4 (RCS):	Telegram types via FIFO 4 to server2 **Default setting: 22 ("Tankinventory")**
FIFO5 (recipe):	Telegram types via FIFO 5 to server2 **Default setting: <empty>**
FIFO6 (general):	Telegram types via FIFO 6 to server2 **Default setting: 3, 5, 6, 18, 19**

*) Description of telegram types, refer to the section FIFO1 to FIFO6 - PCU system data general (Page 286)

See also

Basics (Page 513)

3.3.6.3 Deleting a PCU

With the "Delete" menu item in the shortcut menu of the PCU or in the "Area / PCU..." menu, you can delete the selected PCU in the configuration.

Initially, this does not affect the corresponding project folder "<proj-path>\Pcu.nnn".

Only if the "Checking PCU subdirectories" option is selected in the following "Save" dialog can this project folder be deleted following renewed confirmation.

3.4 AREA-Selection dialog inside the applications

3.4.1 AREA-Selection on application-start

On starting the multiclient-capable application the following dialog-sequence takes place:

- First, a selection dialog appears with the projected AREAs
- Selection of the area (default = last area selected) and OK button starts the application and connects to the respective AREA server.
- Selection of the option box "Hide dialog" saves the area-selection and the dialog is hidden for all applications from now on
- Activating the dialog once more takes place via <Ctrl>+double-click on the application icons

3.4.2 AREA-Selection on file open

In all applications that start with empty Workspace per default, the area selection dialog occurs first during the file opening.

Notes:

- Default setting before an AREA reference was selected is the own AREA, later the at last selected AREA, that means the AREA selection is stored.
- In the case of several opened instances of an application the at last opened determines the active AREA.

3.5 PCU Server

3.5.1 General

Core statement

The application PCU Server is responsible for the coupling between AS and OS or OS and OS. This application should run on every Server or Client.

The application is automatically started when BRAUMAT is started.

The PCU Server has the following tasks:

- Communication functions
  A central interface is provided via Windows Sockets and TCP/IP for the OS-OS (= Client/Server) communication
  A central interface of the subordinated standard communication system SIMATIC NET is provided for the AS-OS communication. This provides the coupling types "ISO-Industrial Ethernet", "TCP/IP" or "MPI".
- Server functions.
  The server functions are connected as so-called 'Miniports' to the PCU server, through which the PCU server is enhanced functionally depending on the IOS station type. The Miniports are implemented as Windows DLLs.

Note

Configuration file "PCU_SERV.ini"

The respective miniports are activated and configured automatically with the IOS settings in the configuration tool 'SiteCfg'. In this respect, from Version V7 the entries are no longer available in the configuration file "PCU_SERV.INI" and do not have to be checked.

PCU server security: Monitoring of the TCP/IP addresses of server and client

The PCU server and the Trend Manager only accept incoming Windows Socket connections from the configured TCP/IP address. The address must match one of the LAN adapters of the IOS server. If a second LAN adapter with in-house network connection (or even Internet connection) exists in addition to the actual terminal bus adapter, which is prohibited according to the security recommendations, an "unwanted" client connection over the second LAN adapter is prevented through the following measures:

- The TCP/IP ports on the server are only opened for the local TCP/IP address configured in the configuration (SiteCfg).
- The IP address used is displayed in the status bar of the PCU server and Trend Manager.

- In case of an incorrect IP address in the configuration, the PCU server outputs a corresponding note when it is started and the IP address has a yellow background in the status bar.
- The PCU server and the Trend Manager only accept client TCP/IP addresses that are defined in the configuration.

Note

Please note the following for upgrading as well as for configuration changes:

The TCP/IP address configuration must be **set up and distributed** without errors and consistent with the LAN adapter configuration on all IOS stations during plant configuration.

3.5.2 Miniports

3.5.2.1 Communication ports

The communication functions of the system are realized through the following PCU_SERVER miniports:

- S7PLUS.DLL
  Coupling for SIMATIC S7 **(S7-400 and S7-1500)** via the "S7ONLINE" access point. The interface through which "S7ONLINE" communicates is defined with the function 'Set PG/PC interface' in the Windows system control.
- S7_LOGPORT.DLL
  If the "Process data logging" function is enabled in an IOS server, the S7 Logging Port is loaded.
  This ensures the cyclic requesting of the data points stored in the TAG list per released PCU and passes them on to the subsequent processing and archiving.
- REPLAYPORT.DLL
  If the "Replay mode" option is selected for an IOS, the replay port is loaded.
  With variable source requests of one or more HMI applications in replay mode this allows values from the data log archive to be supplied instead of online values from the S7 controller.
- CLIENT.DLL
  In this DLL the connection to different PCU Servers is realized via TCP/IP.
- DATAPORT.DLL
  This DLL is required for obtaining the instance data for the visualization objects.

3.5.2.2 Server functions

The server functions of the system are realized through the following PCU_SERVER miniports:

- TA.DLL
  In this DLL a copy of the AS sequencer instances is listed. The copy is supplied with data via the message 13.
- STATUSPORT.DLL
  In this DLL a copy of the tank status AS instances is listed. The copy is supplied with data via the message 22.
- RECCTRL.DLL
  In this DLL the sequencer-crossed control (Synchronizations) is realized.
- BALIDATA.DLL
  In this DLL the server functionality of the job system is realized.
- CAS.DLL
  In this DLL the functions for starting batches are realized.
- RCS_PORT.DLL
  In this DLL the Runtime functions of the Route control are implemented.
- ENTPORT.DLL
  In this DLL the functions for the ICM functions which are in the same context as Entities are realized.
- GLOBLOG.DLL
  In this DLL the convergence of the decentrally registered change protocols is realized.
- SIM.DLL
  In this DLL elemental simulation functions are realized for demonstrations and offline operation.
- MAINTSUPV.DLL
  This DLL implements the cyclic monitoring of the switch alterations and operating hours. Further information about configuring these miniports can be found in chapter Miniport for generating maintenance messages (Page 1005).
- OFFLINEPORT.DLL
  In this DLL the offline engineering is supported centrally.

3.5.2.3 PCU Server coupling types

The PCU server can run the following coupling types:

- System→ Normal system operation with coupling to the automation system
- Simulation - offline mode with simulated automation system for demonstration purposess

The miniports to be activated in the respective coupling type are prescribed by the system. They are determined by the IOS settings (client, server, recipe system, Route Control System), which have been configured in 'SiteCfg'. The user does not have to make any further settings here.

The current coupling type can be displayed and manually changed via the menu item 'Coupling types' PCU server window.

See also

Coupling type Simulation Recipe/Route Control Servers (Page 128)

Coupling type Simulation Recipe/Route Control Servers

This coupling type is available for live presentations or process image creation where no real PCUs are available. Elementary write and read access to data blocks of PCUs are intercepted by the "Sim.dll" miniport and redirected to the offline data blocks. The recipe server is activated and thus essential functions of a simulated batch process are provided.

- Braumat applications are showing ‘Simulation’ in the caption line
- Batches may be created
- Batches are running in simulation mode with user-configurable "StepTime"
- The simulation reads the data from the offline data blocks and writes the value of operations in the blocks.
- No further actions of technological function blocks in the AS are simulated (this could however be achieved for at least one AS with the PLCSIM software package).
- The feedback inputs of ICM blocks maybe simulated (without delay)
- The setpoint/actual values of DFMs may be adopted from the control recipe (valid for all enabled PCUs commonly)
- On selecting a recipe alternative, the result producing RUP will be paused; the user may set the alternative result manually by editing this value in the according sequencer data record within the PARAM application
- With Route control only the server miniport is started including simulated general PCU query but without route running; so that RCS online may be started and fixed but inactive routes (16 per PCU) may be viewed; concerning that, the RCS datablocks DB100 and DB404 should be present in the Offline DB folder of each PCU

Conditions for a useful Simu operation are:

- The Offline DBs should be present in the PCU folder
- In the PCU_SERV.INI file, the sections `[SimTA]` and `[SimICM]` must be preset.
  The preassignment of the delivery state "`<sys-path>\!BM_Proj\sys\PCU_SERV.INI`" is reflected in the following:

```
[SimTa]
;Enable sequence simulation
Enable=1
;Cycle time for sequence simulation in seconds
Cycle=5
;define an PCU-number which will be simulated by PLC sim
PLCSimPCU=-1
;Enable DFM Simulation,AlterResult(PCU < V7: DB725.DBB177, PCU V7:
DB725.DBB193)
EnableDFMSimu=0
;Processing time for steps in cycles
StepTime=2
[SimICM]
;Syntax is as follows: (PCU-number, ICM group number, ICM instance
number); [next description]
;Wildcards are allowed and reasonable ;)
;
;Examples
;--------
;Simulate just everything:     (*,*,*)
;Simulate all ICMs belonging to PCU1:   (1,*,*)
;Simulate all ICMS belonging to PCU1 or PCU2:  (1,*,*); (2,*,*)
;Simulate only the first two groups of all PCUs: (*,1,*); (*,2,*)
ICMsToBeSimulated=(1,*,*);
```

Note

- The ICM simulation in section `[SimICM]` only works in manual mode and in conjunction with the actual ICM simulation in the instance data set.
  → To this end, the flag "Simulate feedback" has to be set in the faceplate.
- The coupling type "Simulation" and "Replay mode" exclude each other.
- If Simulation is activated all the PCU server ports (S7.DLL, S7_LOGPORT.DLL, REPLAYPORT.DLL) loaded for Online and Replay mode are unloaded. Replay mode is prevented in the process diagrams.

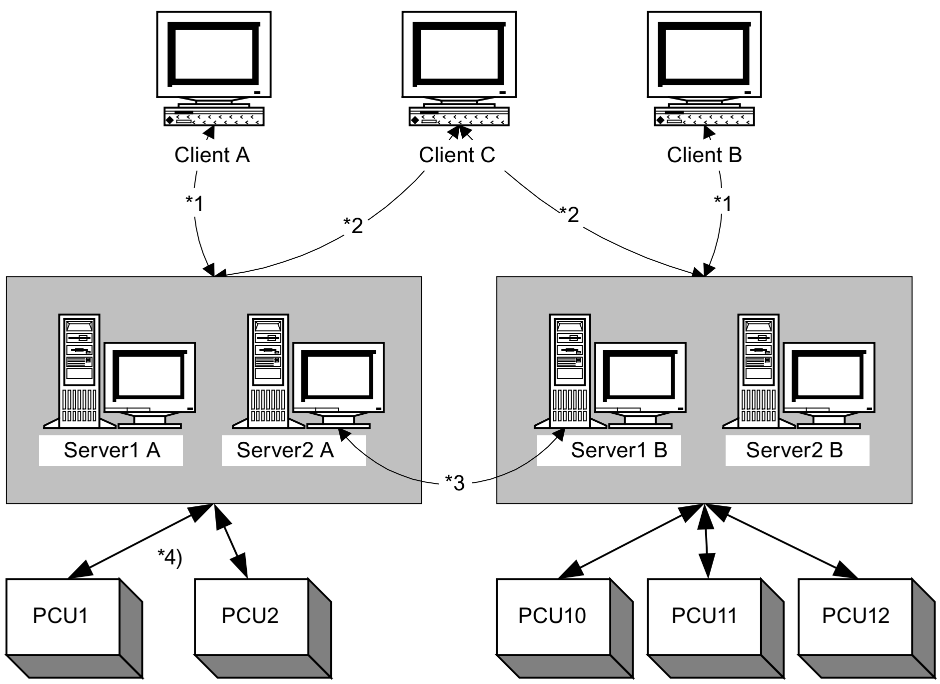
3.5.3 Interprocess communication between the IOSs

The possible IOS-IOS constellations are displayed in the image below.

These are:

- Client-Server constellation (*1)
  Online data for the runtime applications are supplied by 1 server couple.
- MultiClient–Server constellation (*2)
  Online data for the runtime applications are supplied by more than just 1 server couple.
- Server-Multi Client constellation (*3)
  Online data for the runtime applications at the server (now as client) are supplied by the local process connection as well as from different servers from other areas.
- Via the PCU-Server connection *4) a Server IOS normally requests online values from the PCUs to which it has a direct online connection (e.g. via ISO Industrial Ethernet).

The required settings for the various constellations can be made very conveniently for the user using the configuration tool "SITECFG.EXE" (see System configuration 'SiteCfg' (Page 108)).



3.5.4 Disabling specific PCU

With this function, it is possible to prevent write access to certain PCUs with reference to messages and operation.

It is possible to send certain operations to the connected PCUs either directly or via TCP/IP connection. This means that specified data areas in the control systems are overwritten.
For example, if you want to allow data to be visualized, i.e. read, from a remote area (remote PCU via client connection) but wish to prevent remote operation.
You can now block operation of specific PCUs from any IOS (user-definable), e.g. the

"Acknowledge ICM/Horn" function. The relevant switches in the PCU_SERV.INI file are as follows:

- [Trans_NT]
- NoWrite = <PCU-Nrn.,....>

This also applies to messages from various PCUs. These can also be disabled in order, for example, to prevent messages from a remote area being displayed. The list of PCUs is also stored in the PCU_SERV.INI file under:

- [Mld]
- NoMsg = <PCU-Nrn.,....>

3.5.5 Recipe Server

3.5.5.1 Activating the order/recipe system

- For the IOS settings in the configuration tool 'SiteCfg' the 'Enable recipe control' must be set.
- In the PCU server, the recipe server must be activated via the menu command "Program / Recipe server".
- The same function can be activated by pressing the button  in the toolbar.

Note

- The recipe server only boots if it receives an acknowledgment from all PCUs allocated to it in 'SiteCfg' .
- If one of the parameterized PCUs fails to respond, it remains in a standby state.
- This behavior can be changed → See also: Startup Recipe Server (Page 133)
- If a PCU server which was previously an active recipe server is restarted, this IOS will automatically become a recipe server again.
- By starting up the server, a file synchronization is executed with the backup server.

3.5.5.2 Procedure in the case of recipe server failure

First it must be stated if the recipe server has really failed. If this is the case, an attempt should be made to boot this IOS again.

If you are unsuccessful, the order and recipe system must be activated on the reserve recipe server.

The failure of the recipe server can also be signaled if the recipe server has its full functionality.

This message may be caused by these events:

- The connection to one or more PCUs has been aborted
- One or more PCUs are in STOP state
- The IOS is heavily overloaded; the monitoring messages could not be processed in the parameterized time.

The default is no automatic activation of a standby recipe server (delivery state). This is also the safe operation type, as a switchover always indicates an interruption to the production sequence.

3.5.5.3 Recipe server switchover

Basic principles

As any recipe server switchover represents a major intervention in the recipe control, this function should only be implemented if absolutely necessary (e.g.: server failure or server upgrading during continued production!).

If there is no server failure, it has to be ensured that the plant is idle during the switchover.

This means that either no batch is running or the running batch is in safe condition from the recipe view.

Sequence of the Server start

- Set PCU server on the backup recipe server to coupling 'system' if it hasn't yet been set.
- Press the Apply button on the PCU server; a dialog appears indicating whether this IOS should become the recipe server.
- Confirm the dialog box with OK
- If the window of the recipe control is opened, the following steps are performed:
  - FIFO Check:
    Read out the FIFO configuration of the PCUs entered in area.ini
  - FIFO Set:
    Reconfiguration of the FIFOs for the new recipe server IOS
  - KillTeleWait:
    Wait for killer messages → active recipe server is switched to 'STANDBY'!
  - FileSync:
    File synchronization is executed by the recipe server. Do not abort!
  - TA/CAS Setup:
    CAS and sequencer data are read
  - BALIDAT Setup:
    BaliData.dll is enabled
  - RECCTRL Setup:
    RecCtrl.dll is enabled

The reserve recipe server is now the active recipe server.

Note

- If BRAUMAT applications are open which require this coupling (order system, plant overview, recipes, etc.) these are held with a corresponding message box and can only be used after an application new start!
- The clients are converted automatically according to the IOS settings / server sequences in "SiteCfg".

3.5.5.4 Startup Recipe Server

Basic principles

After the standard installation, the Recipe Server runs only if all parameterized PCUs are running and can be accessed via the communication. In the case of large installations it is normal for one or more PLC not to be running. In order to allow startup in this case some inputs in the file recipe.ini should be made.

Release of the function

File:	`<proj-path>\sys\recipe.ini`
Entry:	`[RecCtrl]`
Key:	`EnablePCUSkipping=1`

Release message for missing PCU during startup

If, during recipe server startup, some PCUs are missing and these PCUs are skipped, a message can be generated. This message will be displayed on all IOSs.

File:	`<proj-path>\sys\recipe.ini`
Entry:	`[RecCtrl]`
Key:	`SkippedMessage=1`

BRAUMAT/SISTAR Lean on IPC 427D Microbox RTX

Note

When the "Lean" product license is activated, the number of active UNITs is limited to a maximum of 16.

To meet the license conditions, it is necessary to reduce the number of sequences to a maximum of 16 in the parameter assignment of the relevant PCU. This is checked when the recipe server starts up.

If the parameter assignment is incorrect, the message "License Error" appears in the "Recipe control" diagnostics window of the PCU server.

For this reason, ensure the following parameter assignment before starting BRAUMAT:

- Application parameter assignment / Class "Sequences" / Data source "Online (PCU)" / Global data / Set attribute "Count" ≤ 16

3.5.5.5 File synchronization during startup

Basic principles

One of the last steps during recipe server startup is the file synchronization. The default behavior is to synchronize all files that are necessary for the recipe server.

File synchronization also on the standby server

The file synchronization can also be executed on the standby server. This function should be released.

File:	`<proj-path>\sys\recipe.ini`
Entry:	`[RecCtrl]`
Key:	`EnableSyncOnStandBy=1`

3.5.6 Load distribution

The redundancy concept of BRAUMAT allows a load distribution for the runtime tag visualization. The load distribution enhances performance in normal operation without losing the redundancy functions.

For the load distribution the 'server sequence' entries in the 'SITECFG.EXE ' utility should be configured differently for the client IOSs. This is performed via the dialog 'IOS settings' (context menu->parameters) in the server sequences tab in the field 'default'.

Example:

For the clients 1, 2 and 3 the first server is set to '1' and the backup server is set to '2'. And vice versa for the clients 4, 5 and 6. In case of loss of one server the total load is assumed by one server. In normal operation each server should handle only 3 clients.

3.5.7 Diagnostics

3.5.7.1 Basics

Diagnostic options in case of coupling problems

The following information applies to the diagnostics of data paths to other coupling partners and is usually not of interest for fault-free production operation. In case of problems with the collection and delivery of data from and to the PCUs, IOS servers and IOS clients of the entire plant, however, the information can provide important clues regarding the cause of the problem, especially for requests sent to Customer Support.

Information in the status bar:

The status bar of the PCU server window contains the following information:

- IOS name
- Currently open view of the application (see "Display" menu)
- Computer name and TCP/IP address

  Note

  In case of an incorrect IP address in the configuration, the PCU server outputs a corresponding note when it is started and the IP address has a yellow background.

- Plant - PCU configuration and project ID

  Note

  The project ID is mandatory! An empty project ID or one that does not match blocks the PCU server connections between the IOS stations.

  In the case of different project IDs between redundant servers, this field has a yellow background.

- Date / time

Further diagnostic views

The application includes different views in the "Display" menu, the most important of which are described in the following sections.

3.5.7.2 Coupling status

Core statement

"Coupling status" view

This view shows an overview of the connections from the PCU server. These are permanently defined through the plant configuration ("SiteCfg") for this IOS station.

1. S7Plus Port: S7 connections for online data
2. S7 Logging Port: S7 connections for "Replay mode" / when the option "Process Data Logging" is activated
   For each PCU with data logging activated 1...4 connections are established; per connection a maximum of 3400 PATags/3400 PITags can be registered.

Column	Contents	Description
Area	AREA number	
PCU	PCU number	
Type	S7-400 S7-1500	SIMATIC CPU generation
Ethernet/IP address	xxxxxxxxxxxx nnn.nnn.nnn.nnn ---	MAC address 6 bytes hexadecimal ISO connection IP address 4 bytes numerical (1...255) TCP connection with SIMULATION
Local TSAP External TSAP	ISO-TSAP WinSock port ---	Transport service access point ISO connection windows IP socket port TCP connection with SIMULATION
PA	nnnn [... sec]	Number of registered variable process TAGs [... sec] = Update cycle
PI	nnnn [... sec]	Number of registered tags Trend Manager [... sec] = Update cycle
Connection state	Online - CPU on RUN (green) Online - CPU on STOP! (red) Try to connect... (blue) SIMULATION	Online mode and CPU is in RUN Online mode and CPU is in STOP Connection establishment active Coupling type SIMULATION is active

Total row with the number of: Connections, PaTags, PiTags, TotalTags

3. Client Port: Connections to the so-called "virtual PCUs" which means the data categories within the IOS server stations (local or remote AREA)
   8 categories are shown for each connection partner. The following format is applicable:

Column	Contents	Description
Area	AREA number	
PCU	PCU number DATA RCS: ETY IMG FMS BALX BAL REC	PCU Indirectly over AREA server DATA - general instance data RCS - Route Control System ETY - Entity port (maintenance groups) IMG - (no longer relevant) FMS - (no longer relevant) BALX - Order system BAL - Order system REC - Recipe system
Ethernet/IP address	AREA: nnn.nnn.nnn.nnn	AREA no: IP address 4 bytes numerical (1...255)
Local TSAP	WinSock port	windows IP socket port TCP connection
External TSAP	"[-]" "[M]"	No messages can be received Logged in to receive messages
PA	nnnn	Number of registered tags for process picture visualization
PI	nnnn	not relevant
Connection state	Connected (green) Try to connect... (blue) Server with invalid version: System-Version=..., IpSchema=..., requested IpSchema... Server refused connection (invalid system version, project id or site configuration)!	OK Partner server not available Partner server available - but error

   Total row with the number of: Areas, Connections, PaTags, PiTags, TotalTags

4. Offline port: Access to the offline DBs via application parameter assignment
   One row per connected IOS: IOSxxx: PaTags, PiTags
   Total row: Number of: Clients, PaTags, PiTags, TotalTags

5. Replay port: Access
   One row per connected IOS: IOSxxx: InternalTags=xxx, PaTags=xxx[<cycletime in sec>], Cycle (Interval in sec), Replay-Date/Time, State = RUNNING/HELD/IDLE
   Total row with the number of: Clients, Internal Tags, PaTags
   Additional information is available in the section Properties of the replay port (Page 529) .

3.5.7.3 Connections

Core statement

"Connections" view

This view shows an overview of the incoming TCP/IP connections (Win Socket connections) to the PCU server.

- These can come from the partner IOS server of this AREA or from the server IOS of another AREA.
- In addition, additional dynamic connections to the PCU server are established when applications are started locally or remotely.

Column	Contents	Description
AREA	AREA ID	Area number of the connection partner
Host	<Windows computer name>	
Name	IOS<nnn> "Applicationxyz"	IOS number from plant configuration For local connection
Version	"0x7nnn"	BRAUMAT system version of the connection partner (→ should always be uniform)
Address	"nnn.nnn.nnn.nnn" "local"	IP address 4 bytes numerical (1...255) For local connection
Port	16382	TCP/IP port number
Socket	nnnn	Windows TCP/IP Socket no.
PA#	nnnn	Number of registered tags for process picture visualization
PI#	nnnn	Number of registered tags for measured value visualization (Trend Manager)
PA ID	65535 nnn	No tags requested Request counter of the application
PI ID	65535 nnn	No tags requested Request counter of the application
State	IDLE IDLE [M] BST_READ ... BST_WRITE ...	Waiting Waiting with messages being received Read block Write block

The final information is a totals line with the following information:

Category	Subcategory	Description
Sum PA: Sum PI:	current=<xxx/yyy>	Currently requested PA tags xxx = combined by PCU server yyy = requested by the applications
	max=<xxx/yyy>	Maximum requested PA tags xxx = combined by PCU server yyy = requested by the applications
Sum PI:	clients=<nnn>	Clients currently logged in for trend data
IPConn:	ListenOn= "nnn.nnn.nnn.nnn"	Local IP address 4 bytes numerical (1...255) for incoming connections

3.6 Redundancy

3.6.1 Overview

BRAUMAT can be configured fully redundantly. This can be configured for the Server as well as for the Client functions. The total solution of redundancy influences the performance and flexibility of the system.

The plant engineer and the operator should decide how to use the redundancy for the plant. The system can be configured in such a way that all data and functions are available redundantly.

Exceptions that should be saved by the project manager:

- STEP 7 Projects
- RCS offline configuration, if this is not in the project folder

3.6.2 Hardware Redundancy

3.6.2.1 Redundant process bus

Each PCU may support up to 2 communication processors of type CP443-1. With this a redundant process bus may be configured. Further information about the parameterization of the IOS (Server) and PCU configurations is included in the description of the configuration tool "SiteCfg.exe" in the "Connection configuration display" chapter.

3.6.2.2 Redundant Terminal Bus

Redundant Terminal Bus

Functionality

The terminal bus connects the servers with the clients of the process control system. A redundant terminal bus is set up using two identical network segments, or using linked terminal bus rings (double ring). The network components plus PC LAN adapters ensure unrestricted operation of the terminal bus. If a terminal bus fails, communication remains possible over the second terminal bus.

Redundant communication solution

The following solutions are available to guard against failure of the terminal bus:

Two redundant fault-tolerant terminal bus rings.

These are set up with switches of the SCALANCE series. Optical, electrical and combined networks can be configured.

Suitable SCALANCE Switches

Bus components	Description
SCALANCE X414-3E	Modular switch with standby function, transmission rates up to 1 Gbps possible (suitable media module required for 1 Gbps). SCALANCE X414-3E can be used as a redundancy manager.
SCALANCE X204-2 (6GK5204-2BB10-2AA3)	Transmission rates up to 100 Mbps possible
SCALANCE X2..IRT	Modular Switch with Standby Function
SCALANCE X408-2	Link networks optically or electrically, 1Gbps

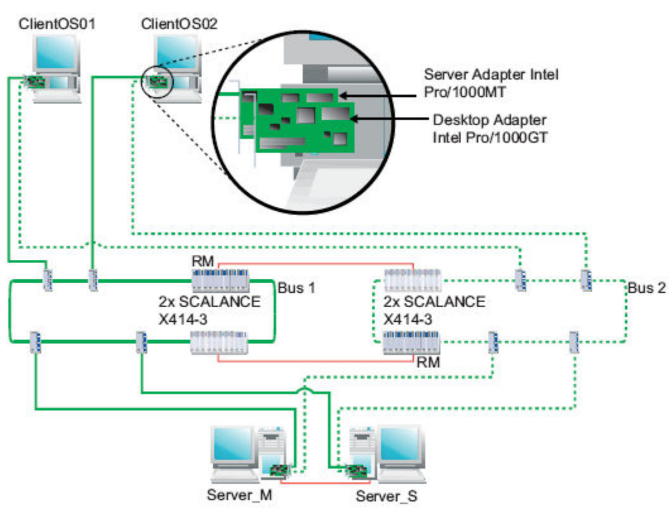
The following two network cards are used in each server or client PC station that is to be connected to the terminal bus:

- Intel Pro/1000MT server adapter
- Intel Pro/1000GT desktop adapter

These network cards work in a "team mode" with only one TCP/IP network address. Each network card is connected to one of the redundant terminal bus rings. All network components are redundant.

In a network segment (ring), a redundancy manager (RM) is configured to enable ring redundancy. The link between the redundant network segments (rings) is implemented using two switches (100 Mbps) in each network.

The following figure shows this configuration.



Linking Redundant Network Segments (Rings)

Switches and data links (network cable) connect the redundant networks. For a redundant link between the redundant network segments (standby link), you need to configure two switches within a network segment for synchronization of the network segments. You can find additional information about this in the operating instructions Industrial Ethernet Switches SCALANCE X-400. The two switches connected in the configuration exchange data frames and thereby synchronize their operating status (one device becomes the master and the other the slave).

When operating without fault, the data link to the other network segment is active only for the master. If this link section fails (for example, due to a defective device), the slave activates its data link as long as the fault exists.

Note

The redundant link of two network segments is supported only with switches that act as redundancy manager (e.g. SCALANCE X414-3E).

Configuration of the switches

You can find detailed information about configuring switches in the configuration manual Industrial Ethernet Switches SCALANCE X-400:

- Configuration using Web-Based Management and Command Line Interface
- Configuration and Diagnostics over SNMP

How to configure the network cards for the Terminal Bus

Requirements

- The following two network cards should be installed in each PC that is to be connected to the terminal bus (for example, OS server, OS client, domain controller):
- Intel Pro/1000MT server adapter
- Intel Pro/1000GT desktop adapter
- Note:
  Only the standard Windows services and the TCP/IP protocol are allowed to be active on the ‘Teaming Mode’ LAN adapters. The ISO transport protocols Simatic Industrial Ethernet (ISO) has to be disabled after installing a new LAN adapter for the terminal bus in teaming mode.

Procedure – Installing and Configuring Drivers

- Install the driver for the redundant network cards. This can be found in the manufacturer's Support/Download area at www.intel.com
- Start the installation package and carry out the installation steps according to the accompanying manufacturer documentation.

- After installing the driver, the two network adapters required for the redundant operation are identified by the operating system.
- These now have to be combined into one group. To do this, carry out the required steps according to the manufacturer documentation.
- For the operating mode of the group, select the entry "Switch Fault Tolerance" in the "Select a team mode" list.

Note

Operating system "Windows Server 2012 R2 (64Bit)"

Instead of the installation and configuration of the INTEL™ driver software, the operating-system-specific NIC-Teaming-mode can be activated in the Server Manager here.
- In the Server Manager, select the local server
- Under “Properties”, set entry "NIC Teaming" to "Enabled" ... the NIC Teaming dialog opens
- Create a new team and assign the 2 applicable adapters from the list of available adapters to the team
- The default setting "Switch independent" should be retained for the operating mode.

3.6.3 Software Redundancy

3.6.3.1 Redundant data maintenance

- For system configurations with redundant IOS servers and corresponding configuration in "SiteCfg", BRAUMAT saves all data redundantly to two servers (parallel redundancy).
- The system saves all data in single files on the hard disk.
- Each change in configuration, batch planning and recipes is filed immediately on both servers.

3.6.3.2 Route monitoring

Basic principles

Access to the paths is monitored cyclically both on the Servers and on the Clients.

In general the system configuration should be configured completely in the "Management" tab for correct function. The access paths and IP addresses of computers must be entered there.

Cycle time of the monitoring

File:	PathSupv.ini
Section:	[Config]
Key:	Sleep=30

With the value in seconds the monitoring cycle time is defined

Unhide and hide window

The function normally runs in the background without any display. The path supervising can be displayed in a window for testing and for diagnosis. The window can be closed. After closing it won’t be displayed for the following starts of the PCU Server either. The window may be activated again only by changing the file 'pathsupv.ini'.

File:	PathSupv.ini
Section:	[Config]
Key:	ShowWindow=1

Disable paths in runtime operation

Path accesses of all BRAUMAT applications are disabled if access via network is no longer possible. The system then makes this access via the next entered path in the AREA.INI . Local paths will never be disabled. This function runs in the background.

When a path is locked or enabled, a message will be entered in the message archive.

3.6.3.3 Variable visualization

The variable visualization is executed via TCP/IP. The allocated AREAs can be configured for each client. The changeover will be executed in the background.

In a multi-client configuration, the TCP/IP addresses are generated from the server sequence and the addresses of the server IOS.

3.6.3.4 Parallel recording

All messages, step protocols, curve values and free protocols will be sent to the two servers and recorded in parallel. The corresponding entries in the AS-FIFO blocks are configured in the delivery state for this.

3.6.3.5 Recipe and Route control

Server Standby

A Server and Standby concept has been realized for the function Recipe and Route control. The active control server runs each IOS Server. On Server 1 recipe control can run by preference and on Server 2 route control by preference. In case of redundancy (failure of one server) both can run on one server.

Activating the automatic server switchover

In case of failure of the recipe server, the other (standby) server is activated as the recipe server automatically after a user-definable time (about 6 minutes with the default settings). The following key should be enabled in the file "`<proj-path>\sys\RECIPE.INI`" on both servers of a redundant server pair:

```
[Serversupervising]

AutomaticServerActivation = 1

...
```

General query with Server start

By starting the IOS server, a general query of all relevant PCU data will be executed. Afterwards the data will be updated via message from the PCU.

File synchronization with Server start

Data are synchronized by starting the server. The synchronization guarantees that all necessary files for the recipe system are equal on both servers.

The configuration of synchronization files is necessary for the file synchronization. For starting the active and the standby servers, different files can be configured.

The emphasis should be placed in the recipe data for the active server. This causes a rapid (but more secure) startup.

For the standby server an extensive synchronization should be configured.

Synchronization of the process visualization

The process images and the configured text and object data are filed redundantly. Due to the high amount of data the synchronization of the process images should only be executed on the standby server.

After the installation as decision should be made as to whether an automatic synchronization of the process image is necessary at all.

3.7 Option pack “Route control" (RCS)

3.7.1 General information

The Route Control system is used for controlling and monitoring production routes in industrial process plants. Simple route settings up to a variety of complex route combinations are possible which depends on the plant complexity. Route control is mainly a tool for simplifying and standardizing configurations, processing and diagnosis of route settings. After a request for a new route (from source via partial route to destination) the assigned route can be determined, tested, controlled, monitored and watched with the help of the Route Control system.

The documentation for the RCS option pack also covers the following manuals:

Chapter	Subject
"Route control configuration"	This chapter describes the engineering of the route control system.
"Route control and monitoring process mode"	This chapter describes the online functions of the route control system.
"Route control AS blocks"	This chapter describes the online functions of the AS blocks.

3.7.2 Installation notes

3.7.2.1 AS Installation

PCU or AS installation is described in chapter AS Installation (Page 102). The PCU blocks of the RCS optional package are grouped together in separate program folders of the AS project supplied. After basic installation, the blocks in folders BM_RCS_SYS and BM_RCS_USR have to be copied into all PCUs which are involved in route control (the "Blocks" folder in each case).

Further information about integrating the RCS blocks into the AS process structure is contained in sub-chapter System block Calls (Page 1159).

3.7.2.2 OS Installation

Basics

The installation of route control is already included in the basic OS installation of BRAUMAT.

A separate ALM license is required to operate route control (refer to the section Licensing of software (Page 94))

IOS settings

For online operation of RCS, it is necessary to enable route control on the IOS server stations. This is described in the sectionModifying IOS settings (Page 114).

PCU settings

In order to enable the PCU functions and the coupling to the IOS server, it is necessary to enable route control on the PCUs. This is described in the sectionModifying PCU settings (Page 121).

3.8 "SQL adapter" option package

3.8.1 Installation and configuration

Installation and configuration

The installation of the component "BRAUMAT SQL Adapter IOS" is included in the basic installation.

Configuration is only possible with appropriate licensing (see the section Licensing of software (Page 94)) and is performed with the 'SiteCfg' system configuration tool according to the following steps:

- Enabling the SQL adapter in the IOS settings in "SiteCfg" (refer to the section . Modifying IOS settings (Page 114))
- Configuration of the SQL adapter "SiteCfg" (proxy connection and enables functions, refer to the section Configuring the SQL adapter (Page 118))

Note

Documentation on the Database Host

There is a separate manual for the Database Host page which describes the installation and configuration of the SQL proxy service and SQL archiver service services as well as the table structures. This manual can be found on the BRAUMAT DVD in the directory:

- _Manuals\Deutsch\SQL_DBHosta.pdf
- _Manuals\English\SQL_DBHostb.pdf

System administration

4

4.1 Basic Menu

4.1.1 Functionality

The application 'Basic menu' is used for the overview and selection of all selectable applications of the system.

The Client Area is divided into two areas:

1. Display of the bitmap `'<proj-path>\bilder\logo.bmp'`
   The representation shows the current delivery state.
2. View of the group tab with the respective applications

- For the plant operation
- For the system operation (by activating the "Expanded“ box)

Depending on which licenses are available for the current IOS, certain disabled applications are shown with a banned icon.

4.1.2 Working with the application

Working with the 'Main menu' application includes the following activities:

Selection of main menu

The 'Main menu' application can be opened in various ways:

- A double click on the desktop icon 'Application Center'
- Call-up in the Windows Start menu 'All Programs / Siemens Automation / BRAUMAT/ SISTAR Application Center'
- Call-up from other applications or process pictures.
- Automatically after a system start (see chapter WINDOWS System settings (Page 90))

Only 1 instance of the application can exist. Each selection of the 'Main menu' application from other applications results in the first instance of the application being activated.

Selecting the applications

The applications are opened from the main menu in three different ways:

- Double-clicking on the text of the selected application
- Clicking on the application and pressing the Enter key
- Clicking on the application and selecting the menu items 'Program' 'Run'

4.1.3 Register of the Application

The structure of the main menu is prescribed by the system and can be modified project-specifically as required.

Note

Project-specific modification of the main menu

- For this the definition file "BM_AppCenter.ini" must be copied from directory "`<sys-path>\Texte.x\...`" to project directory "`<proj-path>\Texte.x\...`" (x = Language number, e.g. Text.0 = German texts) and adapted appropriately.
- Following the start of the "ApplicationCenter", if the "BM_AppCenter.ini" file is found in the project directory, it is read in and interpreted instead of the system's own configuration file.
- If any existing user-specific adjustments to previous versions ("MenuV460.ini") are transferred during upgrading, these have to be entered into the new configuration file "`<proj-path>\Texte.x\BM_AppCenter.ini`" correspondingly.

The division into tabs and selection of the "Extended" check box adds the following input possibilities:

Table 4-1 Description of sections

Entry	Meaning
[Global]	Global adjustment
Small=1,2	Register user page
Extended=3,4,5	Register System functions
Bitmap=logo.bmp	Image in the main menu
ExtendedName=Extended	Name of the check box
[Chapter_1]	Register
Name=	Name of the Register
Desc=	Description of the Register
Bitmap=	Displayed image of the register
Applications=	Number of applications in the register
App1=	List of individual applications
App2=	List of individual applications
App3=	List of individual applications

4.1.4 Selection of the application with password check

A password check can be carried out before calling an application. This is also configured in the file "BM_AppCenter.ini".

Example:

The change and message archive with a password level 255 is opened with the following command line:

- `App3=Changes and messages;passwchk.exe 255 ArchiveViewer.exe`

4.2 Language transfer

4.2.1 Multilingual facility

Switchover of user interface languages

The system can run in different languages When the system language is changed in the main menu, the following actions become active after a restart of the system:

- The language DLLs in subdirectory `'<sys-path>\DLL.nnn\*.dll'` (nnn=language number) are loaded.
- When text files are accessed, the corresponding subdirectory `'<proj-path>\TEXTE.nnn\*.*'` (nnn=language number) is selected.

Languages available in delivery version

In the delivery version of the system, 'German', 'English', 'Spanish' and 'Chinese' are available. Upon request, other languages can also be created and supplied. The Language libraries and text files must be stored into corresponding folders. In addition you must configure these languages within the system in file `'<proj-path>\sys\SYS.INI'` under:

[Languages]

Numbers=0,1,<other language numbers>

0=German

1=English

2=Spanish

14=Chinese

:

<new number>=<language>

After restarting the system the operator can select new languages via the system settings.

After installation, the system directory contains the following subdirectories for the language resources:

...\ DLL.0	Index 0 = German
...\ DLL.1	Index 1 = English
...\ DLL.2	Index 2 = Spanish
...\ DLL.14	Index 14 = Chinese

Note

Additional user interface languages

- Additional subdirectories can be created by installing additional languages (available or to be implemented on project-specific basis).
- However, this manual installation is possible only under a Windows administrator login, because the system directory is write-protected for normal Windows users.
- The following subfolders are required (nnn = new language number):
  `dll.nnn` → Resources of the system user interface
  `help.nnn` → Online help system (can also be copied from language 1="English", if necessary)
  `texte.nnn` → Global system texts (can also be copied from language 1="English", if necessary)

4.2.2 Engineering language

Basic principles

Note

The engineering languages should be defined with the ‘SiteCfg’ Tool in the menu area > plant settings. Further information is available in chapter Modifying Plant settings (Page 109).

Default Setting:

Default adjustment = no changeover of the engineering language. The system texts are located in the folder "<Project-Path>\PCU.nnn\texte\...".

Dynamic Engineering language setting

In this case, the engineering language is switched over with the system language according to the allocation in the dialog "Factory settings (see above)."

The system texts are located in the folder "<Project-Path>\PCU.nnn\texte.x\...".

4.3 Access authorization, Password

4.3.1 General

The system offers the possibility to have all operations and inputs subjected to a staggered password protection. The protection is guaranteed by allocating a level area (1 to max. 255) per user and comparing against defined function levels.

- The function level and user / user level settings are configured in the application 'user administration'.
  When the system is delivered, the accompanying level "Sysini_P"=1 is set, i.e. enabled for all users.
- After the first configuration, the user and the level areas should be set to level "Sysini_P" > 200 (= Superuser).
- Before executing a protected function, the level area of the active user is compared against the function level saved for the requested operation. Only after a successful comparison is the requested operation or input possible.
- The system automatically requests the input of the username and password if the currently valid level area is not sufficient for the requested function.

Note

Central user administration

The user and function level definitions are only located on the redundant servers of an AREA and are therefore AREA-specific.

They are synchronized by the system between the two IOS servers (master / standby). A transfer or synchronization by the system between different AREAs is not supported.

Migration from earlier versions takes place as follows:

- The function levels are automatically imported from the "sys.ini" file when the server is started for the first time.
- The user definitions can be imported through a separate note dialog when the user administration on a server is called for the first time.

For multi-client access as well as for process pictures with access to different AREAs, the user rights are taken into account by the respective IOS server.

IOS-related user and/or function level definitions (analogous to 'IOS-related pictures') can be set up as follows:

1. Copy the existing definition file "<proj-path>\Admin\bm_user.dat" and/or "bm_levels.ini" manually to the servers into a separate IOS directory ("<proj-path>\IOS.xxx\Admin\").
2. Then edit this file from the respective IOS with the "User administration" application.

The following user-staggering is useful. It is possible to enter several users under each user group. The combination of the actions can also be adapted accordingly:

Group	Short Name (Example.)	Level	Actions
Super user	Su	1-255	Execution of all functions permitted
Operator user	Bed	1-100	Only selection of the user functions possible (left side in the menu). No configuration, no system functions.
Program user	Prog	1-200	Selection of the user and configuration function. No system functions possible.
Special user	Spez	<very special Level or Level section>	Only authorized for one or several very special function(s).

Release areas of function level IOS related

With this function it is possible to specify a level area for which there is no checking of rights. The configuration is made with a manual entry in the local definition file ""<proj-path>\sys\sys.ini" on the relevant IOS for which the area will be released. The configuration of the function level itself and the user can continue to be performed centrally.

Configuration file "<proj-path>\sys\sys.ini"

```
[PC]

...

;Passwordlevel after Startup/Reset is
StartupLevelMin..StartupLevelMax

StartupLevelMin=x

;Passwordlevel after Startup/Reset is
StartupLevelMin..StartupLevelMax

StartupLevelMax=y
```

4.3.2 User query

- The system automatically requests the input of the password if the currently valid level is not sufficient for the requested operation.
- After the correct input of the username / password a new password can be entered for this user or the current password can be changed with the ‘New password’ button.
- Only letters and digits are permitted in the password dialog.
- The new password now has to be confirmed again. If both password inputs are identical, the new password is valid.
- If there are different inputs for the new password, the following note box 'The new password is not active' will appear.

Basics

The operators of the process control system (not Windows users) are set up with the "User Manager" application. The application is called either manually from the main menu → Administration Tab → User Manager or automatically after login if users/password definitions are not (yet) found.

Note

When the "User Manager" is called, an administrator login is requested. An administrator login is therefore requested in a User Account Control (UAC) dialog window before changes can be made.

Creating/modifying a user

The **'User / Define'** button opens a dialog in which users already defined are listed.

- A new user with the name and the corresponding level section is entered via the button **'New'**
- The check box **'Password is set'** is grayed out with a new input, which means the new user must first enter a unique password when calling an application for the first time.
- You can change the user name and the level area with the **'Change'** button.
- The maximum **user name** length is **20 characters**.
- The password can be deleted by deselecting the check box **'Password is set'**. The user has to enter a new password when next logging in (important if a password is lost).
- The user's function level area is defined in the section Level.
- You delete the user with the **'Delete'** button.

Enter password

- When the "User" window shows asterisks in the "Password" column, the user has already entered a password.
- The user is queried in the applications / function calls, which are assigned a function level. If a user wants to change their password, they have to enter their password into the user dialog via the button **'New password'** . After a security query the password is active.
- The maximum length is **32 characters.**
- The minimum length can be set from **0 ... 9 characters.** With minimum length=0, an empty password can be set for this user. In this case, a password must not be entered in the login dialog.

User logoff

The user can logoff by means of the following methods, which are independent of one another:

- Automatic user login/logout
  When this option is enabled, the user is automatically logged off when the application is ended.
- User logoff after ...
  A time in minutes is defined with this setting. After this time has expired, all currently logged-in users are logged off, regardless of active applications. The time is restarted automatically after the next login procedure. If the time value "0" is entered, the method is inactive.
- Manually by the user
  It is possible for the user to log off at any time via the "Key icon" in the toolbar. Manual logoff is always possible, independent of the automatic methods.

Set function level

'Function level / Define' button opens the dialog, where the function levels can be assigned for each application.

- Button **'Change'** allows the function level to be changed within the range of 1...255.
- Multiple selection of text lines is possible via "Switch" + mouse click / "CTRL" + mouse click.

The following table contains the function keywords with their brief description and the default level assignments.

Program	Function	Description	Level
BALIEDIT.EXE BM_BatMan.exe	Bali_Chgpa	Order system: • Change parameters • Abort batch	1
BALIEDIT.EXE BM_BatMan.exe	Bali_Insba	Order system: • Insert batch • Attach batch • Move batch	1
BALIEDIT.EXE BM_BatMan.exe	Bali_Newor	Batch/Order System - Enter order	1
BALIEDIT.EXE BM_BatMan.exe	Bali_Proj	Batch/Order System - Configuration	1
BALIEDIT.EXE BM_BatMan.exe	Bali_RelBa	Batch/Order System - Change status	1
BALIEDIT.EXE BM_BatMan.exe	Batch_PIL_Chg	Batch/Order System - Changing batch process input lists	1
LZSYS.EXE	BEDIEN	Process diagrams - Open file Exit the system with LZSYS as shell	1
BIKO.EXE	BIKO	Image design - Application call	1
BLREDIT.EXE	BlrEdit-AdvOperator	Advanced BLR configuration	1
BLREDIT.EXE	BlrEdit-Engineer	BLR configuring	1

Program	Function	Description	Level
ANALOG DIGITAL CTRL3 ICM3.OCX	Commissioning	Commissioning (Analog/Digital/ICM3/CTRL parameters)	1
DBEDIT.EXE	DBEDIT	DB Editor – Value change	1
ENTITYDEF.EXE	EntityExplorer	Entity explorer access	1
ICM3.OCX	EntityOperator	ICM - Entity Edit TAB operator	1
SeqCtrl Unit_Ctrl.ocx	EOPAbort	Plant overview/UnitCtrl.ocx - EOP Abort	1
SeqCtrl Unit_Ctrl.ocx	EOPHold	Plant overview/UnitCtrl.ocx - EOP Hold	1
SeqCtrl Unit_Ctrl.ocx	EOPPause	Plant overview/UnitCtrl.ocx - EOP Pause	1
SeqCtrl Unit_Ctrl.ocx	EOPStart	Plant overview/UnitCtrl.ocx - EOP Start	1
SeqCtrl Unit_Ctrl.ocx	EOPStop	EOP Stop	1
ESG.EXE	ESG	ICM - Application User access	1
MS-Excel Export.dll	ExcelExport	Excel - Export function	1
MS-Excel/Export.dll	ExcelImport	Excel - Import function	1
ICM3.ocx	ICMOperator	ICM3 – User access (Manual-Auto, On-Off); seat-lifting	1
ICM3.ocx	Improvement	ICM3 - Entity maintenance mode details	1
SeqCtrl UnitCtrl.ocx	KETTE	Plant overview/UnitCtrl.ocx - Inputs (old) Skip synchronization / alternative	1
SeqCtrl UnitCtrl.ocx	KETTE_HAL	Plant overview/UnitCtrl.ocx - Operation 'Hold' (A-)	1
SeqCtrl	KETTE_KPO	Plant overview - Edit unit	1
SeqCtrl UnitCtrl.ocx	KETTE_SCR	Plant overview/UnitCtrl.ocx - Operation 'Step selection'	1
SeqCtrl UnitCtrl.ocx	KETTE_STA	Plant overview/UnitCtrl.ocx - Operation 'Start'	1
SeqCtrl UnitCtrl.ocx	KETTE_STO	Plant overview/UnitCtrl.ocx - Operation 'Stop'	1
SeqCtrl UnitCtrl.ocx	KETTE_SW	DFM overview/UnitCtrl.ocx - Input free setpoints	1
EditRec ControlRec SeqCtrl UnitCtrl.ocx	KETTE_SW1	Plant overview/Edit-/ControlRec/UnitCtrl.ocx – Setpoints 1-5	1

Program	Function	Description	Level
EditRec ControlRec SeqCtrl UnitCtrl.ocx	KETTE_SW2	Plant overview/Edit-/ControlRec/UnitCtrl.ocx – Setpoints 6-10	1
EditRec ControlRec SeqCtrl UnitCtrl.ocx	KETTE_SW3	Plant overview/Edit-/ControlRec/UnitCtrl.ocx – Setpoints 11-15	1
EditRec ControlRec SeqCtrl UnitCtrl.ocx	KETTE_SW4	Plant overview/Edit-/ControlRec/UnitCtrl.ocx – Setpoints 16-20	1
EditRec ControlRec SeqCtrl UnitCtrl.ocx	KETTE_SW5	Plant overview/Edit-/ControlRec/UnitCtrl.ocx – Setpoints 21-...	1
SeqCtrl UnitCtrl.ocx	KETTE_WEI	Plant overview/UnitCtrl.ocx - Operation 'Resume' (A+)	1
SeqCtrl UnitCtrl.ocx	KETTE_ZUA	Plant overview/UnitCtrl.ocx - Aux/user bit Off	1
SeqCtrl UnitCtrl.ocx	KETTE_ZUE	Plant overview/UnitCtrl.ocx - Aux/user bit On	1
KURVEIN.EXE	KURVEIN	Curve Entry - Save	1
KURVEN.EXE	KURVEN_BEA	Trend visu.- Editing trend	1
SEQCTRL.EXE	LeaveInterState	Leave Intermediate State of EOP manually	1
WARTDAT.EXE	MAINT	Maintenance data - Acknowledge and reset order	1
WARTDAT.EXE	MAINT_P	Maintenance data - Parameter input and setpoint	1
ANALOG DIGITAL CTRL3 ICM3.OCX	Maintenance	Controls - Maintenance access, parameter, note	1
EDITREC.EXE	Mat_Ass	Recipe system – Material assignment	1
EDITREC.EXE	Mat_Man	Recipe system - Material management	1
EDITREC.EXE	Mat_SetActSt	Recipe system - Setting current material stock	1
ARCH_MAN.EXE	Meldarch	Archive manager - Delete archive	1
NEWMENU.EXE	Menu_E	Main menu - Exit application	1
NEWMENU.EXE	Menu_M	Main menu - Move	1
NEWMENU.EXE	Menu_P	Main menu – Windows Explorer access	1
EDITREC.EXE	MR_Status_Change	Recipe system - Changing status of the master recipe	1
CTRL3.ocx	Operator	CTRL3 – User access (Manual-Auto, Internal-External)	1
SEQCTRL.EXE	OpReqAck	Plant Overview - Acknowledgement of operator prompts	1

Program	Function	Description	Level
PARAM.EXE	Param	Parameter assignment - SW changes	1
PARATXT.EXE	Paratxt	Text Parameterization - Edit/Save	1
PCU_SERV.EXE	PcuServ	PCU server- General Functions (old)	1
PCU_SERV.EXE	PcuServ_Dbl	PCU server - without menu bar	1
PCU_SERV.EXE	PcuServ_End	PCU server - End	1
PCU_SERV.EXE	PcuServ_Ini	PCU server - Edit INI files	1
PCU_SERV.EXE	PcuServ_Kop	PCU server - Coupling	1
PCU_SERV.EXE	PcuServ_Opt	PCU server - Options	1
PCU_SERV.EXE	PcuServ_Smm	PCU server - Mini/Size/Move	1
PCU_SERV.EXE	PcuServ_Win	PCU server - Restart Windows	1
QUEUING.EXE	QueuingExplorer	Queuing explorer access.	1
RCS_ONLINE.EXE ROUTECON.EXE	RCS_L1	RCS Online: Manual Stop	1
RCS_ONLINE.EXE	RCS_L2	RCS Online: Request, Start, Pause, Acknowledge	1
RCS_ONLINE.EXE	RCS_L3	RCS Online: Route properties, configuration	1
EDITREC.EXE	Rec_Cl	Recipe system - Components list Edit	1
EDITREC.EXE	Rec_Head	Recipe system - Recipe header	1
EDITREC.EXE	Rec_Main	Recipe editor - Editing process parameter definition	1
EDITREC.EXE	Rec_New	Recipe system - Create new recipes	1
EDITREC.EXE	Rec_Proj	Recipe system – EOP definition, DFM definition, equipment configuration, weigher definition in storage locations	1
EDITREC.EXE	Rec_Struct	Recipe system – Change recipe structure	1
REGLER.EXE	Regler_P	Controller operation - Parameter input	1
REGLER.EXE	Regler_T	Controller operation - value input, key operation	1
LZSYS.EXE	REPLAYMODE	Display and replaying of recorded data	1
EDITREC.EXE	Rezepte	Recipe system – Save Recipe	1
CONTROLREC.EXE RECCONTR.EXE	Rezepte_on	Recipe system - Online editing	1
CONTROLREC.EXE RECCONTR.EXE EDITREC.EXE	Rezepte_sw	Recipe system – Change setpoint definition	1
EDITREC.EXE CONTROLREC.EXE	Rezepte_SWx	Recipe system - Substitution or scaling of recipe setpoints	1
EDITREC.EXE	RP_Status_Change	Recipe system - Changing status of recipe procedures	1
SITECFG.EXE	SiteCfgSave	Area.ini: Saving with SiteCfg.exe	255
SONDWERT.EXE	Sondwt_N	Special values - Edit Note text	1
SONDWERT.EXE	Sondwt_p	Special values - Parameter assignment	1
SONDWERT.EXE	Sondwt_w	Special values - Value input	1

Program	Function	Description	Level
Sistar_ADP.exe	SQL_ADP_END	Sistar_ADP - Exit application	1
SRP_NEU.EXE	Srproted	Step protocols - Delete	1
STATUS.EXE	Status	Status.exe - Opening the application	1
STEUERN.EXE	Steuern	Forcing – Value change	1
CONFIG.EXE	Sysini_I	System Settings - Save	1
ARCH_MAN.EXE	Sysini_L	Archive - Define new export drive	1
USERMAN.EXE	Sysini_P	User Manager - Password/Level	255
NEWMENU.EXE PCU_SERV.EXE	Sysini_S	Main menu, PCU Server Language/IOS settings	1
TRACEVWR.EXE	TRACESVR	Trace server operation	1
TRANSBST.EXE	Transbst	Block transfer - Transfer start	1
TRENDMAN.EXE	Trendman	Trend Manager - Exit application	1
ZLSYNCHRONIZ-ER.EXE	ZLSynchronizer	ZLSynchronizer - edit	1

4.4 Data backup

4.4.1 Import/Export, Delete Archive files

The application 'Archive Manager' is used in the system to import/export archive files. Here the project engineer can define their individual archive structures and sections in order to realize definite file movements.

In the application, so-called export drives first have to be specified. There must either be a network connection to these drive letters or external USB data carriers can be used, for example.

4.4.2 Archive Manager

4.4.2.1 Functionality

The application 'Archive Manager' is used for the administration of archive files.

These are created continuously during operation and filed on the hard disk of the IOS. Filed are e.g. messages, free protocols, step protocols, curves, etc.

The archive types to be administrated by the Archive Manager can be freely configured. The application allows the swapping out of archive files to export disk drives, their deletion or re-import to the system disk in a convenient and time-saving manner.

The application is started under Basic menu -> Administration Tab -> Archive Manager.

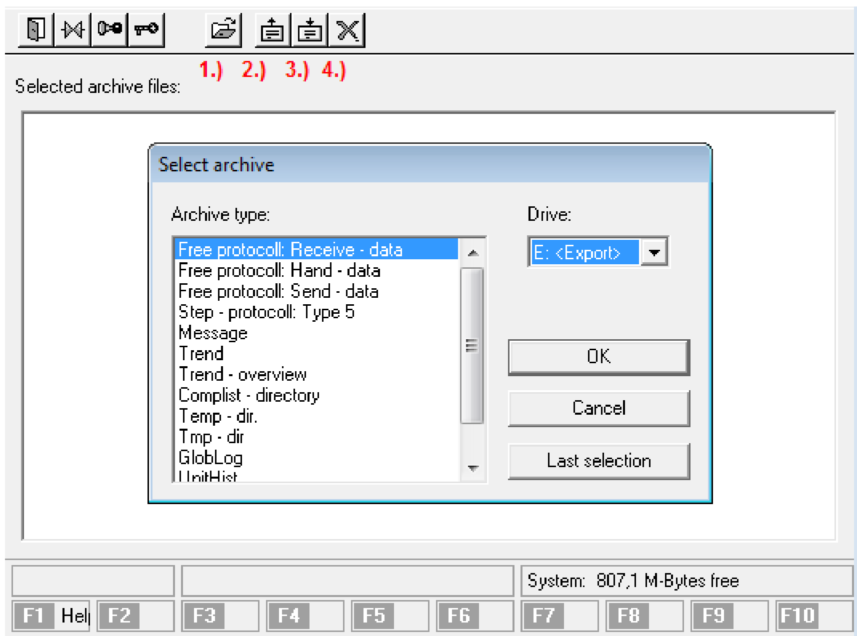
The Client Area includes a list box "Selected Archive files“. There the currently selected archive files are displayed completely with their total path. The list box is used exclusively to control the selection.

The status line displays the user-defined disk capacity of the system disk drive and of a selectable export disk drive.

4.4.2.2 Working with the application

Basic principle

The following figure shows the screen of the application:



Menu item Program:

Use according to usual system applications.

Menu item functions:

This item covers application-specific functions.

- "Selection" or 1.) Click Icon
  A dialog box for selection of the archive type and the disk drive appears. With the button "Previous selection" the last selection can be called again. After you have selected, continue with "OK". Then a new dialog box is opened for any further selections according to archive-specific indications.
  With "OK“ it will branched out to the next dialog box, with "All" all files can be selected in the current path.
  The last dialog box is used for the selection of the requested files. It includes two list boxes: "Available Archive files" and "Selected Archive files“. With the buttons "-->" or "<--" the corresponding files which are marked can be transferred into the opposite list box. The buttons ">>" and "<<" displace all included files in the different list box. The corresponding file can also be moved by a double-click.
  With "OK" the file selection is ended and the dialog box is closed.
  The selected files appear with their complete path in the list box "Selected Archive files" of the Client Area.
- "Drive setting"
  Export drives for inputting or outputting archive data can be defined here.
- "Export" or 2.) Click Icon
  Export the selected archive files to the destination drive. A dialog box appears for the choice of destination disk drive. Afterwards the selected disk drive must be confirmed again in a message box. During the export procedure a progress bar informs about the status of the activity. After the export the list box is emptied in the field of activity.
- "Import" or 3.) Click Icon
  Import the selected archive files on the archive-specific system disk drive. The selection should be confirmed in a message box.
  It should be considered:
  As a system disk drive with the corresponding path, the application uses the indications entered in the "SiteCfg" for the corresponding archive type. During the import procedure a progress bar informs about the the progress. After the import the list box is emptied in the field of activity.
- "Delete" or 4.) Click Icon
  Delete the selected archive files. First there is a password query. Afterwards the selection should be confirmed again. During the deleting procedure a progress bar informs about the status of the activity. After deleting the list box is emptied in the section of activity.

Note

- The import is only possible from the export disk drives. Destination disk drive is automatically set to the system disk drive of the corresponding archive type which is indicated by the "SiteCfg".
- It is only possible to export to the defined export disk drives.
- The destination path to the export drive also contains the ISO name of the source data. This makes it possible, for example, to export a multi-client IOS from the archives of several other IOS stations to the export drive

---

Menu item 'Options':

The switches for displaying or hiding the toolbar, status bar and function keys are located in this menu.

Furthermore, the application 'archive manager' makes a menu item 'display' available with which the disk drive should be selected whose free capacity should be displayed in the status line.

After having clicked, a dialog appears in which all export disk drives are available for selection. The display can also be deactivated by the selection "none".

4.4.2.3 Archpath.txt

Basic principles

With the function 'selection' various archive files can be selected. The dialog-controlled selection refers to the search criteria and path description which is stored in the text file ARCHPATH.TXT.

The path/file names stored in the text file are supplemented successively by the user selections. The selection is made in a list box which presents the found files to the user according to the list inputs or wildcard characters.

After a successful selection of the archive files the complete paths with file names are viewed in the list box "Selected Archive files“ in the Client Area of the application.

The text file is located at `<proj-path>\SYS\ARCHPATH.TXT`.

Configuration:

In the file ARCHPATH.TXT a dedicated section must be entered for each archive type, as described below. At the beginning of the text file the following section must be indicated one time.

[ANZ]		
Number	= <Number>	Amount of entered archive types

In the following the syntax for the configuration of a search path is listed for one archive type:

[TAB<Nr>]		Number of the Archive type
Name	= <Archive name>	Archive name
PathDat	= <Path/File name with wild-card character>	Data file with complete path indication and possible wildcard characters.

[TAB<Nr>]		Number of the Archive type
List<List-Nr>	= <Name Listbox>	• List No: Number of wildcard character block • Name List box: List box headline, which is output by selection of the corresponding wildcard character. If there is no input available, no selection list box is displayed and branched out to the next selection. The next selection (List No.) starts with the next wildcard character block.
From<List-Nr>	= <Char number>	character number from which the selection starts If it should not be searched from the first question mark, the corresponding character number should be entered.
To<List-Nr>	= <Char number>	character number, from which the selection starts If it should not be searched to the last question mark, the corresponding character number should be entered.
Ref<List-Nr>	= <Reference to List No.>	Reference to the list input which has already been executed. If there is no list input, the same input is executed as for referenced list no. If there is a list and ref input for a number, the wildcard character block is filled starting from the left. The next step is to select the rest of the signs in a list box, whereby the list input should be used in connection with From/To. References to several list inputs are possible. The numbers should be separated by '/'. The wildcard character block is filled consecutively starting from the left with the reference indications.
Sec	= <End>	If secondary files should be selected with the same paths and names at the same time, but with a different extension, the secondary extension can be entered here.
Typ	= <File type>	File types are, e.g. 0=Curves 1=Fr.Prot 2=Write out step protocol 3=Messages

The following table shows the standard parameterization when the system is delivered: This may be configured at project-specific level.

Section	Path/File structure
[ANZ] Anzahl = 13	13 Archive definitions to follow
[TAB1] ;Name = Freies Protokoll: Empfangs-Daten Name = Free protocoll: Receive - data PathDat = \FRPROT\FRJA_??\REZ_???\FE_??_??.DBF List1 = year List2 = recipe type Ref3 = 1 List4 = month Typ = 1	…<poj-path>\FRPROT – folder FRJA_?? → subfolder year 2-chars (00=2000) REZ_??? → subfolder Recipe type 3-chars FE_??_??.DBF → FE file for Year/ Month in format <yy_mm>
[TAB2] ;Name = Freies Protokoll: Hand-Daten Name = Free protocoll: Hand - data PathDat = ..\FRPROT\FRJA_??\REZ_???\FH_??_??.DBF List1 = year List2 = recipe type Ref3 = 1 List4 = month Typ = 1	…<poj-path>\FRPROT – folder FRJA_?? → subfolder year 2-chars (00=2000) REZ_??? > subfolder Recipe type 3-chars FH_??_??.DBF → FH file for Year/ Month in format <yy_mm>
[TAB3] ;Name = Freies Protokoll: Sende-Daten Name = Free protocoll: Send - data PathDat = ..\FRPROT\FRJA_??\REZ_???\FS_??_??.DBF List1 = year List2 = recipe type Ref3 = 1 List4 = month Typ = 1	…<poj-path>\FRPROT – folder FRJA_?? → subfolder year 2-chars (00=2000) REZ_??? → subfolder Recipe type 3-chars FS_??_??.DBF → FS file for Year/ Month in format <yy_mm>
[TAB4] ;Name = Schrittprotokoll Typ 5 Name = Step - protocoll: Type 5 PathDat = ..\SRPROT\SRJA_??\REZ_???\ANR?????\S???????.DBF List1 = year List2 = recipe type List3 = block List4 = Order - no. From4 = 3 To4 = 7 Ref4 = 1 Typ = 2	…<poj-path>\SRPROT – folder SRJA_?? → subfolder year 2-chars (00=2000) REZ_??? → subfolder Recipe type 3-chars ANR????? → subfolder order no. – range (e.g. ANR00100 = Anr.00000 .. 00100, etc.) S???????.DBF → file for order no. in format <yynnnnn>

Section	Path/File structure
[TAB5] ;Name = Meldungen Name = Message PathDat = ..\MELD\ME_??_??\MA??????.TXT List1 = year List2 = month List3 = day From3 = 5 To3 = 6 Ref3 = 1/2 Typ = 3	…<poj-path>\MSG – folder ME_??_?? → subfolder Year/Month in format <yy_mm> MA??????.TXT → file per day in format <yymmdd>
[TAB6] ;Name = Kurven Name = Trend PathDat = ..\TREND\DATA.???\KW????\????T???.H List1 = block List2 = week/year Ref3 = 2 List4 = No. Sec = S Typ = 0	…<poj-path>\TREND – folder DATA.??? → subfolder measuring value No. range KW???? → subfolder calendar week in format <wwyy> ????T???.H → Header files in format <wwyy>T<meas.value.no> ????T???.S → Data files
[TAB7] ;Name = Kurven-Übersicht Name = Trend - overview PathDat = ..\TREND\DATA.???\BA??T???.H List1 = block List2 = year List3 = No. Typ = 0	…<poj-path>\ TREND – folder DATA.??? → subfolder measuring value No. range BA??T???.H → Batch curves header files in format <yy>T<meas.value.no>
[TAB8] ;Name = Komponentenliste Name = Complist - directory PathDat = .\recipe\complist\clja_??\rez_???\anr????? \CI_?????.DBF List1 = year List2 = recipe - type List3 = order - no. List4 = batch - no. Typ = 4	…<poj-path>\RECIPE\COMPLIST – folder clja_?? → subfolder year 2-chars (00=2000) REZ_??? → subfolder Recipe type 3-chars ANR????? → subfolder order no. CI_?????.DBF → complist for batch
[TAB9] Name = Temp - dir. PathDat = ..\TEMP\????????.??? List1 = name List2 = type Typ = 5	…<user-temp-lw>\TEMP folder
[TAB10] Name = Tmp - dir PathDat = ..\TMP\????????.??? List1 = name List2 = type Typ = 5	…<poj-path>\TMP – folder

Section	Path/File structure
[TAB11] Name = GlobLog PathDat = ..\Logging\GlobLog\Gl_??\GL??????.dbf List1 = year List2 = month List3 = day From3 = 5 To3 = 6 Ref3 = 1/2 Typ = 3	…<proj-path>\LOGGING\GlobLog – folder Gl_?? → subfolder year 2-chars (00=2000) GL??????.dbf → Global log file per day in format <yymmdd>
[TAB12] Name = UnitHist PathDat = ..\Logging\UnitHist\Uh_??\UH??????.dbf List1 = year List2 = month List3 = day From3 = 5 To3 = 6 Ref3 = 1/2 Typ = 3	…<proj-path\LOGGING\UNITHIST – folder UH_?? → subfolder year 2-chars (00=2000) UH??????.dbf → Unit history log file per day in format <yymmdd>
[TAB13] Name = UserLog PathDat = ..\Logging\Userlog\Ul_??\UL??????.dbf List1 = year List2 = month List3 = day From3 = 5 To3 = 6 Ref3 = 1/2 Typ = 3	…<proj-path>\LOGGING\USERLOG – folder UL_?? → subfolder year 2-chars (00=2000) UL??????.dbf → User log file per day in format <yymmdd>

4.5 Block transfer

4.5.1 Functionality

The application **'Block transfer'** has the following functionality:

- Transfer of PCU blocks from the AS to the hard disk of the IOS.
- Transfer of PCU blocks from the hard disk of the IOS to the AS.

While processing the same file simultaneously by several users (IOSs) the consistency of the data will be guaranteed by attaining validity for the last saved file.

The application is started under Basic menu -> Administration Tab -> Block transfer.

4.5.2 Client Area

The Client Area is divided into two areas:

right:	Buttons for using the functions: Save, load, process blocks
left:	Listing the transfer blocks
Column 1:	Area
Column 2:	Block
Column 3:	Block number

4.5.3 Working with the application

Create new transfer list

After selecting the menu item **New** in the **File** menu or clicking the icon **New** a new transfer list will be created in the toolbar.

Open transfer list

After selecting the function **Open** in the menu **File** or after clicking the icon **OPEN** in the toolbar a transfer list can be opened. A modal dialog box for the selection of drives, directories, file name and file format will be displayed.

The files are in the directory: <proj-path>\TRANS

Block transfer:
The files for the block transfer have the format "*.trs".

File transfer list

After selecting the function **Save** in the menu **File** or after clicking the icon **SAVE** in the toolbar the current transfer list will be filed.

To save the transfer list under a different name use the **Save as...** function in the menu **File**. A modal dialog box for the selection of drives, directories, file name and file format will be displayed.

The data files are filed in the directory `<proj-path>\TRANS`.

Block transfer:
The files for the **block transfer** have the format "*.trs".

Edit transfer list

After selecting the function **Process** in the menu **Functions**, after clicking the button **Process** in the Client Area or after activating the function key **F5** the transfer list can be edited. A modal dialog box will be displayed.

Single recipes/blocks can be deleted from the transfer list with **Delete**, with **Add** recipes/blocks can be added or with the function **Delete all** recipes/blocks can be deleted from the Recipe/ Block list. When you assign numbers, you can define individual recipes/blocks, or an area (x to y), or you can use wildcards "*" to select all available recipes/blocks.

Load blocks

After selecting the function **Load** in the menu **Functions**, after activating the function key **F4** or after clicking the button **IOS->PCU** in the Client Area a dialog box will be displayed if no transfer list is opened. After the input of **Area** and **Block** and activating OK the transfer will start. If a transfer list is opened the block transfer starts immediately and **without any feedback**. The currently transferred block is displayed.

Save blocks

After selecting the function **Save** in the menu **Functions**, after activating the function key **F3** or after clicking the button **PCU->IOS** in the Client Area a dialog box will be displayed if no transfer list is opened. After the input of **Area** and **Block** and activating **OK** the transfer will start. If a transfer list is opened the block transfer starts immediately and **without any feedback**. The block that is transferred currently is displayed.

Options -> Settings

You can customize the following settings:

- Report -> always generate log file
  Generates the ASCII format log file "Report.txt" in the folder `<proj-path>\TRANS`. An existing file will be overwritten.
- Report -> Print automatically
  The generated log file is output on the default printer
- Error handling -> Blocks not found
  When this function is enabled, all non-existent blocks will also be logged. Otherwise, the log indicates only the transferred blocks.
- Error handling -> Cancel transfer
  The transfer of data is automatically canceled if a block is not found in the list.

Special handling of the FIFO and SendPu DBs

NOTICE
The following data blocks are excluded from the IOS → PCU transfer ("Load blocks"):
DB 670, 671, 672, // FIFO1-FIFO3
DB 674, 675, 676, // SendPu1-SendPu6
DB 690, 691, 692, // FIFO4-FIFO6
DB 694, 695, 696 // SendPu4-SendPu6

PCU type S7-1500

Unlike for PCU type S7-400, only data blocks (DB *) can be transferred for PCU type S7-1500. This is true for both transfer directions (PCU → IOS and IOS → PCU). As a result, both the engineering data and the complete runtime data of an S7-1500 PCU can be saved on the OS. Moreover, this also means that basic engineering using Microsoft Excel is supported.

4.5.4 Starting automatic block transfer at the command line prompt

Basics

The program "Transbst.exe" is suitable for generating automatic backups of the PCU blocks. The user may configure a periodic call of this program in "Prosched.exe" see chapter Scheduler (Page 176), in order to start this cyclic block transfer.

Calling "Transbst"

Call	Function
Transbst /u:upload	"transbst.exe" executes the commands in <poj-path>/trans/upload.trs. The application window of "transbst.exe" is visible during the execution of the program.
Transbst /u:upload /s	"transbst.exe" executes the commands in <poj-path>/trans/upload.trs. The application window of "transbst.exe" is hidden during the execution of the program.

Example:

The sample configuration and the corresponding INI files below describe the syntax:

- An automatic weekly backup of the AS blocks is to be configured.
  "Transbst.exe" fetches all blocks (FB, FC, OB, DB, SDB) from PCU14 and PCU16, e.g. every Sunday at 12:15 AM.
- The <poj-path>/trans/upload.trs file is needed for this. This file must be configured appropriately with the application beforehand (see above)

- **Prosched.ini**
```
[Global]
Processes=TransBst1, ...
 [TransBst1]
Application=transbst.exe /u:upload
;to be started weekly, each sunday at 00h:15m:00s
Type=WEEKLY
WeekDay=0
ReqTime=00:15
```
- **Transbst.ini**
```
[Options]
;Break transmission when an error occurs
BreakOnErr=0
;i.e. the transfer is not canceled if the block is not found
```

Result:

Every week on Sunday at 00h:15m:00s, "ProSched.exe" starts the block transfer. The function generates the block folders with the current block files in `<proj-path>\pcu.xxx`. The ProSched window indicates the start and the time of the start of "transbst.exe".

Further switches the user can set for calling the program "Transbst.exe":

- /U:<File name> .... Upload with configuration file <File name>
- /D:<File name> .... Download with configuration file <File name>
- /S ….Program execution with hidden window

4.6 Synchronization

4.6.1 Functionality

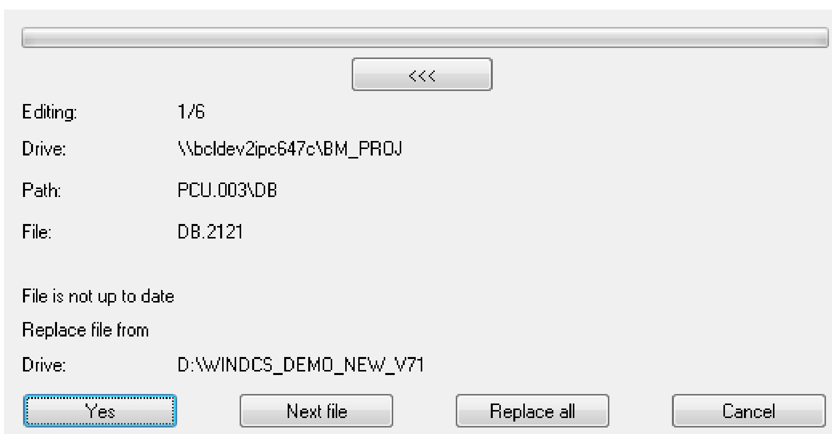
The application synchronizes the files and folder sub-trees within any group of network shared drives. In this way the latest version of a file is used as the source.

The application is started from the main menu > Administration tab > Synchronization.

The application is used in three steps:

- File definitions in file 'synchro.def' lesen
- Compare files (results in file 'synchrox.tmp')
- Copy files (results in file 'synchrox.err')

After the comparison the following dialog box appears for controlling the synchronization process appears:



The file to be synchronized, together with source and destination paths, is indicated.

The following actions are available using the buttons:

- Yes → the indicated file will be synchronized.
- Next file → the indicated file will be skipped.
- Replace all → the indicated file and all following files will be synchronized
- Cancel → the synchronization will be cancelled

4.6.2 Application call with parameter:

Automatically start: /a

If the application is started with the parameter '/a', all functions are carried out fully automatically.

There is no query!

All non-existent files or obsolete files are copied without any queries!

4.6.3 Call without parameter

Basics

- The application is started automatically of the OS works as recipe server. A synchronization occurs with every start of the Recipe Server.
- After having called the application it is initially queried as to whether the synchronization should be carried out.
- If this box is answered with 'Yes', all files are compared.
- The paths are synchronized with each other, are specified with the configuration tool 'SiteCfg' during the system configuration.
- The result is stored in the '<windowsuser-temp-path>\...\synchrox.tmp' file
  x can take the values 0 - 4; the file with the latest date is the result of the last synchronization run.
  (e.g. "`c:\users\TestUser1\AppData\Local\Temp\BRAUMAT\Synchro0.tmp`")
- If errors occur while copying, these are stored in the '<windowsuser-temp-path>\...\synchrox.err' files
  (e.g. "`c:\users\TestUser1\AppData\Local\Temp\BRAUMAT\Synchro0.err`")

Example 'synchrox.err'

```
 *** Line: 1 Path: pcu.*\ Depth: 1
```

```
File 'V:\.........\pcu.004\DIRINFO' can't be copied to '\\WIN7-32-DB-BM7\BM_PROJ\pcu.004\', error=53
```

Structure:

Comment lines start with '***'.

- Any further lines have the structure:
  File <Source drive> <Path> <File name> <Relationship> <Destination drive> <Windows error>
- <Relations> can be:
  "older" File exists but is out of date
  "does not exist" File does not exist on the destination drive
  "can't be copied to" File could not be copied to the destination drive

4.7 Scheduler

4.7.1 General

A dedicated Scheduler program is integrated in the system.

The Scheduler has the functions:

- hourly, daily or weekly start of applications
- Monitoring of applications and restart, if necessary.

4.7.2 Start of the Scheduler

The Scheduler can be started

- via the basic menu
- via the Autostart group of Windows,
- manually
- from the PCU Server. To do this, the following INI switch must be set in PCU_SERV.INI :

```
[Pcu_serv]
....
;Start and Check for running of ProcessScheduler "ProSched.exe"
RunProcessScheduler=1
```

The Start type depends on which applications should be started and monitored with the Scheduler.

The name of the application is 'prosched.exe' and it is located in the system directory.

Note

- When the scheduler is started, the configured start conditions are checked immediately and the applications are started accordingly.
- In case of the start types 'daily' and 'hourly' PROSCHED.EXE determines whether the previous start did not happen within the configured time period and tries to make up for this.
- In case of start type 'weekly' an application is started only if the requested day of week matches the actual day.

4.7.3 Configuration of the Scheduler

The Scheduler will be configured via the file prosched.ini in the system directory.

4.7.3.1 Scheduler Configuration

Display Scheduler window

Key:	[Config]
Input:	ShowWindow=0

Scheduler Cycle

The cycle for examining and starting applications can be configured. The value is indicated in seconds.

Key:	[Config]
Input:	CycleTime=30

4.7.3.2 Configure processes

List of processes

Key:	[Global]
Input:	Processes=XXProc

A list of processes can be entered here. The inputs should be separated by comma. The name allocated in the list need not conform with the starting applications.

One more section should be made in the file prosched.ini for each of the processes. The key name of the section should conform explicitly with the processes names in the process list. (Example: [XXProc])

Sections of applications

Key:	[xxProc]
Input:	Application=

The input can be made with path indication and application parameters. If no path is indicated the application will be searched according to Windows Standard.

Start types and parameters

Type=HOURLY, DAILY, WEEKLY	Facilitates the application call • Hourly • Daily (a particular minute of the day) • Weekly (a particular day of the week) For further definitions see 'Configuring Start time' below.
Type=PERMANENT	The inputs for day and time are not relevant for this type but should be defined as "Weekday=0" and "Reqtime=00:00".
Type=ONCE	Single call

Configure Start time

Type WEEKLY

Input:	WeekDay=6

This input is only relevant for the Start type weekly. The week day will be indicated. Sunday has the value 0, Monday the value 1 and Saturday the value 6. The inputs for hour and minute are not relevant for the WEEKLY type but should be defined as 'Reqtime=00:00'. The app. is started as soon as the correct weekday has been reached (starting at 00:00 hours) and the day of the last start is not the actual day.

Type DAILY

Input:	ReqTime=00:10

With the input the requested start time is configured. The time is indicated in hour:minute. The app. is started as soon as the correct time has been reached and the day of the last start is not the actual day.

Type HOURLY

Input:	ReqTime=00:10

For the Start type 'hourly' the hour value is not relevant but should be defined in the range of 0-23. The app. is started as soon as the correct minute has been reached and the day/hour of the last start does not match the actual time.

Exit application

Entry:	ShutDown=1

You can set the switch to determine whether or not the started application is exited at the end of the scheduler.

Parameterization of technology objects 5

5.1 Individual control

5.1.1 General to the individual control

Tasks of the individual control are triggering, interlocking, monitoring, and signaling of final controlling elements like valves, motors, pumps, flags, slide valves, etc.

Triggering commands can come from

- The equipment operations of the Units
- Connection programs
- Any other automatic programs (which aren't part of the System) or
- Screen operation
-

Interlocking programs guarantee that the necessary operation interlockings are considered.

These are effective for each type of triggering, i.e., in the automatic- and in the manual operation.

Basic logics like the interlocking of product- and cleaning valves, the dry safety of pumps, agitators, etc., need to be filed only one time. Equipment operations, connection programs, and any others concerning the individual final controlling are shorter and have much more overview by that.

However, the basic interlocking isn't a substitute for safety interlockings, which are to be realized outside of the control.

With a non-fulfilled operation, interlocking commands are sent to all final controlling elements for an "OUT"- position:

- Motor, Pump: "OUT"
- Valve, flap, solid: "Closed"

Triggering commands aren't considered.

On each position command the actuator must react within an individually preset time by a corresponding acknowledgement (1..255 Seconds). Either this reaction is executed

- Not at all
- delayed, or
- the reaction signal disappears without the corresponding command again

an actuator fault is recognized. This one is filed in the actuator state for the display and a disturbance message is sent for the released operating systems (OSs).

5.1.2 System functions

The System makes the block ICM individual final controlling element available. This one includes the controlling-, locking-, monitoring-, and signal logic. The available instances of this block are organized as follows (<Number of PCU before V7> / <Number of PCU V7>):

- ICM1: 255/256 final control elements of ICM group 1
- ICM2: 255/256 final control elements of ICM group 2
- ICM3: 255/256 final control elements of ICM group 3
- ICM4: 255/256 final control elements of ICM group 4

Furthermore there is the possibility to configure closing and off delays for controlling ICMs. These delays are only active if the final controlling element is in the automatic operation.

The configuration of the closing- and off delays has to be done in the data record of the ICM block.

Independent of the individual control by the System, an external manual level is realizable.

The System enables the bumpless transfer from and to the external manual level.

5.1.3 Interrelations

The operation of the ICMs is executed by the OS.

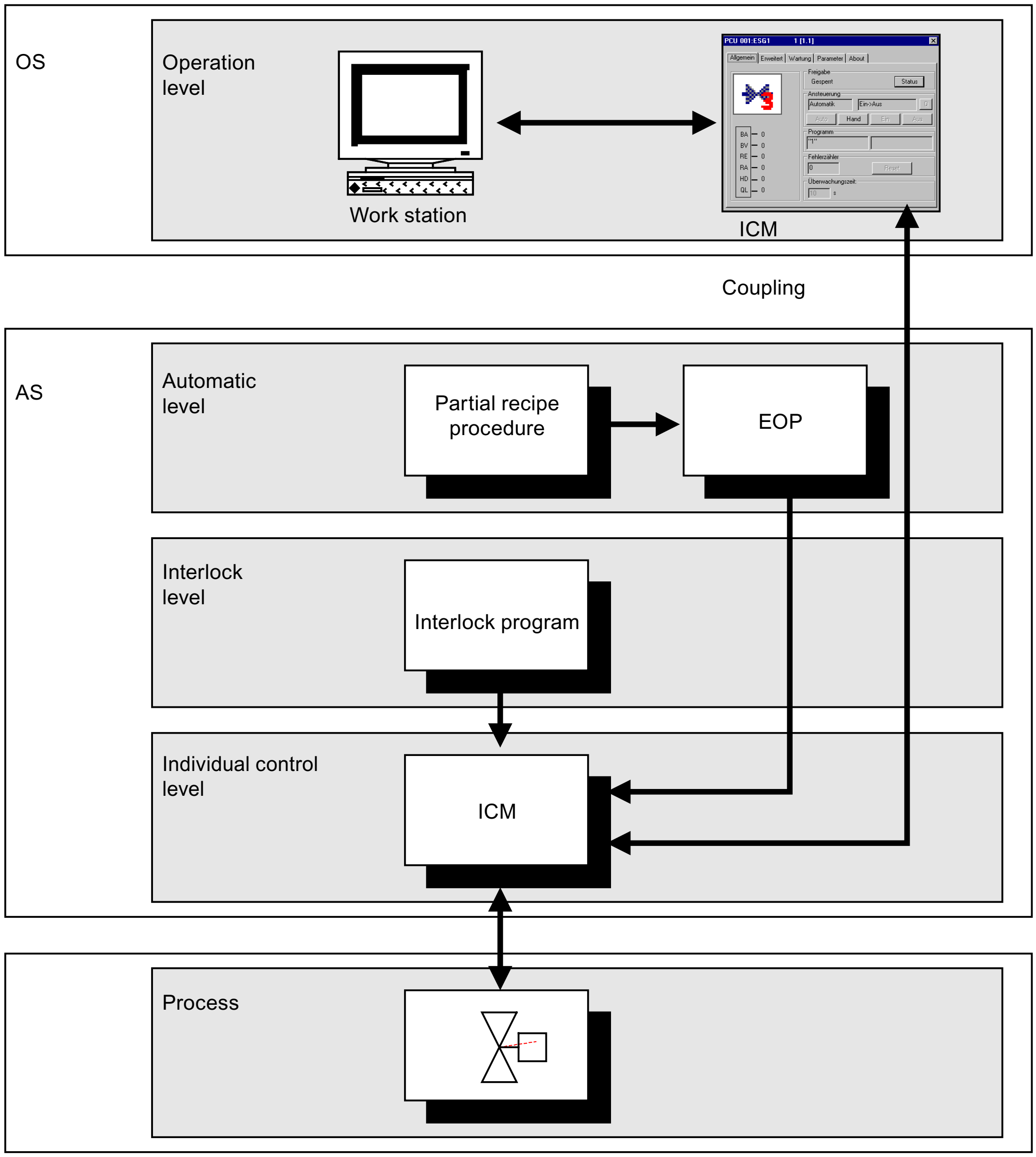
Controlling the ICM – blocks is executed in the AS:

- In the automatic program of the Units by the equipment operations
- Per operation via the screen
- By any other connection programs

In the ASs control-independent connection programs guarantee that the necessary interlocks are considered.

The conditions of the final controlling element are transferred to the operation System (OS) by the coupling for visualization.

The operation System (OS) itself transmits operation commands for the final controlling elements via coupling to the underlaid Systems (AS).


OS
Operation level
Work station
ICM
Coupling
AS
Automatic level
Partial recipe procedure
EOP
Interlock level
Interlock program
Individual control level
ICM
Process

5.2 Analog functions

5.2.1 General

For processing analog sizes in the frame of process automation the System makes a sequence of standard blocks available. By the corresponding configuration of the blocks and interconnection of blocks among each other the most different functions are realizable.

The following blocks that process analog values are included in the system:

ANA (DB734)	Memory location for an analog value. Serves, for example as the interconnection partner for other blocks.
ANAU (FB731)	Converts an allocated setpoint in electrical units and transfers it by the analog output module.
DFM0 (FB736)	Implements counters (analog value interface to the recipe system). Processing must be started explicitly (FC736).
DFM1, DFM2, DFM3 (FB737)	Timers, limit values/setpoints, decoders among other things (analog value interface to the recipe system) Processing must be started explicitly (FC737 f.).
DREIP (FB744)	Block for three point controller for simple arrangements.
INKU (FB729)	It is calculated the duration of open- or close commands and starts one of two timers accordingly.
KURVSW (FB747)	Allocation of setpoint curves (represented graphically or in tabular form). These setpoints can be transmitted to, e.g., a controller by curve scanning.
MEKO (FB728)	Monitors an allocated value on two limits.
MESS (FB727)	Reads a value from an analog input module and makes it available as an adapted physical variable to other blocks.
MULT (FB732)	Depending on the selected function an output value is formed by two input values or a flag bit is manipulated.
PID (FB730)	Controller block, which forms from setpoint W, actual value X, and influencing quantity Z a manipulated variable Y.
POLY (FB735)	Adapts an input value via six support couples. It is used for linearizing or as a trend calculator for indirect measurements.
VMON (FB750)	Comparison of target and actual value with the limit value Two hysteresis values and delay times can be defined for the upper/lower limits The setpoint and actual values to be compared are taken from another block (ANA, MESS, MULT, ...) via an interconnection to it.

Note

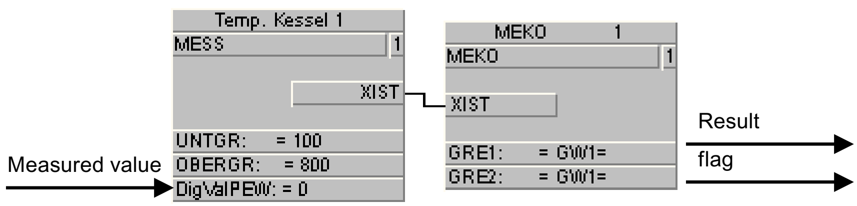
Analog value processing

All analog signal processing blocks of PCU V7 work with the data type REAL.

5.2.2 Examples

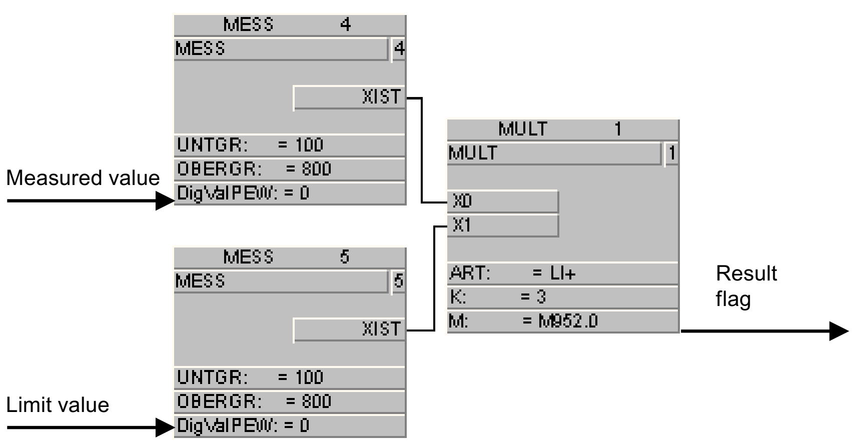
Monitoring a measured value of 2 fixed limits

Solution by connection of a block MESS with a block MEKO. Both comparison results (Limit 1 is violated or limit 2 is violated) are available as flags for step enabling processing.



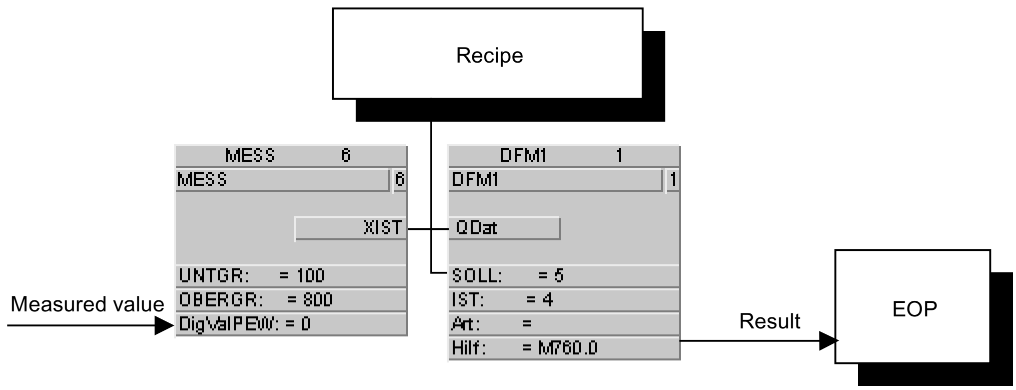
Monitoring of a measured value on an external limit

Solution by connection of a block MULT with the function LI+ with two blocks MESS for actual value and limiting value. The comparison result is available as flag.



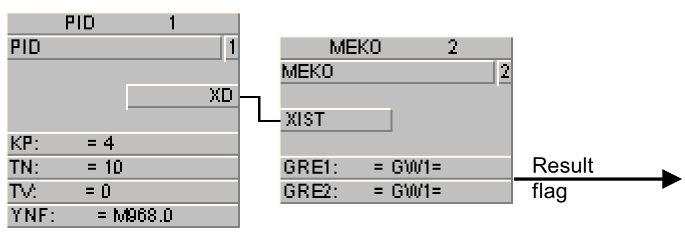
Monitoring a measured value on a recipe-dependent limit and formation of a step enabling criterion

Solution by interconnection of a MESS block with a DFM block and evaluation of the comparison in a technical operation.



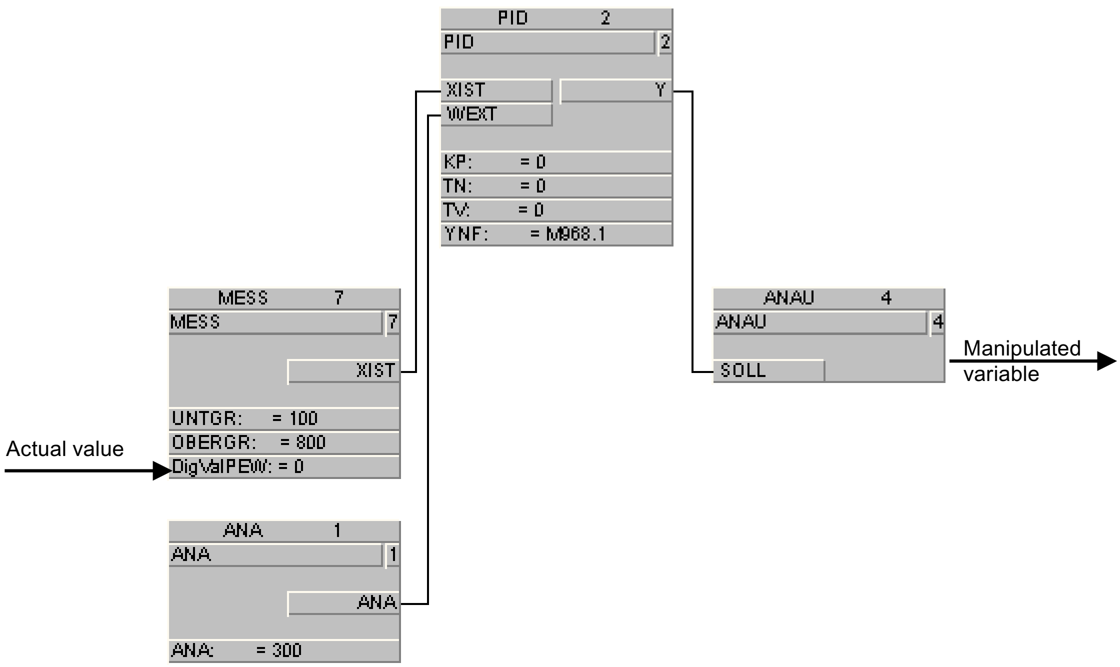
Monitoring of the deviation on a fixed limit

Solution by connection of the block PID with the block MEKO.



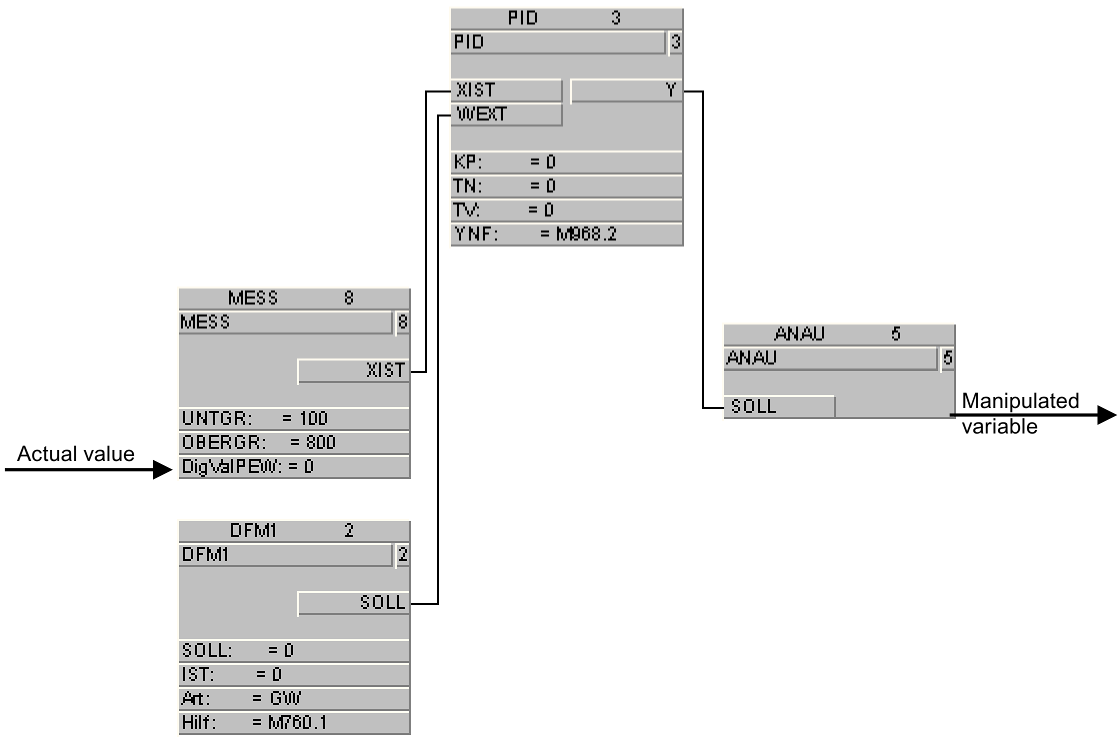
Continuous controller with a fixed setpoint

Solution by the usage of a block MESS for the actual value, of a block AFIX for the setpoint, of a PID-block and a block ANAU for the manipulated variable output.



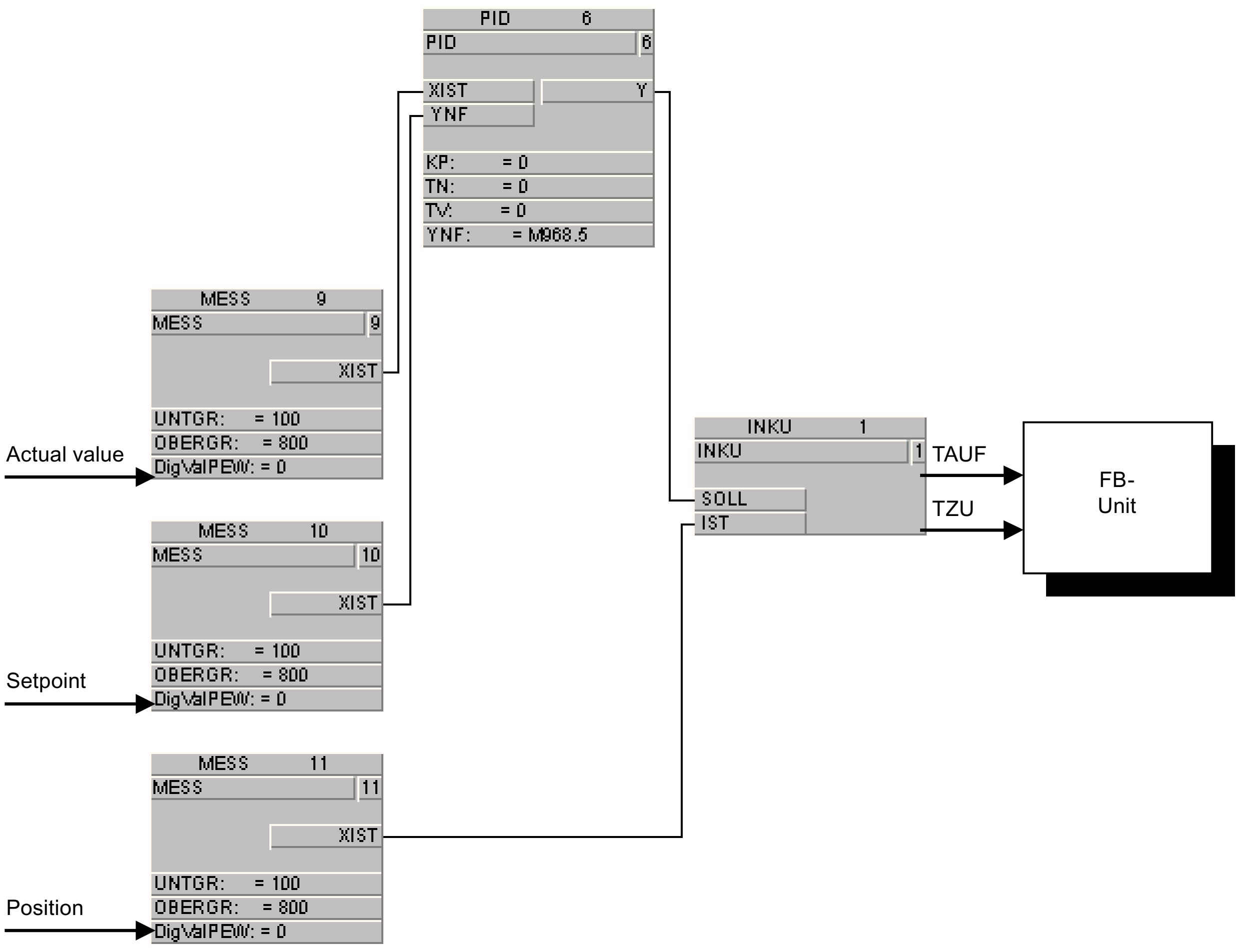
Continuous controller with recipe-dependent setpoint

Solution by the usage of a block MESS for the actual value, of a block DFM for the setpoint, of a PID-block and a block ANAU for the manipulated variable output.



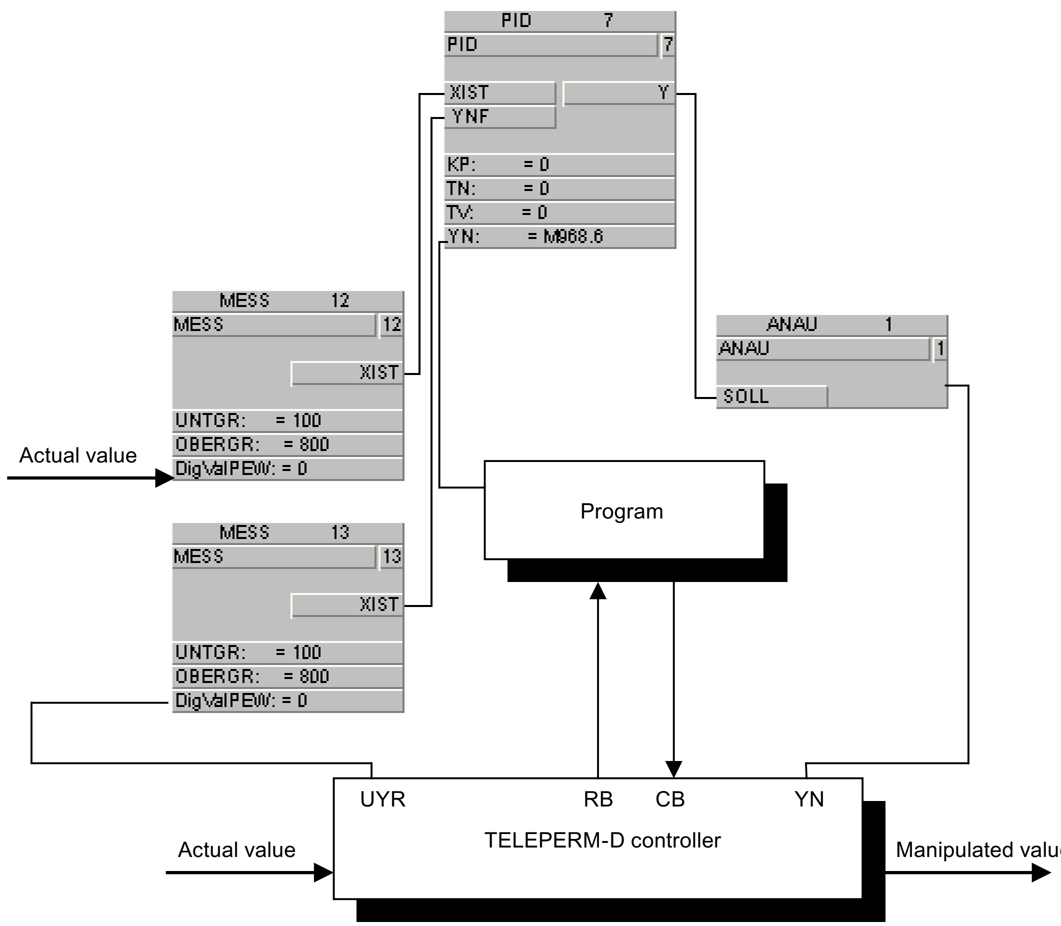
Step controller with external setpoint

Solution by the usage of three blocks MESS for the actual value, setpoint, and position acknowledgement, of a PID-block, of a block INKU, and one part of the program.



Controller with Back-Up

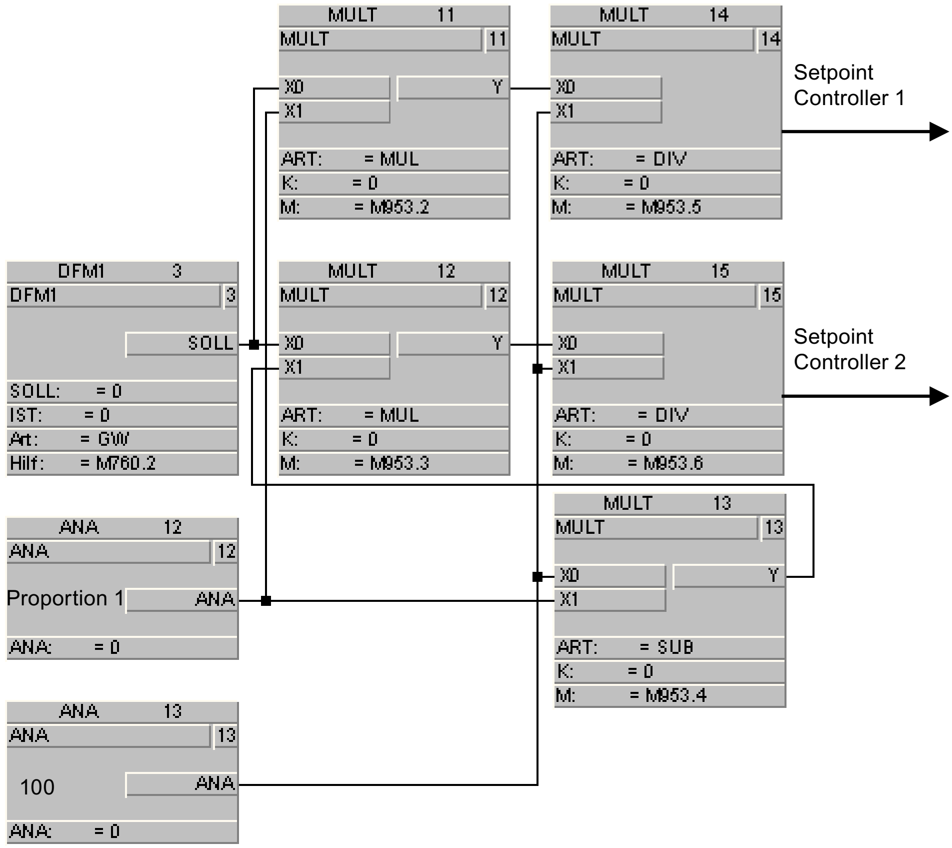
Solution by using two blocks AIN for the actual value and compensating value, of a PID-block, of an ANAU-block for the allocation of the manipulated variable, and of an external hardware controller.



Ratio Control

It is requested that two controllers should accept one adjustable proportion of a setpoint each.

Solution with two blocks AFIX (one for the number 100 and one for the setpoint of the controller 1), five blocks MULT (two DIV, two MUL, one SUB).



Explanation of the circuit:

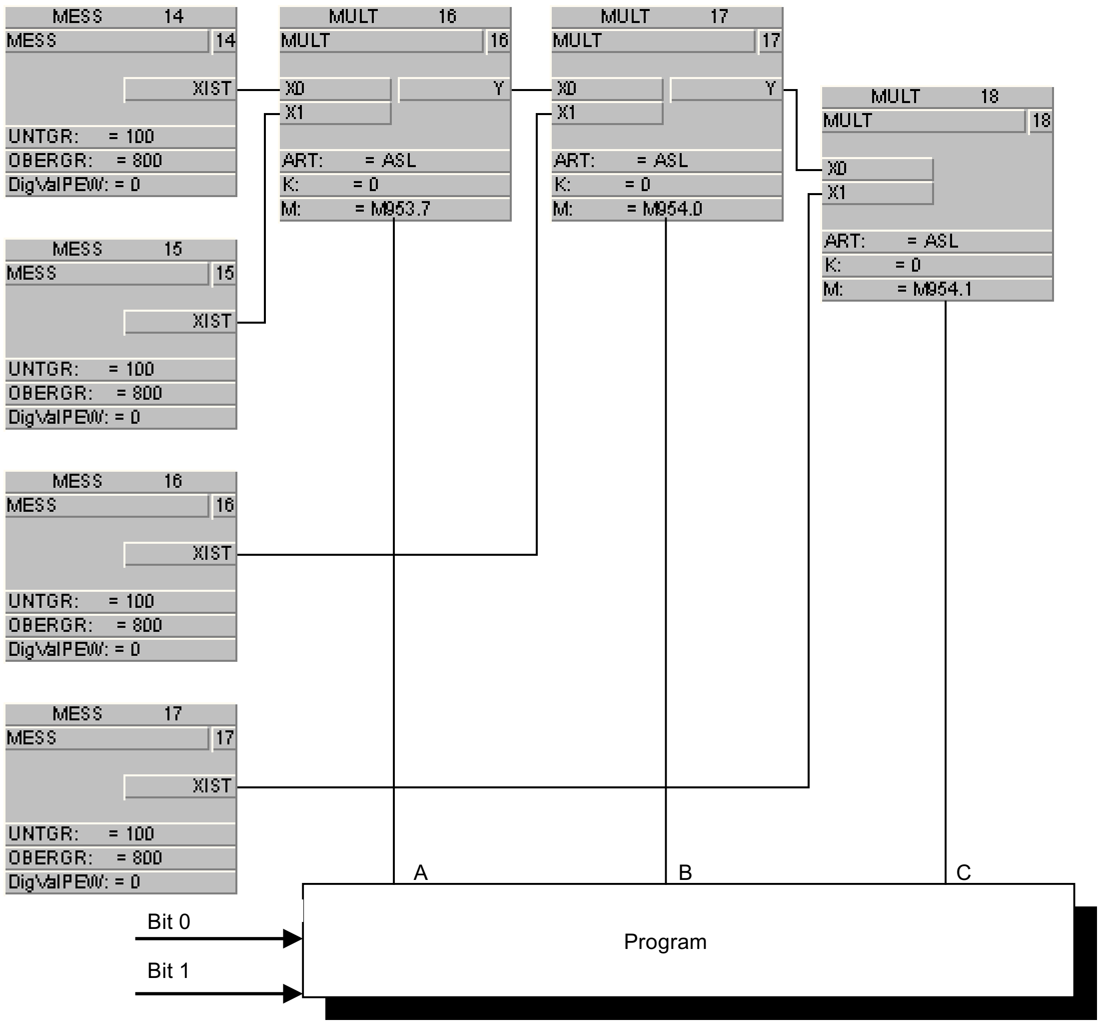
- As proportion 1 + proportion 2 has always to be 100, the proportion 1 is filed as a value and can be changed as operation. The operation program has to consider the limit 0 ≤ proportion 1 ≤ 100 has to consider the operation program.

Setpoints are calculated in the following form:

- Setpoint Controller 1 = Setpoint x proportion1 / 100
- Setpoint Controller 2 = Setpoint x (100 - proportion1) / 100

Selection of one out of 4 measured values, dependent on a selection counter

Solution with four blocks MESS, three blocks MULT in the function ASL, and one part of the program.

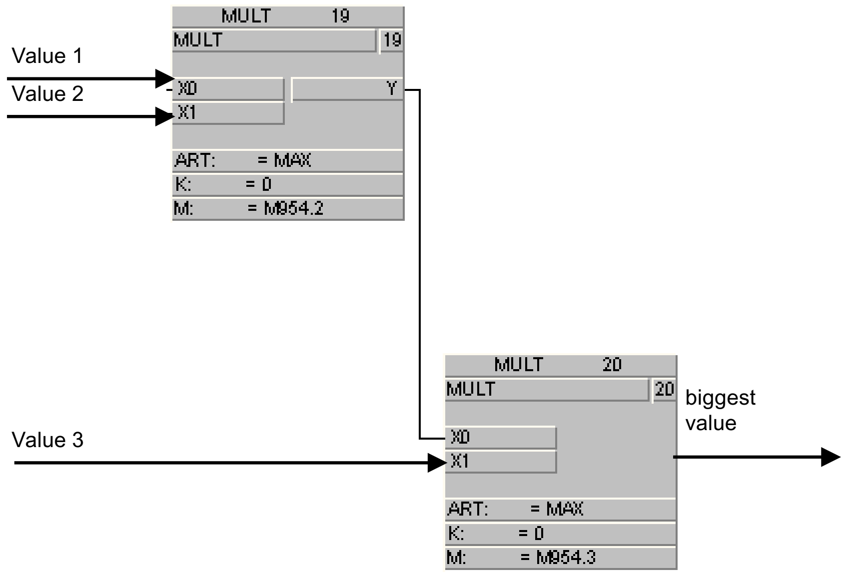


The following logic is valid for the program:

Bit 0	Bit 1	MESS	A	B	C
0	0	14	0	0	0
0	1	15	1	0	0
1	0	16	X	1	0
1	1	17	X	X	1

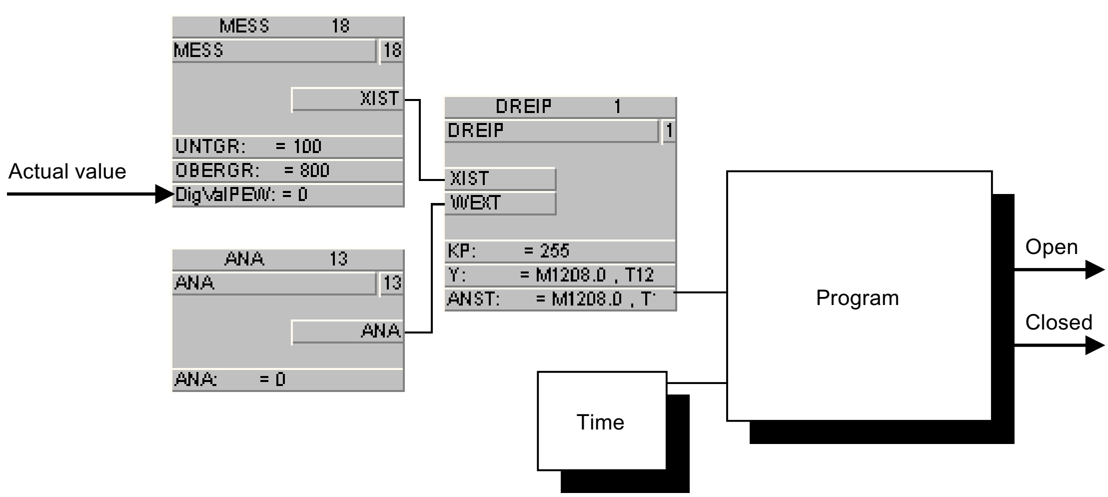
Selection of the biggest value of three

Solution with two blocks MULT with the function MAX.



Temperature- or pressure control with switching controller (On/Off or Open/Closed)

Solution by the usage of a block MESS for the actual value, one block ANA for the setpoint, of a THREP-block, and of a user program.



In the user program one of the outputs OPEN or CLOSED is controlled by queries of the Timers Control-bits (ANST).

Example:

Timer runs and ANST = 1	OPEN = 1, CLOSED = 0
Timer runs and ANST = 0	OPEN = 0, CLOSED = 1
Timer doesn't run	OPEN = 0, CLOSED = 0

5.3 Basic principles of parameterization

5.3.1 Overview configuration

One of the basic properties of the system is that different attributes of blocks, constant values, etc. are stored in certain lists.

These lists are structured. In a dialog, the process cell-specific attributes can be adapted in the lists. It is possible to access either the online values (SIMATIC data block) or offline values (IOS file) of the attributes.

Process cell specific attributes include: Names of individual control modules, ranges of measured values, sort names, dimensions, etc.

5.3.2 Structure of the classes

A BRAUMAT/SISTAR class consists of:

- A number of instances of the class
  All instances of a class are stored in a SIMATIC data block. The attributes can be entered using the "Parameterization" application. An instance is specified uniquely by PCU number, class and instance number. The instance number starts with 1 and goes to the maximum value.
- An FB in the PCU that processes the instances (exceptions: e.g. FIXV, SPEVAL, ...)
- One name for each instance
  All instance names of a class are stored in a text data file on the IOS. You could enter text via the application "Text parameterization". There is one line in a text file for each instance name. The line number corresponds to the instance number.
- One description of each instance
  All descriptions of a class are stored in a text data file on the OS. You could enter text via the application "Text parameterization".
- An address description of the input / output assignment of each instance
- A location description within the process cell of each instance
- One assigned process image to each instance
- One note to each instance

5.3.3 Structure of the parameter records

The parameter records of a class put together in a list correspond to a data block in the SIMATIC system.

Although the user doesn't access individual parameters by the help with data block number (DBn) and the data word number (DWn), the general list structure should be explained.

Global data of the "AIN" class:

	Name	D.Type	A.T...	Value	Comment	Address
1	MessAnz	INT	ENG	512	Number of AIN records	DB 727.DBW 8
2	DS_Len	INT	SYS	108	Data set length	DB 727.DBW 4
3	Offset	INT	SYS	300	Offset first DS	DB 727.DBW 2
4	MaxDS	INT	SYS	512	Maximum number of DS	DB 727.DBW 6
5	OffsRun	INT	SYS	100	Offset runtime copy	DB 727.DBW 10
6	Commissioning	BOOL	ENG	TRUE	Commissioning is active	DB 727.DBX 14.0

The "MessAnz" attribute specifies how many instances have been switched active for this class. The length of the dataset is 108 bytes, i.e., one parameter record is 108 bytes long. The Offset to the 1.DS indicates the starting point of the first data record. The maximum DS amount indicates the maximum amount of the possible objects of the class. The Offset to the running time copy indicates the address in the data block where the corresponding data record is copied that is processed at the running time.

Instance data of the "AIN" class:

The figure below shows the parameter record of the first "AIN" instance.

	Name	D.Type	A.T...	Value	Comment	Address
1	Sequence	ENUM	ENG	<0>	Assigned unit	DB 727.DBB 360
2	DigValRef	GREF	ENG	PEW 512	Reference to measured value	DB 727.DBL 300
3	DigValStart	INT	ENG	0	digital start value	DB 727.DBW 318
4	DigValEnd	INT	ENG	27648	digital end value	DB 727.DBW 320
5	ConvFuncNo	INT	ENG	0	Conversion-FB (-1=Raw value...	DB 727.DBW 316
6	ProcValStart	REAL	ENG	0.000	Process value begin	DB 727.DBD 322
7	ProcValEnd	REAL	ENG	120.000	Process value end	DB 727.DBD 326
8	ProcValAct	REAL	RT	0.000	Current process value	DB 727.DBD 330
9	UpperLimit	REAL	ENG	0.000	Upper limit	DB 727.DBD 344
10	UpperLimitH...	REAL	ENG	0.000	Hysteresis upper limit	DB 727.DBD 352
11	UpperLimitH...	BOOL	ENG	TRUE	Hysteresis upper limit, ope...	DB 727.DBX 362.3
12	LowerLimit	REAL	ENG	0.000	Lower limit	DB 727.DBD 340
13	LowerLimitH...	REAL	ENG	0.000	Hysteresis lower limit	DB 727.DBD 348
14	LowerLimitH...	BOOL	ENG	TRUE	Hysteresis lower limit, ope...	DB 727.DBX 362.2
15	LimitErrDel...	INT	ENG	0	Delay time error message (s...	DB 727.DBW 356
16	SimOn	BOOL	ENG	FALSE	Simulation	DB 727.DBX 362.0
17	OnValErrUse...	BOOL	ENG	FALSE	In case of fault set actual...	DB 727.DBX 362.5
18	EnableMsgErrUL	BOOL	ENG	FALSE	Enable error output upper l...	DB 727.DBX 362.6
19	EnableMsgErrLL	BOOL	ENG	FALSE	Enable error output lower l...	DB 727.DBX 362.7
20	MsgLock	BOOL	ENG	FALSE	No error message for underf...	DB 727.DBX 363.0
21	LimitProcVal	BOOL	ENG	FALSE	Limit to start and end value	DB 727.DBX 363.1
22	SmoothProcVal	BOOL	ENG	FALSE	Smooth measured value	DB 727.DBX 362.4
23	ResetSmoothPVs	BOOL	ENG	TRUE	Reset measured value bufer	DB 727.DBX 364.4
24	ValueGrpError	BOOL	RT	FALSE	Measured value fault	DB 727.DBX 362.1
25	ErrUpperLimit	BOOL	RT	FALSE	Error upper limit	DB 727.DBX 363.3
26	ErrLowerLimit	BOOL	RT	FALSE	Error lower limit	DB 727.DBX 363.4

5.3.4 Connection of parameters

Core statement

Some parameters of the system blocks are designed for data transport. Via these interconnectable parameters, values can be taken from other data sources or written to other data sinks. The volume and data direction depends on the PCU version:

- PCU <V7
  The QUEL data type only allows the reading of parameters of other blocks. QUEL source access is only possible at the IOS end to attributes of technical objects, absolute addressing (DB.DBW, PIW, PQW) is not possible.
- PCU V7
  The QUEL data type is still available for reasons of compatibility but is being replaced technically by references. References can be read (GREF = Get REFerence) or write ((PREF = Put REFerence) oriented and are not restricted in terms of the addressable memory area.

Sources and references are created in the "Parametrization" application. If you double click on the value of a GREF/PREF parameter, the "Select data source" dialog opens. Here, there are two tabs in which you can either select an object or select an absolute address; the possible variants need to be adapted depending on the parameters and the PCU version.

Since symbolic object referencing (e.g. "AIN,1,XIST") is only known on the IOS and the actual data transport is handled by the PCU runtime system, internally interconnection partners are always addressed absolutely (for example: DB727.DBW330).

Format descriptions

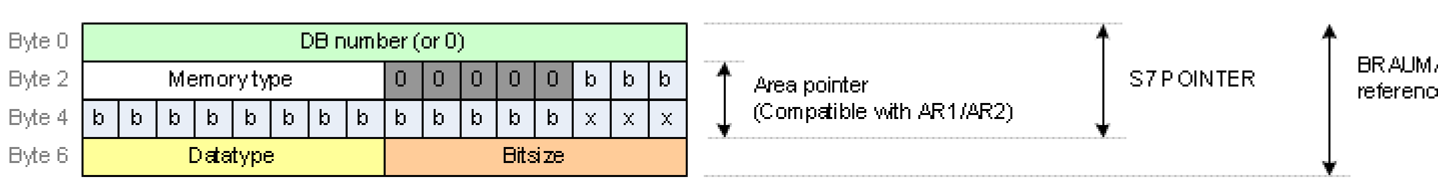
Sourcing PCU <V7

Sourcings are 4 bytes long:

Byte 0	DB number
Byte 2	Byte offset

Reference PCU V7

References are 8 bytes long:



You will find a description of the memory type (byte 2) and the data type (byte 6) in the online help of the SIMATIC manager in "Format of the Parameter Type ANY".

Note

Access to I/O areas

In contrast to the S7 pointer or S7 ANY pointer, references enable access to peripheral inputs/outputs. For this, the interconnection dialog sets bit 3, which is not used by Simatic, in byte 2 (memory type). The AS runtime functions take this bit into consideration, and it is masked out before loading to an address register. If you yourself want to resolve references, you may need to take access to I/O addresses into account.

Recommended implementation:

```
      // Detect periphery access (not an official area pointer
element!)
      L     Ref.dwPtr
      L     DW#16#08000000 // Mask for "periphery access"
      UD
      L     0
      <>D
      =     xUsesPI

      L     Ref.dwPtr
      L     DW#16#F7FFFFFF // Inverted mask "periphery access"
      UD
      T     dwPtr
      LAR2
      ...
```

Note

Periphery access errors in CPU diagnostics buffer

If GREF/PREF references are interconnected to non-existent I/O addresses, corresponding error entries are made in the CPU diagnostic buffer, caused by system function **FC769 "BmSetRefValGenericFC"**.

The purpose of the following information is to help you troubleshoot incorrectly configured GetREF and PutREF attributes in the system block classes:

- Determine the access address and access type read / write / word / bit in the diagnostic buffer.
- Browse the possible parameter assignments for the access type and check the access address.
- The table below shows all references of the block classes that could be effective.

List of classes with GREF/PREF attributes (PCU V7):

Block class	Reference type GetREF / PutREF	Attributes	Possible P access error
AOUT	GREF	SetpointRef	read / word
AOUT	PREF	DigValRef	write / word
DFM 1...8	GREF	ActualValueRef	read / word
THRESTEP	GREF	ProcValRef, ExtSpRef	read / word
ICM 1...4	GREF	RefInterlock, RefFbk0, RefFbk1	read / binary
ICM 1...4	PREF	RefCtrl	write / binary
INCO	GREF	SetpointRef, ActualValueRef	read / word
CURVSCAN	GREF	ExtPtr	read / word
LINE	GREF	MaterialIDRef	read / word
MVC	GREF	ActualValueRef	read / word
AIN	GREF	DigValRef	read / word
MULT	GREF	ProcVal1, ProcVal2	read / word
PID	GREF	ProcValRef, SpExtRef, DisturbValRef, FollowUpValRef	read / word
POLY	GREF	ProcValRef	read / word
PULSE	PREF	Pxx_PulseSigRef (xx= 01...32)	write / binary
SEQUENCES	GREF	PermCondRef	read / binary
SLB	GREF	Top_PulseSigRef, Top_CmdActivateRef, Top_FbkRef, Top_InterlockRef Btm_PulseSigRef, Btm_CmdActivateRef, Btm_FbkRef, Btm_InterlockRef	read / binary
SLB	PREF	Top_CtrlRef, Btm_CtrlRef	write / binary
TANK	GREF	FullDetectorRef, EmptyDetectorRef, Tx_IsValidRef (x=1...8)	read / binary
TANK	GREF	Tx_ValueRef (x=1...8), PressureRef, QuantityRef, MaterialIDRef, QuantitySPRef	read / word
TANK	PREF	TankStatusOutRef	write / word
TIMER_01, TIMER_02	GREF	RefIn	read / binary
TIMER_01, TIMER_02	PREF	RefOutPos, RefOutNeg	write / binary
VMON	GREF	SourceSP, SourceAV	read / word

Error evaluation of references and display in signal status (S7-1500)

When references are resolved, a local error handling takes place using the S7-1500 system function "GET_ERR_ID". If an access error occurs, its identifier can be found in the corresponding "_ST" attribute (if necessary, the "extended attributes" must be shown and the SYS filter must be active directly underneath the reference in the parameter assignment).

For further details, refer to the TIA Portal documentation for this system function.

The following table shows the possible values that can be output in the signal status of the references:

xxxRef_ST (hexadecimal)	Description
0	No error
2503	Invalid pointer
2520	Invalid STRING:
2522	Read error: Operand is outside the valid area
2523	Write error: Operand is outside the valid area
2524	Read error: Invalid operand
2525	Write error: Invalid operand
2528	Read error: Data alignment
2529	Write error: Data alignment
252C	Invalid pointer
2530	Write error: Data block
2533	Invalid reference used
2538	Access error: DB not available
2539	Access error: Incorrect DB used
253A	Global data block does not exist
253C	Incorrect specification or function does not exist
253D	System function does not exist
253E	Incorrect specification or function block does not exist
253F	System block does not exist
2550	Access error: DB not available
2551	Access error: Incorrect DB used
2575	Error in the program nesting depth
2576	Error in the local data distribution
2577	Block property "Parameter passing via registers" is not activated.
25A0	Internal error in TP
25A1	Tag is write protected
25A2	Invalid numeric value of tag
2942	Read error: Input
2943	Write error: Output

See also

System blocks and parameters (Page 214)

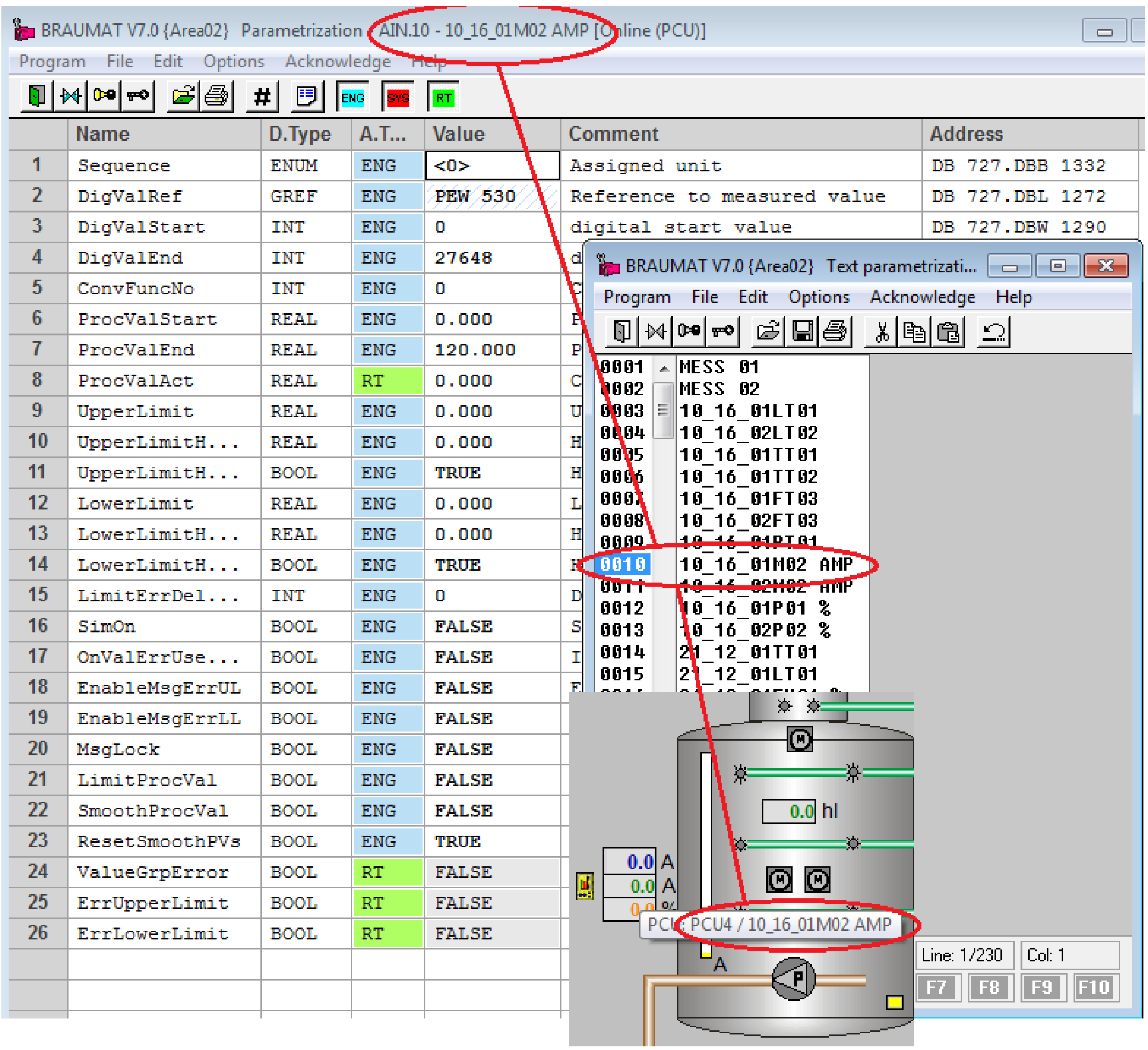
5.3.5 Block names

Each instance of a technological block or class can be given a process cell-related name in the system. This means that the instances can be identified easily by the technologists.

The instance names are stored in a class-related text file in the project directory on the IOS.

The input of these names can be executed individually as well as directly in the text file.

The following figure shows the relationship between the text parametrization and various display positions of an instance name.



5.4 Application parameterization

5.4.1 Working with the application

Basic principle

Working with the **Configuration** application involves the following activities:

Selection of the class file

You have the following options for selecting the class file:

File menu	Function key	Toolbar
Open	F2	

In the dialog box **Open class file** the following possibilities for selection are:

- **Data source** Online (PCU) or offline (hard drive)
- **Area** Symbolic names of the PCUs
- **Class** e.g. AIN, ICM1, sequences etc.

Selection of a data set

In a list box all available data sets are listed with their data set names.

A dataset is selected using the **Select dataset** function in the **Edit** menu, by pressing function key **F3** or by clicking the following icon in the toolbar

Selection of next data set

You select the next dataset using the **Next dataset** function in the **Edit** menu or by pressing the **NUM+** key.

Selection of previous data set

You select the previous dataset using the **Previous dataset** function in the **Edit** menu or by pressing the **NUM-** key.

Update values

Switching values on and off is selected by the function **Update value** in the menu **Processing** or by activating the function key **SHIFT+F3.**

Edit data set name

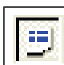
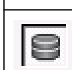
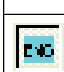
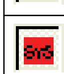
After selecting the function **Data set name** in the menu **Processing** or by activating the function key **SHIFT+F5** the name of the current data set can be edited in a dialog box

Parameterize data sets

As soon as a data set is selected, its individual parameters/attributes are displayed as a table in the Client area of the application as follows:

Column	Meaning	Value range
Name	Attribute name	System name – not editable
D.Type	Data type	System type – not editable
		Refer to the section Parameter types (Page 204)
A.Type	Access type	System type – not editable "ENG" = Configurable attribute "SYS" = System attribute "RT" = Runtime attribute
Value	Attribute value	Value format according to "D.Type"; Editing only possible for A.type"ENG" D.Type "Quell" / "GREF" / "PREF": after selection, a dialog box appears for selecting the source / destination object. which is collected from class, instance and attribute D.Type "Step": Input of Step STL operation "UM x.y"
Comment	Attribute description	System description – not editable
Address	Address in PCU	Data block / Data element offset

Filter functions in the Icon bar

	Show/hide extended attributes Via the menu item **Options Extended attributes** or with this button these attributes can be switched on/off. This is also valid for the call in the Global Data.
	Display LOG attributes With the menu item **Show LOG attributes** in the **Options** menu and with this button, the attributes with data logging for the "Replay-Mode" can be filtered out. When doing this all other attribute switches should also be set otherwise you may not obtain the full LOG attribute list.
	Show/hide engineering attributes
	Show/hide system attributes
	Show/hide runtime attributes

Access type

The possible accesses to the data set attributes within a block class are fixed. The block descriptions in chapter System blocks and parameters (Page 214) therefore include the extra column "Access", which contains one of the following labels for each parameter:

Label	Description
NL	Normal: • Always visible • Read and write access
RD	Read: • Always visible • Read-only access
HI	Hidden: • Only visible if the "Show extended attributes" filter is active • Read and write access
HR	Hidden & Read: • Only visible if the "Show extended attributes" filter is active • Read-only access

Print configuration

You have the following options for printing the parameterization:

File menu	Function key	Toolbar
Print	F4	

The following can be printed:

- the entire class,
- the current data set or
- several data sets (from ... to).

Copying of data sets

Copying DS to a destination DS:	• The current data set can be copied to a destination data set. • The selection of the function occurs via Copy to ... in the menu processing or by activating the function key F8. A dialog box appears. • In a list box one can be selected from the available data sets.
Copying from a source DS to a destination DS:	• You can copy from a selectable source data set to the current data set. • The function is selected using Copy from... in the Edit menu or by actuating function key Shift+F8. A dialog box appears. • In a list box one can be selected from the available data sets.

Generate data set

By calling the function **Generate data set** in the menu **Processing** or by activating the function key **F5** data sets can be generated new.

Text file

By calling the function **Text file** in the menu **Processing** you will be forwarded to the corresponding text configuration.

Configuration

The configuration is selected via the function **Data set number** in the menu item **Processing** or by the function key **F6.** A dialog box appears. The data set number can be preset with the configuration, increased or reduced. However, it should be indicated in the value range, otherwise the input with an error message is refused.

The configuring is only then permissible if during the opening of the class file the data source "Offline (hard drive)" was selected.

Global data

The selection occurs via the **menu item Data set** in the menu **Processing**. A dialog box appears in which the **Global Data** should be activated.

The function **Global Data** causes the signal of class-specific data.

Memo function

The selection of the note field occurs via the function **Memo** in the menu **Options** or by activating the function key **F10**.

Texts can now be entered for each data set. For each single data set there is the corresponding file:

- ...\PCU.xxx\NOTETXT.y\class\class.nnn

If there is no file for a data set yet, a new one is created.

<y>	Language setting (0=German, 1=English, etc.)
<class>	Object class
<nnn>	Data set number

The editing is protected by a password. After a successful password query the text editor can be used. After activating the button **OK**, the new file is stored and the dialog box is closed.

5.4.2 Parameter types

D.Type PCU before Version V7	D.Type PCU from Version V7	Lower limit	Upper limit	Description
BOOL	BOOL	FALSE	TRUE	1 Bit
BYTE	BYTE	B#16#00	B#16#FF	Hexa number 8 bit
WORD	WORD	W#16 #0000	W#16 #FFFF	Hexa number 16 bit
DWORD	DWORD	DW#16 #00000000	DW#16 #FFFFFFFF	Hexa number 32 bit
USINT	USINT	0	255	'Unsigned short integer' 8 bits
INT	INT	-32768	+32767	Integer number 16 bit
DINT	DINT	-2 147 483 648	+2 147 483 647	Double integer number 32 bit
REAL	REAL	-99 999 997 952	999 999 995 904	Floating-point number 32 bit
STEP	STEP			Logic command in the Notation of STL The following are supported. A <I, Q, M, DBnnn.DBX> x.y AN <I, Q, M, DBnnn.DBX> x.y
QUEL	QUEL			Address of a data source class/attribute (e.g.. "ICM0,12, Stat") **Note:** You can only use flag bits in the range from 0.0 to 2047.7 for source connection
---	GREF PREF			The address of a data source generally replaces QUEL as of PCU version V7 (G=Get, P=Put) Default = (NULL); double-clicking opens the "Select data source" dialog; the reference is deleted with "Del".
---	OREF			Reference to object instance
---	ENUM	0	MAX_DINT	Enumeration, text string of a whole number
---	DT			Date and time
---	STRING[n]			Character string with length <n> (max. 254)
---	BLOB			Binary data of the length <n> bytes (max. 1024)

5.5 Application text parameterization

5.5.1 Functionality

The application **Text configuration** is used for configuring the user texts and for printing the text configuration.

Client area

In the Client area the content of the selected user text file via which the current line can be edited is represented line by line.

Column 1: Continuous numbering of the single lines

Column 2: User text

5.5.2 Working with the application

Basics:

Working with the **Text configuration** application involves the following activities:

Open a text file

You have the following options for opening the text file:

File menu	Function key
Open	F2

A modal dialog box appears following selection. The following selection can be made in different list boxes:

Language:	Language with global or PCU-related texts,
PCU:	[Global] → Global texts '<proj-path>\texte.x\...' PCUnnn → PCU-related texts

Type	Type with PCU-related texts • Instance → '<proj-path>\pcu.nnn\texte\...' or '<proj-path>\pcu.nnn\texte.x\...' (see "SiteCfg/Plant settings/Projection languages") • Instance.X → '<proj-path>\pcu.nnn\texte.x\...' • Address → '.<proj-path>\pcu.nnn\Address.x\...' • Description → '<proj-path>\pcu.nnn\Desc.x\...' • Picture file '<proj-path>\pcu.nnn\IMAGE\..." • Location → '<proj-path>\pcu.nnn\Location.x\...'
File name:	Text file '...\<Class>.txt'

Confirm with **OK** to load the selected text file.

Note

PCU-related text attribute

Additional text attributes of the objects can be configured in addition to the module instance names. These are displayed in the process image with opened faceplate of the module instance in the "About" tab.

With these attributes it is an additional object description for the operator using the following attributes

- Description → long text/explanation of the object
- Address → one or more pieces of address information on input / output signals or flag bits
- Location → describes the location of the installation
- In addition a note can be created and displayed for every object.

Saving a text file

After selecting the **Save** menu item in the **File** menu, by activating the **F3** function key or clicking the **Save** icon, the called text file is saved.

Print the text parameters

You have the following options for printing:

File menu	Function key
Print	F4

A modal dialog box appears following selection. The following can be printed:

- the current file
- the current area (= all text files of the selected area)

If you activate the button **Installation**, the dialog box **Printer setup** appears, where you can select a printer, change the size of paper, etc.

File parameter

After selecting **Parameter** in the menu **File** the formatting parameters of the current file can be changed. A dialog box appears with the following information:

- Characters per line
- Number of lines (max. number of instances)
- Display first line (Yes / No)

This formatting information is stored in a text file "Deftext.ini" for each type and language subdirectory. If no entries found in the corresponding "Deftext.ini" when opening a certain text file (type, class and language) (or "Deftext.ini" is missing), the system prompts you to set these parameters.

Process the file

The following functions can be selected under the menu item **Processing** in the text configuration by activating the corresponding function keys or by selecting the icon in the button bar:

Edit menu	Function key
Back	CTRL+Z
Cut out	F5
Copy	F6
Insert	F7
Delete	F8
Insert line	Shift+F7
Delete line	Shift+F8

Engineering language:

If you want to use more than one engineering language then the corresponding language-specific files need to be configured for the different types and text files for the engineering languages configured (selection via "Language" listbox).

5.6 STEP 7 Use symbol table

5.6.1 Extract data from the Step 7 symbol table

Extracting text attributes from STEP 7 symbol table/PLC tag table

The symbol definitions in the SIMATIC S7 programming package (STEP 7 V5.x / STEP 7 (TIA-Portal)) that arise during the course of configuring the system are completely separate from the instance texts in the text parametrization initially. For this reason, without measures for export and import of symbols, the acquisition effort would be doubled.

With the help of the "ZL Synchronizer" application, an exported symbol table of the STEP 7 programming package can now be transferred step by step to the text parametrization of the object classes. Conversely, certain information from the text parametrization can be written back to the STEP 7 symbol table.

Note

Behavior of PLC tag table in the TIA Portal:

The back transfer of instance texts from BRAUMAT/SISTAR to the PLC tag table is subject to the following limitations:

- During the import, existing entries are overwritten if they have the same name as the entries being imported.
- If reference to PLC tags or constants are already present in the project, these references are automatically updated during the import. The references are updated based on the name of the PLC tags and constants.
- An import of externally modified symbols for existing and utilized PLC tags would lead to duplicate entries and be reported as an error.

Behavior of the symbol table in STEP 7 V5.x:

- In the case of S7-400/STEP 7 V5.x, it is possible to choose the import behavior in overwrite mode (address or symbol).

Conclusion:

- In the case of S7-1500 / TIA-Portal, only the **S7 Project → BRAUMAT/SISTAR Project transfer** can be fully utilized.
- The back transfer should be used at most for changes in the comment column.

Requirements

- The PCU-related assignment list/PLC tag table is used to display the symbol information. It must be exported in the symbol editor (STEP 7 V5.x) or the PLC tag table (TIA Portal) after changes.
- The mnemonic setting in STEP 7 / TIA Portal must match the currently valid system engineering language (if an engineering language is not selected, this is the current system language).
  Engineering language = German → STEP 7 mnemonic = Deutsch
  Engineering language ≠ German → STEP 7 mnemonic = English/International

	PCU type S7-400 with STEP 7 V5.x	**PCU type S7-1500 with STEP 7 TIA Portal**
Export/import file and format	s7_sym.seq	PLCTags.xml
Path when a change of engineering language is not available	'<proj-path>\pcu.nnn**s7_sym.seq**' or '<proj-path>\pcu.nnn**texte \s7_sym.seq**'	'<proj-path>\pcu.nnn**PLCTags.xml**' or '<proj-path>\pcu.nnn**texte \PLCTags.xml**'
Path when a change of engineering language is available	'<proj-path>\pcu.nnn**texte.x \s7_sym.seq**'	'<proj-path>\pcu.nnn**texte.x \PLCTags.xml**'

- The "ZL_Synchronizer" application is capable of handling multiclients, i.e. the text parameterization of any PCU can be edited from an IOS station. For this the associated area IOS servers must be reachable via the network.

Starting the application and selecting the symbol and text files

The application is called in the main menu on the "Engineering" tab.

Initially a selection dialog for the subsequent text comparison is opened using the 'File / Open' menu.

The entries in the dialog window "Select S7 symbol file" are saved using 'OK' and are therefore available again when you next open the file. The following selection fields are available:

1. Area:
   Selecting the AREA in the listbox
2. PCU:
   Selecting the PCU in the listbox
3. File:
   The specified symbol file based on the above table is specified by the application.
4. Class:
   Selecting the object class

5. Parameter:
   Selection of the "leading" object attribute in the symbol table. This alone is used as a reference for the comparison in both lists. The standard shortcuts specified upon system delivery can be adjusted to the abbreviations used in the project's symbol table where required.
   The following block classes and parameters are available:

Class	Parameter	Comment
SEQS	Flag	Trigger flag
DFM0	Flag (output) Flag (input)	Result flag Counter input
DFM1 ... DFM8	Flag	Result flag
THRESTEP	Flag	Direction flag
ICM1 ... ICM4	Command automatic Operation interlock Load output Feedback Off Feedback On	BA flag BV flag QL output RA input RE input
MVC	Flag limit value 1 Flag limit value 2	Limit value flag 1 Limit value flag 2
MSG	Flag	Trigger flag
AIN	Upper limit flag Lower limit flag	Upper limit value flag Lower limit value flag
MULT	Flag	Control/result flag
PID	Flag	Adjustment flag
EOP	Function **(only available for S7-400!)**	EOP-FC
Sequences	TA-FB **(only available for S7-400!)** Process control flag Permanent condition flag Manual mode flag	TA user function block TA process flag TA permanent condition Manual group flag
TIMER01,02	Flag input Flag negative edge Flag positive edge	Trigger flag Output flag negative Output flag positive

6. Address:
   The address area of the selected parameter is displayed in the 'From' / 'To' fields. Initially these are the standard addresses of the relevant object attribute specified with system delivery. In the event that these have been changed in the project symbol table then the new addresses must be entered here. The default address assignment of the system can be restored with the "Default" button.

Following confirmation by pressing OK the selection criteria selected are displayed in the main view. Based on this selection, the corresponding texts from the symbol table and from the text parametrization are assembled for the selected object in the table below.

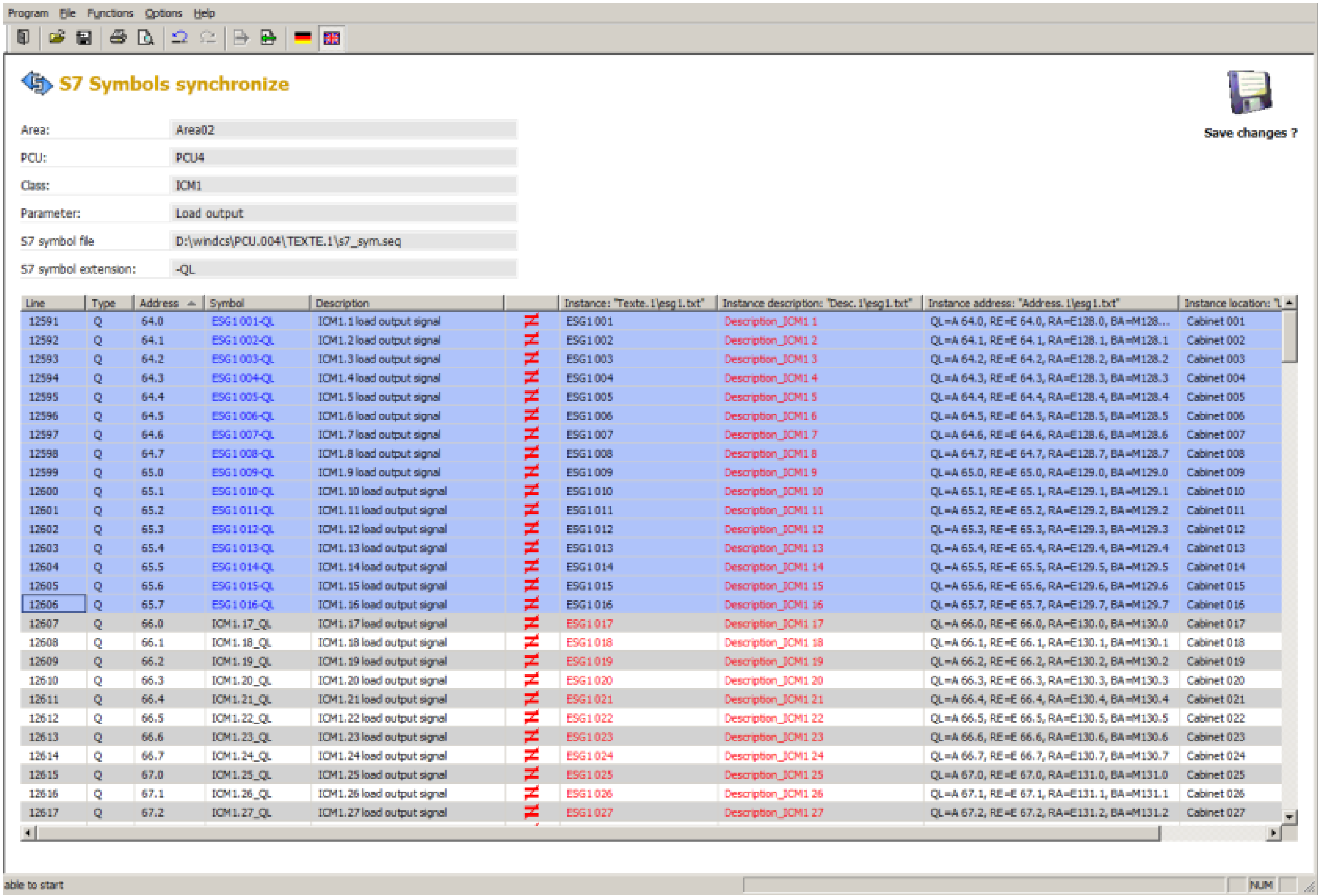


Table 5-1 Description of the table columns and the possible operations

Left table area S7 project (symbol table)	Right table area BRAUMAT/SISTAR project (text parametrization)	Associated text files	Possible operation
Line no. of the symbol file	Instance address	'<proj-path>\pcu.nnn\Address.x\...'	Display only
Type (E / A / M)			
Address (bit address)			
Symbol	Instance name	'<proj-path>\pcu.nnn\texte.x\...'	Edit & transfer
Description	Instance description	'<proj-path>\pcu.nnn\Desc.x\...'	Edit & transfer
-/-	Instance location	'<proj-path>\pcu.nnn\Location.x\...'	Edit

- The "equal to/'unequal to" column shows the result of the comparison between the STEP 7 symbol table and the BRAUMAT/SISTAR text parametrization. For this the symbol/ instance names and the description/instance description columns are compared.
  - For this comparison the type shortcut in the symbol name (e.g. '-QL' ...) is **not** also taken into account. The type shortcut of the parameter selected currently can be adjusted in the subsequent selection dialog.
  - = → identical content in both columns
  - ≠ → difference in at least one column
- Font color red → marking of the differences
- Font color blue → Text change following input in text buffer or through manual editing
- Font color black → Content agrees

Sequence for table comparison and text transfer

- Marking of a line area in the table view which should be transferred with the subsequent synchronization command into the application's text buffer.
- With the menu commands 'Functions' / 'Synchronize right' ('Ctrl-R') or 'Synchronize left' ('Ctrl-L') a dialog appears for selecting the columns to be transferred.
- Only the columns Symbol↔Instance and Description ↔Instance description can be transferred.
  When the instance name is transferred to the Step 7 symbol table (transfer direction "←"), the "Reduce to 24 characters" option can be selected. The configuration engineer must ensure that the Step 7 symbols (on the left) are unique.
- In addition to this, the S7 symbol extension (e.g. "-QL" for load output) can be adapted at the request of the user so that it differs from the default:
  - With 'Synchronize to right', this is removed
  - With 'Synchronize to left', this is added
- The following comparison functions are thereby available:

S7 project	Transfer direction	BRAUMAT/SISTAR project	Max. string length
Symbol	→	Instance name (optional with symbol extension)	STEP 7 V5.x: 24 characters TIA Portal: 32 characters
Symbol	←	Instance name (optional with symbol extension)	STEP 7 V5.x: 24 characters TIA Portal: 32 characters
Description	→	Instance description	STEP 7 V5.x: 80 characters TIA Portal: 128 characters
Description	←	Instance description	STEP 7 V5.x: 80 characters TIA Portal: 128 characters

- Following confirmation of the selection dialog the selected source texts are input into the text buffer of the table view and changes are marked in blue - but they are not saved yet.
- If any changes have been detected then the symbol "Save changes?" appears at the top right.
  By clicking the mouse on this symbol the text marked in color is saved. This can also be done via the "File / Save" menu command or with "Ctrl-S".
- The following text fields can also be edited directly in the table view - changes are marked in blue:
  - S7 project symbol table / Symbol
  - S7 project symbol table / Description
  - Text parameterization / Instance name
  - Text parameterization / Instance description
  - Text parameterization / Instance location
- Text changes can only be made alternately in the right or left table area, a change to the other side is only possible after "Save" or "Reject".
- If any text changes (text color blue) are active on one side then the synchronization command is only enabled for this side; any change to the other side is only possible after "Save" or "Reject".

Additional menu commands

- File / Print
  The Windows standard dialog 'Print' is opened. The comparison result in the page view can be printed out using this.
- File / Page view
  The comparison result is opened in the print view.
- File / Printer setup
  The Windows standard dialog 'Print settings' is opened.
- Function / Undo
  The last text change / synchronization is canceled
- Function / Restore
  The last text change / synchronization is repeated

5.7 System blocks and parameters

5.7.1 FIXV – Fixed analog values

Basics

This block provides up to 255 (V6) / 256 (V7) selectable analog values as a source for other blocks.

Parameter sets ANA / DB 734

Parameter set

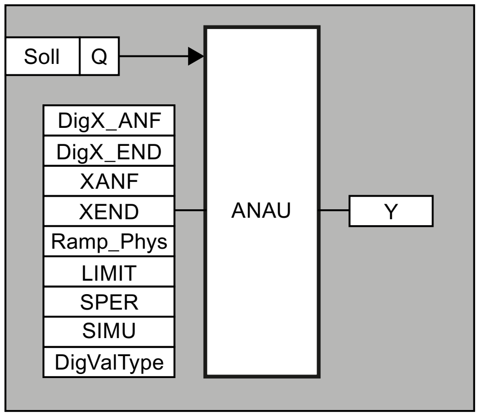
Name	Access	D.Type	A.Type	Value	Comment
ANA	NL	INT	ENG	0	ANA value INT
ANA_DINT	NL	DINT	ENG	0	ANA value DINT
ANA_REAL	NL	REAL	ENG	0.000	ANA value REAL

The analog value can be entered in integer, DINT or REAL format. Each display format has its own memory location in the dataset.

5.7.2 AOUT - Analog output

Basics

- This block allows up to 256 setpoints to be output via the analog output modules to the process.
- The setpoint is specified as a physical variable of other blocks (PID controller, sequences etc.) via sourcing. The block converts this specified setpoint into electrical units and forwards it to the I/O devices via the analog output modules.
- At the same time, the physical and the digital start and limit range values may be configured.
- The preset of digital range limits has the advantage of adapting several analog output modules e.g. SIMATIC S7, and other vendors.
- The setpoint is converted into electrical units linearly within the specified scale (`XANF/ProcValStart`, `XEND/ProcValEnd`) with taking into account the digital limits (`DigX_ANF/DigValStart`, `DigX_END/DigValEnd`).
- The block works independently of various physical output formats (0..10 V, -10..+10 V, 0..20 mA, 4..20 mA).



To avoid jumps, the maximum setpoint change per second can be specified using a ramp (`RAMP_PHYS/Ramp`). The output value is counted up by a 1 sec cycle whereas the full value is output if the RAMP value is zero.

For Test mode without analog peripherals, per instance a block bit (`SPER/DisableOutput`) can be used to suppress the output of the value. **However, the last valid output value remains in the I/O output word !**

To allow simulation (`SIMU/SimDigVal`) a value (`Y/DigValNorm`) can be set for each setpoint. The interconnection is then inactive.

Parameter sets AOUT / DB 731

PCU parameter set before Version V7 (max. 256 per PCU)

Name	Access	D.Type	A.Type, Log	Value	Comment
SOLL	NL	QUEL	ENG	ANA,1,ANA	Source of physical target value
DigX_ANF	NL	INT	ENG	0	Digital start value
DigX_END	NL	INT	ENG	27648	Digital end value
XANF	NL	INT	ENG, LOG	0	Physical start value
XEND	NL	INT	ENG, LOG	1000	Physical end value
RAMP_PHYS	NL	INT	ENG, LOG	100	Max. change per sec. of physical value
DigValType	NL	BYTE	ENG	B#16#00	AO format: 0=S7, 5=S5, 6=S5 (sign/amount)
SIMU	NL	BOOL	ENG	FALSE	Simulation: 0/1 = no/yes
SPER	NL	BOOL	ENG	FALSE	Output disable: 0/1 = no/yes
LIMIT	NL	BOOL	ENG	TRUE	Limit XANF<=Y<=XEND
Y	NL	INT	RT, LOG	0	Hardware-independent digital AO value
Y_Raw	HI	INT	RT	0	Hardware-dependent digital AQ value
ISOLL	NL	INT	ENG, LOG	0	Setpoint
Status	HR	INT	RT, LOG	2048	Status as word
SIMU2	NL	BOOL	RT	FALSE	Simulation2: 0/1 = no/yes

PCU Version V7 parameter set (max. 256 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
SetpointRef	NL	GREF	ENG	NULL	Reference to setpoint
S7-1500: Setpoin-tRef_ST	HR	WORD	SYS	W#16#0000	Signal status
SetpointPv	NL	REAL	RT, LOG	0.000	Setpoint
ProcValStart	NL	REAL	ENG, LOG	0.000	Physical start value
ProcValEnd	NL	REAL	ENG, LOG	120.000	Physical end value
SetpointAct	RD	REAL	RT	0.000	Effective setpoint
S7-400: ConvFuncNo	NL	INT	ENG	0	Conversion FB (-1=raw value, 0=S7-AI)
S7-1500: Conversion	NL	ENUM	ENG	0	Measured value conversion (0=Raw value, 1=S7-AI, 2=User)
Ramp	NL	REAL	ENG, LOG	10.000	Max. change per sec. of physical value
SimDigVal	NL	BOOL	ENG	FALSE	Simulate digital output value
DisableOut-put	NL	BOOL	ENG	FALSE	Output disable
LimitDigVal	NL	BOOL	ENG	TRUE	Limit output value
SetpointOp	NL	BOOL	RT	FALSE	Setpoint by operator
DigValStart	NL	INT	ENG	0	Digital start value
DigValEnd	NL	INT	ENG	27648	Digital end value
DigValNorm	NL	INT	RT, LOG	0	Normalized digital output value
DigValRaw	RD	REAL	RT	0.000	Hardware-dependent digital AQ value
DigValRef	NL	PREF	ENG	PAW 512	Target reference for output value
S7-1500: DigValRef_ST	HR	WORD	SYS	W#16#0000	Signal status
ErrDeNorm-Failed	HR	BOOL	RT	FALSE	De-normalization failed
ErrRefGrpEr-ror	HR	BOOL	RT	FALSE	At least one reference is invalid
NoteIsActive	HR	BOOL	RT	FALSE	Note is available
Status	HR	WORD	RT, LOG	W#16#0800	(Internal)

Note

Features of V7-PCU

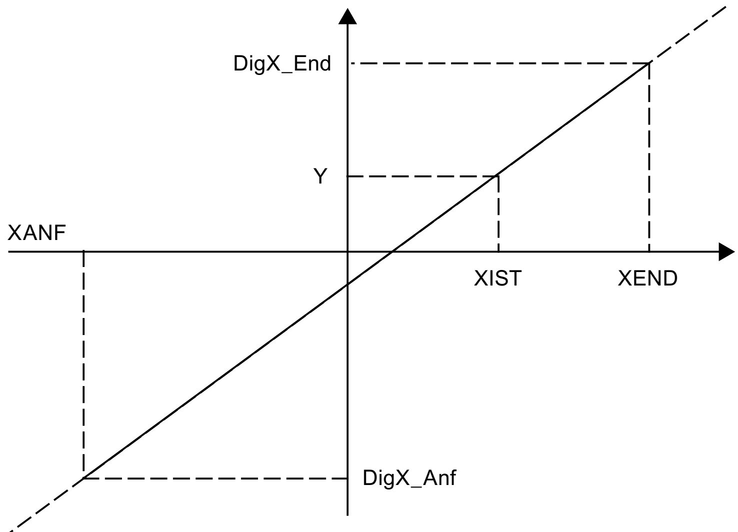
- The hardware-dependent raw value is output via the reference `DigValRef`, it is no longer necessary to configure ANAU_PW. The requirements for the memory width (PQB, PQW, PQD) for the respective, connected peripheral are set in the data source dialog of the reference.
- Conversion of the process value `SOLL/SetpointAct` into the hardware-dependent raw value is achieved with the `"ConvFuncNo"` parameter:
- The BOOL attribute `"Commissioning"` has been added to the global data. In the block AOUT, the bit causes an output disable for all instances, analogous to the "local" `SPER/DisableOutput` attribute.

ConvFuncNo	Description
-1	The process value (REAL) is copied unchanged to the destination of the reference (note memory width)
0	The process value is converted internally for S7 analog output
[FB]	The process value can be manipulated by a user FB before output. The FB3001 („`BmUsrConvValOutFB`") can be used as a template for this.

Attribute	Value	Description
S7-400: `ConvFuncNo`	-1	The process value (REAL) is copied to the destination unchanged (note memory width DWORD)
	0	The process value is converted internally for an S7 analog output
	[FB]	The process value will be manipulated by the user FB specified here before the output. The `FB3001 („BmUsrConvValOutFB")` can be used as a template for this.
S7-1500: `Conversion`	0	The process value (REAL) is copied to the destination unchanged (note memory width DWORD)
	1	The process value is converted internally for an S7 analog output
	2	The process value is processed by user block `BmUsrCallIfAoutConvValueFC (FC733)` before the output. This block is located in the Step 7 (TIA Portal) project folder ".../Runtime-Interfaces" and serves as a template for the process value conversions to be included by the user.
`ErrDeNormFailed`	0 / 1	Measured value denormalization failed (unable to derive a usable digital output value DigValNorm from the real value). This attribute can be set by the user conversion block (S7-400: FB3000, S7-1500: FC732). For the standard conversion (S7 I/O or unmanipulated), it is always = 0
`ErrRefGrpError`	0 / 1	At least one reference is invalid

Relations:

(see examples in chapter "Analog functions", AOUT)



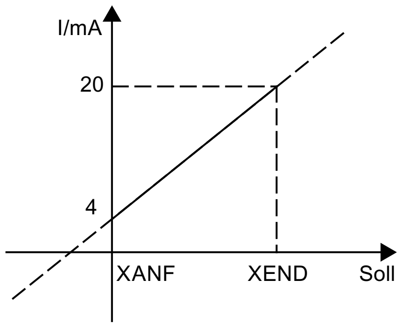
Examples:

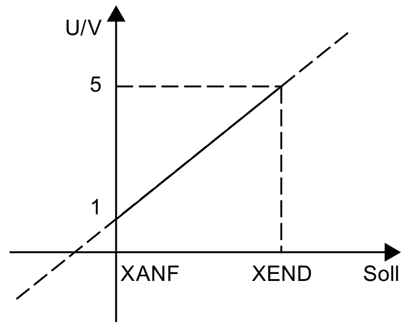
- Example 1:
  Application of AOUT with an analog output module with (0 to 20 mA or 0 to 10 V)

Presets/Adjustments	PCU < V7; S7-AA module
Parameterization hardware	Change of hardware configuration in S7 project: Type of output: I or U Output range: 0 to 20 mA or 0 to 10 V
Settings in the parameterization	DigX_ANF = 0 (preset) DigX_END = 27648 (preset) XANF = 0 (0.0%) (preset) XEND = 1000 (100.0%) DigValType = 0 (S7-AA)

- Example 2:
  Application of AOUT with analog output module with (4 to 20 mA or 1 to 5 V)

Presets/Adjustments	PCU V7; S7-AA module
Parameterization hardware	Change of hardware configuration in S7 project: Type of output: I or U Output range: 4 to 20 mA or 1 to 5 V
Settings in the parameterization	DigValStart = 0 (preset) DigValEnd = 27648 (preset) ProcValStart = 0 (0.0%) (default) ProcValEnd = 1000 (100.0%) ConvFuncNo = 0 (S7-AA)





Significance of parameters SIMU/SimDigVal and SPER/DisableOutput for the AOUT block

SIMU SimDigVal	SPER Disable Output	Functional properties
0	0	Output to peripherals. Derive value `Y/ProcValNorm` from value SOLL (normal operation).
1	0	Output to peripherals. • Bit `SIMU/SimDigVal` `Y/ProcValNorm` value is not formed from `SOLL/ProcValNorm` but can be set in parameter assignment (dig. output value). • Bit `SIMU2/SetpointOp` Simulated value is entered in `iSoll/SetpointPv` by analog faceplate and converted from phys. dig. output value by the ANAU block.
0	1	No output to peripherals. Derive value `Y/ProcValNorm` from value `SOLL/ProcValNorm`.
1	1	No output to peripherals. …"SIMUx" see above

5.7.3 AOUT_PW

Basics

Only applies to PCU versions before V7!

For PCU version from V7, the analog value is output directly in the ANAU block.

The peripheral addresses of the ANAU blocks need not be organized in consecutive order.

The system may be notified by the class **ANAU_PW** of max. 25 ranges to which it outputs the analog values (starting with range 1).

Each range is defined by the peripheral start address and the number of rare values.

Parameter sets ANAU_PW / DB 731

PCU parameter set (max. 25 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
PW	NL	INT	ENG	512	Startadr. PAW area (-1=List end)
Num	NL	INT	ENG	192	Range length in PAW (0=List end)

The ranges have to be occupied starting with range 1,

which is preset in the delivery state with:

- PW = 512
- Num = 192
- i.e. the first 192 analog values are output from PAW512.

If the system finds a range PW = -1 then the input of rare values is stopped as well as at Num <= 0.

5.7.4 SEQS - Sequence start block

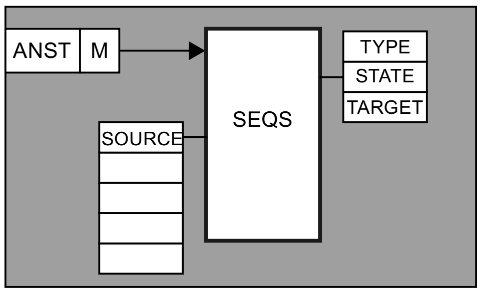
Basics

This block allows sequences to be started with simultaneous specification of the recipe type, the recipe number, the job number, and the batch number.

Functions

- The production year and job number are used for reports.
- Sequences can be started by other sequences or any other user applications. It is also possible to start a sequence via a coupling from a partner PCU. In this case, a data set must only be parameterized in the source PCU, not in the target PCU. The sequence is then started via a coupling bit. The block XC_ASTA_RCV may be used in the destination PCU for diagnostic purposes.
- For each start, one of 96 possible data blocks is required in which the corresponding source and target data can be specified.
- When starting sequences, the following occurs in the target sequence data set:
- No start occurs if the sequence is running (step<>0), if the permanent condition is not true or the sequence is in Manual mode
- Before start:
  Year, recipe type, recipe number, order number, and batch number are entered in accordance with the parameterization in the data set.

- Start:
  Sequence start occurs (setting step 1). The SequenceIsRunning flag may be queried by the user to determine the success of the initiated start.
- Each parameter set is assigned an impulse flag (M 672.0 - M 683.7). The control (set/delete) of the trigger flag is performed by the user in the respective block of the sequence or in the basic operations.



The block checks the start conditions of the target sequence. If the start conditions "Target sequence released"", "Not started", "No Manual mode", "Permanent conditions present" are not found, the "Start Error" message will be given.

If the start conditions are fulfilled, the target sequence is started with step 1.

Parameter sets for block SEQS / DB742

PCU parameter set before Version V7 (max. 96 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Type	NL	USINT	ENG	0	SEQS-Type (0..6,9..14)
MsgSEQ	NL	USINT	ENG	0	allocated sequence for message
SrcSEQ	NL	USINT	ENG	0	Source sequence
DestSEQ	NL	USINT	ENG	0	No. of destination sequence
DestPCU	NL	USINT	ENG	0	Destination PCU No.
Year	NL	USINT	ENG	0	Year for RType, JobNo, BatchNo
RecipeType	NL	USINT	ENG	0	Recipe type
Recipe	NL	INT	ENG	0	Recipe number
Order	NL	INT	ENG	0	Order number
Batch	NL	INT	ENG	0	Batch number
ASTA_M	HI	BOOL	RT	FALSE	Trigger flag from user
ASTA_FP	HI	BOOL	RT	FALSE	Positive edge
ParamError	NL	BOOL	RT	FALSE	Parametrization error
ParamErrorMsg	HI	BOOL	RT	FALSE	Message parametrization error to IOS
Error	NL	BOOL	RT	FALSE	Error at sequence start
ErrorMsg	HI	BOOL	RT	FALSE	Message error at sequence start to IOS
RemoteError	HI	USINT	RT	0	0: OK; 1..255: Error
SrcPCU	HI	USINT	ENG	0	Source PCU no.

Name	Access	D.Type	A.Type	Value	Comment
SrcASTA	HI	USINT	ENG	0	Source SEQS no. in source PCU
E4SendRecv	HI	BOOL	ENG	FALSE	0=Level7 Get/Put; 1=Level4 Send/ Recv

PCU parameter set V7 (max. 96 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Type	NL	ENG	USINT	0	SEQS-Type (0..6,9..14)
MsgSEQ	NL	ENG	USINT	0	allocated sequence for message
SrcSEQ	NL	ENG	USINT	0	Source sequence
DestSEQ	NL	ENG	USINT	0	No. of destination sequence
DestPCU	NL	ENG	USINT	0	Destination PCU No.
Year	NL	ENG	USINT	0	Year for RType, JobNo, BatchNo
RecipeType	NL	ENG	USINT	0	Recipe type
Recipe	NL	ENG	INT	0	Recipe number
Order	NL	ENG	INT	0	Order number
Batch	NL	ENG	INT	0	Batch number
ASTA_M	HR	RT	BOOL	FALSE	Trigger flag from user
ASTA_FP	HR	RT	BOOL	FALSE	Positive edge
ParamError	RD	RT	BOOL	FALSE	Parametrization error
ParamErrorMsg	HR	RT	BOOL	FALSE	Message parametrization error to IOS
Error	RD	RT	BOOL	FALSE	Error at sequence start
ErrorMsg	HR	RT	BOOL	FALSE	Message error at sequence start to IOS
RemoteError	HR	RT	USINT	0	0: OK; 1..255: Error
SrcPCU	HI	ENG	USINT	0	Source PCU no.
SrcASTA	HI	ENG	USINT	0	Source SEQS no. in source PCU

Process interface: DB 614: Trigger bit

DB614 sequence permanent condition

DB614	
DBB10	SEQS 8 ... 1
DBB11	SEQS 16 ... 9
DBB12	SEQS 24 ... 17
...	...
DBB21	SEQS 96 ... 89

User interface for block SEQS

No.	Flag	No.	Flag	No.	Flag	No.	Flag
1	M 672.0	25	M 675.0	49	M 678.0	73	M 681.0
2	M 672.1	26	M 675.1	50	M 678.1	74	M 681.1
3	M 672.2	27	M 675.2	51	M 678.2	75	M 681.2
4	M 672.3	28	M 675.3	52	M 678.3	76	M 681.3
5	M 672.4	29	M 675.4	53	M 678.4	77	M 681.4
6	M 672.5	30	M 675.5	54	M 678.5	78	M 681.5
7	M 672.6	31	M 675.6	55	M 678.6	79	M 681.6
8	M 672.7	32	M 675.7	56	M 678.7	80	M 681.7
9	M 673.0	33	M 676.0	57	M 679.0	81	M 682.0
10	M 673.1	34	M 676.1	58	M 679.1	82	M 682.1
11	M 673.2	35	M 676.2	59	M 679.2	83	M 682.2
12	M 673.3	36	M 676.3	60	M 679.3	84	M 682.3
13	M 673.4	37	M 676.4	61	M 679.4	85	M 682.4
14	M 673.5	38	M 676.5	62	M 679.5	86	M 682.5
15	M 673.6	39	M 676.6	63	M 679.6	87	M 682.6
16	M 673.7	40	M 676.7	64	M 679.7	88	M 682.7
17	M 674.0	41	M 677.0	65	M 680.0	89	M 683.0
18	M 674.1	42	M 677.1	66	M 680.1	90	M 683.1
19	M 674.2	43	M 677.2	67	M 680.2	91	M 683.2
20	M 674.3	44	M 677.3	68	M 680.3	92	M 683.3
21	M 674.4	45	M 677.4	69	M 680.4	93	M 683.4
22	M 674.5	46	M 677.5	70	M 680.5	94	M 683.5
23	M 674.6	47	M 677.6	71	M 680.6	95	M 683.6
24	M 674.7	48	M 677.7	72	M 680.7	96	M 683.7

Significance of the TYPE parameter in the SEQS block

TYPE	Function	Source	Target	Miscellaneous
0	Element undefined	-	-	
1	Start via sequence	Sequence	Sequence	Recipe type, recipe number, job number, and batch number from the source sequence
2	Start of a CIP sequence	-	Sequence	Recipe type, recipe number, job number, and batch number taken from the SEQS data set
4	Start of a CIP sequence	-	Sequence	Recipe type and recipe number taken from SEQS data set Job number and batch number not changed in target sequence
5	Start of a sequence	-	Sequence	Recipe type and recipe number taken from SEQS data set Year=current year Order number=calendar week Batch number=(1...n) beginning with 1 each calendar week, incremented by 1 for each start within the same calendar week
6	Start of a sequence	-	Sequence	User interface "SEQS_USER_FC"

TYPE	Function	Source	Target	Miscellaneous
9	Start of a sequence in another PCU	Sequence	PS in other PCU	Like type 1, however, target sequence is in another PCU
10	Start of a CIP sequence in another PCU	-	PS in other PCU	Like type 2, however, target sequence is in another PCU
12	Start of a CIP sequence in another PCU	-	PS in other PCU	Like type 4, however, target sequence is in another PCU
13	Start of a sequence in another PCU	-	PS in other PCU	Like type 5, however, target sequence is in another PCU
14	Start of a sequence in another PCU	-	PS in other PCU	Like type 6, however, target sequence is in another PCU

Recommendation for types 2/4/10/12

- In order to get optimal protocols and archives, the parameters Year, Order, Batch should varied prior to each sequence start.
- For this several variants exist:
  - Direct access to data set:
    e.g. variation of parameter year

```
L SYS.byYear
T ASTA.au[13].u.bYear
```

Designation of parameters can be taken from DB742 or UDT742

- Use of type 5/13
- Use of type 6/14

Function SEQS_USER_FC:

Variation of type 5/13 can be prescribed by calling ASTA_USER_FC .

```
interface of the FC:
VAR_INPUT
iRecord : INT;                    //number of SEQS data set (1 to n)
bType : BYTE;                     //SEQS type: 6 (local), 14 (remote)
END_VAR
VAR_OUTPUT
boRetVal : BOOL;                  //result False=OK, TRUE=error.
                                  //if error no
                                  //sequence start is initiated!
END_VAR
VAR_IN_OUT                        //Partner-PCU
                                  ///destination sequence in partner PCU
                                  //All of the following parameters are preset with
                                  //current values from SEQS data set
                                  //and can be varied by the
                                  //user program in SEQS_USER_FC
                                  //.
```

```
 interface of the FC:
    bDestPCU : BYTE;
    bDestSEQ : BYTE;
    bYear : BYTE;
    bRecipeType : BYTE;
    iRecipe : INT;
    iOrder : INT;
    iBatch : INT;
END_VAR
```

In the delivery version of ASTA_USER_FC a program code is deposited, which takes the year from DB701, sets iOrder=month*100+day and counts iBatch consecutively from 1..n. The batch number starts with 1 on each new order number. Further, the SEQS data set number and type are not evaluated.

Note

This code is intended as a typical, however it can be used in practice without changes.

5.7.5 BLR - Binary connection logic

5.7.5.1 BLR1 to 9 - binary connection logic

Basics

The "binary connection logic" available from PCU version V7 provides the user (system integrator/project manager and system driver/operator) with configuration and online representation of connection logics based on FUP. For each PCU, the system provides 9 groups with 256 instances each that you can use freely.

Functional description

- The logic underlying the BLR objects is interpreted in runtime. Changes can be made at any time.
- BLR objects have to be configured graphically. There is no automatic reinterpretation of the AWL codes from predecessor versions.
- The quantity structure of a BLR object is limited to:
  - 32 pins (of which 31 are user pins)
  - 32 gates (of which 31 are user gates)
- BLR objects can identify the origin of an output signal change (first trigger diagnostics).
- Each pin participating in the connection logic can be bridged at the runtime with an adjustment value
- The result output of a BLR object can be simulated:

Suggested configuration

If you accept the following configuration suggestion, the first time you create a reference to a BLR object you will be supported by the system and the suitable object will be preset.

Initial object			Object to be logically linked		
ICM1	1 to 256	RefInterLock	BLR1	1 to 256	Result
ICM2	1 to 256	RefInterLock	BLR2	1 to 256	Result
ICM3	1 to 256	RefInterLock	BLR3	1 to 256	Result
ICM4	1 to 256	RefInterLock	BLR4	1 to 256	Result
Sequences	1 to 128	PermCondRef	BLR5	1 to 128	Result

Result of logic operation and update

The result of the logic operation of a BLR can be queried passively or actively in the user program:

- Passive query of result bit
  This query type is appropriate when it is assured that the respective BLR object was processed before the query. This is the case for BLR groups 1 ... 5 because attribute `"PeriodicUpdate" = TRUE` is generally always set in the parameter assignment data record (see section "Call-up system of BLR classes" further below). For BLR groups 6...9, the result of logic operation is only updated for running EOPs, i.e., the passive query is only appropriate within the EOP FCs.
  Example S7-400: `U "BLR3".s_aBlr[42].xResult`
  Example S7-1500: `U "BLR3".s_rgRecords[42].s_BlrObj.xResult`
- Active query based on a system function
  This is appropriate for BLR objects that are not cyclically processed, e.g. user logic operations or EOP next step conditions, if attribute `"PeriodicUpdate" = FALSE`.
  Please note that the parameters differ between S7-400 and S7-1500.
  Example S7-400, BLR,6,42:
  ```
  CALL "BmGetBlrResultFC"(
    BlrIDB      := 2116,
    BlrInstance := 42,
    RET_VAL     := xRetVal,
    Result      := xResult);
  ```
  Example S7-1500 (SCL):
  ```
  "BmGetBlrResultFC" (
    BlrGroup := 6,
    BlrInstance := 42,
    Result => #xResult);
  ```

Call-up system of BLR classes

- All BLR groups are generally embedded in the AS runtime environment (AS Scheduler)
- The processing of single instances is controlled using attribute `"PeriodicUpdate" = TRUE` in the parameter assignment data record.
- For the classes for the ICM interlocks BLR1 ... BLR4 and the permanent sequence condition BLR5, processing is activated for all instances (delivery state).

- For classes BLR6 ... BLR9 intended for use as a next step condition, this attribute has been set to `= FALSE`, because it is not necessary in these applications. Here, the processing and querying take place automatically in the context of the sequence (when EOP is configured accordingly). The formulation of a next step condition within an EOP is not required in this case. In particular, no "`BmGetBlrResultFC`" call has to be made by the user (this occurs internally).
- **Accesses to the global flag interface (MB 99 ... 107) of the sequences**
  - Please note that you must not access the global flag interface of the units for formulation of the unit permanent condition because the processing of BLR5 takes place outside of the unit processing. In the logic connections of the BLR data sources dialog, only the status bits of the sequence data record (e.g. "`Sequenzen.nnn.SeqRun`", etc.) may be used for this purpose.
  - In the context of the EOP step conditions BLR6 ... BLR9, use of global flags is generally possible. However, because they are used jointly by all unit instances, their process value is "random" from the perspective of the visualization, which can lead to undesired effects (e.g. flashing of inputs or logic elements).

5.7.5.2 BLR configuration

Basics

The settings and parameters of a BLR object and the connection code itself are configured using the editor described here ("BlrEdit").

The symbol names of the input pins are obtained as follows assuming they are available:

- References to object attributes represent the instance texts from the text parameterization.
- References to absolute addresses (especially I, Q, M references) represent the texts from the symbol table (STEP 7) / PLC tag table (TIA Portal).

Symbols from STEP 7 / TIA Portal

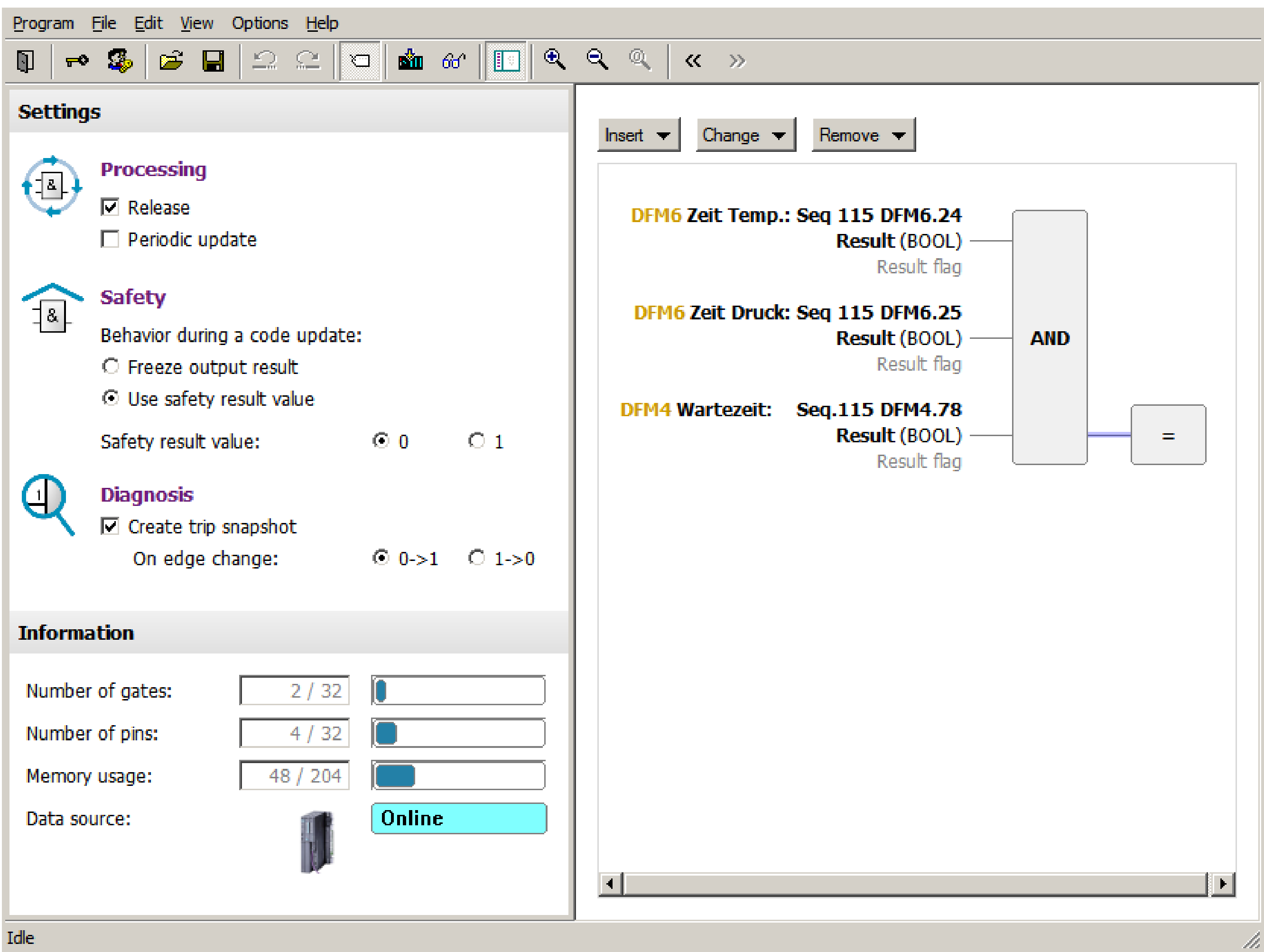
The PCU-related assignment list/PLC tag table is used to display the symbol information. It must be reexported in each case in the STEP 7 Symbol editor (V5.x) or PLC tag table (TIA Portal) after changes. It is expected with the following formats and in the following system directories:

	PCU type S7-400 with STEP 7 V5.x	**PCU type S7-1500 with STEP 7 TIA Portal**
Export/import file and format	s7_sym.seq	PLCTags.xml
Path when a change of engineering language is not available	'<proj-path>\pcu.nnn**s7_sym.seq**' or '<proj-path>\pcu.nnn**texte\s7_sym.seq**'	'<proj-path>\pcu.nnn**PLCTags.xml**' or '<proj-path>\pcu.nnn**texte\PLCTags.xml**'
Path when a change of engineering language is available	'<proj-path>\pcu.nnn**texte.x\s7_sym.seq**'	'<proj-path>\pcu.nnn**texte.x\PLCTags.xml**'

The mnemonic setting in STEP 7 / TIA Portal must match the currently valid system engineering language (if an engineering language is not selected, this is the current system language).

- Engineering language = German → STEP 7 mnemonic = Deutsch
- Engineering language ≠ German → STEP 7 mnemonic = English/International

Working with the application



Menu functions

Menu	Command	Shortcut key	Function
File	Open ...	CTRL+O	Open dialog 'Open block' • Select from area, PCU, class (Blr group), instance, origin (online/ offline)
	Save	CTRL+S	Saves the current settings and the connection logic of the instance in the offline container
	Save As ...	F12	Open dialog 'Save as ...' • Select from area, PCU, class (Blr group), instance, origin (online/ offline)
	Load	CTRL+L	Transfers the current instance to the PCU
	Next instance	NUM +	Switches to the next BLR instance
	Previous instance	NUM -	Switches to the previous BLR instance
	Exit		Exits the editor
Edit	Back	CTRL+Z	Reverses the last change
	Redo	CTRL+Y	Repeats the last change
	Delete	Del	Delete (remove) the entire logic operation of the instance
View	Toolbar		Show/hide toolbar
	Status bar		Show/hide status bar
	Monitor	CTRL+F7	Switch logic view to monitor mode (see chapter "Monitor block")
	Show Symbols	CTRL+Q	Show/hide data source symbols
	Show reference IDs	CTRL+R	Show/hide reference IDs
	Maximize view		Zoom in on logic operation area
	Minimize view		Zoom out of logic operation area
	Reset view		Reset logic operation area to original size
Tools	Export object	CTRL+E	Export current instance to CSV text file This can be processed further using MS-Excel (mass data engineering)
	Duplicate objects	CTRL+D	Read processed CSV text file In this way, object instances can be reproduced (mass data engineering)
	Confirm loading		Get AS confirmation before each download to the AS (**Safety function!**)

Settings and information

Group	Adjustments	Function
Editing	Release	Enables the processing of the block instance in the PCU;
	Periodic updating	Switches the cyclical processing of the instance on/off. The **following recommendation** applies here: • ON - for use as a permanent sequence condition • OFF - for use as a step condition (opened automatically in the sequencer) • ON - for use as ICM interlocking

Group	Adjustments	Function
Security	Behavior during code updating	Presetting of the local output value while the "Load block" command updates the code in the AS (can take from 1 to several seconds): • 'Freeze' result output (keep last state) • Use security output value (required security output value = 1 or = 0)
Diagnostics	Create signal snapshot	Activates the trigger for a memory map of the current logic state (input signals, interim results, output signal). **Notice: Function is only enabled if the BLR instance is not in simulation mode.** • The trigger is activated by an alternating edge (0 → 1 or 1 → 0) of the link result (can be configured). • In monitor mode, this image can be displayed in place of the online value • The image is deleted with the reset command and the trigger reactivated.
Information	Number of gates	Currently used / max. gates of the instance Default: 1 / 32 (1 gate for system, 31 gates for user configuration)
	Number of pins	Currently used / max. pins of the instance Default: 1 / 32 (1 pin for system, 31 pins for user configuration)
	Memory consumption	Currently used / max. memory (Bytes) of the instance Default: 8 / 204 (8 bytes for system)
	Data source	Data source / destination of the displayed / configured parameters. This is selected in the "Open block" dialog. • Offline = Hard drive • Online = AS memory

Important: Changes to settings are only written to the AS and activated with the command "Load block".

Connection logic area

The structure of the connection logic and the link to references is created with a few commands using the following table.

Menu	Command	Function
Insert	Pin (before, after)	Insert new pin (connection) before, after the current selection
	Gates (AND, OR, NAND, NOR, XOR)	Insert new gate <Type> at the selected pin
	Reference	Link selected pin to data source
Change	Invert pin	Invert selected pin
	AND, OR, NAND, NOR, XOR	Change selected gate to <Type>
	Edit	Change selected pin / data source

Menu	Command	Function
Remove	Pin	Remove selected pin • All activated gates (if any) are removed
	Gate	Remove selected gate • The input of the following gate is connected to the first activated gate / first data source.

Important: Changes to settings are only written to the AS and activated with the command "Load block".

5.7.5.3 BLR mass configuration

Basic principles

A BLR object can be used as a template and reproduced for very similar objects. The structure of the connection logic is retained while the references used are exchanged on the basis of a CSV file.

On the Procedure:

- Create and export template BLR object (Tools → Export object)
- Expand CSV file by some lines according to the syntax below
- Import CSV file and reproduce objects (Tools → Reproduce objects)

Import/Export file

Basic Structure:

- Line 1: Header for information
- Line 2: Template (original) object
- All subsequent lines: Destination objects
- Column description

Column	Description
BLR object	<PCU number>, <BLR class name>, BLR instance
DataOrigin	Data source: "online" or "offline"
RefX	Description of the reference at position X. The address can be object-based, symbolic or absolute. • Object-based: <Class name>, <Instance number>, <Attribute name> • Symbolic (from exp. STEP7 allocation list): "Icon" • Absolute (in usual STEP syntax): M 4000.7

For better orientation, the reference IDs can be shown in the editor (View → Display reference IDs).

5.7.5.4 BLR diagnosis/Online

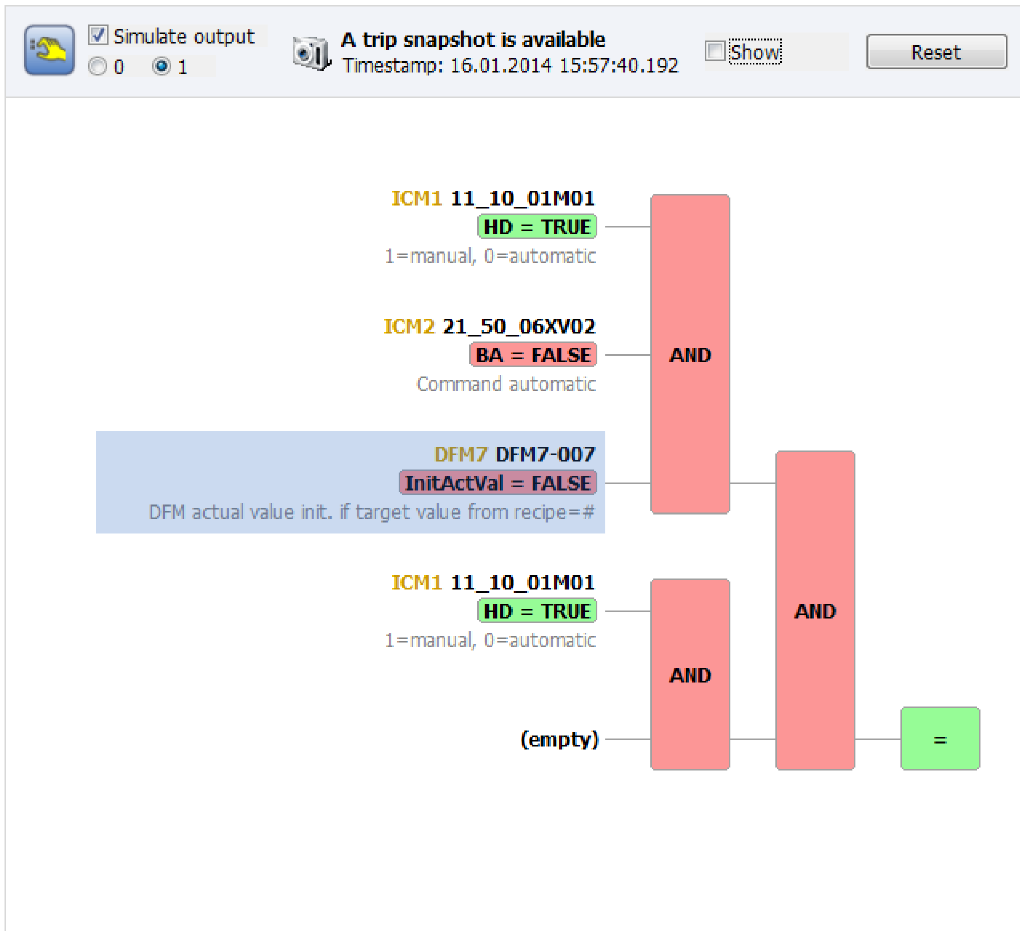
Basics

The application "BLR Editor" can also be used for fast diagnostics and troubleshooting of the logic operation status (inputs, interim results/gate, output) when it is switched to "Monitor mode". Depending on the context of the use, it is opened in one of the following applications:

- When configuring the logic itself, it is opened in the "View→Monitor" menu (Ctrl+F7).
- When used as an ICM interlock, it is opened in the 'ICM control' control dialog using the Status button.
  (For ICM configuration, refer to the section ICM - Individual Control Modules (ICM1 ICM2 ICM3 ICM4) (Page 269))
- When as step enabling condition, it is opened in the "System overview" using the "Status step condition" command (CTRL+F4).
  (For EOP configuration, refer to the section Dialog 'Project equipment operation' (Page 577))
- When used as a permanent sequence condition, it is opened in the "System overview" using the 'Start and permanent condition' command (CTRL+F4).
  (For SEQUENCE configuration, refer to the section Sequences / TeilAnl - sequence control (Page 345))

Operator control and monitoring of blocks

In operator control and monitoring mode, there is cyclic updating of the signal and logic statuses:



- Display of signals / gates
  - Signal state/Connection result = `FALSE` → red
  - Signal state/Connection result = `TRUE` → green
- Bridging the input signal (after pin selection):

  Note

  The binary signals involved in logic operations can be bridged at runtime ("signal bridging"). Please note that if there is active signal bridging, no changes to the logic operation code can be loaded. This also applies to the "Save as (online)" function.

  - Shortcut menu Pin → "Control value to 1" (Ctrl+1)
  - Shortcut menu Pin → "Control value to 0" (Ctrl+0)
  - Shortcut menu Pin → "Deactivate value controlling" (Ctrl+B)

- Simulate result output
  The logic output value can be prescribed for simulations irrespective of the signal state (0 / 1).

Note

The simulation control elements are only active when there is currently no signal snapshot being displayed.

The function "Create signal snapshot" is disabled with enabled simulation.

- Snapshot available (only visible if there is a memory map)
  - Show
    Shows the last memory image instead of the online status
  - Reset snapshot
    Deletes the memory image, returns to the online display and enables the trigger for a new memory image

Opening the archive viewer via the shortcut menu

It is possible to branch to the message archive via the "Operator inputs and messages ..." shortcut menu in operator control and monitoring mode. In so doing, the currently selected BLR instance is transferred as the filter criterion. As a result, you receive a list of all operator inputs and messages recorded for this BLR instance.

5.7.6 DFM - Digital Function Modules

5.7.6.1 Overview DFM

Basics

The specification of digital target values is necessary for controlling the process. For this purpose, there are a maximum of 24 (20 for PCU versions up to V5) setpoints as Digital Function Modules (DFM) available for the unit for each equipment operation (EOP) plus the runtime monitoring "EopMonTmError". Every setpoint can be assigned a function type in an EOP.

The DFMs are divided into the following groups:	
DFM0	255 counters
DFM1 - DFM4	255 timers, limit value levels, decoders (DINT)
DFM5 - DFM8	255 timers, limit value levels, decoders (REAL)

Types of DFM group

Type	Function	DFM1 - DFM4	DFM5 - DFM8
1	Restarting timer	✓	✓
2	Consecutive timer	✓	✓
3	Down timer	✓	✓
4	Limiter	✓	✓
5	Setpoint	✓	✓
6	Bit mask	✓	
7	Decoder	✓	
8	Transfer	✓	
10	Setpoint / actual value cell	✓	✓
11	Time	✓	✓
14	Limiter simple	✓	✓

The results (e.g. time expired, counter value reached, limit value exceeded) are available as binary signals (DFM bit memory) in the EOPs. or as switching condition in the basic operations.

The setpoints of the function modules are stored in the recipe lists for every EOP after the number of the EOP and are loaded into the function modules when processing the step by the unit control.

The following is valid for all function types:

If the special character "#" is entered as the setpoint in the recipe list, the setpoint of the function module will not be overwritten when the EOP starts; the setpoint from the previous steps will be retained.

In this case, the "`InitActVal`" parameter can be used to specify how the corresponding actual value behaves:

- `InitActVal=1`: actual value, initialized with 0 at each step change
- `InitActVal=0`: actual value is retained

The control program for binary signals of the function module (time release, counter cycle, target value block) are entered into the respective sequence function block.

If no block "DisableStepProt" report has been programmed, a run report is automatically written when processing the unit (e.g. batch log) in which the EOP, the starting time of the EOP as well as the setpoints and actual values of all function modules are entered.

The starting time of the EOP (date HH.MM) is found in the first column of the run report, whereas the second column contains the values of the digital function modules.

The printout of the run report can be started automatically after processing the last EOP of the last subprocess. or can be started by the operator.

To achieve a meaningful run report and archiving, the batch numbers of all sequences should be incremented at each sequence start. Batch numbers can be assigned from 1...32767 per sort block.

5.7.6.2 DFM0 - Counter

Basics

255 counters can be parameterized per PCU.

To update the DFMs and form the DFM result flag, DFM processing needs to be started explicitly in the unit or in the EOP. To do this, the BmDfmCallFC (FC636) function needs to be called with suitable parameters:

```
CALL FC636
(
    iDfmType := [DFM group],
    iDfm     := [DFM instance]
)
```

Parameter sets for the DFM0 / DB 736 block

PCU parameter set before Version V7 (max. 255 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
SOLL	NL	INT	ENG	0	Setpoint low word
SOLL_DINT	NL	DINT	ENG, LOG	L#0	Setpoint double integer
IST	NL	INT	RT	0	Actual value low word
IST_DINT	NL	DINT	RT, LOG	L#0	Actual value double integer
InitActVal	NL	BOOL	ENG	FALSE	DFM actual value init. if target value from recipe=#
Grenze_DINT	NL	DINT	ENG	L#0	Limit double integer
Richtung	NL	BOOL	ENG	FALSE	0=up counter, 1=down counter
Summation	NL	BOOL	ENG	FALSE	0 = non-accumulating, 1 = accumu-lating
Art	NL	BOOL	ENG	FALSE	0=PSK is increment, 1 = PSK is re-duction
PSK	NL	INT	ENG	1	Increment/reduction
Hilf	HI	INT	RT	0	Help cell
Error	HI	BOOL	RT	FALSE	Parametrization error

PCU version V7 parameter set (max. 255 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Setpoint	NL	DINT	ENG, LOG	L#0	Setpoint (DINT)
ActualValue	NL	DINT	RT, LOG	L#0	Actual value (DINT)
Limit	NL	DINT	ENG	L#0	Limit (DINT)
InitActVal	NL	BOOL	ENG	FALSE	DFM actual value init. if target value from recipe=#
Reverse	NL	BOOL	ENG	FALSE	0=up counter, 1=down counter

Name	Access	D.Type	A.Type	Value	Comment
Cumulative	NL	BOOL	ENG	FALSE	0 = non-accumulating, 1 = accumu-lating
ParamUsage	NL	ENUM	ENG	Inkrement	0=Increment, 1=Reduction
Param	NL	INT	ENG	1	Increment/reduction
ParamError	HR	BOOL	RT	FALSE	Parametrization error

General information on counters

To register counter pulses it is necessary to assign the counter inputs to flags. One flag (M 984.0 ... M 1015.6) per counter is specified for this purpose. The assignment is performed depending on the counter frequency in the program of the sequence or, for fast pulse sequences, in user program FB1224.

The maximum counter frequency is 2.5 Hz.(100 ms sampling). However, it is possible to run the pulse inputs with a higher sampling rate of 200 ms to reduce the AS cycle time, which provides a max. counting frequency of 1.25 Hz. Thereby the AS cycle time decreases.

Mode	Description
DB 736 DBX12.0=0	Delivery state: Reduction 100 ms -> 200: No
DB 736 DBX12.0=1	Reduction 100 ms -> 200: Yes

To update the counter readings and to form the DFM result memory bit, FC 736 must be called in the unit or in the EOP with the counter number (1 to 255) as the parameter.

The `Richtung/Reverse` parameter specifies whether it is an up counter or a down counter.

If parameter `Art/ParamUsage` = "0" is specified, the counter counts up or down at each pulse by the value of the `PSK/Param` parameter.

If parameter `Art/ParamUsage` = "1" is specified, the counter counts up or down by one after each x-th pulse (x = value in `PSK/Param`).

Example (only relevant parameters):

Some common DFM applications are shown below.

Up counter

- Up counter for liquid influx: The counter is to be increased by 1 hl at each pulse. The counter should be configured as DFM No. 5.
- When processing the counter and at the same time starting an EOP, the parameterized switching limit is loaded into the actual value cell of the counter. The DFM result flag is set when the setpoint specified in the recipe list is reached or exceeded.

Parameter assignment

Parameter	Value	Description
`Richtung/Reverse`	0	0 = up counter
`Art/ParamUsage`	0	0 = `PSK/Param` is increment
`PSK/Param`	1	Increment

Assignment of counter input in the sequence FB:

	A	I3.1	Pulse counter input IDM
	A	I65.2	ICM-11 RE
	=	M 984.4	Counter input DFM0 no. 5

DFM block call in the sequence FB:

	AN	M 5.3	Release flag
	JC	NEXT	Skip FC call, else
	CALL	FC 736	Processing group DFM0
	iDfm:=	5	DFM number = 5
NEXT			

Up counter accumulating (through several steps)

Up counter accumulating for solid substance taking: The counter is to be increased by 50 kg at each pulse. The display occurs in unit 1. The counter should be configured as DFM No. 6.

When starting the sequence the parameterized switching limit is loaded into the actual value cell. As opposed to the non-accumulating up counter, the actual value is not overwritten by the switching limit parameterized in the PCU at every step. The DFM result flag is set when the setpoint specified in the recipe list is reached or exceeded.

Parameter assignment

Parameter	Value	Description
`Grenze_DINT/Limit`	0	Limit double integer
`Richtung/Reverse`	0	0 = up counter

Parameter	Value	Description
`Summation/ Cumulative`	1	1 = accumulating
`Art/ParamUsage`	0	0 = `PSK/Param` is increment
`PSK/Param`	50	Increment

Assignment of counter input in the sequence FB:

	A	I2.5	Pulse counter input IDM
	=	M 984.5	Counter input DFM0 no. 6

DFM block call in the sequence FB:

	AN	M 656.0	Plant sect. 1 not running
	JC	NEXT	Skip FC call, else
	CALL	FC 736	Processing group DFM0
	iDfm:=	6	DFM number = 6
NEXT			

Down counter

Down counter for solid product dosing. At every fifth pulse the counter value is to be reduced by 1 m^3. The display occurs in unit 4. The counter should be configured as DFM No. 7.

When processing the counter and starting an EOP at the same time, the setpoint is loaded into the actual value cell. Upon reaching or dropping below the parameterized switching limit (Grenze_L) the DFM result flag is set.

Parameter assignment

Parameter	Value	Description
`Grenze_DINT/Limit`	0	Limit
`Richtung/Reverse`	1	1 = Down counter
`Art/ParamUsage`	1	1 = `PSK/Param` is reduction ratio
`PSK/Param`	5	Reduction

Assignment of counter input in the sequence FB:

	A	I13.7	Pulse counter input IDM
	=	M 984.6	Counter input DFM0 no. 7

DFM block call in the sequence FB:

	CALL	FC 736	Processing group DFM0
	iDfm:=	7	DFM number = 7

Down counter accumulating (through several steps)

Down counter accumulating for liquid addition. At every pulse the counter is to be reduced by 10 l. The display occurs in unit 8. The counter should be configured as DFM No. 8.

When first processing the counter, the switching limit is loaded into the actual value cell of the counter. As opposed to the non-accumulating down counter, the switching limit parameterized in the PCU is not loaded into the actual value cell at every step. Upon reaching or dropping below the parameterized switching limit (Grenze_L) the DFM result flag is set.

Parameter assignment

Parameter	Value	Description
`Grenze_DINT/Limit`	0	Limit double integer
`Richtung/Reverse`	1	1 = Down counter
`Summation/Cumulative`	1	1 = accumulating
`Art/ParamUsage`	0	0 = `PSK/Param` is increment
`PSK/Param`	10	Increment

Assignment of counter input in the sequence FB:

	A	I22.3	Pulse counter input IDM
	A	I67.3	ICM-26 RE
	=	M 984.7	Counter input DFM0 no. 8

DFM block call in the sequence FB:

	CALL	FC 736	Processing group DFM0
	iDfm:=	8	DFM number = 8

Assignment of counter inputs and DFM result flags of counters

Counter number								Counter inputs	DFM result flag
Bit address								Byte address	Byte address
.0	.1	.2	.3	.4	.5	.6	.7	Flag	Flag
1	2	3	4	5	6	7	8	984	728
9	10	11	12	13	14	15	16	985	729
17	18	19	20	21	22	23	24	986	730
25	26	27	28	29	30	31	32	987	731
33	34	35	36	37	38	39	40	988	732
41	42	43	44	45	46	47	48	989	733
49	50	51	52	53	54	55	56	990	734
57	58	59	60	61	62	63	64	991	735
65	66	67	68	69	70	71	72	992	736
73	74	75	76	77	78	79	80	993	737
81	82	83	84	85	86	87	88	994	738
89	90	91	92	93	94	95	96	995	739
97	98	99	100	101	102	103	104	996	740
105	106	107	108	109	110	111	112	997	741
113	114	115	116	117	118	119	120	998	742
121	122	123	124	125	126	127	128	999	743
129	130	131	132	133	134	135	136	1000	744
137	138	139	140	141	142	143	144	1001	745
145	146	147	148	149	150	151	152	1002	746
153	154	155	156	157	158	159	160	1003	747

Counter number								Counter inputs	DFM result flag
161	162	163	164	165	166	167	168	1004	748
169	170	171	172	173	174	175	176	1005	749
177	178	179	180	181	182	183	184	1006	750
185	186	187	188	189	190	191	192	1007	751
193	194	195	196	197	198	199	200	1008	752
201	202	203	204	205	206	207	208	1009	753
209	210	211	212	213	214	215	216	1010	754
217	218	219	220	221	222	223	224	1011	755
225	226	227	228	229	230	231	232	1012	756
233	234	235	236	237	238	239	240	1013	757
241	242	243	244	245	246	247	248	1014	758
249	250	251	252	253	254	255		1015	759

Example:

DFM 0.44	Counter input:	M 989.3
	DFM result flag	M 733.3

5.7.6.3 DFM1 - DFM4 - timers, limits, decoders (DINT)

Basics

Per PCU, three (PCU < V7) or four (PCU V7) groups each with 255 instances of digital function modules for timers, limiters and decoders in the format DINT can be used.

To update the DFMs and form the DFM result flag, DFM processing needs to be started explicitly in the unit or in the EOP. To do this, the BmDfmCallFC (FC636) function needs to be called with suitable parameters:

```
CALL FC636
(
    iDfmType := [DFM group],
    iDfm     := [DFM instance]
)
```

Parameter sets for the DFM1, 2, 3, 4 / DB 737, 738, 739, 2104 (only V7) block

PCU parameter set before Version V7 (max. 255 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
SOLL	NL	INT	ENG	0	Setpoint low word
SOLL_DINT	NL	DINT	ENG, LOG	L#0	Setpoint double integer
IST	NL	INT	RT	-23250	Actual value low word
IST_DINT	NL	DINT	RT, LOG	L#42286	Actual value double integer

Name	Access	D.Type	A.Type	Value	Comment
InitActVal	NL	BOOL	ENG	FALSE	DFM actual value init. if target value from recipe=#
Art	NL	INT	ENG	1	1=t_v 2=t_sum 3=t_r 4=GW 5=SW 6=mask 7=dec 8=trans 14=GW32
PSK	NL	INT	ENG	6	type1-3:divider in s type4:Hysteresis bottom type7:Group type8:DB
Hilf	NL	INT	ENG	0	Help cell (for type8: target data word, type4: Hysteresis above)
QBit	NL	STEP	ENG	U M102.0	type1-3:enable, type5:disable
QDat	NL	QUEL	ENG	ANA, 1,ANA	type4:Actual source
Error	HI	BOOL	RT	FALSE	Parametrization error
HyError	HI	BOOL	ENG	FALSE	Hysteresis above/below exceeded
HyUp	HI	BOOL	ENG	FALSE	Display in the upper hysteresis bands
HyDown	HI	BOOL	ENG	FALSE	Display in the lower hysteresis bands
boResult	HI	BOOL	RT	TRUE	DFM result BIT
HyUpError	HI	BOOL	ENG	FALSE	Hysteresis above exceeded
HyDoError	HI	BOOL	ENG	FALSE	Hysteresis below underrun

Table 5-2 Description of the function details

TYPE	Function	QBit	PSK	Help cell	QDat	DFM result flag = 1 when
1	Restarting timer	Release	Reduction	-	-	ACTUAL ≥ setpoint
2	Consecutive timer	Release	Reduction	-	-	ACTUAL ≥ setpoint
3	Down timer	Release	Reduction	-	-	ACTUAL ≤ 0
4	Limiter	-	Hysteresis	-	Actual source	ACTUAL ≥ setpoint
5	Setpoint	Lock	Substitute value	-	-	-
6	Bit mask	-	Rel. decoder no.	-	-	Setpoint ≤ 0

TYPE	Function	QBit	PSK	Help cell	QDat	DFM result flag = 1 when
7	Decoder	-	Rel. decoder no.	-	-	Setpoint ≤ 0
8	Transfer	-	Target DB	Target DW	-	-
14	Limiter simple	-	Hysteresis below	-	Actual source	ACTUAL ≥ setpoint

PCU version V7 parameter set (max. 255 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Function	NL	ENUM	ENG	...	Function type
Setpoint	NL	DINT	ENG, LOG	0	Setpoint
... function-dependent parameters, see function descriptions below					
ActualValue	NL	DINT	RT, LOG	0	Actual value
InitActVal	NL	BOOL	ENG	FALSE	Initialize actual value if setpoint=#
... function-dependent parameters, see function descriptions below					
ParamError	HR	BOOL	RT	FALSE	Parametrization error
Result	HR	BOOL	RT	FALSE	Result bit

Result flag of the DFM modules DFM1, DFM2, DFM3 and DFM4

DFM number								DFM1	DFM2	DFM3	DFM4
Bit address								Byte address			
.0	.1	.2	.3	.4	.5	.6	.7				
1	2	3	4	5	6	7	8	760	792	824	1688
9	10	11	12	13	14	15	16	761	793	825	1689
17	18	19	20	21	22	23	24	762	794	826	1690
25	26	27	28	29	30	31	32	763	795	827	1691
33	34	35	36	37	38	39	40	764	796	828	1692
41	42	43	44	45	46	47	48	765	797	829	1693
49	50	51	52	53	54	55	56	766	798	830	1694
57	58	59	60	61	62	63	64	767	799	831	1695
65	66	67	68	69	70	71	72	768	800	832	1696
73	74	75	76	77	78	79	80	769	801	833	1697
81	82	83	84	85	86	87	88	770	802	834	1698
89	90	91	92	93	94	95	96	771	803	835	1699
97	98	99	100	101	102	103	104	772	804	836	1700
105	106	107	108	109	110	111	112	773	805	837	1701
113	114	115	116	117	118	119	120	774	806	838	1702
121	122	123	124	125	126	127	128	775	807	839	1703
129	130	131	132	133	134	135	136	776	808	840	1704
137	138	139	140	141	142	143	144	777	809	841	1705
145	146	147	148	149	150	151	152	778	810	842	1706
153	154	155	156	157	158	159	160	779	811	843	1707

DFM number								DFM1	DFM2	DFM3	DFM4
Bit address								Byte address			
.0	.1	.2	.3	.4	.5	.6	.7				
161	162	163	164	165	166	167	168	780	812	844	1708
169	170	171	172	173	174	175	176	781	813	845	1709
177	178	179	180	181	182	183	184	782	814	846	1710
185	186	187	188	189	190	191	192	783	815	847	1711
193	194	195	196	197	198	199	200	784	816	848	1712
201	202	203	204	205	206	207	208	785	817	849	1713
209	210	211	212	213	214	215	216	786	818	850	1714
217	218	219	220	221	222	223	224	787	819	851	1715
225	226	227	228	229	230	231	232	788	820	852	1716
233	234	235	236	237	238	239	240	789	821	853	1717
241	242	243	244	245	246	247	248	790	822	854	1718
249	250	251	252	253	254	255		791	823	855	1719

Examples:

DFM 1.44	DFM result flag	M 765.3
DFM 2.85	DFM result flag	M 802.4
DFM 4.222	DFM result flag	M 1715.5

Function 1: Restarting timer

- When processing the DFM as an up timer and simultaneously starting the EOP, a value of zero is loaded into the actual value cell. The DFM result flag is set when the setpoint specified in the recipe list is reached or exceeded.
- For setting the time base, a reduction factor is specified in the `Reduction` control constants related to the time base cycle 1 second (e.g. `Reduction` = 1 --> time base = 1 second, `Reduction` = 6 --> time base = 1/10 minute), i.e. at `Reduction` = 6 the actual value is increased by 1 after 6 seconds.
- The "running" of the timer is only enabled when a query result of "1" is set in the `Enable` parameter. At a query result of "0" the timer is "stopped."

Example

The counted value of an up timer is to be incremented by 1 every 6 seconds and displayed with one decimal place. The timer is only required in EOP 68 and it should be set as DFM no. 12 in the DFM1 group.

Parameter assignment

Parameter	Value	Description
Function	Restarting timer	Function type
Setpoint	0	Setpoint
ActualValue	0	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#

Parameter	Value	Description
Reduction	6	Reduction
Enable	U M3.1	Enable execution

Enabling a timer in an EOP FC

AN	M 102.3	Pulse step end
AN	M 761.3	Result flag DFM1.12
=	M 3.1	Enable flag

Calling a DFM block in an EOP FC

CALL	FC 636	Processing group DFM
iDfmType:=	1	DFM group = 1
iDfm:=	12	DFM number = 12

Function 2: Consecutive up timer (over several steps)

- When processing the DFM as a consecutive up timer for the first time the value zero is loaded into the actual value cell. As opposed to the non-consecutive up timer, the actual value is not set to zero at each step. The DFM result flag is set when the setpoint specified in the recipe list is reached or exceeded.
- For setting the time base, a reduction factor is specified in the `Reduction` control constants related to the time base cycle 1 second (e.g. `Reduction` = 1 --> time base = 1 second, `Reduction` = 6 --> time base = 1/10 minute), i.e. at `Reduction` = 6 the actual value is increased by 1 after 6 seconds.
- The "running" of the timer is only enabled when a query result of "1" is set in the `Enable` parameter. At a query result of "0" the timer is "stopped."

Example

The counted value of a consecutive up timer will be incremented by 1 each minute and displayed without a decimal place. The display occurs in unit 3. The timer is to be set as DFM no. 13 in group DFM1.

Parameter assignment

Parameter	Value	Description
Function	Consecutive timer	Function type
Setpoint	0	Setpoint
ActalValue	0	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
Reduction	6	Reduction
Enable	U M5.1	Enable execution

Enabling a timer in a sequence FB

A	M 656.2	Unit 3 running
AN	M 761.4	Result flag DFM1.13
=	M 5.1	Enable flag

Calling a DFM block in a sequence FB

CALL	FC 636	Processing group DFM
iDfmType:=	1	DFM group = 1
iDfm:=	13	DFM number = 13

Function 3: Down timer

- When processing the DFM as a down timer and simultaneously starting the EOP, the setpoint is loaded into the actual value cell from the recipe list. The DFM result flag is set when the value reaches or falls below zero.
- For setting the time base, a reduction factor is specified in the `Reduction` control constants related to the time base cycle 1 second (e.g. `Reduction` = 1 --> time base = 1 second, `Reduction` = 6 --> time base = 1/10 minute), i.e. at `Reduction` = 6 the actual value is increased by 1 after 6 seconds.
- The "running" of the timer is only enabled when a query result of "1" is set in the `Enable` parameter. At a query result of "0" the timer is "stopped."

Example

The counter value of a time step backward is to be reduced by one every second. The display occurs in unit 42. The timer is to be set as DFM no. 35 in group DFM2.

Parameter assignment

Parameter	Value	Description
Function	Down timer	Function type
Setpoint	0	Setpoint
ActualValue	0	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
Reduction	1	Reduction
Enable	U M23.4	Enable execution

Enabling a timer in a sequence FB

A	M 661.1	Sequence-42 running
AN	M 796.2	Result flag DFM2 Nr. 35
=	M 23.4	Enable flag

Calling a DFM block in a sequence FB

CALL	FC 636	Processing group DFM
iDfmType:=	2	DFM group = 2
iDfm:=	35	DFM number = 35

Function 4: Limiter

- When the EOP starts, the setpoint is loaded into the setpoint cell.
- The source of the actual value is set in the `ActualValueRef` parameter.
- The upper and lower hysteresis band is set in the `HysterAbove, HysterBelow` parameters. The actual value set in the data source is compared with the setpoint. The following parameters indicate the status of the actual value.
  - `AvOutsideHyst` – indicates that the process value is outside the two hysteresis bands (above and below)
  - `AvInsideHystAbove` – indicates that the process value is inside the upper hysteresis band
  - `AvInsideHystBelow` – indicates that the process value is inside the lower hysteresis band
  - `AvOutsideHystAbove` – indicates hysteresis above exceeded
  - `AvOutsideHystBelow` – indicates hysteresis below underrun
  - `Result` – `TRUE`: Actual value > setpoint
  - `Result` – `FALSE`: Actual value < setpoint - `HysterBelow`
- These parameters can be made visible in the parameter settings in Options / Hidden attributes. The bits that belong to the parameters can be read in from the datasets.

Example

The temperature of the MBPF (AIN 8) should be displayed with one decimal place in unit 4. The limiter should be set in DFM no. 58 of group DFM1.

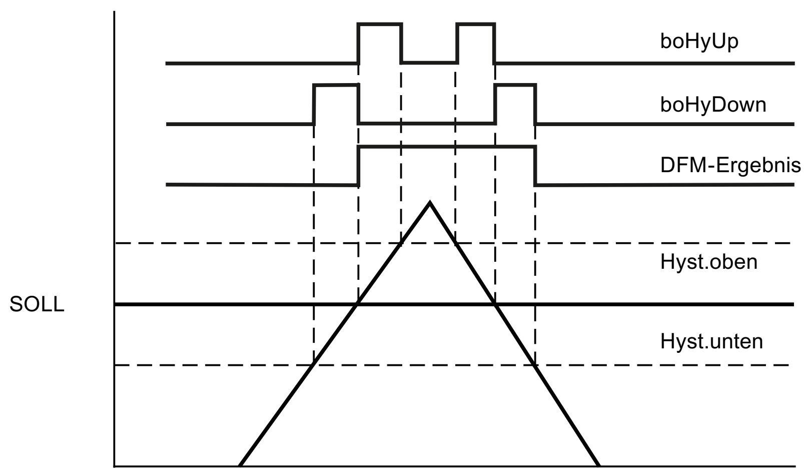
Parameter assignment

Parameter	Value	Description
Function	Limiter	Function type
Setpoint	0	Setpoint
ActualValue Ref	MESS, 8, ProcValAct	Reference to actual value
ActualValue	0	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
HysterAbove	6	Hysteresis above
HysterBelow	0	Hysteresis below

Call in a sequence FB

CALL	FC 636	Processing group DFM
iDfmType:=	1	DFM group = 1
iDfm:=	58	DFM number = 58

Function



Function 5: Setpoint

- The setpoint function is used to specify setpoints (for analog output from the recipe list). When the EOP starts, the setpoint is loaded into the setpoint cell.
- In parameter `UseSubstValue` , the STEP 7 command to query user flags for the use of the substitute value is set. If the query result is "1", the set `SubstValue` substitute value is loaded into the actual value cell. If the query result is "0" the setpoint specified in the recipe list is loaded into the actual value cell.
- The DFM result flag is irrelevant.

Example

A flow volume needs to be set as the setpoint. If the pump is not running, a setpoint of 0 hl/h should be specified. The display occurs in unit 15. The setpoint step should be set in DFM no. 31 of group DFM1.

Parameter assignment

Parameter	Value	Description
Function	Setpoint	Function type
Setpoint	0	Setpoint
ActualValue	0	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
SubstValue	0	Substitute value
UseSubstValue	U M23.4	Use substitute value

Call in a sequence FB

	AN	M 657.6	Unit 15 not running
	SPB	NEXT	Skip FC call, else
	CALL	FC 636	Processing group DFM
	iDfmType:=	1	DFM group = 1
	iDFM:=	31	DFM number = 31
NEXT:			

Function 6: Bit mask

- Mask 32 of 32 supports recipe-controlled selection functions or route switches.
- When the EOP starts, the setpoint is loaded into the setpoint cell. The setpoint is depicted in the actual value cell. The result flag is not set.
- Depending on the relative decoder number (0 ... 2) , the setpoint `SpDint` is transferred to the first 32 bits of the flag area of the selected decoder. The individual bits can then be evaluated in the EOP.
- The corresponding flag bytes are occupied from the high byte to the low byte.
  Flag area for the relative decoder numbers 0 ... 2:
  Decoder no. = 0: MB 688 ... 691
  Decoder no. = 1: MB 696 ... 699
  Decoder no. = 2: MB 704 ... 707

Example

16 bits from the bit mask are to be used for a route switch and the setpoint in decoder no. 1 will be mapped. The call takes place only in EOP 312. The bit mask should be set in DFM no. 73 of group DFM1.

Parameter assignment

Parameter	Value	Description
Function	Bit mask	Function type
Setpoint	0	Setpoint
ActualValue	0	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
DecoderNo	1	Decoder number

Call in an EOP FC

	U	M 102.3	Pulse step end
	SPB	NEXT	Skip FC call, else
	CALL	FC 636	Processing group DFM
	iDfmType:=	1	DFM group = 1
	iDfm:=	73	DFM number = 73
NEXT:			

Function 7: Decoder

- This decoder supports recipe-controlled selection functions or route switches and operating modes.
- When the EOP starts, the setpoint is loaded into the setpoint cell. The setpoint is depicted in the actual value cell. At a setpoint ≤ 0 the DFM result flag is set.
- Depending on the relative decoder number (0 ... 2), by setting the setpoint `SpDint` (value 1 ... 64), the relevant flag is set in the specified decoder. This can then be evaluated in the EOP.
- The corresponding flag bytes are occupied from high byte to low byte.
  Flag range for rel. decoder numbers 0 ... 2:
  Decoder no. = 0: MB 688 ... 695
  Decoder no. = 1: MB 696 ... 703
  Decoder no. = 2: MB 704 ... 711

Example 1

For the silo selection (silo 0...23), decoder 1 of 64 is to be used. Depending on the setpoint, a flag is set in decoder no. 0. The call takes place only in EOP 11. The decoder should be set in DFM no. 74 of group DFM1.

Parameter assignment

Parameter	Value	Description
Function	Decoder	Function type
Setpoint	0	Setpoint
ActualValue	0	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
DecoderNo	0	Decoder number

Call in an EOP FC

	A	M 102.3	Pulse step end
	SPB	NEXT	Skip FC call, else
	CALL	FC 636	Processing group DFM
	iDfmType:=	1	DFM group = 1
	iDfm:=	74	DFM number = 74
NEXT:			

Example 2

A text list should be used to select which steam valves are to be opened when heating up. Dependent on the text number in the text list, a flag is set in decoder no. 2 by decoder 1 of 64. The call takes place only in EOP 36. The decoder should be set in DFM no. 75 of group DFM1.

Parameter assignment

Parameter	Value	Description
Function	Decoder	Function type
Setpoint	0	Setpoint

Parameter	Value	Description
ActualValue	0	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
DecoderNo	2	Decoder number

Call in an EOP FC

	A	M 102.3	Pulse step end
	SPB	NEXT	Skip FC call, else
	CALL	FC 636	Processing group DFM
	iDfmType;=	1	DFM group = 1
	IDfm:=	75	DFM number = 75
NEXT:			

Example text list '\PCUxxx\TEXTE\DAMPF.TXT'

Line	Contents	Description
Line 0:		Blank line
Line 1:	Steam valve D101	Upon selection: SW = 1 → M 704.0 = 1
Line 2:	Steam valve D102	Upon selection: SW = 2 → M 704.1 = 1
Line 3:	Steam valve D101+D102	Upon selection: SW = 3 → M 704.2 = 1

Function 8: Transfer

- When processing the DFM as a routing block, a data block is opened.
- The number of the data block (DB) is specified in the `TargetDB` parameter. The setpoint is transferred from the recipe list to this data block as a double word. The target address is stored in the `TargetDBD` parameter. In addition, the `TargetDBD` parameter is increased by 4 bytes and the double word addressed in this way is loaded is loaded from the data block as the actual value in the actual value cell. The DFM result flag is irrelevant.

Example

The transfer function is used to transfer the setpoint of a temperature to DB 62, DBD 8. The call takes place only in EOP 25. The decoder should be set in DFM no. 126 of group DFM2.

Parameter assignment

Parameter	Value	Description
Function	Transfer	Function type
Setpoint	0	Setpoint
ActualValue	0	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
TargetDB	62	Target DB
TargetDBD	8	Destination address

Call in an EOP FC

	O	M 102.3	Pulse step end
	ON	M 54.7	Flank flag
	SPB	NEXT	Skip FC call, else
	CALL	FC 636	Processing group DFM2
	iDfmType:=	2	DFM group = 2
	iDFM:=	126	DFM number = 126
NEXT:			

Function 10: Setpoint / actual value cell

- The setpoint / actual value cell is used to set a setpoint from the recipe list and to display the actual value from the process cell. This, for example, allows a controller setpoint to be set and the actual value of the controlled system is displayed as the return value.
- When the EOP starts, the setpoint is loaded into the setpoint cell.
- The source of the actual value is set in the QDat parameter.
- The DFM result flag is irrelevant.

Example

The flow is specified as a setpoint for the controller for a frequency controlled pump. The current value of the flow is displayed as the actual value. The flow (AIN 8) will be displayed with one decimal place in unit 15. The target-/actual value cell should be parameterized in DFM no. 42 of group DFM1.

Parameter assignment

Parameter	Value	Description
Function	Setpoint / actual value cell	Function type
Setpoint	0	Setpoint
ActualValue Ref	MESS, 8, ProcValAct	Reference to actual value
ActualValue	0	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#

Call in a sequence FB

	AN	M 657.6	Unit 15 not running
	SPB	NEXT	Skip FC call, else
	CALL	FC 636	Processing group DFM
	iDfmType:=	1	DFM group =1
	iDfm:=	42	DFM number = 42
NEXT:			

Function 11: Time

- The setpoint cell is used to set a time (value in seconds since 1.1.1970) from the recipe list as the setpoint `SpDint`.
- If the current CPU time on the AS exceeds this setpoint, this is signaled in the DFM event flag and in the `Result` parameter.

Parameter assignment

Parameter	Value	Description
Function	Time	Function type
Setpoint	0	Setpoint
ActualValue	0	Actual value

Function 12: Bit mask setpoint/actual

- When the EOP starts, the setpoint is loaded into the setpoint cell.
- The source of the actual value is set in the `ActualValueRef` parameter.
- **Difference from Function 6: Bit mask**: The result of the comparison is displayed using parameter `Result` (= DFM flag bit):
  - `Result` – TRUE: Actual value = Setpoint
  - `Result` – FALSE: Actual value ≠ Setpoint

Function 14: Limiter simple

- When the EOP starts, the setpoint is loaded into the setpoint cell.
- The source of the actual value is set in the `ActualValueRef` parameter.
- In the `HysterBelow` parameter, a low hysteresis band can be set. The actual value set in the data source is compared with the setpoint. The `Result` parameter (= DFM flag bit) indicates the status of the actual value.
  - `Result` – TRUE: Actual value > setpoint
  - `Result` – FALSE: Actual value < setpoint - `HysterBelow`
- These parameters can be made visible in the parameter settings in Options / Hidden attributes. The bits that belong to the parameters can be read in from the datasets.

Example

The temperature of the MBPF (AIN 8) should be displayed with one decimal place in unit 4. The limiter should be set in DFM no. 58 of group DFM1.

Parameter assignment

Parameter	Value	Description
Function	Limiter simple	Function type
Setpoint	0	Setpoint

Parameter	Value	Description
ActualValue Ref	MESS, 8, ProcValAct-Dint	Reference to actual value
ActualValue	0	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
HysterBelow	6	Hysteresis below

Call in a sequence FB

CALL	FC 636	Processing group DFM
iDfmType:=	1	DFM group = 1
iDfm:=	58	DFM number = 58

5.7.6.4 DFM5 - DFM8 - timers, limits, decoders (REAL)

Basics

DFM5 to DFM8 are only available as of PCU V7

Per PCU, four groups each with 255 instances of digital function modules for timers, limiters and decoders in the format REAL format can be used.

To update the DFMs and form the DFM result flag, DFM processing needs to be started explicitly in the unit or in the EOP. To do this, FC636 needs to be called with suitable parameters:

```
CALL FC636
(
    iDfmType := [DFM group],
    iDfm     := [DFM instance]
)
```

Parameter sets for the DFM5, 6, 7, 8 / DB 2105, 2106, 2107, 2108 block

PCU version V7 parameter set (max. 255 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Function	NL	ENUM	ENG	...	Function type
Setpoint	NL	REAL	ENG, LOG	0.000	Setpoint (raw format)
... function-dependent parameters, see function descriptions below					
ActualValue	NL	REAL	RT, LOG	0.000	Actual value
InitActVal	NL	BOOL	ENG	FALSE	Initialize actual value if setpoint=#
... function-dependent parameters, see function descriptions below					
ParamError	HR	BOOL	RT	FALSE	Parametrization error
Result	HR	BOOL	RT	FALSE	Result bit

Result flag of the DFM modules DFM5, DFM6, DFM7 and DFM8

DFM number								DFM5	DFM6	DFM7	DFM8
Bit address								Byte address			
.0	.1	.2	.3	.4	.5	.6	.7				
1	2	3	4	5	6	7	8	1720	1752	1784	1816
9	10	11	12	13	14	15	16	1721	1753	1785	1817
17	18	19	20	21	22	23	24	1722	1754	1786	1818
25	26	27	28	29	30	31	32	1723	1755	1787	1819
33	34	35	36	37	38	39	40	1724	1756	1788	1820
41	42	43	44	45	46	47	48	1725	1757	1789	1821
49	50	51	52	53	54	55	56	1726	1758	1790	1822
57	58	59	60	61	62	63	64	1727	1759	1791	1823
65	66	67	68	69	70	71	72	1728	1760	1792	1824
73	74	75	76	77	78	79	80	1729	1761	1793	1825
81	82	83	84	85	86	87	88	1730	1762	1794	1826
89	90	91	92	93	94	95	96	1731	1763	1795	1827
97	98	99	100	101	102	103	104	1732	1764	1796	1828
105	106	107	108	109	110	111	112	1733	1765	1797	1829
113	114	115	116	117	118	119	120	1734	1766	1798	1830
121	122	123	124	125	126	127	128	1735	1767	1799	1831
129	130	131	132	133	134	135	136	1736	1768	1800	1832
137	138	139	140	141	142	143	144	1737	1769	1801	1833
145	146	147	148	149	150	151	152	1738	1770	1802	1834
153	154	155	156	157	158	159	160	1739	1771	1803	1835
161	162	163	164	165	166	167	168	1740	1772	1804	1836
169	170	171	172	173	174	175	176	1741	1773	1805	1837
177	178	179	180	181	182	183	184	1742	1774	1806	1838
185	186	187	188	189	190	191	192	1743	1775	1807	1839
193	194	195	196	197	198	199	200	1744	1776	1808	1840
201	202	203	204	205	206	207	208	1745	1777	1809	1841
209	210	211	212	213	214	215	216	1746	1778	1810	1842
217	218	219	220	221	222	223	224	1747	1779	1811	1843
225	226	227	228	229	230	231	232	1748	1780	1812	1844
233	234	235	236	237	238	239	240	1749	1781	1813	1845
241	242	243	244	245	246	247	248	1750	1782	1814	1846
249	250	251	252	253	254	255		1751	1783	1815	1847

Example

DFM 5.44 DFM result flag M 1725.3

DFM 6.85 DFM result flag M 1762.4

Function 1: Restarting timer

- When processing the DFM as an up timer and simultaneously starting the EOP, a value of zero is loaded into the actual value cell. The DFM result flag is set when the setpoint specified in the recipe list is reached or exceeded.
- For setting the time base, a reduction factor is specified in the `Reduction` control constants related to the time base cycle 1 second (e.g. `Reduction` = 1 --> time base = 1 second, `Reduction` = 6 --> time base = 1/10 minute), i.e. at `Reduction` = 6 the actual value is increased by 1 after 6 seconds.
- The "running" of the timer is only enabled when a query result of "1" is set in the `Enable` parameter. At a query result of "0" the timer is "stopped."

Example

The counted value of an up timer is to be incremented by 1 every 6 seconds and displayed with one decimal place. The timer is only required in EOP 68 and it should be set as DFM no. 12 in the DFM5 group.

Parameter assignment

Parameter	Value	Description
Function	Restarting timer	Function type
Setpoint	0.000	Setpoint
ActualValue	0.000	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
Reduction	6	Reduction
Enable	U M3.1	Enable execution

Enabling a timer in an EOP FC

AN	M 102.3	Pulse step end
AN	M 1721.3	Result flag DFM5.12
=	M 3.1	Enable flag

Calling a DFM block in an EOP FC

CALL	FC 636	Processing group DFM
iDfmType:=	5	DFM group = 5
iDfm:=	12	DFM number = 12

Function 2: Consecutive up timer (over several steps)

- When processing the DFM as a consecutive up timer for the first time the value zero is loaded into the actual value cell. As opposed to the non-consecutive up timer, the actual value is not set to zero at each step. The DFM result flag is set when the setpoint specified in the recipe list is reached or exceeded.
- For setting the time base, a reduction factor is specified in the `Reduction` control constants related to the time base cycle 1 second (e.g. `Reduction` = 1 --> time base = 1 second, `Reduction` = 6 --> time base = 1/10 minute), i.e. at `Reduction` = 6 the actual value is increased by 1 after 6 seconds.
- The "running" of the timer is only enabled when a query result of "1" is set in the `Enable` parameter. At a query result of "0" the timer is "stopped."

Example

The counted value of a consecutive up timer will be incremented by 1 each minute and displayed without a decimal place. The display occurs in unit 3. The timer is to be set as DFM no. 13 in group DFM5.

Parameter assignment

Parameter	Value	Description
Function	Consecutive timer	Function type
Setpoint	0.000	Setpoint
ActualValue	0.000	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
Reduction	6	Reduction
Enable	U M5.1	Enable execution

Enabling a timer in a sequence FB

A	M 656.2	Unit 3 running
AN	M 1721.4	Result flag DFM5.13
=	M 5.1	Enable flag

Calling a DFM block in a sequence FB

CALL	FC 636	Processing group DFM
iDfmType:=	5	DFM group = 5
iDfm:=	13	DFM number = 13

Function 3: Down timer

- When processing the DFM as a down timer and simultaneously starting the EOP, the setpoint is loaded into the actual value cell from the recipe list. The DFM result flag is set when the value reaches or falls below zero.
- For setting the time base, a reduction factor is specified in the `Reduction` control constants related to the time base cycle 1 second (e.g. `Reduction` = 1 --> time base = 1 second, `Reduction` = 6 --> time base = 1/10 minute), i.e. at `Reduction` = 6 the actual value is increased by 1 after 6 seconds.
- The "running" of the timer is only enabled when a query result of "1" is set in the `Enable` parameter. At a query result of "0" the timer is "stopped."

Example

The counter value of a time step backward is to be reduced by one every second. The display occurs in unit 42. The timer is to be set as DFM no. 35 in group DFM6.

Parameter assignment

Parameter	Value	Description
Function	Down timer	Function type
Setpoint	0.000	Setpoint
ActualValue	0.000	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
Reduction	1	Reduction
Enable	U M23.4	Enable execution

Enabling a timer in a sequence FB

A	M 661.1	Sequence-42 running
AN	M 1756.2	Result flag DFM6.35
=	M 23.4	Enable flag

Calling a DFM block in a sequence FB

CALL	FC 636	Processing group DFM
iDfmType:=	6	DFM group = 6
iDfm:=	35	DFM number = 35

Function 4: Limiter

- When the EOP starts, the setpoint is loaded into the setpoint cell.
- The source of the actual value is set in the `ActualValueRef` parameter.

- The upper and lower hysteresis band is set in the `HysterAbove, HysterBelow` parameters. The actual value set in the data source is compared with the setpoint. The following parameters indicate the status of the actual value.
  - `AvOutsideHyst` – indicates that the process value is outside the two hysteresis bands (above and below)
  - `AvInsideHystAbove` – indicates that the process value is inside the upper hysteresis band
  - `AvInsideHystBelow` – indicates that the process value is inside the lower hysteresis band
  - `AvOutsideHystAbove` – indicates hysteresis above exceeded
  - `AvOutsideHystBelow` – indicates hysteresis below underrun
  - `Result – TRUE`: Actual value > setpoint
  - `Result – FALSE`: Actual value < setpoint - `HysterBelow`
- These parameters can be made visible in the parameter settings in Options / Hidden attributes. The bits that belong to the parameters can be read in from the datasets.

Example

The temperature of the MBPF (AIN 8) should be displayed with one decimal place in unit 4. The limiter should be set in DFM no. 58 of group DFM5.

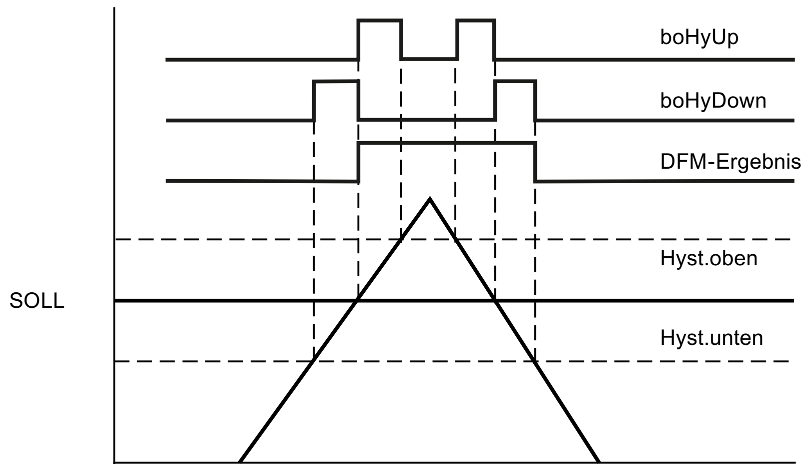
Parameter assignment

Parameter	Value	Description
Function	Limiter	Function type
Setpoint	0.000	Setpoint
ActualValue Ref	MESS, 8, ProcValAct	Reference to actual value
ActualValue	0.000	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
HysterAbove	6	Hysteresis above
HysterBelow	0	Hysteresis below

Call in a sequence FB

CALL	FC 636	Processing group DFM
iDfmType:=	5	DFM group = 5
iDfm:=	58	DFM number = 58

Function



Function 5: Setpoint

- The setpoint function is used to specify setpoints (for analog output from the recipe list). When the EOP starts, the setpoint is loaded into the setpoint cell.
- In parameter `UseSubstValue` , the STEP 7 command to query user flags for the use of the substitute value is set. If the query result is "1", the set `SubstValue` substitute value is loaded into the actual value cell. If the query result is "0" the setpoint specified in the recipe list is loaded into the actual value cell.
- The DFM result flag is irrelevant.

Example

A flow volume needs to be set as the setpoint. If the pump is not running, a setpoint of 0 hl/h should be specified. The display occurs in unit 15. The setpoint is to be configured in DFM no. 31 in group DFM5.

Parameter assignment

Parameter	Value	Description
Function	Setpoint	Function type
Setpoint	0.000	Setpoint
ActualValue	0.000	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
SubstValue	0	Substitute value
UseSubstValue	U M23.4	Use substitute value

Call in a sequence FB

	AN	M 657.6	Unit 15 not running
	SPB	NEXT	Skip FC call, else
	CALL	FC 636	Processing group DFM
	iDfmType:=	5	DFM group =5

	iDfm:=	31	DFM number = 31
NEXT:			

Function 10: Setpoint / actual value cell

- The setpoint / actual value cell is used to set a setpoint from the recipe list and to display the actual value from the process cell. This, for example, allows a controller setpoint to be set and the actual value of the controlled system is displayed as the return value.
- When the EOP starts, the setpoint is loaded into the setpoint cell.
- The source of the actual value is set in the QDat parameter.
- The DFM result flag is irrelevant.
- `Result - TRUE:` Setpoint = actual value

Example

The flow is specified as a setpoint for the controller for a frequency controlled pump. The current value of the flow is displayed as the actual value. The flow (AIN 8) will be displayed with one decimal place in unit 15. The setpoint / actual value cell should be set in DFM no. 42 of group 5.

Parameter assignment

Parameter	Value	Description
Function	Setpoint / actual value cell	Function type
Setpoint	0.000	Setpoint
ActualValue Ref	MESS, 8, ProcValAct	Reference to actual value
ActualValue	0.000	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#

Call in a sequence FB

	AN	M 657.6	Unit 15 not running
	SPB	NEXT	Skip FC call, else
	CALL	FC 636	Processing group DFM
	iDfmType:=	5	DFM group =5
	iDfm:=	42	DFM number = 42
NEXT:			

Function 11: Time

- The setpoint cell is used to set a time (value in seconds since 1.1.1970) from the recipe list as the setpoint `SpDint`.
- If the current CPU time on the AS exceeds this setpoint, this is signaled in the DFM event flag and in the `Result` parameter.

Parameter assignment

Parameter	Value	Description
Function	Time	Function type
Setpoint	0.000	Setpoint
ActualValue	0.000	Actual value

Function 14: Limiter simple

- When the EOP starts, the setpoint is loaded into the setpoint cell.
- The source of the actual value is set in the `ActualValueRef` parameter.
- In the `HysterBelow` parameter, a low hysteresis band can be set. The actual value set in the data source is compared with the setpoint. The `Result` parameter (= DFM flag bit) indicates the status of the actual value.
  - `Result` – TRUE: Actual value > setpoint
  - `Result` – FALSE: Actual value < setpoint - `HysterBelow`
- These parameters can be made visible in the parameter settings in Options / Hidden attributes. The bits that belong to the parameters can be read in from the datasets.

Example

The temperature of the MBPF (AIN 8) should be displayed with one decimal place in unit 4. The limiter should be set in DFM no. 58 of group DFM5.

Parameter assignment

Parameter	Value	Description
Function	Limiter simple	Function type
Setpoint	0.000	Setpoint
ActualValue Ref	MESS, 8, ProcValAct	Reference to actual value
ActualValue	0.000	Actual value
InitActVal	FALSE	Initialize actual value if setpoint=#
HysterBelow	6	Hysteresis below

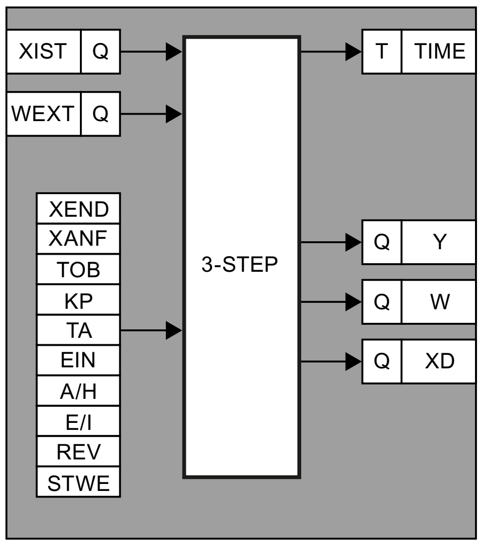
Call in a sequence FB

CALL	FC 636	Processing group DFM
iDfmType:=	5	DFM group = 5
iDfm:=	58	DFM number = 58

5.7.7 THRESTEP - Three-step controller

Basics

The block contains all necessary functions for a max. of 96/128 controllers per PCU. The controller is suitable for fixed setpoint control in temperature and pressure control loops.



From an actual value `XIST/ProcValRef` available as a source parameter and either an external setpoint `WEXT/ExtSpRef` available as a source parameter or an internal setpoint `W/EffExtSP` the control error `XD/Deviation` is formed.

Parameter sets for the THRESTEP / DB 744 block

PCU parameter set before Version V7 (max. 96 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
TEILANL	NL	USINT	ENG, LOG	0	Assigned unit
XIST	NL	QUEL	ENG	ANA, 1,ANA	Source actual value
WEXT	NL	QUEL	ENG	ANA, 1,ANA	Source external target value
W	NL	INT	RT, LOG	0	Effective setpoint
XEND	NL	INT	ENG, LOG	1000	End limit
XANF	NL	INT	ENG, LOG	0	Start limit
XD	NL	INT	RT, LOG	0	Control deviation
TOB	NL	INT	ENG, LOG	0	Dead band
KP	NL	INT	ENG, LOG	255	Response time amplification factor

Name	Access	D.Type	A.Type	Value	Comment
Y	NL	INT	RT, LOG	0	Manipulated variable in 100 ms
TA	NL	INT	ENG, LOG	1	Sampling time in seconds (0=Lock)
EIN	NL	BOOL	ENG	FALSE	Controller ON/OFF = 1/0
A/H	NL	BOOL	ENG	FALSE	Mode AUTO/MANUAL = 0/1
E/I	NL	BOOL	ENG	FALSE	Mode EXTERNAL/INTERNAL = 0/1
REV	NL	BOOL	ENG	FALSE	Reverse mode NO/YES = 0/1
MES	NL	BOOL	ENG	FALSE	Measured value monitoring
ANST	NL	BOOL	RT	FALSE	Direction CLOSED/OPEN = 0/1
STWE	NL	INT	ENG, LOG	255	Variable in manual operation in 100 msec.
AUF	NL	BOOL	RT	FALSE	Adjustment start OPEN = 1
ZU	NL	BOOL	RT	FALSE	Adjustment start CLOSED = 1
X	NL	INT	RT, LOG	0	Actual value
Wex	NL	INT	RT	0	External setpoint
Error	HI	BOOL	RT	FALSE	Processing error
INT_US	HR	INT	SYS	0	Internal
Status	HR	INT	RT, LOG	0	Internal

PCU version V7 parameter set (max. 128 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Sequence	NL	ENUM	ENG, LOG	<0>	Assigned unit
ProcValRef	NL	GREF	ENG	NULL	Source actual value
S7-1500: ProcVal-Ref_ST	HR	WORD	SYS	W#16#0000	Signal status
ProcValAct	RD	REAL	RT, LOG	0.000	Actual value
ExtSpRef	NL	GREF	ENG	NULL	Source external target value
S7-1500: ExtSpRef_ST	HR	WORD	SYS	W#16#0000	Signal status
ExpSpVal	HR	REAL	RT	0.000	External setpoint
EffExtSp	NL	REAL	RT, LOG	0.000	Effective setpoint
UpperLimit	NL	REAL	ENG, LOG	100.000	End limit
LowerLimit	NL	REAL	ENG, LOG	0.000	Start limit
Deviation	HR	REAL	RT, LOG	0.000	Control deviation
DeadBand	NL	REAL	ENG, LOG	0.000	Dead band
ActTmGain	NL	REAL	ENG, LOG	1.000	Response time amplification factor

Name	Access	D.Type	A.Type	Value	Comment
ActTmManVal	NL	INT	ENG, LOG	10	Response time in manual operation in 100 ms
ActTmEffVal	RD	INT	RT, LOG	0	Response time in 100 ms
ActTmMinVal	NL	INT	ENG	30	Minimum switchover time in 100 ms (wobble hold-off)
SamplingTime	NL	USINT	ENG, LOG	1	Sampling time in seconds (0=Lock)
ControllerAc-tive	NL	BOOL	ENG	FALSE	Activate controller
ManualMode	NL	BOOL	ENG	FALSE	Mode: 0=Auto, 1=Manual
InternalSp	NL	BOOL	ENG	FALSE	Mode: 0=External, 1=Internal
Reversing-Mode	NL	BOOL	ENG	FALSE	Reversing mode
ValueOutside-DeadBand	NL	BOOL	ENG	FALSE	Display measured value has left dead band
CtrlOn	RD	BOOL	RT	FALSE	Actuating direction: 0=Close, 1=Open
CmdOn	HR	BOOL	RT	FALSE	Trigger OPEN (manual)
CmdOff	HR	BOOL	RT	FALSE	Trigger CLOSED
Error	HR	BOOL	RT	FALSE	Processing error
NoteIsActive	HR	BOOL	RT	FALSE	Note is available
Status	HR	WORD	RT, LOG	W#16#0000	(Internal)

Note

New for PCU version V7

- A minimum on/off time can be set using the parameter `ActTmMinVal` (flutter inhibit)

Operation of 3-step controller

Automatic mode

The control error (`XD/Deviation`) is compared with a dead band (`TOB/DeadBand`).

- If the absolute value for `XD/Deviation` is lower than `TOB/DeadBand`, no actuating commands are generated.
- Otherwise, a time step is started with a value which is calculated using the formula:
  <V7: `Y` [in 100 msec] = `KP` [actuating time gain factor in 100ms] * 10 * `ABS (XD / (XEND - XANF`))
  =V7: `ActTmEffVal` [in 100 msec] = `ActTmGain` [actuating time gain factor in 100ms] * 10 * `ABS (Deviation / (UpperLimit - LowerLimit`))
  There is no position feedback. The direction bit (`ANST/CtrlOn`) is derived from the sign of the control error. In reversing mode (`REV/ReversingMode` =1) the direction bit is inverted.

Manual mode

To allow operation of the controller in Manual mode, an adjustable time value "STWE/ActTmManVal" (manipulated variable/manual operation) is output when necessary by the timer.
The output is triggered by setting the bit "ZU/CmdOff" or "AUF/CmdOn":

- When "ZU/CmdOff" has the value 1, the bit "MES/ValueOutsideDeadBand" is set and the direction bit ("ANST/CtrlOn") adopts the value 0.
- When "AUF/CmdOn" has the value 1, the bit "MES/ValueOutsideDeadBand" is set and the direction bit ("ANST/CtrlOn") adopts the value 1.
- If the timer is not running in Manual mode Y/ActTmEffVal changes to = 0 and bit "MES/ValueOutsideDeadBand" is set to = 0.

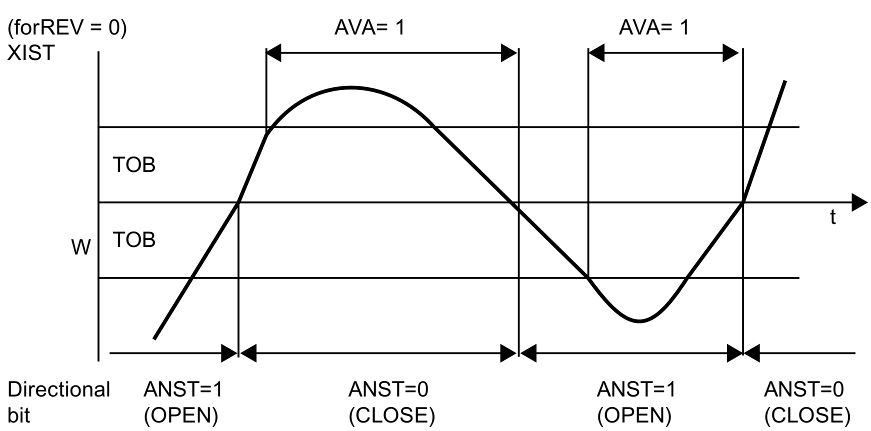
User program

The time step can be called up from a user program and the outputs for "more" or "less" can be actuated with the direction bit.

The user program must run at the 100 ms rate.

Using the bit for measured value monitoring (MES/ValueOutsideDeadBand) it is possible to detect whether the actual value has left the dead band upwards or downwards (MES/ValueOutsideDeadBand = 1). The bit becomes zero, when the actual value reaches the setpoint (MES/ValueOutsideDeadBand = 0).

If TOB/DeadBand = 0, MES/ValueOutsideDeadBand becomes = 1 as soon as XIST/ProcValRef >< W/EffExtSp (see figure)



Operation modes 3-step controller

Operating mode	Function
ON/OFF	In the case of "OFF" the time step is loaded with the value 0. The bit "`MES/ValueOutsideDeadBand`" is set to "0" `XD/Deviation` is calculated. In the case of "ON", control operation takes place insofar as "AUTO" is also present.
MANUAL/AUTO	In the case of "MANUAL", the time step is only actuated when an OPEN or CLOSE operation takes place. The actuating time in manual operation "`STWE/ActTmManVal`" is taken as the time value. In the user program the control elements can be actuated. In the "AUTO" mode, control operation takes place.
EXTERNAL/INTERNAL	If "EXTERNAL is set, "`WEXT/ExtSpRef`" is used as the setpoint, if "INTERNAL" is set, the setpoint "`W/EffExtSp`" is used, which can, for example, be set by the operator.

Assignment of time step block THRESTEP

No.	Time step	No.	Time step	No.	Time step	No. *)	Time step *)
1	T 128	33	T 160	65	T 192	97	T 224
2	T 129	34	T 161	66	T 193	98	T 225
3	T 130	35	T 162	67	T 194	99	T 226
4	T 131	36	T 163	68	T 195	100	T 227
5	T 132	37	T 164	69	T 196	101	T 228
6	T 133	38	T 165	70	T 197	102	T 229
7	T 134	39	T 166	71	T 198	103	T 230
8	T 135	40	T 167	72	T 199	104	T 231
9	T 136	41	T 168	73	T 200	105	T 232
10	T 137	42	T 169	74	T 201	106	T 233
11	T 138	43	T 170	75	T 202	107	T 234
12	T 139	44	T 171	76	T 203	108	T 235
13	T 140	45	T 172	77	T 204	109	T 236
14	T 141	46	T 173	78	T 205	110	T 237
15	T 142	47	T 174	79	T 206	111	T 238
16	T 143	48	T 175	80	T 207	112	T 239
17	T 144	49	T 176	81	T 208	113	T 240
18	T 145	50	T 177	82	T 209	114	T 241
19	T 146	51	T 178	83	T 210	115	T 242
20	T 147	52	T 179	84	T 211	116	T 243
21	T 148	53	T 180	85	T 212	117	T 244
22	T 149	54	T 181	86	T 213	118	T 245
23	T 150	55	T 182	87	T 214	119	T 246
24	T 151	56	T 183	88	T 215	120	T 247
25	T 152	57	T 184	89	T 216	121	T 248
26	T 153	58	T 185	90	T 217	122	T 249

No.	Time step	No.	Time step	No.	Time step	No. *)	Time step *)
27	T 154	59	T 186	91	T 218	123	T 250
28	T 155	60	T 187	92	T 219	124	T 251
29	T 156	61	T 188	93	T 220	125	T 252
30	T 157	62	T 189	94	T 221	126	T 253
31	T 158	63	T 190	95	T 222	127	T 254
32	T 159	64	T 191	96	T 223	128	T 255

*) applies to PCU version V7

Assignment of the direction bit "`ANST/CtrlOn`" THRESTEP block

Bit address OPEN/CLOSED flag for actuator open/close THRESTEP								Byte
.0	.1	.2	.3	.4	.5	.6	.7	Flag
1	2	3	4	5	6	7	8	1208
9	10	11	12	13	14	15	16	1209
17	18	19	20	21	22	23	24	1210
25	26	27	28	29	30	31	32	1211
33	34	35	36	37	38	39	40	1212
41	42	43	44	45	46	47	48	1213
49	50	51	52	53	54	55	56	1214
57	58	59	60	61	62	63	64	1215
65	66	67	68	69	70	71	72	1216
73	74	75	76	77	78	79	80	1217
81	82	83	84	85	86	87	88	1218
89	90	91	92	93	94	95	96	1219
97	98	99	100	101	102	103	104	1220 *)
105	106	107	108	109	110	111	112	1221 *)
113	114	115	116	117	118	119	120	1222 *)
121	122	123	124	125	126	127	128	1223 *)

*) applies to PCU version V7

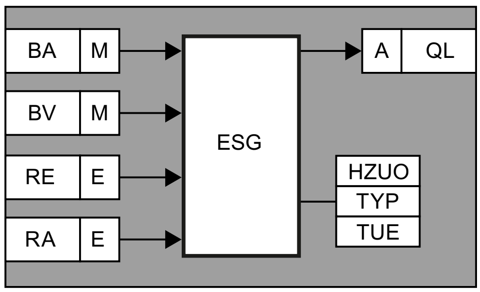
Allocation program time step to outputs

```
 Example:
A    T128              // Time step 3-step 1
A    M 1208.0          // Trigger bit "Open" = 1
=    Q x.x             // Set output for actuator open

A    T128
AN   M 1208.0          // Trigger bit for "CLOSE" = 0
=    Q x.y             // Set output for actuator close
```

5.7.8 ICM - Individual Control Modules (ICM1 ICM2 ICM3 ICM4)

Basics

The individual control module (ICM) handles the interlocking, control and monitoring of up to 1024 actuators such as valves, motors, etc per PCU. Each individual actuator type is assigned a parameter set containing the actuator type, the monitoring time etc.



Defined flags form the interface to the automatic programs (e.g. basic operations) and to an interlocking program provided for each ICM.

For each ICM a switch-on and switch-off delay may be configured. This is only active, if the ICM is in Automatic mode.

The signal of the load output `QL/RefCtrl` can be inverted by suitable parameter assignment.

With ICMs without feedback On `RE/Fbk1` or feedback Off `RA/Fbk0`, the feedback is simulated internally.

Error time:

- Error time monitoring is independent of the ON / OFF control, i.e. it starts when the feedback does not match the control.
- Exiting the feedback matching the control is only reported once the error time expires (0-15 s). This filters out, for example fast signal changes ("chatter" of end switches).
- If the monitoring time `TUE/MonTmSv` is longer than the error time `FltTime/FaultTimeSv`, the `TUE/MonTmSv` monitoring time is taken into account when turning on/off.

Note

Activation of the ICM groups 3 and 4 (only necessary for PCU < V7)

In order to use ICM groups 3 and 4, they have to be released in the scheduler. In the project supplied for the PCU V6, the corresponding parameter assignment is already included in the "SCHEDULE / DB720" block. This can be activated later in the user project as follows:

- Set the "Disabled" flag of the corresponding SCHEDULE datasets to "FALSE".
  ICM group 3 → data set 11, 12, and 13: Parameter Disable = "FALSE"
  ICM group 4 → data set 21, 22, and 23: Parameter Disable = "FALSE"
- Please make sure that all three data sets have been enabled for each ICM group.

Parameter sets for block ICM1, 2, 3, 4 / DB 726, 743, 748, 749

PCU Version before V7 parameter set (max. 4 x 255 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
STAT	HI	INT	RT	40	ICM status bits
TA	NL	USINT	ENG	1	Assigned unit
HZUO	NL	USINT	ENG, LOG	1	0 none, 1..64 manual group, >64 SeqManMode=1
TYP	NL	USINT	ENG, LOG	49	Type 8..13,16..21,32..38,48..53,128=frozen
TUE	NL	USINT	ENG, LOG	10	Monitoring time in seconds
TUEI	HI	USINT	RT, LOG	0	ACTUAL value monitoring time in seconds
QL	HI	BOOL	RT	FALSE	Load output
BSP	HI	BOOL	RT	FALSE	Command memory
QSP	HI	BOOL	RT	FALSE	Error memory
BV	HI	BOOL	RT	TRUE	Operation interlock
BA	HI	BOOL	RT	FALSE	Command automatic
RA	HI	BOOL	RT	TRUE	Feedback Off
RE	HI	BOOL	RT	FALSE	Feedback On
HD	HI	BOOL	RT	FALSE	1=manual, 0=automatic
M	HI	BOOL	RT	FALSE	Maintenance
ABM	HI	BOOL	RT	FALSE	Restricted release
REQM	HI	BOOL	RT	FALSE	Request M
ResREQM	HI	BOOL	RT	FALSE	Request: Reset 'Deactivation'
REQABM	HI	BOOL	RT	FALSE	Request ABM
ResREQABM	HI	BOOL	RT	FALSE	Request: Reset 'Hidden deactivation'
SETM	HI	BOOL	RT	FALSE	Set M
RESETM	HI	BOOL	RT	FALSE	Reset M
DIAGINFO	HI	BOOL	RT	FALSE	Diagnostics
Anz	HI	BOOL	RT	FALSE	Monitoring ICM display
ERRCOUNT	HI	DINT	RT, LOG	L#0	Error counter: double integer
OldQSP	HI	BOOL	RT	FALSE	Old value QSP
Warning	HI	BOOL	ENG	FALSE	Warning by start
Evz	HI	BOOL	RT	FALSE	Power-on delay running
BSPEvz	HI	BOOL	RT	FALSE	BSP power-on delay
Avz	HI	BOOL	RT	FALSE	Power-off delay running
BSPAvz	HI	BOOL	RT	FALSE	BSP power-off delay
BSPAvz	HI	BOOL	RT	FALSE	BSP power-off delay
SollEvz	NL	INT	ENG, LOG	0	Setpoint power-on delay
IstEvz	HI	INT	RT	0	Actual value power-on delay

Name	Access	D.Type	A.Type	Value	Comment
SollAvz	NL	INT	ENG, LOG	0	Setpoint power-off delay
IstAvz	HI	INT	RT	0	Actual value switch-off delay
InvertQL	NL	BOOL	ENG	FALSE	Load output to invert
NoRetSig	NL	BOOL	ENG	FALSE	ICM without feedback
T100MS	HI	USINT	ENG	0	0=1sec >0 time 100 ms ICM
WarnON	NL	BOOL	ENG	FALSE	Warning by start
ForceEna	NL	BOOL	ENG	FALSE	Forcing enable
SIM	NL	BOOL	ENG	FALSE	Simulation RE/RA
StatusDD	HR	DWORD	RT, LOG	DW#16 #00010028	Status as DWord
FltTime	NL	BYTE	ENG	B#16#00	Error time in sec (Bit0-3:setpoint , Bit4-7:actual value)
FltTRunO	HI	BOOL	RT	FALSE	Fault time run out

PCU Version V7 parameter set (max. 4 x 256 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Sequence	NL	ENUM	ENG	SE-QUENCE 001	Assigned unit 1
Sequence2	NL	ENUM	ENG	0	Assigned unit 2
ManualGroup	NL	USINT	ENG, LOG	0	0: none, 1..128: Manual group, >128: SeqManMode=1
Type	NL	USINT	ENG, LOG	0	Type 8..13,16..21,32..38,48..53,128=frozen
MonTmSv	NL	USINT	ENG, LOG	10	Monitoring time start value [sec]
MonTmAv	HR	USINT	RT, LOG	0	Monitoring time actual value [sec]
Ctrl	HR	BOOL	RT	FALSE	Load output
CmdMem	HR	BOOL	RT	FALSE	Command memory
ErrMem	HR	BOOL	RT	FALSE	Error memory
Interlock	HR	BOOL	RT	TRUE	Operation interlock
Cmd	HR	BOOL	RT	FALSE	Command automatic
Fbk0	HR	BOOL	RT	TRUE	Feedback Off
Fbk1	HR	BOOL	RT	FALSE	Feedback On
ManualMode	HI	BOOL	RT	FALSE	1=manual, 0=automatic
MsgLock	NL	BOOL	ENG	FALSE	Lock messages
MaintActive	HI	BOOL	RT	FALSE	Maintenance
Affected ByMaint	HI	BOOL	RT	FALSE	Affected by maintenance
ReqMaint	HI	BOOL	RT	FALSE	Request maintenance
ResetReq-Maint	HI	BOOL	RT	FALSE	Reset request maintenance

Name	Access	D.Type	A.Type	Value	Comment
ReqAffected ByMaint	HI	BOOL	RT	FALSE	Request 'affected by maintenance'
ResetReq AffectedBy-Maint	HI	BOOL	RT	FALSE	Reset request 'affected by mainte-nance'
FireSetMaint	HI	BOOL	RT	FALSE	Dispatch datagram 'set mainte-nance'
FireResetMaint	HI	BOOL	RT	FALSE	Dispatch datagram 'reset mainte-nance'
NoteIsActive	HR	BOOL	RT	FALSE	Note is available
ReadyToStart	HR	BOOL	RT	TRUE	Ready for power on
FbkTracking	HI	BOOL	RT	FALSE	Monitoring ICM display
ErrorCounter	HI	DINT	RT, LOG	L#0	Error counter
StartingAlarm IsActive	HI	BOOL	ENG	FALSE	Starting alarm is active
DelayOnRun-ning	HI	BOOL	RT	FALSE	Delay on is active
CmdMem DelayOn	HI	BOOL	RT	FALSE	Command on delay
DelayOffRun-ning	HI	BOOL	RT	FALSE	Off delay is active
CmdMem DelayOff	HI	BOOL	RT	FALSE	Command off delay
DelayOnSv	NL	INT	ENG, LOG	0	Start value delay on
DelayOnAv	HI	INT	RT, LOG	0	Actual value delay on
DelayOffSv	NL	INT	ENG	0	Delay off
DelayOffAv	HI	INT	RT	0	Actual value delay off
InvertCtrl	NL	BOOL	ENG	FALSE	Load output to invert
NoFbkAvaila-ble	NL	BOOL	ENG	FALSE	ICM without feedback
StartingAlarm	NL	BOOL	ENG	FALSE	Enable starting alarm
BypassInter-lock	NL	BOOL	ENG	FALSE	Bypass interlock
SimFbk	NL	BOOL	ENG	FALSE	Simulation RE/RA
STAT	HR	WORD	RT	W#16# 0028	ICM status bits
StatusDD	HR	DWORD	RT	DW#16# 00010028	(internal)
StatusLW	HR	LWORD	RT, LOG	DW#16#000 00000 00010028	(internal)
FaultTimeSv	NL	USINT	ENG	0	Start value error time [s]
FaultTimeAv	HR	USINT	RT	0	Actual value error time [s]
FaultTimeEx-pired	HR	BOOL	RT	FALSE	Fault time run out

Name	Access	D.Type	A.Type	Value	Comment
RefInterLock	NL	GREF	ENG	NULL	Source for interlocking
S7-1500: RefInter-Lock_ST	HR	WORD	SYS	W#16#0000	Signal status
RefFbk0	NL	GREF	ENG	NULL	Source for feedback off
S7-1500: RefFbk0_ST	HR	WORD	SYS	W#16#0000	Signal status
RefFbk1	NL	GREF	ENG	NULL	Source for feedback on
S7-1500: RefFbk1_ST	HR	WORD	SYS	W#16#0000	Signal status
RefCtrl	NL	PREF	ENG	NULL	Target for load output
S7-1500: RefCtrl_ST	HR	WORD	SYS	W#16#0000	Signal status
SafePosOn-SlActive	HI	BOOL	ENG	FALSE	Assume safety position during active seatlifting
SafePosOn-SlError	HI	BOOL	ENG	FALSE	Assume safety position in case of seatlifting feedback errors

Note

Features of V7-PCU

- Alternatively, the signals involved in the process connection (`RefInterLock, RefFbk0, RefFbk1, RefCtrl`) can be connected via references (D.type = GREF/ PREF) .
- If the respective references are interconnected, standard routing no longer takes place.
- If the interlock is interconnected to the result bit "Result" of a BLR object, the respective logic can be controlled and monitored in the IMC picture block.
(For more information on the BLR object, refer to the section BLR diagnosis/Online (Page 232))
- The new attribute "`ReadyToStart`" signalizes the readiness for activation by summarizing the following conditions:
  – No errors required acknowledgment
  – Operating interlocking fulfilled
  – Automatic mode enabled
- The instance number has been increased from 255 to 256 instances per ICM group. Consequently, the assignment of signalization bits for flags and process interfaces has also been increased by one bit. The address information in the following tables refers to the instance number of V7 PCU (=256), i.e. the last bit is invalid for PCU Version < V7.
- Using the engineering attribute "`MsgLock`" the generation of messages by the instance can be suppressed.
The operational messages are suppressed.
  – Switchover man/auto
  – Forcing on/off
  – Simulation on/off
  – On/off in manual mode
- Additional feedback status bits are available in the data record of the ICM instance:
  – xFbk0WithError - Feedback off/closed, error active
  – xFbk1WithError - Feedback on/opened, error active
  – xFbk0NoError - Feedback off/closed, no error
  – xFbk1NoError - Feedback on/opened, no error

Assignment of actuator type - Type for the ICM block

The parameter type consists of bits with the following meaning:

Bit0:	Feedback with positive logic
Bit1:	No Feedback OFF
Bit2:	Motor
Bit3:	-
Bit4:	"Internal" manual control level
Bit5:	Without override

Bit6:	-
Bit7:	Disable editing

Type	Actuator	RE/Fbk1	RA/Fbk0	Manual level
				Hardware level subordinated (bumplessness when Manual → Auto)
8	Valve	0 = open		External, with overwrite
9	Valve	1 = open		External, with overwrite
11	Valve	1 = open	1 = closed	External, with overwrite
12	Motor	0 = on		External, with overwrite
13	Motor	1 = on		External, with overwrite
				Manual level in PCU (bumplessness when Manual → Auto)
16	Valve	0 = open		Internal, with overwrite
17	Valve	1 = open		Internal, with overwrite
19	Valve	1 = open	1 = closed	Internal, with overwrite
20	Motor	0 = on		Internal, with overwrite
21	Motor	1 = on		Internal, with overwrite
				Hardware level subordinated (no bumplessness when Manual → Auto)
32	Valve	0 = open		External, without overwrite
33	Valve	1 = open		External, without overwrite
35	Valve	1 = open	1 = closed	External, without overwrite
36	Motor	0 = on		External, without overwrite
37	Motor	1 = on		External, without overwrite
				Manual level in PCU (no bumplessness when Manual → Auto)
48	Valve	0 = open		Internal, without overwrite
49	Valve	1 = open		Internal, without overwrite
51	Valve	1 = open	1 = closed	Internal, without overwrite
52	Motor	0 = on		Internal, without overwrite
53	Motor	1 = on		Internal, without overwrite
128-255	ICM blocked			ICM is not processed

When the actuator is of type "Valve", the load output QL remains active in the event of a fault; when the type is "Motor", the load output is shut down.

Type	Description
Type 8 .. 13	With subordinated manual level (e.g. emergency control or C1 level) and bumpless switching Manual → Auto. During manual operation, the QL is switched off and the BSP/CmdMem is tracked to RE. After switching to the system (Auto), the control state triggered by the subordinated level is maintained. In Automatic mode, the ICM can be operated via the screen.
Type 16 .. 21	Manual level in system and bumpless switching Manual → Auto. During manual operation in Manual or Auto mode, the QL is tracked to the BSP/CmdMem (with overwrite in Automatic mode). During manual operation the BA/Cmd has no influence. Manual operation is carried out via the screen or with a separate FB which reverses the BSP/CmdMem depending on the keys of the manual inputs. When the operation mode has been switched from Manual to Automatic, the switching status of the actuator is maintained
Type 32 .. 37	With subordinated manual level (e.g. emergency control or C1 level) and no bumpless switching Manual → Auto. During manual operation, the QL is switched off and the BSP/CmdMem is tracked to RE. After switching over to the system (Auto), the status of the BA/Cmd is adopted in the BSP/CmdMem.
Type 48 .. 53	Manual level in system and no bumpless switching Manual → Auto. During manual operation the QL is tracked to BSP/CmdMem. During manual operation the BA/Cmd has no influence. Manual operation is carried out via the screen or with a separate FB which reverses the BSP/CmdMem depending on the keys of the manual inputs. After switching over from Manual → Auto, the status of the BA/Cmd is adopted in the BSP/CmdMem. Override is not possible in the Automatic mode. **Notice:** **After acknowledging a fault while in automatic mode, a pending** BA/Cmd **is adopted immediately in the** BSP/CmdMem **and the relevant actuator is switched on.**

Process interface for ICM block

- All system technology function blocks do not directly access inputs, outputs, and flags.
- In the standard delivery state, the FC726 routing block ensures the routing to the I/O interface.
- This function block is left open to allow the user to adapt the I/O assignment to his own requirements (e.g. for an electrical terminal block).
- The following DBs are provided as an interface:

DB no.	Function
DB601	ICM-BA
DB602	ICM-BV
DB603	ICM-RE (Feedback On)
DB604	ICM-RA (Feedback Off)
DB605	ICM-QL

The structure is the same for all data blocks.

The assignment of the individual ICM interface bits to the bit addresses of the data block is defined as follows:

ESG1.1 - ESG1.256 (V6: 255) = DBX 10.0 - DBX 41.7 (41.6)

ESG2.1 - ESG2.256 (V6: 255) = DBX 42.0 - DBX 73.7 (73.6)

ESG3.1 - ESG3.256 (V6: 255) = DBX 74.0 - DBX 105.7 (105.6)

ESG4.1 - ESG4.256 (V6: 255) = DBX 106.0 - DBX 137.7 (137.6)

Interlocks for the ICM block

As an alternative to the interlock logic with BLR elements, the use of ICM interlock blocks, as was standard in earlier system versions, is also possible. The delivered projects contain the corresponding block templates for this. The assignment of ICM interlocks to the interlock blocks results from the following tables:

PCU type S7-400	PCU Version before V7	PCU Version from V7
FB1226	ICM1 - 1 ... 128	ICM1 - 1 ... 256
FB1227	ICM1 - 129 ... 255	ICM2 - 1 ... 256
FB1228	ICM2 - 1 ... 128	ICM3 - 1 ... 256
FB1229	ICM2 - 129 ... 255	ICM4 - 1 ... 256
FB1230	ICM3 - 1 ... 128	
FB1231	ICM3 - 129 ... 255	
FB1232	ICM4 - 1 ... 128	
FB1233	ICM2 - 129 ... 255	

PCU type S7-1500	ICM group / instance
BmUsrCallIfIcm1LockFC [FC910]	ICM1 - 1 ... 256
BmUsrCallIfIcm2LockFC [FC911]	ICM2 - 1 ... 256
BmUsrCallIfIcm3LockFC [FC912]	ICM3 - 1 ... 256
BmUsrCallIfIcm4LockFC [FC913]	ICM4 - 1 ... 256

Application:

For both PCU types, the interlocks are routed to flags M 256.0 ... M 383.7 (for details, see following section).

Each interlock network must end with the interlock flag assignment:

In the parameter set of the respective ICM instance, the "RefInterLock" reference must not be set (= NULL).

User interface for ICM block

The connection of the individual control modules with the system, the user programs, and procedures is performed by the following signals:

		ICM1	ICM2	ICM3	ICM4
BV	Operation interlock For interlocking of actuator operation 0/1 = lock/release	M 256.0 - M 287.7	M 288.0 - M 319.7	M 320.0 - M 351.7	M 352.0 - M 383.7
BA	CommandAutomatic For the control of the final controlling elements out of basic operations or user-written programs	M 128.0 - M 159.7	M 160.0 - M 191.7	M 192.0 - M 223.7	M 224.0 - M 255.7
RE	Feedback On Control unit feedback for the state On or Open. (for details see table "Assignment of actuator type")	E 64.0 - E 95.7	E 96.0 - E 127.7	E 192.0 - E 223.7	E 224.0 - E 255.7
RA	Feedback Off Actuator acknowledgment of the status "off" or "closed" (for details see table "Assignment of actuator type")	E 128.0 - E 159.7	E 160.0 - E 191.7	E 256.0 - E 287.7	E 288.0 - E 319.7
QL	Load output For the control of the final controlling elements	A 64.0 - A 95.7	A 96.0 - A 127.7	A 128.0 - A 159.7	A 160.0 - A 191.7
HornErrorICM	Hooter operation Activated when there is an ICM error, processing and reset by user	M 99.5	M 99.5	M 99.5	M 99.5
HornGrpError	Horn operation group flag Activated when there is an ICM, AIN, and SEQU error, processing and reset by user	M 107.1	M 107.1	M 107.1	M 107.1

Status bits ICM block

The ICM block has the following internal bits:

ICM	
BSP CmdMem	Command memory Indicates the target circuit state of the actuator. • In automatic mode (HD = 0): The BSP/CmdMem is changed at a signal change of the BA/Cmd or by operator input on the screen. Both types of activation have the same rights. By changing the BSP/CmdMem, the monitoring time is triggered. • In manual mode: The BSP/CmdMem is tracked to the Feedback On. If for a pending BA/Cmd, the actuator is switched off via the BV/Interlock and subsequently released again, the BA/Cmd command is transferred again, i.e. the actuator is controlled again.
QSP ErrMem	Error memory is activated when the runtime monitoring responds or if there is double feedback and can be reset using the "QUITT" key if BSP/CmdMem and feedback match. Automatic report via message archive and horn operation flag (M99.5), if error type ICM is released.
HD ManualMode	Manual mode indicates whether Manual mode (HD = 1) or Automatic mode (HD = 0) is activated. The bit is manipulated by the system block "Manual Signal Distributor".

Example:

Actuator triggering

The operation of the individual control modules from the block PROGRAM USER or basic operations is carried out via the standardized communications interface.

BA: Command Automatic
ICM1: M 128.0 -M 159.7
ICM2: M 160.0 -M 191.7
ICM3: M 192.0 -M 223.7
ICM4: M 224.0 – M 255.7

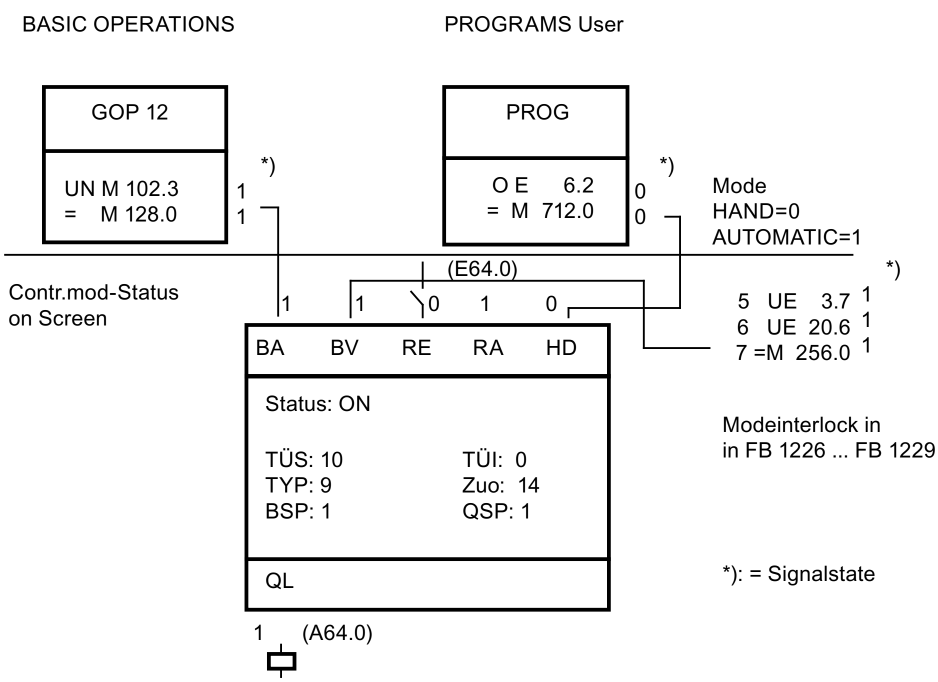
The BA flag is for triggering the actuators from the basic operations (switching on/off) or the block USER PROGRAM.

Signal occupancy CA for the actuators 1 to 256: see signal occupancy table.

Example:

Assignment group 1:	
ICM1	= M 128.0
ICM2	= M 128.1
ICM9	= M 129.0
ICM256	= M 159.7

Example ICM1:



Signal allocation to ICM block

Signal allocation for the control module groups 1 and 2

ICM number								ICM group 1					ICM group 2				
Bit address								BV	BA	RE	RA	QL	BV	BA	RE	RA	QL
.0	.1	.2	.3	.4	.5	.6	.7	(F)	(F)	(I)	(I)	(Q)	(F)	(F)	(I)	(I)	(Q)
1	2	3	4	5	6	7	8	256	128	64	128	64	288	160	96	160	96
9	10	11	12	13	14	15	16	257	129	65	129	65	289	161	97	161	97
17	18	19	20	21	22	23	24	258	130	66	130	66	290	162	98	162	98
25	26	27	28	29	30	31	32	259	131	67	131	67	291	163	99	163	99
33	34	35	36	37	38	39	40	260	132	68	132	68	292	164	100	164	100
41	42	43	44	45	46	47	48	261	133	69	133	69	293	165	101	165	101
49	50	51	52	53	54	55	56	262	134	70	134	70	294	166	102	166	102
57	58	59	60	61	62	63	64	263	135	71	135	71	295	167	103	167	103
65	66	67	68	69	70	71	72	264	136	72	136	72	296	168	104	168	104
73	74	75	76	77	78	79	80	265	137	73	137	73	297	169	105	169	105
81	82	83	84	85	86	87	88	266	138	74	138	74	298	170	106	170	106
89	90	91	92	93	94	95	96	267	139	75	139	75	299	171	107	171	107
97	98	99	100	101	102	103	104	268	140	76	140	76	300	172	108	172	108
105	106	107	108	109	110	111	112	269	141	77	141	77	301	173	109	173	109
113	114	115	116	117	118	119	120	270	142	78	142	78	302	174	110	174	110
121	122	123	124	125	126	127	128	271	143	79	143	79	303	175	111	175	111
129	130	131	132	133	134	135	136	272	144	80	144	80	304	176	112	176	112

ICM number								ICM group 1					ICM group 2				
Bit address								BV	BA	RE	RA	QL	BV	BA	RE	RA	QL
.0	.1	.2	.3	.4	.5	.6	.7	(F)	(F)	(I)	(I)	(Q)	(F)	(F)	(I)	(I)	(Q)
137	138	139	140	141	142	143	144	273	145	81	145	81	305	177	113	177	113
145	146	147	148	149	150	151	152	274	146	82	146	82	306	178	114	178	114
153	154	155	156	157	158	159	160	275	147	83	147	83	307	179	115	179	115
161	162	163	164	165	166	167	168	276	148	84	148	84	308	180	116	180	116
169	170	171	172	173	174	175	176	277	149	85	149	85	309	181	117	181	117
177	178	179	180	181	182	183	184	278	150	86	150	86	310	182	118	182	118
185	186	187	188	189	190	191	192	279	151	87	151	87	311	183	119	183	119
193	194	195	196	197	198	199	200	280	152	88	152	88	312	184	120	184	120
201	202	203	204	205	206	207	208	281	153	89	153	89	313	185	121	185	121
209	210	211	212	213	214	215	216	282	154	90	154	90	314	186	122	186	122
217	218	219	220	221	222	223	224	283	155	91	155	91	315	187	123	187	123
225	226	227	228	229	230	231	232	284	156	92	156	92	316	188	124	188	124
233	234	235	236	237	238	239	240	285	157	93	157	93	317	189	125	189	125
241	242	243	244	245	246	247	248	286	158	94	158	94	318	190	126	190	126
249	250	251	252	253	254	255	256	287	159	95	159	95	319	191	127	191	127

Signal allocation for the control module groups 3 and 4

ICM number								ICM group 3					ICM group 4				
Bit address								BV	BA	RE	RA	QL	BV	BA	RE	RA	QL
.0	.1	.2	.3	.4	.5	.6	.7	(F)	(F)	(I)	(I)	(Q)	(F)	(F)	(I)	(I)	(Q)
1	2	3	4	5	6	7	8	320	192	192	256	128	352	224	224	288	160
9	10	11	12	13	14	15	16	321	193	193	257	129	353	225	225	289	161
17	18	19	20	21	22	23	24	322	194	194	258	130	354	226	226	290	162
25	26	27	28	29	30	31	32	323	195	195	259	131	355	227	227	291	163
33	34	35	36	37	38	39	40	324	196	196	260	132	356	228	228	292	164
41	42	43	44	45	46	47	48	325	197	197	261	133	357	229	229	293	165
49	50	51	52	53	54	55	56	326	198	198	262	134	358	230	230	294	166
57	58	59	60	61	62	63	64	327	199	199	263	135	359	231	231	295	167
65	66	67	68	69	70	71	72	328	200	200	264	136	360	232	232	296	168
73	74	75	76	77	78	79	80	329	201	201	265	137	361	233	233	297	169
81	82	83	84	85	86	87	88	330	202	202	266	138	362	234	234	298	170
89	90	91	92	93	94	95	96	331	203	203	267	139	363	235	235	299	171
97	98	99	100	101	102	103	104	332	204	204	268	140	364	236	236	300	172
105	106	107	108	109	110	111	112	333	205	205	269	141	365	237	237	301	173
113	114	115	116	117	118	119	120	334	206	206	270	142	366	238	238	302	174
121	122	123	124	125	126	127	128	335	207	207	271	143	367	239	239	303	175
129	130	131	132	133	134	135	136	336	208	208	272	144	368	240	240	304	176
137	138	139	140	141	142	143	144	337	209	209	273	145	369	241	241	305	177
145	146	147	148	149	150	151	152	338	210	210	274	146	370	242	242	306	178
153	154	155	156	157	158	159	160	339	211	211	275	147	371	243	243	307	179

ICM number								ICM group 3					ICM group 4				
Bit address								BV	BA	RE	RA	QL	BV	BA	RE	RA	QL
.0	.1	.2	.3	.4	.5	.6	.7	(F)	(F)	(I)	(I)	(Q)	(F)	(F)	(I)	(I)	(Q)
161	162	163	164	165	166	167	168	340	212	212	276	148	372	244	244	308	180
169	170	171	172	173	174	175	176	341	213	213	277	149	373	245	245	309	181
177	178	179	180	181	182	183	184	342	214	214	278	150	374	246	246	310	182
185	186	187	188	189	190	191	192	343	215	215	279	151	375	247	247	311	183
193	194	195	196	197	198	199	200	344	216	216	280	152	376	248	248	312	184
201	202	203	204	205	206	207	208	345	217	217	281	153	377	249	249	313	185
209	210	211	212	213	214	215	216	346	218	218	282	154	378	250	250	314	186
217	218	219	220	221	222	223	224	347	219	219	283	155	379	251	251	315	187
225	226	227	228	229	230	231	232	348	220	220	284	156	380	252	252	316	188
233	234	235	236	237	238	239	240	349	221	221	285	157	381	253	253	317	189
241	242	243	244	245	246	247	248	350	222	222	286	158	382	254	254	318	190
249	250	251	252	253	254	255	256	351	223	223	287	159	383	255	255	319	191

Example:

Determination of the signals, e.g. for the actuator 172 (group 1):

- Search for actuator no. 172 in the left-hand table field
- There are located the byte addresses for BV, RE, RA, QL in the same line of the right table field
- The column heading which is part of the actuator no. represents the bit address

For actuator no. 172, the following applies:

- IL → M 277.3
- CA → M 149.3
- RE → I 85.3
- RA → I 149.3
- LO → O 85.3

Manual group assignment of the ICM

The actuators and unit controls for a system can be classified into up to 64/128 manual groups per PCU. The classification is of any kind but will generally comply with the technological requirements.

The allocation of an actuator is defined by the parameter HZUO of the ICM block.

HZUO (<V7) / ManualGroup (V7)	Significance for ICM 1/2, SEQUENCE

0	No routing of manual signals to the data set. In this case, the manual signal can be assigned by the user program. Example: ICM1.123:
	`By program`
	`=ESG1.au[123].boHD (<V7)`
	`=ESG1.au[123].xManualMode (V7)`

1...64 / 1...128	SeqManMode bit comes after the status of the relevant manual group
>64 / >128	SeqManMode bit is always = 1 !

The manual group releases can be assigned in the user program via defined flags (flag = "1" means "manual").

The system completes the distribution to ICM and sequence blocks.

- Allocation of the manual group 1...64 to MB 712...719
- Allocation of the manual group 65...128 to MB 720...727 (from PCU V7)

Bit address per manual group								Byte
.0	.1	.2	.3	.4	.5	.6	.7	Flag
1	2	3	4	5	6	7	8	712
9	10	11	12	13	14	15	16	713
17	18	19	20	21	22	23	24	714
25	26	27	28	29	30	31	32	715
33	34	35	36	37	38	39	40	716
41	42	43	44	45	46	47	48	717
49	50	51	52	53	54	55	56	718
57	58	59	60	61	62	63	64	719
65	66	67	68	69	70	71	72	720 *)
73	74	75	76	77	78	79	80	721 *)
81	82	83	84	85	86	87	88	722 *)
89	90	91	92	93	94	95	96	723 *)
97	98	99	100	101	102	103	104	724 *)
105	106	107	108	109	110	111	112	725 *)
113	114	115	116	117	118	119	120	726 *)
121	122	123	124	125	126	127	128	727 *)

*) as of PCU V7

Switch unit assignment by manual group

It is possible to change the unit assignment by FC calls for all actuators which are assigned to the same manual group.

CALL	FC723	Change unit assignment
	iTA := 12	New unit assignment
	IHGrp := 10	For all ICMs with HZUO=10

"Warning by start" function

- The function is only available in automatic mode.
- In DB701 (PCU_GEN), the setpoint of the warning time
- The bit `WarnON/StartingAlarm` in the ICM data set (see above) must be set
- **PCU < V7:** FC728 must be called in cyclic 1000 ms user FB (FB1225).
- **PCU V7:** FC728 is called automatically in the AS runtime system!

Sequence:

1. If an ICM with configured WarnOn function is started automatically, M99.0 (request bit) is set.
2. This request triggers the warning signal M99.1.
3. After the warning time has expired, the enable flag M99.2 is set for 10 sec. With the enable flag, the load output of the requesting ICM is enabled too. During the 10 second enable time, all other ICMs with configured "warning by start" function may be started directly without warning signal.
4. After the enable time (10 seconds) has expired, M99.2 is reset so a new warning signal is necessary on ICM start.

ICMs without configured "warn by start" function continue starting directly.

ICM functions for maintenance groups

If an ICM needs to be set to "Maintenance", all ICMs that belong to this partial route must be in the safe status (valves must not open, motors or pumps not run etc.). This changeover throughout the group is ensured by the PCU server of the IOS. ICMs are assigned to individual maintenance groups using the application of the same name in the Application Center / 'Engineering' tab. The ICMs are stored in the maintenance group PCU-related.

For the maintenance groups function, the following attributes in the ICM parameter set are used:

- **Bit SETM / FireSetMaint**
  If this bit has the value 1, the SetM telegram will be sent. If the entry in FIFO is successful, the bit will be taken back.
- **Bit RESETM / FireResetMaint**
  If this bit has the value 1, the ResetM telegram will be sent. If the entry in FIFO is successful, the bit will be taken back.

- **Bit REQABM / ReqAffectedByMaint**
  - If this bit is set and the BSP/CmdMem flag is set to 1, the AckABM telegram (neg) will be sent and the bit reset.
  - If this bit is set and the BSP/CmdMem flag is set to 0, the bit ABM/AffectedByMaint will be set and the AckABM (pos) telegram will be sent and the bit will be reset.
- **Bit REQM / ReqMaint**
  - If this bit is set and the bit ABM/AffectedByMaint is set, an AckM telegram (pos) will be sent and the bit will be reset; additionally, the M bit will be set in the ICM data set.
  - If this bit is set and the bit ABM/AffectedByMaint is not set, an AckM telegram (neg) will be sent and the bit will be reset.
- **Bit ABM / AffectedByMaint or M / MaintActive**
  If one of these bits is set, the CA flag will not be used anymore. The ICM always has 0 at the output.
  Manual operation is also not possible.
- **Bit ResREQM / ResetReqMaint**
  If this bit is set, the AckResM (pos) telegram will be sent and the bit will be reset; in addition to this, the M bit will be reset in the ICM data set.
- **Bit ResREQABM / ResetReqAffectedByMaint**
  - If this bit is set and the bit M is not set, an AckResABM (pos) telegram will be sent and the bit will be reset; in addition, the ABM bit will be reset in the ICM data set.
  - If this bit is set and the bit M is set, an AckResABM (neg) telegram will be sent and the bit will be reset.

Other IMC functions

- **Bit ForceEna / BypassInterlock**
  If this bit is set, the BV flag is ignored. The release is always given.
- **Bit SIM / SimFbk**
  This bit simulates the feedback of the ICM.

"Fast ICM" function

The response time of a PCU-dependent number of ICMs can be reduced by also operating them outside standard processing. Ideally, this is done in the alarm interface blocks (if available):

Cycle time	PCU < V7	PCU V7
100 ms	`FB 1222 (TIMER_100MS_BEG_USR_FB)`	`FB 1222 (BmUsr100msBeginFB)`
200 ms	`OB 34`	`FB 1230 (BmUsr200msBeginFB)`
500 ms	`OB 33`	`FB 1231 (BmUsr500msBeginFB)`

Cycle time	PCU < V7	PCU V7

Call	`CALL FC 727   // ICM_ALARM_FC`
	`iMode   := 35 // OB-Number`
	`iICM_GR := 1  // ICM group`
	`iICM_NR := 42 // ICM instance`
		`CALL FC 727      // BmIcmUsrCallFC`
		`IcmGroup    := 1  // ICM group`
		`IcmInstance := 42 // ICM instance`

Note

For PCUs <V7, the synchronization errors resulting from direct calls from the operating blocks OB33 and OB34 have to be unmasked according to the following PCU configuration:

```
AUF "SYS"
Call SFC 36 (Quittungsfehler ausblenden)
PRGFLT_SET_MASK := DBD160
ACCFLT_SET_MASK := DBD164
RET_VAL :=#iRetVal
PRGFLT_MASKED := DBD168
ACCFLT_MASKED := DBD172
CALL FC727 (...)
```

If an extra ICM call is removed, you will have to use the application "Parameterization" to switch back the ICM to cycle time in seconds:

- Set hidden attribute "T100MS" = 0.

See also

SLB - Seat lifting for double seat valves (Page 357)

5.7.9 FIFO1 to FIFO6 - PCU system data general

General

The block enables you to determine for each of the two servers which telegram types should be sent from the PCU to the corresponding server.

- FIFOs 1 to 3 (DB670, 671, 672) are assigned fixed to server 1
- FIFOs 4 to 6 (DB690, 691, 692) are assigned fixed to server 2

Note

- This FIFO parameterization generally takes place via the system configuration tool 'SiteCfg'. (See also section Modifying PCU settings (Page 121))
- Normally, it may not be changed by the user.
- The following description is therefore used for diagnostics only.

Description of telegram types

Type	Description	Message handler
Type 0	Locked	
Type 1	Free protocols (old) (PI5500)	
Type 2	Step protocols (old)	Srprot2.dll
Type 3	Messages	Prot_003.dll
Type 4	Recipe order V2	Gr.dll
Type 5	Step report time + 20/24 DFM	Srprot5.dll
Type 6	Free protocol: Received data	Prot_006.dll
Type 7	Date/time synchronization	Pcu_serv.exe
Type 8	Recipe order V3	Prot_008.dll/ recctrl.dll
Type 9	Batch order data: Batch start	Cas.dll
Type 10	Reserved for system extensions	
Type 11	Reserved for system extensions	
Type 12	Reserved for system extensions	
Type 13	Unit status	Ta.dll
Type 14	Free protocol order	Prot_014.dll
Type 15	Batch status	BaliData.dll
Type 16	"Life/killer" telegram for redundancy	Recctrl.dll
Type 17	Transaction store	Recctrl.dll
Type 18	Trigger status change (S7 only)	S7.dll
Type 19	ICM pipe-entity control	Entity.dll
...	Reserved for system extensions	
Type 22	Tank status frames	Statusport.dll
...	Reserved for system extensions	
Type 30	RCS: Route update	Rcs_port.dll
Type 31	RCS: Element update	Rcs_port.dll
Type 32	RCS: Request dynamic route ID	Rcs_port.dll
Type 33	RCS: Error triggering elements	Rcs_port.dll
Type 34	RCS: General request	Rcs_port.dll
Type 35	RCS: reserved	Rcs_port.dll
Type 36	RCS: "Life/killer" telegram for redundancy	Rcs_port.dll

Type	Description	Message handler
Type 37 to 127	Reserved for system extensions	
Type 128 ... 255	Free for user allocation	

5.7.10 GRUP_TA - Group block

General

Messages of technological objects (ICMs, AIN, PID, etc.) should be referenced to specific batches for several reasons. This is done using the `TA/Sequence` parameter in the data set for this object. If an event that needs reporting occurs, batch information is obtained depending on the value of the parameter as follows:

Case 1: Unit assignment

The parameter value corresponds to the number of the sequence data set from which the recipe type, job number, and batch number will be read. This assignment is selected for objects which are running inside of a sequence control (e.g. control modules, analog inputs/outputs, controllers of vessels).

- `TA` = 1 ... 64 for PCU version before V7
- `Sequence` = 1 ... 128 for PCU version V7

Case 2: No unit assignment

A pseudo batch assignment can be defined for objects which are not involved in a step control. Batch information is not taken from a sequence data set but from a GRUP_TA data record. The number of the GRUP_TA data set is calculated as follows:

- `TA` = 101 ... 255 for PCU version before V7 → GRUP_TA 1 ... 155
- `Sequence` =129 ... 255 for PCU version V7 → GRUP_TA 1 ... 127

GRUP_TA parameterization:

The GRUP_TA data records can be varied by parameterization or at runtime per program.

```
Example:
    L    SYS.u.byYear;
    T    GRP_TA.au[127].bYear;     //year
    L    255;
    T    GRP_TA.au[127].bReCType;  //recipe type
    L    254;
    T    GRP_TA.au[127].iRecipe;   //recipe number
    L    253;
    T    GRP_TA.au[127].iOrder;    //job number
    L    252;
```

```
T    GRP_TA.au[127].iBatch;    //batch number
```

Parameter sets for block GRUP_TA / DB 723

PCU parameter set before Version V7 (max. 155 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Jahr	NL	USINT	ENG	14	Year for recipe type, job no., batch no.
RTyp	NL	USINT	ENG	1	Recipe type
RezNr	NL	INT	ENG	0	Recipe number
AuftrNr	NL	INT	ENG	0	Order number
ChargeNr	NL	INT	ENG	0	Batch number

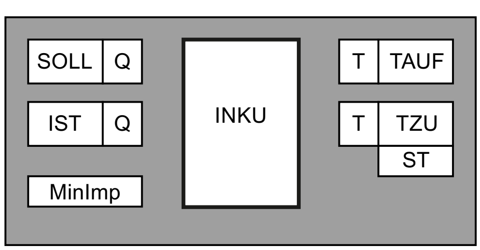
PCU version V7 parameter set (max. 127 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
RecCat	NL	USINT	ENG	1	Recipe category
BatchYear	NL	USINT	ENG	14	Batch year
RecipeNo	NL	INT	ENG	0	Recipe number
OrderNo	NL	INT	ENG	0	Order number
BatchNo	NL	INT	ENG	0	Batch number

5.7.11 INCO - Increment transformer

Basics

The function block is used for calculating opening or closing pulses with corresponding length for up to 16 motorized actuators. The user must take the pulse outputs of the assigned timers to digital outputs in the AS program



The block is interconnected with the manipulated variable (`SOLL/SetpointRef`) and the actuator feedback (`IST/ActualValueRef`). The actuating time (`ST/ActuatingTm`) in seconds of the drive is stored in the parameter set. On the basis of these parameters, INCO calculates the required setpoints for the timers at one second intervals.

```
PCU < V7: T = ST * (SOLL - IST) / 10 (time setpoint in 0.1 s)
```

```
PCU V7:   T = ActuatingTm * (SetpointRef - ActualValueRef) / 10 (time
setpoint in 0.1 s)
```

If the calculated T is greater than the parameterized minimum pulse width, depending on the actuation direction, one of the two time steps is started and the other one is canceled.

Parameter sets for block INCO / DB 729

PCU parameter set before Version V7 (max. 16 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
SOLL	NL	QUEL	ENG	ANA, 1,ANA	Position target value
IST	NL	QUEL	ENG	ANA, 1,ANA	Position actual value
ST_H	HI	INT	ENG	0	Manipulating time in seconds High
ST	NL	INT	ENG	1	Manipulating time in seconds
MinImp_H	HI	INT	ENG	0	Minimum impulse duration (1/10s) High
MinImp	NL	INT	ENG	0	Minimum impulse duration (1/10s)

PCU version V7 parameter set (max. 16 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
SetpointRef	NL	GREF	ENG	NULL	Reference to setpoint
S7-1500: Setpoin-tRef_ST	HR	WORD	SYS	W#16#0000	Signal status
SetpointPv	RD	REAL	RT	0.000	Process value SETPOINT
ActualValueR-ef	NL	GREF	ENG	NULL	Reference to actual value
S7-1500: ActualValueR-ef_ST	HR	WORD	SYS	W#16#0000	Signal status
ActualValuePv	RD	REAL	RT	0.000	Process value ACTUAL
ActuatingTm	NL	INT	ENG	1	Actuating time [s]
MinPulse	NL	INT	ENG	0	Min. pulse duration [100 msec]

Assignment of Simatic S7-CPU Timer ↔ INCO instance

No.	"Open"	"Closed"	No.	"Open"	"Closed"
1	T96	T97	9	T112	T113
2	T98	T99	10	T114	T115
3	T100	T101	11	T116	T117
4	T102	T103	12	T118	T119
5	T104	T105	13	T120	T121
6	T106	T107	14	T122	T123

7	T108	T109	15	T124	T125
8	T110	T111	16	T126	T127

Example:

Routing program timer to outputs

```
A       T96             Timer for INCO 01 "open"
=       Q xx.x          Output for INCO 01 "open"
A       T97             Timer for INCO 01 "closed"
=       Q yy.y          Output for INCO 01 "closed"
```

5.7.12 CURVSCAN - Curve target values

Basics

The system provides the possibility to represent any setpoint course as a curve in a graph. The coordinates are held in a data block. A maximum of 64 curves per PCU may be running at a time.

This block always calculates the current setpoint depending on the time basis. This setpoint "SP_Val" can be switched for further processing, e.g. to a PID controller.

A QBI release command "EnableCmd" can be set in the data set of the curve. The curve is running only if it is enabled. If "EnableCmd" = 0, sampling of the curve is stopped (e.g. in the event of a fault in the sequence)

There are two options for the initiation of the curve sampling:

Start mode	Description
Internal:	This means that the curve is initiated manually. To do this, the curve group number and the relative curve number in the group must be entered in the data set. The curve is then started by the start bit "Start from intern=1" in the data set. This start bit is reset automatically after the curve has finished. The curve can only be started manually if the source parameterized in the curve data set does not specify a curve number.
External:	This means that the curve is initiated via a sequence. In this case, the curve number is entered as a decimal number with two places after the point in a linkable block e.g. a setpoint in a DFM. This block is configured as a source in the curve's data set Place before the decimal point: Curve group number Place after the decimal point: Relative curve group number in the group After the curve is started, setpoint SW = 0 must be entered in the recipe list, as otherwise the curve would be restarted after the first run. As long as the curve number is set, it is started repeatedly. If the specified DFM curve number changes while the curve is in progress, the existing curve is terminated and the curve with the new number is started instead.

Parameter sets for block CURVESCAN / DB 747

PCU parameter set before Version V7 (max. 64 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
CrvGrp	NL	INT	ENG	0	Curve group
Crv	NL	INT	ENG	0	Curve number
ExtPtr	NL	QUEL	ENG	ANA,1, ANA	ext. Start: Source for curve number
ExtCrvGrp	HR	USINT	ENG	0	External curve group
ExtCrv	HR	USINT	ENG	0	External curve number
Start	HI	BOOL	ENG	FALSE	Start
Active	HI	BOOL	RT	FALSE	Curve active
ExtStart	HI	BOOL	RT	FALSE	External start
IntStart	HI	BOOL	RT	FALSE	Internal start
NewCoord	HI	BOOL	RT	FALSE	new coordinates
CurvEnd	HI	BOOL	RT	FALSE	Curve end
NoSyncMore	HI	BOOL	RT	FALSE	no synchronization point left
TimeVal	NL	INT	RT	0	Current time value
SP_Val	NL	INT	RT	0	Current setpoint
SyncVal	NL	INT	RT	0	Current synchronization value
TimeBase	NL	INT	ENG	0	Time base in seconds
TimeCnt	NL	INT	RT	0	Time counting value
EnableCmd	NL	STEP	ENG	U M108.1	Step7 query command for release of curve

Name	Access	D.Type	A.Type	Value	Comment
Enabled	NL	BOOL	RT	TRUE	released
lastTimeVal	NL	INT	RT	0	Last time value of the curve from table
lastSP_Val	NL	INT	RT	0	Last interpolation point of the curve from table
nextTimeVal	NL	INT	RT	0	Next time value of curve from table
nextSP_Val	NL	INT	RT	0	Next interpolation point of the curve from tab.
nextSyncTime	NL	INT	RT	0	Next time value for synchronization
nextSyncVal	NL	INT	RT	0	Next synchronization point from tab.
CoordNoVal	NL	INT	RT	0	No. interpolation pt. pair
CoordNoNextSync	NL	INT	RT	0	Interpolation point number of next synchr. point
ErrorCode	NL	INT	RT	0	Error number

PCU version V7 parameter set (max. 64 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
CrvGrp	NL	INT	ENG	0	Curve group
Crv	NL	INT	ENG	0	Curve number
ExtPtr	NL	GREF	ENG	NULL	ext. Start: Source for curve number
ExtCrvGrp	HR	USINT	ENG	0	External curve group
ExtCrv	HR	USINT	ENG	0	External curve number
Start	HI	BOOL	ENG	FALSE	Start
Active	HI	BOOL	RT	FALSE	Curve active
ExtStart	HI	BOOL	RT	FALSE	External start
IntStart	HI	BOOL	RT	FALSE	Internal start
NewCoord	HI	BOOL	RT	FALSE	new coordinates
CurvEnd	HI	BOOL	RT	FALSE	Curve end
NoSyncMore	HI	BOOL	RT	FALSE	no synchronization point left
TimeVal	NL	INT	RT	0	Current time value
SP_Val	NL	INT	RT	0	Current setpoint
SyncVal	NL	INT	RT	0	Current synchronization value
TimeBase	NL	INT	ENG	0	Time base in seconds
TimeCnt	NL	INT	RT	0	Time counting value
EnableCmd	NL	STEP	ENG	U M108.1	Step7 query command for release of curve
Enabled	NL	BOOL	RT	FALSE	released
lastTimeVal	NL	INT	RT	0	Last time value of the curve from table
lastSP_Val	NL	INT	RT	0	Last interpolation point of the curve from table
nextTimeVal	NL	INT	RT	0	Next time value of curve from table
nextSP_Val	NL	INT	RT	0	Next interpolation point of the curve from tab.
nextSyncTime	NL	INT	RT	0	Next time value for synchronization
nextSyncVal	NL	INT	RT	0	Next synchronization point from tab.

Name	Access	D.Type	A.Type	Value	Comment
CoordNoVal	NL	INT	RT	0	No. interpolation pt. pair
CoordNoNextSync	NL	INT	RT	0	Interpolation point number of next synchr. point
ErrorCode	NL	INT	RT	-2	Error number

Curve group parameter assignment IOS

The curve interpolation points are configured via the IOS application 'Curve input' in the 'Configuration' tab in the basic menu. The description is contained in chapter "Working with the application (Page 964)".

Synchronization points

If any action is to be made dependent on a synchronization value which can be entered in the setpoint curve,

- the block FC647 must be called. This block must be informed (as formal operands) of the curve number, the synchronization value to be queried as well as the recipe system. The block can be called in the EOP or in the user section of OB 1 or OB 35.
- On block FC 647, the data set of the curve (""`iCurveRecord`") for which the synchronization values are to be checked is assigned. Besides the value of the synchronization point ("`iSyncValue`") is handed over.
- The function FC 647 returns the value `"iRetVal"` as a result. If this value equals zero, no error occurred in the function. The transferred RLO of the function determines whether the synchronization point has been reached.
- If the RLO=1, the curve has reached the synchronization point. This result can, for example, be patched to a flag and/or used for the next step enabling condition of an EOP.
- The value range for the synchronization points amounts to Zsyn = 2 ... 32767.

Values 0 and 1 are occupied by the system and can be queried by the user:

- Zsyn = 0 --> curve not running
- Zsyn = 1 --> curve running and the first synchronization point not yet reached.

Example

FC 647 call in a EOP 44 (FC 1044), in which the curve running in data set 8 will be queried for the synchronization value Zsyn = 12. The RLO of FC 647 is assigned to flag M23.4 and can be used, for example, in the step enabling conditions.

FC 1044		
`CALL FC 647`		Processing of synchronization points
iCurveRecord	:=8	Data set in which the curve is processed
iSyncValue	:=12	Synchronization value
iRetVal	= MW 40	= 0, function has been executed without errors.
= M 23.4		RLO of M 23.4 = 1, when the synchronization value is reached

Canceling a running curve

- It is possible to stop and start the setpoint curve using the parameter "`EnableCmd`" on CURVSCAN. However, it is not possible to reset and restart the curve. The setpoint curve has to be run down before restarting is possible.
- As aborting and restart of a sequence cannot often wait until the curve has run down, it has to be killed in the following way: The parameter "`CurvEnd`" (curve end) in the CURVSCAN parameter set has to be set in this case.
- The system resets these bits.
- If the abort bit is set and the assigned curve is not active, the next curve start is aborted!

5.7.13 LINE - Line state

5.7.13.1 LINE - Line status

Area of application and operating principle

This function block manages pipeline states and provides a runtime monitoring function with interfaces and messages for the process status, including an operator and monitoring user interface for this purpose. Due to its general approach, the block can be used for a wide variety of pipeline types.

- The process state allows the handling of up to 255 pipe states (10 at delivery) and can be operated using the OS faceplate and / or control inputs.
- A timer is provided to monitor the remaining runtime for any process state.
- The material is selected in the OS faceplate on the basis of the materials defined in the material configuration dialog.

Parameter sets for block LINE / DB 2101

The instance attributes, such as pipe group, pipe type, unit assignment and references to measured values are configured via the parameterization (block class LINE <Pipe state>).

Parameter set (max. 256 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Sequence	NL	ENUM	ENG, LOG	SEQUENCE 001	Unit
RCSLinkElement	NL	ENUM	ENG	Unknown	RCS link element
LineGroup	NL	ENUM	ENG, LOG	LINEGROUP 1	Line group
LineType	NL	ENUM	ENG, LOG	LINETYPE 1	Line type

Name	Access	D.Type	A.Type	Value	Comment
InLineStatus1	HI	BOOL	RT	FALSE	Ext. signal for line status change on ExtLineStatus1
InLineStatus2	HI	BOOL	RT	FALSE	Ext. signal for line status change on ExtLineStatus2
InLineStatus3	HI	BOOL	RT	FALSE	Ext. signal for line status change on ExtLineStatus3
InLineStatus4	HI	BOOL	RT	FALSE	Ext. signal for line status change on ExtLineStatus4
InLineStatus5	HI	BOOL	RT	FALSE	Ext. signal for line status change on ExtLineStatus5
InLineStatus6	HI	BOOL	RT	FALSE	Ext. signal for line status change on ExtLineStatus6
InLineStatus7	HI	BOOL	RT	FALSE	Ext. signal for line status change on ExtLineStatus7
InLineStatus8	HI	BOOL	RT	FALSE	Ext. signal for line status change on ExtLineStatus8
InLineStatus9	HI	BOOL	RT	FALSE	Ext. signal for line status change on ExtLineStatus9
InLineStatus10	HI	BOOL	RT	FALSE	Ext. signal for line status change on ExtLineStatus10
ExtLineStatus1	NL	ENUM	ENG	(Undefined)	Line status in case InLineStatus1 is valid
ExtLineStatus2	NL	ENUM	ENG	Clean	Line status in case InLineStatus2 is valid
ExtLineStatus3	NL	ENUM	ENG	Product in pipe	Line status in case InLineStatus3 is valid
ExtLineStatus4	NL	ENUM	ENG	Water in pipe	Line status in case InLineStatus4 is valid
ExtLineStatus5	NL	ENUM	ENG	CIP	Line status in case InLineStatus5 is valid
ExtLineStatus6	NL	ENUM	ENG	LINESTATUS x	Line status in case InLineStatus6 is valid
ExtLineStatus7	NL	ENUM	ENG	LINESTATUS x	Line status in case InLineStatus7 is valid
ExtLineStatus8	NL	ENUM	ENG	LINESTATUS x	Line status in case InLineStatus8 is valid
ExtLineStatus9	NL	ENUM	ENG	LINESTATUS x	Line status in case InLineStatus9 is valid
ExtLineStatus10	NL	ENUM	ENG	LINESTATUS x	Line status in case InLineStatus10 is valid
StartLineStatus	NL	ENUM	ENG	(Undefined)	Line status start monitoring time
EndLineStatus	NL	ENUM	ENG	Clean	Line status after monitoring time expired
LineTimeSp	NL	DINT	ENG, LOG	L#3600	Monitoring time

Name	Access	D.Type	A.Type	Value	Comment
LineTimeMon	HI	BOOL	RT, LOG	FALSE	Monitoring time: 1 = monitoring on, 0 = monitoring off
Manual	HI	BOOL	RT, LOG	FALSE	1=Manual mode, 0=Automatic
MaterialManual	HI	BOOL	RT, LOG	FALSE	1=Manual mode for material modification
SetMaterial	HI	BOOL	RT, LOG	FALSE	Apply material ID
RstMaterial	HI	BOOL	RT, LOG	FALSE	Reset material ID in manual mode
NoteIsActive	HR	BOOL	RT, LOG	FALSE	Note is available
MsgLock	HI	BOOL	RT	FALSE	Message lock for commissioning
LineStatusOp	HR	ENUM	RT, LOG	(Undefined)	Line status in manual mode
MaterialIDOp	HR	DINT	RT, LOG	L#0	Material ID manual mode
MaterialIDRef	NL	GREF	ENG	NULL	Reference to material ID in automatic mode
S7-1500: MaterialIDRef_ST	HR	WORD	SYS	W#16#0000	Signal status
MaterialIDPv	HR	DINT	RT	L#0	Material ID automatic mode
LineStatusAct	HR	ENUM	RT, LOG	(Undefined)	Current line status
MaterialAct	HR	DINT	RT, LOG	L#0	Current material ID
LineTimeAct	HR	DINT	RT, LOG	L#0	Current monitoring time
LineRmTimeAct	HR	DINT	RT, LOG	L#3600	Remaining monitoring time
LineStatusAct1	HR	BOOL	RT	TRUE	Line status bit output 1
LineStatusAct2	HR	BOOL	RT	FALSE	Line status bit output 2
LineStatusAct3	HR	BOOL	RT	FALSE	Line status bit output 3
LineStatusAct4	HR	BOOL	RT	FALSE	Line status bit output 4
LineStatusAct5	HR	BOOL	RT	FALSE	Line status bit output 5
LineStatusAct6	HR	BOOL	RT	FALSE	Line status bit output 6
LineStatusAct7	HR	BOOL	RT	FALSE	Line status bit output 7
LineStatusAct8	HR	BOOL	RT	FALSE	Line status bit output 8
LineStatusAct9	HR	BOOL	RT	FALSE	Line status bit output 9
LineStatusAct10	HR	BOOL	RT	FALSE	Pipe state bit output 10
Status	HR	WORD	RT, LOG	W#16#0000	Status bits as a WORD

The text parameters of the line class are defined using the following enumerations and are set for the corresponding block attributes:

Description / Purpose	Text file	Block attribute (type ENUM)
Assignment to RCS link element (PCU - related)	...\PCU.nnn\texte.x\rcs_le.txt	`RCSLinkElement`
Definition of the line type	...\texte.x\LineType.txt	`LineType`
Definition of the line group	...\texte.x\LineGroup.txt	`LineGroup`
Assignment to unit (PCU - related)	...\PCU.nnn\texte.x\Unit.txt	`Sequence`
Definition of the line status, cleaning status	...\texte.x\LineStatus.txt	`ExtLineStatus1 … ExtLineStatus10, StartLineStatus, EndLineStatus, LineStatusOp, LineStatusAct`

5.7.13.2 Functions of 'Line'

General functions

The following general input / output signals are available:

Process state, output

The Line block provides line process status management with up to 255 statuses. The state definitions are contained in the enumeration "LineStatus". The following states are defined and preset here:

Table 5-3 Preset content in the text file 'LineStatus.txt'

Value	Object name
0	(Undefined)
1	Clean
2	Product in pipe
3	Water in pipe
4	CIP
5	...
...	...

Process state, control

The process state can be opened in the following ways:

- In manual mode via the OS faceplate
- In automatic mode, by assigning various block attributes in the user program

AUTO Mode:

If an input bit `InLineStatus1...InLineStatus10` is set, the process state configured at the allocated input `ExtLineStatus1...ExtLineStatus10` is transferred to the output value `LineStatusAct` . If more than one event bit is set, the status with the highest priority at the LineStatusAct output is adopted:

`InLineStatus1` = lowest priority

...

`InLineStatus10` = highest priority

To ensure smooth transition of the automatic / manual switchover, the `LineStatusOp` control value is tracked to the current status in automatic mode.

Process state, runtime monitoring

A monitoring timer is available to influence the process status. The following conditions apply for the handling of the timer:

- The timer setpoint `LineTimeSp` can be specified in the OS faceplate in the format (number of days, hr, min) and the `LineTimeMon` enable can be set.
- The timer remaining time `LineRmTimeAct` can be followed at the OS faceplate.
- Start condition 'Timer':
  If the current process status`LineStatusAct` corresponds to the configured value at `StartLineStatus` and the timer is enabled (`LineTimeMon =1`), it is started. The timer is reset automatically when the starting condition is no longer fulfilled.
- After the end of the configured time, the process state is set to the value configured at the input parameter `EndLineStatus`. In this case, an alarm is generated.

Material management:

The material currently in the line can be controlled as follows:

- In manual mode via the OS faceplate
- In automatic mode, by assigning various block attributes in the user program
- The auto/manual switchover is performed in the OS faceplate.
- Automatic mode (`MaterialManual = 0`):
  If the attribute `SetMaterial` is assigned (pos. signal edge), the value `MaterialIDPv` is transferred to the output `MaterialAct` and displayed on the faceplate. Instead of controlling the direct block attribute in the user program, the reference attribute `MaterialIDRef` can also be interconnected.
- Manual mode (`MaterialManual = 1`):
  In the OS faceplate, a manual material specification can be set for the line. The material is selected from a list box.
  The material is adopted with the Save button. As long as the material correction has not been saved, the "Back" button (`RstMaterial` parameter) can restore the last material that was in the line.

5.7.14 MAINT_ICM - Maintenance data

Basics

Switch alterations and operating hours are seized for every ICM 1023/1024 instances of the function are available.

with a maximum of 5 target values each. The overflow of these values may be output as messages which have to be acknowledged after maintenance has happened. Additional explanations on the configuration and project are contained in chapter Maintenance data (Page 998).

Maint_ICM manages the actual values of the switch alternations and operating hours counters as well as acknowledgements at the overflow of the parameterized target values.

Parameter sets for block MAINT_ICM / DB 682

PCU parameter set before Version V7 (max. 1023 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
STAT	HI	INT	RT, LOG	0	ACK state
SSp_SW1_Ok	NL	BOOL	RT	FALSE	Switching play maintenance work SW1 compl.
SSp_SW2_Ok	NL	BOOL	RT	FALSE	Switching play maintenance work SW2 compl.
SSp_SW3_Ok	NL	BOOL	RT	FALSE	Switching play maintenance work SW3 compl.
SSp_SW4_Ok	NL	BOOL	RT	FALSE	Switching play maintenance work SW4 compl.
SSp_SW5_Ok	NL	BOOL	RT	FALSE	Switching play maintenance work SW5 compl.
Std_SW1_Ok	NL	BOOL	RT	FALSE	Hours - maintenance work SW1 compl.
Std_SW2_Ok	NL	BOOL	RT	FALSE	Hours - maintenance work SW2 compl.
Std_SW3_Ok	NL	BOOL	RT	FALSE	Hours - maintenance work SW3 compl.
Std_SW4_Ok	NL	BOOL	RT	FALSE	Hours - maintenance work SW4 compl.
Std_SW5_Ok	NL	BOOL	RT	FALSE	Hours - maintenance work SW5 compl.
SSp_Wert_H	NL	INT	RT	0	Switching play value high
SSp_Wert_L	NL	INT	RT	0	Switching play value low
SSp_Wert	NL	DINT	RT, LOG	L#0	Switching play double integer
Std_Wert_H	NL	INT	RT	0	Hours value in seconds high
Std_Wert_L	NL	INT	RT	0	Hours value in seconds low
Std_Wert	NL	DINT	RT, LOG	L#0	Hours value in seconds double integer
StartTime	HI	DINT	RT	L#0	Turn on time double integer

PCU version V7 parameter set (max. 1024 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
STAT	HI	INT	RT, LOG	0	ACK state
StartTime	HI	DINT	RT	L#0	Turn on time double integer
MaintSwCy-cles1Ack	NL	BOOL	RT	FALSE	Switching play maintenance work SW1 compl.
MaintSwCy-cles2Ack	NL	BOOL	RT	FALSE	Switching play maintenance work SW2 compl.
MaintSwCy-cles3Ack	NL	BOOL	RT	FALSE	Switching play maintenance work SW3 compl.
MaintSwCy-cles4Ack	NL	BOOL	RT	FALSE	Switching play maintenance work SW4 compl.
MaintSwCy-cles5Ack	NL	BOOL	RT	FALSE	Switching play maintenance work SW5 compl.
MaintO-pHours1Ack	NL	BOOL	RT	FALSE	Hours - maintenance work SW1 compl.
MaintO-pHours2Ack	NL	BOOL	RT	FALSE	Hours - maintenance work SW2 compl.
MaintO-pHours3Ack	NL	BOOL	RT	FALSE	Hours - maintenance work SW3 compl.
MaintO-pHours4Ack	NL	BOOL	RT	FALSE	Hours - maintenance work SW4 compl.
MaintO-pHours5Ack	NL	BOOL	RT	FALSE	Hours - maintenance work SW5 compl.
SwCyclesAV	NL	DINT	RT, LOG	L#0	Switching play double integer
OpHoursAV	NL	DINT	RT, LOG	L#0	Hours value in seconds double integer

5.7.15 MAINT_USR - Maintenance data user

Basics

The switching cycles and operating hours for user aggregates can be recorded with this function. 1023/1024 instances of the function are available.

A maximum of 5 setpoints can be entered in each case for this. If these setpoints are exceeded, this is indicated here only via the maintenance data application.

Maint_USR manages the actual value of the switching cycle counter operating hours counters as well as the acknowledgements when the configured setpoints are exceeded.

Note

Not for PCU type S7-1500!

The MAINT_USR parameter assignment is not available for PCU type S7-1500!

Parameter sets for block MAINT_USR / DB 684

PCU parameter set before Version V7 (max. 1023 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
SSp_SW1_Ok	NL	BOOL	RT	FALSE	Switching play maintenance work SW1 compl.
SSp_SW2_Ok	NL	BOOL	RT	FALSE	Switching play maintenance work SW2 compl.
SSp_SW3_Ok	NL	BOOL	RT	FALSE	Switching play maintenance work SW3 compl.
SSp_SW4_Ok	NL	BOOL	RT	FALSE	Switching play maintenance work SW4 compl.
SSp_SW5_Ok	NL	BOOL	RT	FALSE	Switching play maintenance work SW5 compl.
Std_SW1_Ok	NL	BOOL	RT	FALSE	Hours - maintenance work SW1 compl.
Std_SW2_Ok	NL	BOOL	RT	FALSE	Hours - maintenance work SW2 compl.
Std_SW3_Ok	NL	BOOL	RT	FALSE	Hours - maintenance work SW3 compl.
Std_SW4_Ok	NL	BOOL	RT	FALSE	Hours - maintenance work SW4 compl.
Std_SW5_Ok	NL	BOOL	RT	FALSE	Hours - maintenance work SW5 compl.
SSp_Wert_H	NL	INT	RT	0	Switching play value high
SSp_Wert_L	NL	INT	RT	0	Switching play value low
SSp_Wert	NL	DINT	RT	L#0	Switching play double integer
Std_Wert_H	NL	INT	RT	0	Hours value in seconds high
Std_Wert_L	NL	INT	RT	0	Hours value in seconds low
Std_Wert	NL	DINT	RT	L#0	Hours value in seconds double integer
StartTime	HI	DINT	RT	L#0	Turn on time double integer

PCU version V7 parameter set (max. 1024 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
STAT	HI	INT	RT	0	ACK state
StartTime	HI	DINT	RT	L#0	Turn on time double integer
MaintSwCy-cles1Ack	NL	BOOL	RT	FALSE	Switching play maintenance work SW1 compl.
MaintSwCy-cles2Ack	NL	BOOL	RT	FALSE	Switching play maintenance work SW2 compl.
MaintSwCy-cles3Ack	NL	BOOL	RT	FALSE	Switching play maintenance work SW3 compl.
MaintSwCy-cles4Ack	NL	BOOL	RT	FALSE	Switching play maintenance work SW4 compl.
MaintSwCy-cles5Ack	NL	BOOL	RT	FALSE	Switching play maintenance work SW5 compl.
MaintO-pHours1Ack	NL	BOOL	RT	FALSE	Hours - maintenance work SW1 compl.
MaintO-pHours2Ack	NL	BOOL	RT	FALSE	Hours - maintenance work SW2 compl.

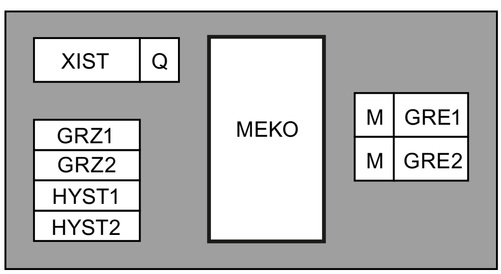
Name	Access	D.Type	A.Type	Value	Comment
MaintO-pHours3Ack	NL	BOOL	RT	FALSE	Hours - maintenance work SW3 compl.
MaintO-pHours4Ack	NL	BOOL	RT	FALSE	Hours - maintenance work SW4 compl.
MaintO-pHours5Ack	NL	BOOL	RT	FALSE	Hours - maintenance work SW5 compl.
SwCyclesAV	NL	DINT	RT	L#0	Switching play double integer
OpHoursAV	NL	DINT	RT	L#0	Hours value in seconds double integer

5.7.16 MVC - Measured Value Control

Basics

The MVC block controls up to 64/128 analog values for limit violations.

The measured value to be checked (`XIST/ActualValueRef`) is taken from another block (AIN, MULT, PID, POLY, SEQUENCE) via an interconnection.



For each measured value, 2 limits (`GRZ1/Limit1`, `GRZ2/Limit2`) are monitored with a different hysteresis band (`HYST1/Hysteresis1`, `HYST2/Hysteresis2`). The hysteresis band can be either above or below the limit in question.

MVC sets or deletes the corresponding event bits (`GRE1/Limit1Hit`, `GRE2/Limit2Hit`) for the measured value to be monitored.

A parameter set is assigned to each measured value to be checked.

Parameter sets for block MVC / DB 728

PCU parameter set before Version V7 (max. 64 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
GRZ1	NL	INT	ENG	0	Limit value 1
HYST1	NL	INT	ENG	0	Hysteresis for limit 1
HYS1	NL	BOOL	ENG	FALSE	Hysteresis band 1: 0/1 = lower/upper
GRZ2	NL	INT	ENG	0	Limit value 2
HYST2	NL	INT	ENG	0	Hysteresis for limit 2
HYS2	NL	BOOL	ENG	FALSE	Hysteresis band 2: 0/1 = lower/upper

Name	Access	D.Type	A.Type	Value	Comment
XIST	NL	QUEL	ENG	ANA,1,ANA	Address of actual value
TEILANL	HI	USINT	ENG	0	Assigned unit
GRE1	NL	BOOL	RT	TRUE	Limit infringement 1: 0/1 = no/yes
GRE2	NL	BOOL	RT	TRUE	Limit infringement 2: 0/1 = no/yes
X	NL	INT	RT	0	Actual value

PCU version V7 parameter set (max. 128 per PCU)

Name	Access*	D.Type	A.Type	Value	Comment
Sequence	HI	ENUM	ENG	<0>	Assigned unit
ActualVa-lueRef	NL	GREF	ENG	NULL	Reference to actual value
S7-1500: ActualVa-lueRef_ST	HR	WORD	SYS	W#16#0 000	Signal status
ActualVa-luePv	NL	REAL	RT	0.000	Process value actual value
Limit1	NL	REAL	ENG	0.000	Limit value 1
Hysteresis1	NL	REAL	ENG	0.000	Hysteresis for limit 1
Hyst1Above	NL	BOOL	ENG	FALSE	Hysteresis band 1: 0=below, 1=above
Limit2	NL	REAL	ENG	0.000	Limit value 2
Hysteresis2	NL	REAL	ENG	0.000	Hysteresis for limit 2
Hyst2Above	NL	BOOL	ENG	FALSE	Hysteresis band 2: 0=below, 1=above
Limit1Hit	RD	BOOL	RT	FALSE	Outside limit 1
Limit2Hit	RD	BOOL	RT	FALSE	Outside limit 2

User interface to MVC block

Event bit assignment - MVC 1..64

No.	GRE1 / Lim-it1Hit	GRE2 / Lim-it2Hit		No.	GRE1 / Lim-it1Hit	GRE2 / Lim-it2Hit
1	M 856.0	M 872.0		33	M 860.0	M 876.0
2	M 856.1	M 872.1		34	M 860.1	M 876.1
3	M 856.2	M 872.2		35	M 860.2	M 876.2
4	M 856.3	M 872.3		36	M 860.3	M 876.3
5	M 856.4	M 872.4		37	M 860.4	M 876.4
6	M 856.5	M 872.5		38	M 860.5	M 876.5
7	M 856.6	M 872.6		39	M 860.6	M 876.6
8	M 856.7	M 872.7		40	M 860.7	M 876.7
9	M 857.0	M 873.0		41	M 861.0	M 877.0
10	M 857.1	M 873.1		42	M 861.1	M 877.1
11	M 857.2	M 873.2		43	M 861.2	M 877.2
12	M 857.3	M 873.3		44	M 861.3	M 877.3
13	M 857.4	M 873.4		45	M 861.4	M 877.4

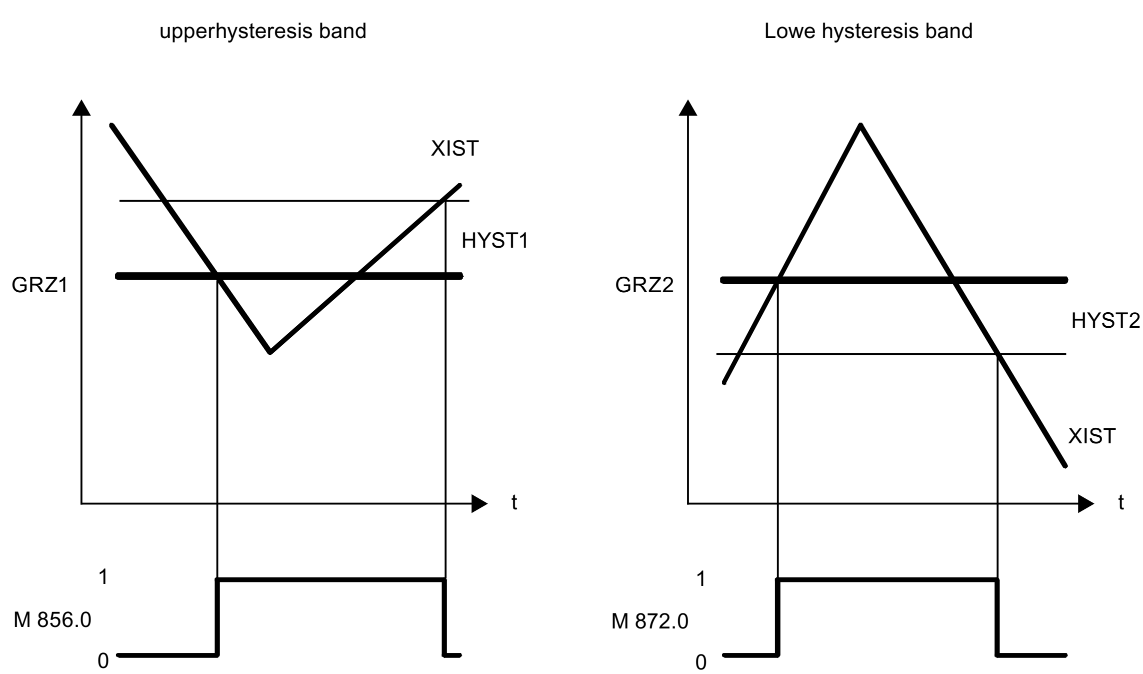
No.	GRE1 / Limit1Hit	GRE2 / Limit2Hit		No.	GRE1 / Limit1Hit	GRE2 / Limit2Hit
14	M 857.5	M 873.5		46	M 861.5	M 877.5
15	M 857.6	M 873.6		47	M 861.6	M 877.6
16	M 857.7	M 873.7		48	M 861.7	M 877.7
17	M 858.0	M 874.0		49	M 862.0	M 878.0
18	M 858.1	M 874.1		50	M 862.1	M 878.1
19	M 858.2	M 874.2		51	M 862.2	M 878.2
20	M 858.3	M 874.3		52	M 862.3	M 878.3
21	M 858.4	M 874.4		53	M 862.4	M 878.4
22	M 858.5	M 874.5		54	M 862.5	M 878.5
23	M 858.6	M 874.6		55	M 862.6	M 878.6
24	M 858.7	M 874.7		56	M 862.7	M 878.7
25	M 859.0	M 875.0		57	M 863.0	M 879.0
26	M 859.1	M 875.1		58	M 863.1	M 879.1
27	M 859.2	M 875.2		59	M 863.2	M 879.2
28	M 859.3	M 875.3		60	M 863.3	M 879.3
29	M 859.4	M 875.4		61	M 863.4	M 879.4
30	M 859.5	M 875.5		62	M 863.5	M 879.5
31	M 859.6	M 875.6		63	M 863.6	M 879.6
32	M 859.7	M 875.7		64	M 863.7	M 879.7

Event bit assignment - MVC 65..128

No.	GRE1 / Limit1Hit	GRE2 / Limit2Hit		No.	GRE1 / Limit1Hit	GRE2 / Limit2Hit
65	M 864.0	M 880.0		97	M 868.0	M 884.0
66	M 864.1	M 880.1		98	M 868.1	M 884.1
67	M 864.2	M 880.2		99	M 868.2	M 884.2
68	M 864.3	M 880.3		100	M 868.3	M 884.3
69	M 864.4	M 880.4		101	M 868.4	M 884.4
70	M 864.5	M 880.5		102	M 868.5	M 884.5
71	M 864.6	M 880.6		103	M 868.6	M 884.6
72	M 864.7	M 880.7		104	M 868.7	M 884.7
73	M 865.0	M 881.0		105	M 869.0	M 885.0
74	M 865.1	M 881.1		106	M 869.1	M 885.1
75	M 865.2	M 881.2		107	M 869.2	M 885.2
76	M 865.3	M 881.3		108	M 869.3	M 885.3
77	M 865.4	M 881.4		109	M 869.4	M 885.4
78	M 865.5	M 881.5		110	M 869.5	M 885.5
79	M 865.6	M 881.6		111	M 869.6	M 885.6
80	M 865.7	M 881.7		112	M 869.7	M 885.7
81	M 866.0	M 882.0		113	M 870.0	M 886.0
82	M 866.1	M 882.1		114	M 870.1	M 886.1
83	M 866.2	M 882.2		115	M 870.2	M 886.2

No.	GRE1 / Limit1Hit	GRE2 / Limit2Hit		No.	GRE1 / Limit1Hit	GRE2 / Limit2Hit
84	M 866.3	M 882.3		116	M 870.3	M 886.3
85	M 866.4	M 882.4		117	M 870.4	M 886.4
86	M 866.5	M 882.5		118	M 870.5	M 886.5
87	M 866.6	M 882.6		119	M 870.6	M 886.6
88	M 866.7	M 882.7		120	M 870.7	M 886.7
89	M 867.0	M 883.0		121	M 871.0	M 887.0
90	M 867.1	M 883.1		122	M 871.1	M 887.1
91	M 867.2	M 883.2		123	M 871.2	M 887.2
92	M 867.3	M 883.3		124	M 871.3	M 887.3
93	M 867.4	M 883.4		125	M 871.4	M 887.4
94	M 867.5	M 883.5		126	M 871.5	M 887.5
95	M 867.6	M 883.6		127	M 871.6	M 887.6
96	M 867.7	M 883.7		128	M 871.7	M 887.7

Hysteresis bands



5.7.17 MSG - Message block

Basics

- The MSG block manages a maximum of 1024/2048 PCU-specific messages
- When they are enabled, messages are sent to the IOS stations enabled for the messages (see parameter assignment block FIFO, telegram type 3). These messages are displayed in the PCU server application window and entered into the message archive.

The message block allows the following message definitions:

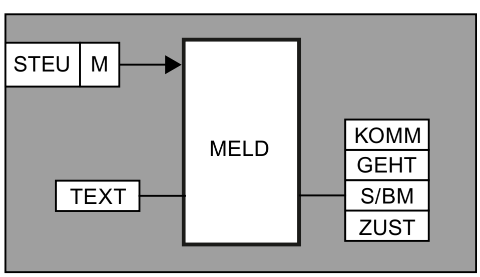
EnableW/Op (<V7) IsOpMsgNotWrn (V7)	S/BM (<V7) IsAlarmMsg (V7)	Message type	Description
0	0	M	Operation message
0	1	F	Fault
1	0	B	Operator message
1	1	W	Warning

For more information on the message configuration, refer to the section Message system (Page 1009).

If the message is configured as a fault message and the horn `FrgHupe/EnableSignalHorn`= 1 is enabled, the group flag HornErrorMsg (M 99.7) is set.

Message output is initiated by a signal change at the trigger flags:

- M 888.0 - M 951.7 for messages 1 to 512
- M 1016.0 - M1079.7 for messages 513 to 1024



Parameter sets for block MSG / DB 733

PCU parameter set before Version V7 (max. 1024 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
TEILANL	NL	USINT	ENG	0	Assigned unit
KOMM	NL	BOOL	ENG	TRUE	Enable coming message:0/1 = no/yes
GEHT	NL	BOOL	ENG	TRUE	Enable going message: 0/1 = no/yes
S/BM	NL	BOOL	ENG	FALSE	0/1 = operating message/error message
FrgHupe	NL	BOOL	ENG	TRUE	Enable horn

Name	Access	D.Type	A.Type	Value	Comment
ZUST	RD	BOOL	RT	FALSE	Message status
EnableW/Op	NL	BOOL	ENG	FALSE	0:S/BM=process/error msg. 1:S/ BM=operator msg./warning

Parameter set PCU Version V7 (max. 2048)

Name	Access	D.Type	A.Type	Value	Comment
Sequence	NL	ENUM	ENG	<0>	Assigned unit
EnableCo-mingMsg	NL	BOOL	ENG	TRUE	Enable coming message:0/1 = no/yes
Enable-GoingMsg	NL	BOOL	ENG	TRUE	Enable going message: 0/1 = no/yes
IsAlarmMsg	NL	BOOL	ENG	FALSE	0/1 = operating message/error mes-sage
EnableSignal-Horn	NL	BOOL	ENG	TRUE	Enable horn
MsgSignal	RD	BOOL	RT	FALSE	Message status
IsO-pMsgNotWrn	NL	BOOL	ENG	FALSE	0:S/BM=process/error msg. 1:S/ BM=operator msg./warning

Example:

Connection of input I3.7 to message 1 (= M 888.0)

A	I3.7	Alarm signal for message
=	M 888.0	Trigger flag

Example:

Connection of input I4.1 to message 1024

A	I4.1	Alarm signal for message
=	M 1079.7	Trigger flag

Parameterization of message texts in IOS

<proj-path>\pcu.nnn\texte.x\MELDKOM.txt			Sets: max. 1024/ 2048 per PCU
No.	**Type**	**Preset**	**Comment**
1	Z48	MESSAGE 01 INCOMING	Message text for incoming message
...	Z48	...	

<proj-path>\pcu.nnn\texte.x\MELDGEH.txt			Sets: max. 1024/ 2048 per PCU
No.	**Type**	**Preset**	**Comment**
1	Z48	MESSAGE 01 OUTGO-ING	Message text for outgoing message
...	Z48	...	

- For each message, 48 characters of configurable text are displayed for incoming and outgoing messages.
- Each message can be set as a fault or operation message with the parameter bit `S/BM/IsAlarmMsg`.

Issuing incoming and outgoing messages:

- Each message direction can be released (= 1) or blocked (= 0) with the assigned parameter bit `KOMM/EnableComingMsg` and `GEHT/EnableGoingMsg`.
- For each fault message, the horn trigger can be enabled or disabled by the parameter bit `FrgHupe/EnableSignalHorn`.

Process interface to MSG block

Assignment of message block to data bit

DB615 MSG_M

Data byte	Assignment DBB x.0 ... x.7 to message number
DBB 10	MSG 1 ... 8
DBB 11	MSG 9 ... 16
...	...
DBB 137	MSG 1017 ... 1024
DBB 138	MSG 1025 ... 1032 *)
...	... *)
DBB 265	MSG 2041 ... 2048 *)

*) applies to PCU V7

User interface to MSG block

The flag interface (trigger flag) is available in both PCU versions for messages 1 ... 1024 only. With PCU V7 , the message trigger for the MSG no. 1025 - 2048 should only be via the process interface (see above).

- The assignment of the flag bit starts with the flag "`M 888.0`" (MSG 1...512) or "`M 1016.0`" (MSG 513...1024).
- The flag "M <Flag byte>.<Flag bit>" for a particular message instance can be determined with the Windows computer (view "Scientific") according to the following method:
  - MSG 1...512: <Flag byte> = `Int` (MSG instance 1 `/` 8) + 888
  - MSG 513...1024: <Flag byte> = `Int` (MSG instance 513 `/` 8) + 1016
  - <Flag bit > = (MSG instance 1) `Mod` 8
- The assignment of the flag bit to the message instance can also be taken from the Step 7 symbol table.

Interface to MSG

HornErrorMsg	Horn MSG error	M 99.7
HornGrpError	Group flag hooter (ICM, SEQU, AIN, MSG)	M 107.1

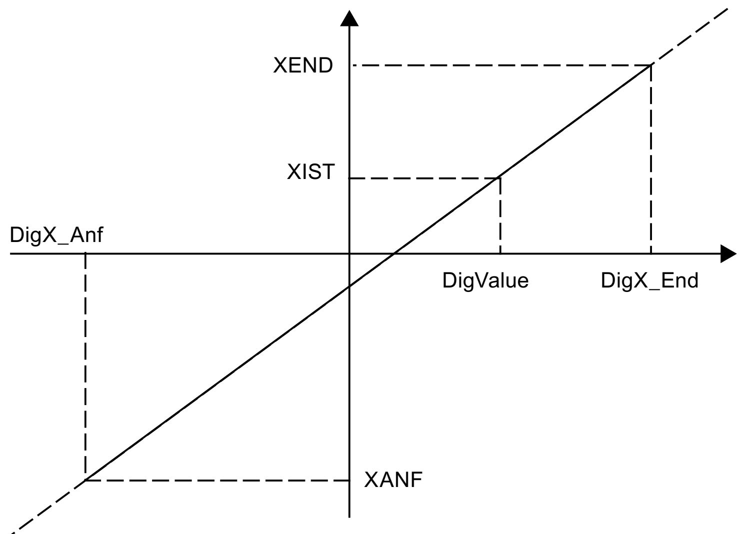
The hooter flag is to be processed and reset by the user.

5.7.18 AIN - Measured value recording

Basics

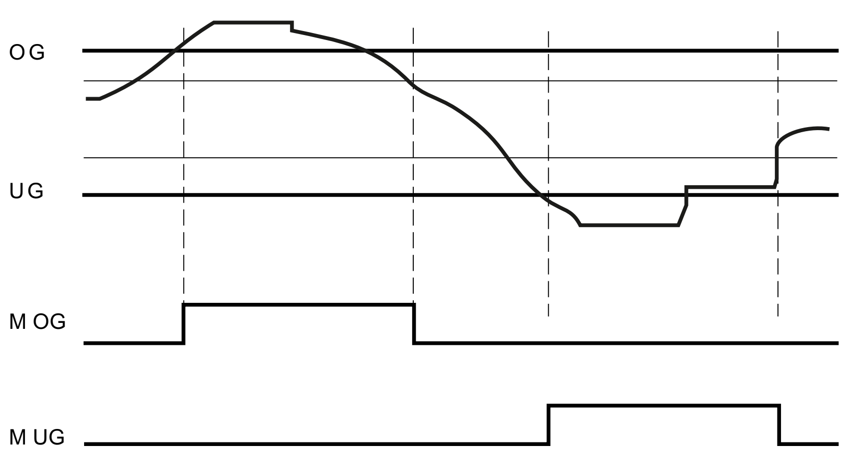
- The AIN block registers and processes up to 256 / 512 analog values per PCU.
- The physical and the digital start and range can be set.
- The acquired analog value is converted into digital units within the set scaling (`XANF/ProcValStart, XEND/ProcValEnd`) according to the digital limits (`DigX_ANF/DigValStart, DigX_END/DigValEnd`).
- The preset of digital range limits has the advantage of adapting several analog input e.g. SIMATIC S7, and other vendors.
- If an error occurs, the analog value is marked as being disturbed (`STOE/ValueGrpError`) and depending on (`STWE/OnErrUseValEnd`) the upper or lower digital limit value will be entered as the substitute value. In addition to this, a message to this effect is entered in the message archive if this is enabled (`MLDG_SPERR/MsgLock`). In addition, the hooter flag (M 99.6) and the group fault flag (M 107.1) are set.

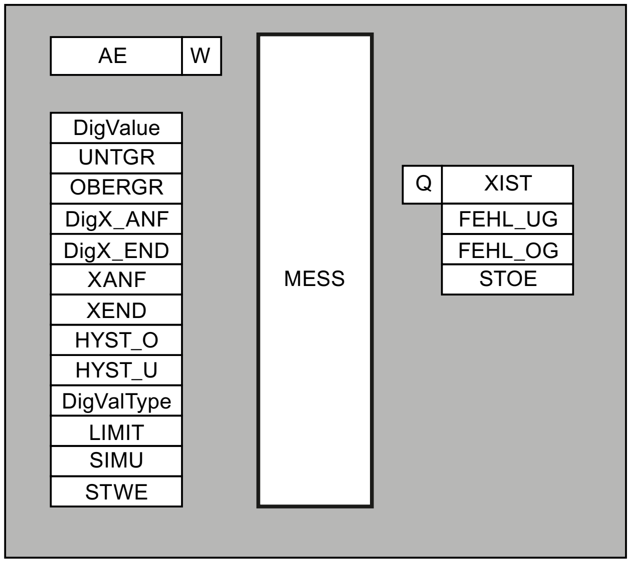
Explanation of a chart:



- Additionally, it is possible to define a lower (`UNTGR/LowerLimit`), upper limit (`OBERGR/UpperLimit`) with different levels of hysteresis. These limit values are checked analogously to the function block MVC, the limit breach is displayed in the corresponding flag.
- For the upper limit, the Hysteresis range is below the trigger value, for the lower limit, the Hysteresis range is above it. In the parameter assignment, a message for limit violation (`FREI_FUG/EnableMsgErrLL`, `FREI_FOG/EnableMsgErrUL`) can be enabled.
- If the simulation bit (`SIMU/SimOn`) is set, there is no access to the module and no adaptation.
- Processing of the measured value is done only if the digital old value and the input digital new value are different. This has to be taken into account as long as no interface modules are connected.

Function of limit value bits





A parameter set is assigned to each analog value.

Parameter sets for block AIN / DB 727

PCU parameter set before Version V7 (max. 256 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
XIST	NL	INT	RT, LOG	1	Physical actual value
DigX_ANF	NL	INT	ENG	0	Digital start value
DigX_END	NL	INT	ENG	27648	Digital end value
XANF	NL	INT	ENG, LOG	0	Start value
XEND	NL	INT	ENG, LOG	1200	End value
UNTGR	NL	INT	ENG, LOG	0	Lower limit
HYST_U	NL	INT	ENG, LOG	0	Hysteresis lower limit
OBERGR	NL	INT	ENG, LOG	0	Upper limit
HYST_O	NL	INT	ENG, LOG	0	Hysteresis upper limit
TEILANL	NL	USINT	ENG	0	Assigned unit
DigValType	NL	USINT	ENG	0	AI format: 0=S7,5=S5 two.compl.,6=S5 VZ+Betr
STWE	NL	BOOL	ENG	FALSE	XIST on error: 0/1 = XANF/ END
SIMU	NL	BOOL	ENG	FALSE	Simulation: 0/0.1 = off/on
STOE	NL	BOOL	RT	FALSE	Fault: 0/1 = No/Yes
FEHL_UG	NL	BOOL	RT	TRUE	Error UG
FEHL_OG	NL	BOOL	RT	TRUE	Error OG
FREI_FUG	NL	BOOL	ENG	FALSE	Enable error output lower limit
FREI_FOG	NL	BOOL	ENG	FALSE	Enable error output upper limit

Name	Access	D.Type	A.Type	Value	Comment
MLDG_SPERR	NL	BOOL	ENG	FALSE	No fault message at under/overflow
LIMIT	NL	BOOL	ENG	FALSE	Limitation XANF<=XIST<=XEND
S5Live	NL	BOOL	ENG	FALSE	S5 type:Open-circuit monitoring < 2.9 mA
Overflow	HI	BOOL	RT	FALSE	Overflow bit
Underflow	HI	BOOL	RT	FALSE	Underflow bit
Error	HI	BOOL	RT	FALSE	Error bit:Underflow or overflow
DigValPEW	HI	INT	RT	0	Hardware dependent dig. AI value
DigValue	HI	INT	RT	0	Hardware independent dig. AI value
DigValOld	HI	INT	RT	0	Digital AI old value
Status	HR	INT	RT, LOG	96	Status as word

PCU Version V7 parameter set (max. 512 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Sequence	NL	ENUM	ENG, LOG	<0>	Assigned unit
DigValRef	NL	GREF	ENG	PEW 512	Reference to measured value
S7-1500: DigValRef_ST	HR	WORD	SYS	W#16#0000	Signal status
DigValRaw	HR	DWORD	RT	DW#16 #00000000	Measured value (raw format)
DigValStart	NL	INT	ENG	0	Digital start value
DigValEnd	NL	INT	ENG	27648	Digital end value
S7-400: ConvFunc-No	NL	INT	ENG	0	Conversion FB (-1=raw value, 0=S7-AI)
S7-1500: Conversion	NL	ENUM	ENG	0	Measured value conversion (0=Raw value, 1=S7-AI, 2=User)
ProcValNorm	HR	REAL	RT	0.000	Standardized process value
ProcValStart	NL	REAL	ENG, LOG	0.000	Process value start
ProcValEnd	NL	REAL	ENG, LOG	120.000	Process value end
ProcValAct	NL	REAL	RT, LOG	0.000	Current process value
ProcValDecPoints	HI	USINT	ENG	0	Number of decimal places for INT/DINT conversion
ProcValActInt	HR	INT	RT	0	Current process value (INT copy)
ProcValActDint	HR	DINT	RT	L#0	Current process value (DINT copy)
UpperLimit	NL	REAL	ENG, LOG	0.000	Upper limit

Name	Access	D.Type	A.Type	Value	Comment
UpperHystValue	NL	REAL	ENG, LOG	0.000	Hysteresis upper limit
UpperHystOpBelow	NL	BOOL	ENG	TRUE	Hysteresis upper limit, effective range below
LowerLimit	NL	REAL	ENG, LOG	0.000	Lower limit
LowerLimitHystValue	NL	REAL	ENG, LOG	0.000	Hysteresis lower limit
LowerLimitOpAbove	NL	BOOL	ENG	TRUE	Hysteresis upper limit, effective range above
LimitErrDelayTm	NL	INT	ENG	0	Delay time error message (start value) [sec]
LimitErrAv	HR	INT	RT	0	Delay time error message (start value) [sec]
SimOn	NL	BOOL	ENG	FALSE	Simulation
OnValErrUseValEnd	NL	BOOL	ENG	FALSE	In case of fault set actual value to: 0=Start value, 1=End value
EnableMsgErrUL	NL	BOOL	ENG	FALSE	Enable error output upper limit
EnableMsgErrLL	NL	BOOL	ENG	FALSE	Enable error output lower limit
MsgLock	NL	BOOL	ENG	FALSE	No fault message at under/overflow
LimitProcVal	NL	BOOL	ENG	FALSE	Limit to start and end value
SmoothProcVal	NL	BOOL	ENG	FALSE	Smooth measured values
ResetSmoothPVs	NL	BOOL	ENG	TRUE	Reset measured values buffer
RefInvalid	HR	BOOL	RT	FALSE	Reference is invalid
NormError	HR	BOOL	RT	FALSE	Normalization error
BadValue	HR	BOOL	RT	FALSE	Group error: Overflow or underflow
WireBreak	HR	BOOL	RT	FALSE	Wire break
ValueGrpError	RD	BOOL	RT	FALSE	Measured value error
ErrUpperLimit	RD	BOOL	RT	FALSE	Error level upper limit
ErrLowerLimit	RD	BOOL	RT	TRUE	Error level lower limit
NoteIsActive	HR	BOOL	RT	FALSE	Note is available
Overflow	HR	BOOL	RT	FALSE	Overflow bit
Underflow	HR	BOOL	RT	FALSE	Underflow bit
Status	HR	DWORD	RT, LOG	DW#16#0C101000	(Internal)

Note

Features of V7-PCU

- It is no longer necessary to configure AIN_PW, the hardware-dependent raw value is input via a reference. The requirements for the memory width (PIB, PIW, PID) for the respective connected peripheral are set in the data source dialog of the reference.
- The conversion of hardware-dependent raw value "`DigValRaw`" to process value "`ProcValAct`" is implemented using parameter `"ConvFuncNo"`.
- The measured value can be smoothed with the "`SmoothProcVal`" parameter: To this end, the last ten measured values are calculated internally. The measured value memory can be reset for smoothing with "`ResetSmoothPVs`". The measured value memory spaces are then repopulated with the current process value.
- The BOOL attribute "`Commissioning`" has been reintroduced to the global data. With the AIN block, the bit means that the references to the measured value (`DigValRef`) are not resolved for all instances, analogous to the "local" `SimOn` attribute.
- The attribute `"ValueGrpError"` consists (OR function) of the individual attributes "`RefInvalid`", "`NormError`", "`BadValue`" and "`WireBreak`".

Attribute	Value	Description
S7-400: `ConvFuncNo`	-1	The process value (REAL) is copied from the source unmanipulated (note memory width DWORD)
	0	The process value is converted internally for an S7 analog input
	[FB]	The process value is manipulated by the user FB specified here before the input. The `FB3000 („BmUsrConvValInFB“)` can be used as a template for this.
S7-1500: `Conversion`	0	The process value (REAL) is copied from the source unmanipulated (note memory width DWORD)
	1	The process value is converted internally for an S7 analog input
	2	The process value is processed by user block `BmUserCallIfAinConvValueFC (FC732)` before the input. This block is located in the Step 7 (TIA Portal) project folder ".../Runtime-Interfaces" and serves as a template for the process value conversions to be included by the user.
`RefInvalid`	0 / 1	The reference to the process input value is invalid.
`NormError`	0 / 1	Measured value normalization failed (unable to derive a usable real value from the raw value). This attribute can be set by the user conversion block (S7-400: FB3000, S7-1500: FC732). For the standard conversion (S7 I/O or unmanipulated), it is always = 0
`BadValue`	0 / 1	= `Overflow` OR `Underflow`
`WireBreak`	0 / 1	Cable break This attribute can be set by the user conversion block (S7-400: FB3000, S7-1500: FC732). For the standard conversion (S7 I/O or unmanipulated), it is always = 0

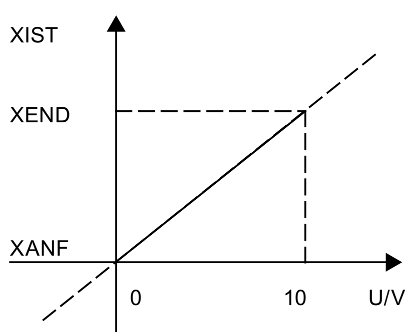
Attribute	Value	Description
ValueGrpError	0 / 1	= NormError OR BadValue OR RefInvalid OR WireBreak
ProcValDecPoints	0...9	The integer copies ProcValActInt, ProcValActDint are generated by multiplying measured value ProcValAct by this factor.

Examples:

Example 1:

Use of AIN with Analog Input Module (0..10 V)

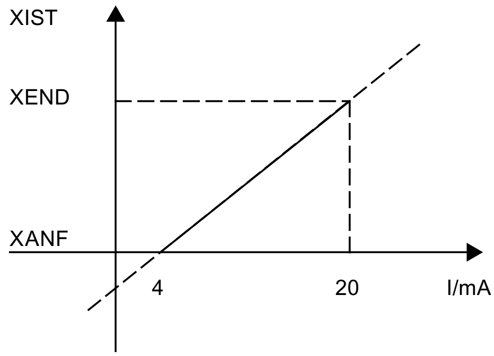
Presets/Adjustments	PCU < V7; S7-AE module
Parameterization hardware	Change of hardware configuration in S7 project: Type of output: A Output range: 0..10V
Settings in the parameterization	DigX_ANF = 0 (default) DigX_END = 27648 (default) XANF = 0 (0.0%) (default) XEND = 1000 (100.0%) DigValType = 0 (S7-AI)



Example 2:

Use of AIN with Analog Input Module (4..20mA)

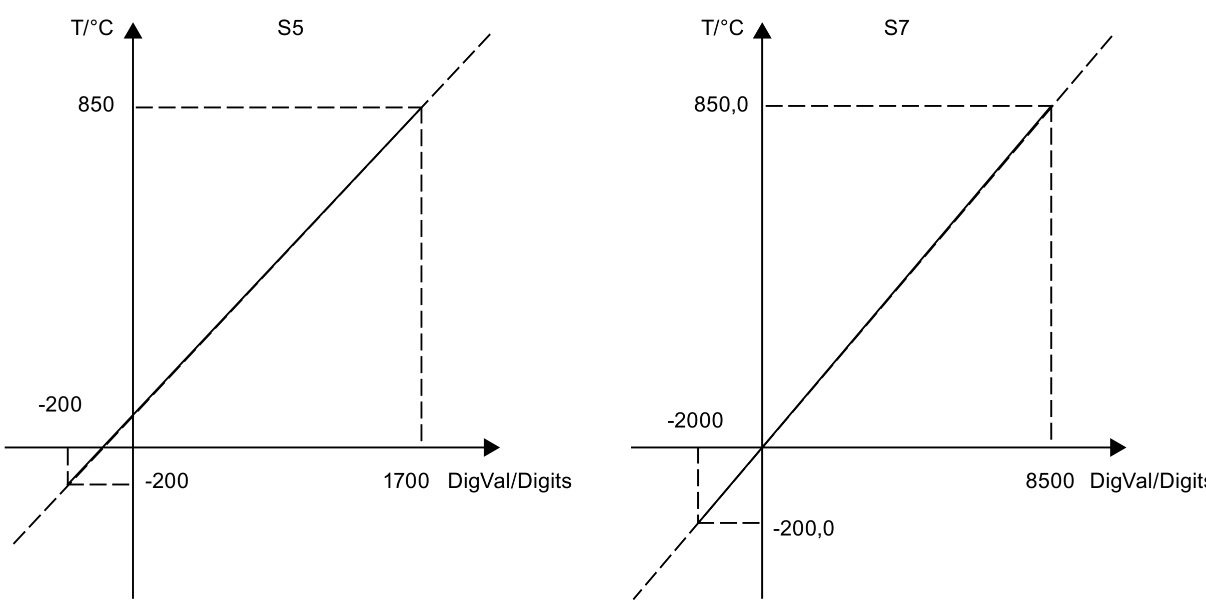
Presets/Adjustments	PCU V7; S7-AE module
Parameterization hardware	Change of hardware configuration in S7 project: Type of output: I Output range: 4..20mA
Settings in the parameterization	DigValStart = 0 (default) DigValEnd = 27648 (default) ProcValStart = 0 (0.0%) (default) ProcValEnd = 1000 (100.0%) ConvFuncNo = 0 (S7-AI)



Example 3:

Use of AIN with a PT 100

Presets/Adjustments	PCU V7
Parameterization hardware	Change of hardware configuration in S7 project: Selection of analog module Adjustment of required values
Settings in the parameterization	`DigValStart` = -2000 `DigValEnd` = 8500 `ProcValStart` = -200.0 (0.0%) `ProcValEnd` = 850.0 (100.0%) `ConvFuncNo` = 0 (S7-AI)



AIN interface		
HornErrorAIN	Hooter flag	M 99.6
HornGrpError	Group fault flag (ICM, SEQUENCES, AIN, MSG)	M 107.1

The hooter flag is to be processed and reset by the user.

Assignment limit value bits - AIN

Table 5-4 AIN 1 ... 256

Bit address								Lower limit	Upper limit
0	1	2	3	4	5	6	7	(MB)	(MB)
1	2	3	4	5	6	7	8	1144	1176
9	10	11	12	13	14	15	16	1145	1177
17	18	19	20	21	22	23	24	1146	1178
25	26	27	28	29	30	31	32	1147	1179
33	34	35	36	37	38	39	40	1148	1180
41	42	43	44	45	46	47	48	1149	1181
49	50	51	52	53	54	55	56	1150	1182
57	58	59	60	61	62	63	64	1151	1183
65	66	67	68	69	70	71	72	1152	1184
73	74	75	76	77	78	79	80	1153	1185
81	82	83	84	85	86	87	88	1154	1186
89	90	91	92	93	94	95	96	1155	1187
97	98	99	100	101	102	103	104	1156	1388
105	106	107	108	109	110	111	112	1157	1189
113	114	115	116	117	118	119	120	1158	1190
121	122	123	124	125	126	127	128	1159	1191
129	130	131	132	133	134	135	136	1160	1192
137	138	139	140	141	142	143	144	1161	1193
145	146	147	148	149	150	151	152	1162	1194
153	154	155	156	157	158	159	160	1163	1195
161	162	163	164	165	166	167	168	1164	1196
169	170	171	172	173	174	175	176	1165	1197
177	178	179	180	181	182	183	184	1166	1198
185	186	187	188	189	190	191	192	1167	1199
193	194	195	196	197	198	199	200	1168	1200
201	202	203	204	205	206	207	208	1169	1201
209	210	211	212	213	214	215	216	1170	1202
217	218	219	220	221	222	223	224	1171	1203
225	226	227	228	229	230	231	232	1172	1204

Bit address								Lower limit	Upper limit
233	234	235	236	237	238	239	240	1173	1205
241	242	243	244	245	246	247	248	1174	1206
249	250	251	252	253	254	255	256	1175	1207

Table 5-5 AIN 257 ... 512 (as of PCU V7)

Bit address								Lower limit	Upper limit
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**(MB)**	**(MB)**
257	258	259	260	261	262	263	264	1624	1656
265	266	267	268	269	270	271	272	1625	1657
273	274	275	276	277	278	279	280	1626	1658
281	282	283	284	285	286	287	288	1627	1659
289	290	291	292	293	294	295	296	1628	1660
297	298	299	300	301	302	303	304	1629	1661
305	306	307	308	309	310	311	312	1630	1662
313	314	315	316	317	318	319	320	1631	1663
321	322	323	324	325	326	327	328	1632	1664
329	330	331	332	333	334	335	336	1633	1665
337	338	339	340	341	342	343	344	1634	1666
345	346	347	348	349	350	351	352	1635	1667
353	354	355	356	357	358	359	360	1636	1668
361	362	363	364	365	366	367	368	1637	1669
369	370	371	372	373	374	375	376	1638	1670
377	378	379	380	381	382	383	384	1639	1671
385	386	387	388	389	390	391	392	1640	1672
393	394	395	396	397	398	399	400	1641	1673
401	402	403	404	405	406	407	408	1642	1674
409	410	411	412	413	414	415	416	1643	1675
417	418	419	420	421	422	423	424	1644	1676
425	426	426	428	429	430	431	432	1645	1677
433	434	435	436	437	438	439	440	1646	1678
441	442	443	444	445	446	447	448	1647	1679
449	450	451	452	453	454	455	456	1648	1680
457	458	459	460	461	462	463	464	1649	1681
465	466	467	468	469	470	471	472	1650	1682
473	474	475	476	477	478	479	480	1651	1683
481	482	483	484	485	486	487	488	1652	1684
489	490	491	492	493	494	495	496	1653	1685
497	498	499	500	501	502	503	504	1654	1686
505	506	507	508	509	510	511	512	1655	1687

5.7.19 AVA_PW

Basic principles

Only applies to PCU versions before V7!

- The peripheral addresses of AIN blocks does not have to be arranged in consecutive order.
- The system may be informed by the class with max. 25 ranges from which it reads out the rare values (starting with range 1).
- The peripheral start address and the number of rare values has to be defined per range.

Parameter sets for block AIN_PW / DB 727

Table 5-6 PCU parameter set

Name	D.Type	A.Type	Value	Comment
PW	INT	ENG	0	Start address PEW range (-1 = list end)
Num	INT	ENG	0	Range length in PEW (0 = list end)

The ranges have to be occupied starting with range 1,

which is preset in the delivery state with:

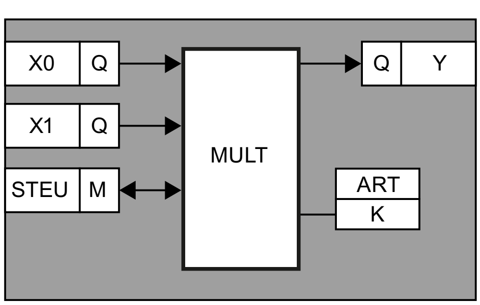
- PW = 512
- Num= 256
- i.e. all measured valued are read from PEW512.

If the system finds a range PW = -1 then the input of rare values is stopped as well as at Num <= 0.

5.7.20 MULT - Multifunction block

Basics

The multifunction block has 2 process value inputs and an output value. The process values can be linked to various math functions and compare functions. Each block occupies one flag bit, which can have various meanings depending on its function.



Parameter sets for block MULT / DB 732

The math functions work with the following number formats:

- < V7: Format INT
- V7: Real REAL

Division by 0 is prevented and the output value retains its previous value.

PCU parameter set before Version V7 (max. 64 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Y	NL	INT	RT	0	Output value
X0	NL	QUEL	ENG	ANA, 1,ANA	1st input value
X1	NL	QUEL	ENG	ANA, 1,ANA	2nd input value
ART	NL	INT	ENG	0	Function type: 0-8
K	NL	INT	ENG	0	Hysteresis
M	HI	BOOL	RT	FALSE	Control input

Description of the parameter "TYPE"

ART	Function	Flag bit	Query	Result
0	Select	Control input	M = 0	Then Y := X0 Else Y := X1
1	Min.	Result	X0 <= X1	Then Y := X0, M := 0 Else Y := X1, M := 1
2	Max.	Result	X0 >= X1	Then Y := X0, M := 0 Else Y := X1, M := 1
3	Add	irrelevant		Y := X0 + X1
4	Sub	irrelevant		Y := X0 - X1
5	Mul	irrelevant		Y := X0 * X1
6	Div	irrelevant		Y := X0 / X1
7	Limit+	Result	X0 > (X1 + K)	then M := 0 otherwise M := 1
8	Limit-	Result	X0 < (X1 - K)	then M := 0 otherwise M := 1

PCU version V7 parameter set (max. 128 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Function	NL	ENUM	ENG	Select	Function type
ProcVal1	NL	GREF	ENG	NULL	Process value 1
S7-1500: ProcVal1_ST	HR	WORD	SYS	W#16#0 000	Signal status
ProcVal2	NL	GREF	ENG	NULL	Process value 2

Name	Access	D.Type	A.Type	Value	Comment
S7-1500: ProcVal2_ST	HR	WORD	SYS	W#16#0000	Signal status
Hysteresis	NL	REAL	ENG	0.000	Hysteresis
Selector	RD	BOOL	RT	FALSE	Control flag: 0=ProcVal1, 1=ProcVal2
ResultFlag	RD	BOOL	RT	FALSE	Result bit
ResultValue	RD	REAL	RT	0.000	Output value

Description of the parameter "Function"

Function	Flag bit	Query	Result
Select	Control input	`Selector = 0`	then: `ResultValue:=ProcVal1` otherwise: `ResultValue:=ProcVal2`
Min	Result	`ProcVal1 <= ProcVal2`	then: `ResultValue:=ProcVal1, ResultFlag := 0` otherwise: `ResultValue:=ProcVal2, ResultFlag := 1`
Max	Result	`ProcVal1 >= ProcVal2`	then: `ResultValue:=ProcVal1, ResultFlag := 0` otherwise: `ResultValue:=ProcVal2, ResultFlag := 1`
Add	irrelevant		`ResultValue:= ProcVal1 + ProcVal2`
Sub	irrelevant		`ResultValue:= ProcVal1 - ProcVal2`
Mul	irrelevant		`ResultValue:= ProcVal1 * ProcVal2`
Div	irrelevant		`ResultValue:= ProcVal1 / ProcVal2`
Limit+	Result	`ProcVal1 > (ProcVal2 + Hysteresis)`	then: `ResultFlag := 0` otherwise: `ResultFlag := 1`
Limit-	Result	`ProcVal1 < (ProcVal2 - Hysteresis)`	then: `ResultFlag := 0` otherwise: `ResultFlag := 1`

Function	Flag bit	Query	Result

<	Result	`ProcVal1 < ProcVal2`	then:
			`ResultFlag := 1`
			otherwise:
			`ResultFlag := 0`

<=	Result	`ProcVal1 <= ProcVal2`	then:
			`ResultFlag := 1`
			otherwise:
			`ResultFlag := 0`

==	Result	`ProcVal1 = ProcVal2`	then:
			`ResultFlag := 1`
			otherwise:
			`ResultFlag := 0`

<>	Result	`ProcVal1 <> ProcVal2`	then:
			`ResultFlag := 1`
			otherwise:
			`ResultFlag := 0`

>=	Result	`ProcVal1 >= ProcVal2`	then:
			`ResultFlag := 1`
			otherwise:
			`ResultFlag := 0`

>	Result	`ProcVal1 > ProcVal2`	then:
			`ResultFlag := 1`
			otherwise:
			`ResultFlag := 0`

User interface: Assignment MULT to flag bit

No.	Flag	No.	Flag	No.	Flag	No.	Flag
1	M 952.0	33	M 956.0	65	M 960.0	97	M 964.0
2	M 952.1	34	M 956.1	66	M 960.1	98	M 964.1
3	M 952.2	35	M 956.2	67	M 960.2	99	M 964.2
4	M 952.3	36	M 956.3	68	M 960.3	100	M 964.3
5	M 952.4	37	M 956.4	69	M 960.4	101	M 964.4
6	M 952.5	38	M 956.5	70	M 960.5	102	M 964.5
7	M 952.6	39	M 956.6	71	M 960.6	103	M 964.6
8	M 952.7	40	M 956.7	72	M 960.7	104	M 964.7
9	M 953.0	41	M 957.0	73	M 961.0	105	M 965.0
10	M 953.1	42	M 957.1	74	M 961.1	106	M 965.1
11	M 953.2	43	M 957.2	75	M 961.2	107	M 965.2
12	M 953.3	44	M 957.3	76	M 961.3	108	M 965.3
13	M 953.4	45	M 957.4	77	M 961.4	109	M 965.4
14	M 953.5	46	M 957.5	78	M 961.5	110	M 965.5
15	M 953.6	47	M 957.6	79	M 961.6	111	M 965.6

No.	Flag	No.	Flag	No.	Flag	No.	Flag
16	M 953.7	48	M 957.7	80	M 961.7	112	M 965.7
17	M 954.0	49	M 958.0	81	M 962.0	113	M 966.0
18	M 954.1	50	M 958.1	82	M 962.1	114	M 966.1
19	M 954.2	51	M 958.2	83	M 962.2	115	M 966.2
20	M 954.3	52	M 958.3	84	M 962.3	116	M 966.3
21	M 954.4	53	M 958.4	85	M 962.4	117	M 966.4
22	M 954.5	54	M 958.5	86	M 962.5	118	M 966.5
23	M 954.6	55	M 958.6	87	M 962.6	119	M 966.6
24	M 954.7	56	M 958.7	88	M 962.7	120	M 966.7
25	M 955.0	57	M 959.0	89	M 963.0	121	M 967.0
26	M 955.1	58	M 959.1	90	M 963.1	122	M 967.1
27	M 955.2	59	M 959.2	91	M 963.2	123	M 967.2
28	M 955.3	60	M 959.3	92	M 963.3	124	M 967.3
29	M 955.4	61	M 959.4	93	M 963.4	125	M 967.4
30	M 955.5	62	M 959.5	94	M 963.5	126	M 967.5
31	M 955.6	63	M 959.6	95	M 963.6	127	M 967.6
32	M 955.7	64	M 959.7	96	M 963.7	128	M 967.7

5.7.21 PCU_GEN - PCU System data in general

Basics

Time master

The PCU can be operated as a time master. For this purpose, the telegram type 7 must be manually padded in FIFO. The telegram's refresh cycle is specified by the parameter `TSynSoll` (<=V6) / `TimeSyncSP` (V7).

Error masking for PCU type S7-400

Synchronization error events can be masked out with the parameters "`PrgFltSetMask` " and "`AccFltSetMask`".

"Synchronous error" means that the error occurs straight after the action that is responsible for triggering it and that it is handled by the PLC.

Examples:

- Programming error, Fault-LED "INTF" with program instruction:
  `TO SPECIAL_DB; // DB not loaded`
- Access error, Fault-LED "EXTF" with program instruction:
  `L PEW 128; // Component not inserted`

The S7 catches these errors and branches into the according error OB. If the OB does not exist, the CPU will enter STOP. If the according mask bit is set, this error is caught and no jump into the error OB takes place. Further, the according fault LED will not be activated.

Startup for PCU type S7-1500

Please note that when technical data blocks are restored using the "Block transfer" application, historical runtime data can also be loaded in the AS and become active and this may lead to an undesired system state. You can prevent this by loading the AS in Stop mode and having the AS runtime system make the appropriate initializations when the CPU is restarted.

For this application case, the following engineering attributes have been introduced in the data record of the PCU_ALG class for PCU type S7-1500:

Attribute	Application
OnRestartStopSequences = TRUE	All unit instances are put into step 0 and "Ready" state on CPU restart.
OnRestartResetICMs = TRUE	All ICM instances are initialized ("OFF" / "Manual") on CPU restart.
OnRestartResetBOS = TRUE	All CAS data records are initialized on CPU restart.

Parameter set of PCU version before V7 (DB 701)

Name	Access	D.Type	A.Type	Value	Comment
BoEsgManualOff	HI	BOOL	ENG	FALSE	ICM message Disable manual on/off
BoEsgManualAutoOff	HI	BOOL	ENG	FALSE	ICM message Disable manual/automatic
BoMsgEsgForcingEnableOff	HI	BOOL	ENG	FALSE	ICM msg. Disable ForcingEnable
BoMsgEsgSimFeedbackOff	HI	BOOL	ENG	FALSE	ICM msg. Disable SimulateFeedback
RepEOPFinalStates	HI	BOOL	ENG	FALSE	EOP final states messaging
RepEOPQuiesStates	HI	BOOL	ENG	FALSE	EOP quiet/intermediate states messaging
RepEOPTransStates	HI	BOOL	ENG	FALSE	EOP transfer states messaging
PCUno	NL	USINT	ENG	4	PCU number
h	HI	USINT	RT	17	Hour
min	HI	USINT	RT	8	Minute
sec	HI	USINT	RT	30	Second
KW	HI	USINT	RT	48	Calendar week
wt	HI	USINT	RT	3	1=MO,..,7=SU
tt	HI	USINT	RT	27	Day
mm	HI	USINT	RT	11	Month
jj	HI	USINT	RT	13	Year
RecServ	HI	INT	RT	3	Active recipe server IOS
KillTele	HI	INT	RT	0	IOS number for killer telegram trigger
SupVTele	HI	INT	RT	0	IOS number for monitoring telegram trigger
ICM_Mode	HI	WORD	RT	W#16#0000	ICM running mode

Name	Access	D.Type	A.Type	Value	Comment
ICM_StartWarningSP	NL	INT	ENG	3079	Setpoint waiting time for ICM start-up warning
TSynSoll	NL	INT	RT	0	Target value for time synchronization
TSynIst	NL	INT	RT	0	Actual value for time synchronization
PrgFltSetMask	NL	DWORD	ENG	DW#16#F0000000	Program Fault Set Mask
PrgFltMasked	HR	DWORD	SYS	DW#16#F0000000	Program Fault Masked
AccFltSetMask	NL	DWORD	ENG	DW#16#0000003C	Access Fault Set Mask
AccFltMasked	HR	DWORD	SYS	DW#16#0000000C	Access Fault Masked
DBNotLoaded	HI	BOOL	ENG	FALSE	PrgFlt: DB not loaded
FC_NotLoaded	HI	BOOL	ENG	TRUE	PrgFlt: FC not loaded
SFC_NotLoaded	HI	BOOL	ENG	TRUE	PrgFlt: SFC not loaded
FB_NotLoaded	HI	BOOL	ENG	TRUE	PrgFlt: FB not loaded
SFB_NotLoaded	HI	BOOL	ENG	TRUE	PrgFlt: SFB not loaded
DBWrErr	HI	BOOL	ENG	FALSE	PrgFlt: DB write error
DI_WrErr	HI	BOOL	ENG	FALSE	PrgFlt: DI write error
DBNumErr	HI	BOOL	ENG	FALSE	PrgFlt: DB number error
DI_NumErr	HI	BOOL	ENG	FALSE	PrgFlt: DI number error
FC_NumErr	HI	BOOL	ENG	FALSE	PrgFlt: FC number error
FB_NumErr	HI	BOOL	ENG	FALSE	PrgFlt: FB number error
AlignErrRd	HI	BOOL	ENG	FALSE	PrgFlt: Alignment error read
AlignErrWr	HI	BOOL	ENG	FALSE	PrgFlt: Alignment error write
BcdConvErr	HI	BOOL	ENG	FALSE	PrgFlt: BCD conversion error
AreaLenErrRd	HI	BOOL	ENG	FALSE	PrgFlt: Area length error read
AreaLenErrWr	HI	BOOL	ENG	FALSE	PrgFlt: Area length error write
AreaErrRd	HI	BOOL	ENG	FALSE	PrgFlt: Area error read
AreaErrWr	HI	BOOL	ENG	FALSE	PrgFlt: Area error write
TimerNumberErr	HI	BOOL	ENG	FALSE	PrgFlt: Timer number error
CounterNumberErr	HI	BOOL	ENG	FALSE	PrgFlt: Counter number error
AccFltRd	HI	BOOL	ENG	TRUE	AccFlt: periph. access fault read error
AccFltWr	HI	BOOL	ENG	TRUE	AccFlt: periph. access fault write error
AccFltRdN	HI	BOOL	ENG	TRUE	AccFlt: periph. access fault read err. (n>1)
AccFltWrN	HI	BOOL	ENG	TRUE	AccFlt: periph. access fault read err. (n>1)

Name	Access	D.Type	A.Type	Value	Comment
TWarnSoll	NL	USINT	SYS	0	Warning time set value – sec
TWarnIst	HI	USINT	SYS	0	Warning time actual value – sec
StartDayOfCWeek	HI	USINT	ENG	0	Start day for a calendar week: 0=Mo, 1=Su.
AdjustSEQYear	HI	BOOL	ENG	FALSE	Automatic reconciliation of the year in SEQ-DS

Parameter set of PCU version V7 (DB 701)

Name	Access	D.Type	A.Type	Value	Comment
PcuNo	RD	USINT	RT	7	PCU number
Hour	HI	USINT	RT	12	Hour
Minute	HI	USINT	RT	26	Minute
Second	HI	USINT	RT	30	Second
CalWeek	HI	USINT	RT	10	Calendar week
WeekDay	HI	USINT	RT	5	1=MO,..,7=SU
Day	HI	USINT	RT	7	Day
Month	HI	USINT	RT	3	Month
Year	HI	USINT	RT	14	Year
StartDayOfCWeek	HI	USINT	ENG	0	Start day for a calendar week: 0=Mo, 1=Su.
AdjustSeqYear	HI	BOOL	ENG	FALSE	Automatic reconciliation of the year in SEQ-DS
RecServ	HI	INT	RT	1	Active recipe server IOS
KillTele	HI	INT	RT	0	IOS number for killer telegram trigger
SupVTele	HI	INT	RT	0	IOS number for monitoring telegram trigger
IcmMode	HI	WORD	RT	W#16#0000	ICM processing method
IcmStartingAlarmSP	NL	INT	ENG	0	Setpoint waiting time for ICM start-up warning
IcmStartingAlarmAV	HI	INT	RT	0	Actual value waiting time for ICM start-up warning
TimeSyncSP	NL	INT	ENG	0	Target value for time synchronization
TimeSyncAV	HI	INT	RT	0	Actual value for time synchronization
DisableIcmMsgManual	HI	BOOL	ENG	FALSE	ICM message Disable manual on/off
DisableIcmMsgModeSwitch	HI	BOOL	ENG	FALSE	ICM message Disable manual/automatic
DisableIcmMsgForcingEnable	HI	BOOL	ENG	FALSE	ICM msg. Disable ForcingEnable
DisableIcmMsgSimFeedback	HI	BOOL	ENG	FALSE	ICM msg. Disable SimulateFeedback

Name	Access	D.Type	A.Type	Value	Comment
ReportEopFinalStates	HI	BOOL	ENG	FALSE	EOP final states messaging
ReportEopQuiesStates	HI	BOOL	ENG	FALSE	EOP quiet/intermediate states messaging
ReportEopTransStates	HI	BOOL	ENG	FALSE	EOP transfer states messaging
S7-1500: OnRestartStopSequences	HI	BOOL	ENG	FALSE	Stop units during startup
S7-1500: OnRestartResetICMs	HI	BOOL	ENG	FALSE	Reset ICMs during startup
S7-1500: OnRestartResetBOS	HI	BOOL	ENG	FALSE	Reset CAS during startup
S7-1500: SysErrorCollection	HR	LWORD	SYS	LW#16#00000000 00000000	(internal)
S7-400: PrgFltSetMask	NL	DWORD	ENG	DW#16#00000000	Program Fault Set Mask
S7-400: PrgFltMasked	HR	DWORD	SYS	DW#16#00000000	Program Fault Masked
S7-400: AccFltSetMask	NL	DWORD	ENG	DW#16#0000003C	Access Fault Set Mask
S7-400: AccFltMasked	HR	DWORD	SYS	DW#16#0000003C	Access Fault Masked
S7-400: DB_NotLoaded	HI	BOOL	ENG	FALSE	PrgFlt: DB not loaded
S7-400: FC_NotLoaded	HI	BOOL	ENG	FALSE	PrgFlt: FC not loaded
S7-400: SFC_NotLoaded	HI	BOOL	ENG	FALSE	PrgFlt: SFC not loaded
S7-400: FB_NotLoaded	HI	BOOL	ENG	FALSE	PrgFlt: FB not loaded
S7-400: SFB_NotLoaded	HI	BOOL	ENG	FALSE	PrgFlt: SFB not loaded
S7-400: DB_WrErr	HI	BOOL	ENG	FALSE	PrgFlt: DB write error
S7-400: DI_WrErr	HI	BOOL	ENG	FALSE	PrgFlt: DI write error
S7-400: DB_NumErr	HI	BOOL	ENG	FALSE	PrgFlt: DB number error
S7-400: DI_NumErr	HI	BOOL	ENG	FALSE	PrgFlt: DI number error
S7-400: FC_NumErr	HI	BOOL	ENG	FALSE	PrgFlt: FC number error
S7-400: FB_NumErr	HI	BOOL	ENG	FALSE	PrgFlt: FB number error
S7-400: AlignErrRd	HI	BOOL	ENG	FALSE	PrgFlt: Alignment error read
S7-400: AlignErrWr	HI	BOOL	ENG	FALSE	PrgFlt: Alignment error write
S7-400: BcdConvErr	HI	BOOL	ENG	FALSE	PrgFlt: BCD conversion error

Name	Access	D.Type	A.Type	Value	Comment
S7-400: AreaLenErrRd	HI	BOOL	ENG	FALSE	PrgFlt: Area length error read
S7-400: AreaLenErrWr	HI	BOOL	ENG	FALSE	PrgFlt: Area length error write
S7-400: AreaErrRd	HI	BOOL	ENG	FALSE	PrgFlt: Area error read
S7-400: AreaErrWr	HI	BOOL	ENG	FALSE	PrgFlt: Area error write
S7-400: TimerNumber-Err	HI	BOOL	ENG	FALSE	PrgFlt: Timer number error
S7-400: CounterNum-berErr	HI	BOOL	ENG	FALSE	PrgFlt: Counter number error
S7-400: AccFltRd	HI	BOOL	ENG	TRUE	AccFlt: periph. access fault read er-ror
S7-400: AccFltWr	HI	BOOL	ENG	TRUE	AccFlt: periph. access fault write er-ror
S7-400: AccFltRdN	HI	BOOL	ENG	TRUE	AccFlt: periph. access fault read err. (n>1)
S7-400: AccFltWrN	HI	BOOL	ENG	TRUE	AccFlt: periph. access fault read err. (n>1)

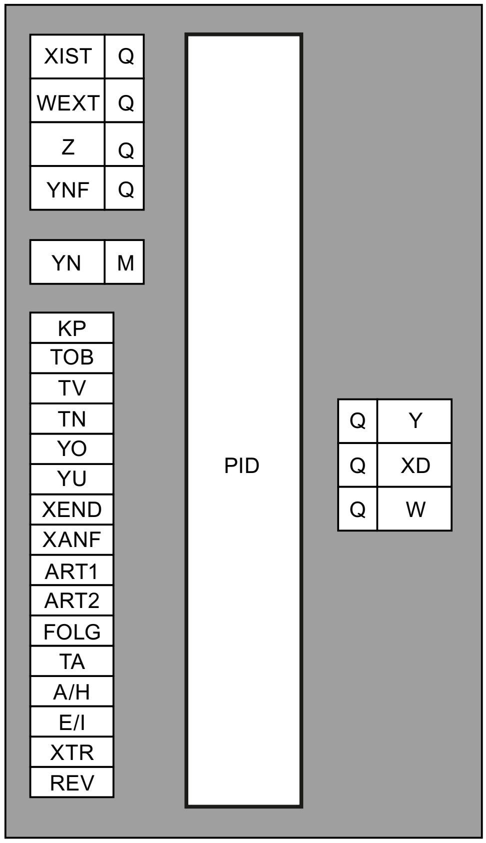
5.7.22 PID controller

Basics

The block contains all necessary functions for a max. of 64 controllers per PCU. The controller is suitable for:

- Fixed value control
- Cascade control

- Ratio control
- Hardware back-up control



The PID controller works according to the position algorithm, i.e. a manipulated variable `Y/ActuatingVal` is calculated. By interconnecting `Y/ActuatingVal` with the AOUT block a continuous controller results, in conjunction with the INKU a step controller.

Parameter sets for block PID / DB 730

PCU parameter set before Version V7 (max. 64 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Y	NL	INT	RT, LOG	0	Manipulated variable
KP	NL	INT	ENG, LOG	200	P-gain 0.00-327.67
TN	NL	INT	ENG, LOG	0	Adjusting factor=TA/TN
TV	NL	INT	ENG, LOG	0	Anticipation factor=Tv/TA
A/H	NL	BOOL	ENG	FALSE	Mode: 0/1 = Auto/Manual
E/I	NL	BOOL	ENG	FALSE	Setpoint: 0/1 = external/internal
W	NL	INT	RT, LOG	0	Effective setpoint
XIST	NL	QUEL	ENG	ANA,1,ANA	Source actual value

Name	Access	D.Type	A.Type	Value	Comment
WEXT	NL	QUEL	ENG	ANA,1,ANA	Source external target value
Z	NL	QUEL	ENG	ANA,1,ANA	Source disturbance value
YNF	NL	QUEL	ENG	ANA,1,ANA	Source tracking value
XD	NL	INT	RT, LOG	0	Control deviation
XANF	NL	INT	ENG, LOG	0	Initial limit for XIST, WEXT, W
XEND	NL	INT	ENG, LOG	1000	Deadline for XIST, WEXT, W
YU	NL	INT	ENG, LOG	0	Lower limit for manipulated variable Y
YO	NL	INT	ENG, LOG	1000	Upper limit for manipulated variable Y
TEILANL	NL	USINT	ENG, LOG	0	Assigned unit
YN	NL	BOOL	ENG	FALSE	YN Marker
ART1	NL	INT	ENG	0	Controller type 1
ART2	NL	INT	ENG	0	Controller type 2
FOLG	NL	INT	ENG	0	Number of following controller
TA	NL	INT	ENG, LOG	1	Sampling time in seconds (0=Lock)
TOB	NL	INT	ENG, LOG	0	Dead band
WIED	NL	BOOL	ENG	FALSE	Hot restart: 0/1 = unchanged/Manual
REV	NL	BOOL	ENG	FALSE	Reversing duty: 0/1 = No/Yes
XTR	NL	BOOL	ENG	FALSE	X-Tracking: 0/1 = No/Yes
X	NL	INT	RT, LOG	0	Actual value
Wex	NL	INT	RT	0	External setpoint
Zex	NL	INT	RT, LOG	0	External disturbance value
YNFex	NL	INT	RT	0	External tracking value
INT_US	NL	INT	SYS	0	internal usage
Status	HR	INT	RT, LOG	0	Status as word

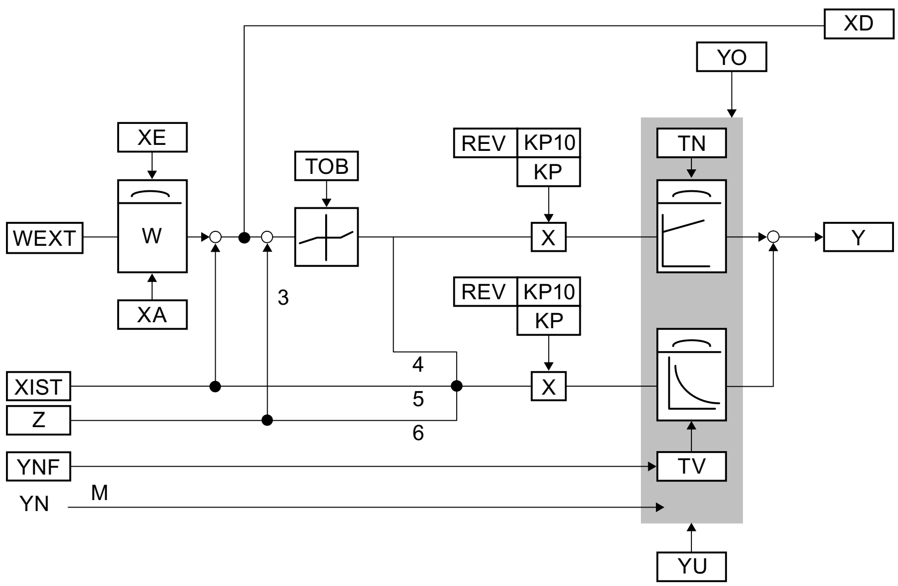
PCU Version V7 parameter set (max. 64 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Sequence	NL	ENUM	ENG, LOG	<0>	Assigned unit
ProcValRef	NL	GREF	ENG	NULL	Source actual value
S7-1500: ProcValRef_ST	HR	WORD	SYS	W#16#0000	Signal status
SpExtRef	NL	GREF	ENG	NULL	Source external target value
S7-1500: SpExtRef_ST	HR	WORD	SYS	W#16#0000	Signal status
DisturbValRef	NL	GREF	ENG	NULL	Source disturbance value

Name	Access	D.Type	A.Type	Value	Comment
S7-1500: DisturbVal-Ref_ST	HR	WORD	SYS	W#16#0000	Signal status
FollowUpValRef	NL	GREF	ENG	NULL	Source tracking value
S7-1500: FollowUpVal-Ref_ST	HR	WORD	SYS	W#16#0000	Signal status
ActuatingVal	NL	REAL	RT, LOG	0.000	Manipulated variable
Gain	NL	REAL	ENG, LOG	2.000	P amplification
PostCtrlFactor	NL	REAL	ENG, LOG	10.000	Integration factor
PreCtrlFactor	NL	REAL	ENG, LOG	0.000	Derivative action factor
ManualMode	NL	BOOL	ENG	FALSE	Mode: 0=Auto, 1=Manual
InternalMode	NL	BOOL	ENG	FALSE	Setpoint: 0=External, 1=Internal
EffSpValue	NL	REAL	RT, LOG	0.000	Effective setpoint
Deviation	NL	REAL	RT, LOG	0.000	Control deviation
ProcValLL	NL	REAL	ENG, LOG	0.000	Start limit
ProcValUL	NL	REAL	ENG, LOG	10.000	Limit value
ActuatingValLL	NL	REAL	ENG, LOG	0.000	Actuating value lower limit
ActuatingValUL	NL	REAL	ENG, LOG	10.000	Upper limit for actuating value
ActuatingVal FollowUp	NL	BOOL	ENG	FALSE	Track actuating value
ModeSwitch1	NL	INT	ENG	0	Controller type 1
ModeSwitch2	NL	INT	ENG	0	Controller type 2
NextController	NL	INT	ENG	0	Number of following controller
SampleTime	NL	INT	ENG, LOG	1	Sample time [sec] (0=Disabled)
DeadBand	NL	REAL	ENG, LOG	0.000	Dead band
SwitchToMan OnReset	NL	BOOL	ENG	FALSE	On restart: 0=Keep mode, 1=Switch to manual
ReversingMode	NL	BOOL	ENG	FALSE	Reversing mode
ProcValTracking	NL	BOOL	ENG	FALSE	Track process value
NoteIsActive	HR	BOOL	RT	FALSE	Note is available
ProcValAct	HR	REAL	RT, LOG	0.000	Actual value
SpExtAct	HR	REAL	RT	0.000	External setpoint
DisturbValAct	HR	REAL	RT, LOG	0.000	External disturbance value

Name	Access	D.Type	A.Type	Value	Comment
FollowUpValAct	HR	REAL	RT	0.000	External tracking value
Status	HR	WORD	RT, LOG	W#16 #0000	(Internal)

Block diagram PID controller:



Parameter type1/position jumpers 3-6:

Art1 / ModeSwitch1	3456 ("1"= jumper in)
0	1100
1	1010
2	1001
3	0001

Operation modes PID controller

- Cascade control:
  The actuating value `Y/ActuatingVal` of the primary controller is fed to the secondary controller as setpoint `WEXT/SpExtRef`. If the secondary controller is switched from INTERNAL mode to EXTERNAL mode, the system changeover is bumpless, i.e. the `Y/ActuatingVal` of the primary controller is adapted to the actual value of the following controller.
- Controller with hardware backup:
  Here, the software controller is cascaded by a discrete controller (see chapter Examples (Page 183)). By means of the flag `YN/ActuatingValFollowUp` (track manipulated variable) the controller program is informed that the hardware controller is switched to DDC mode.
  On the positive edge of the flag `YN/ActuatingValFollowUp`, `Y/ActuatingVal` is tracked to the value applied to the input `YNF/FollowUpValRef` of the controller; the software controller is in Manual mode!
- Operation mode Auto-Manual:
  Beginning with the shift from Automatic to Manual operation mode, the controller does not compute new manipulated variables anymore.
  - The `Y/ActuatingVal` to be output can be changed by operator input. With every processing cycle the controller compares the current `Y/ActuatingVal` to the limits `YO/ActuatingValUL` and `YU/ActuatingValLL` and corrects it accordingly. This means that , even if an illegal value is entered manually, the permitted `Y/ActuatingVal` is corrected to these limits.
  - When changing from Manual to Automatic mode, there is synchronization similar to cascade control; only, its own position `Y/ActuatingVal` is being used as the correction value. The operation mode transfer is printed automatically to the message file.
- Mode External-Internal (E/I):
  In Internal mode, the setpoint `W/EffSpValue` can be set by operator input. To adjust the `W/EffSpValue` to the limits `XANF/ProcValLL` and `XEND/ProcValUL` the same applies as for the actuating value `Y/ActuatingVal` in manual mode. The operation mode transfer is edited automatically by the operation mode printer.
- X tracking:
  In X tracking mode (`XTR/ProcValTracking`), the internal setpoint tracks the actual value `XIST/ProcValRef`. This renders possible a bumpless transfer from External to Internal.
- Logging:
  Each change from auto to manual or external to internal is logged if the messages are enabled.

Controller types PID block

Controller type	Value	Description
ART1 / ModeSwitch 1	0	PID action: is derived from `XD/Deviation + Z/DisturbValRef`
	1	The D action is derived from `XIST/ProcValRef` PI from `XD/Deviation + Z/DisturbValRef`
	2	The D action is derived from `Z/DisturbValRef` PI from `XD/Deviation + Z/DisturbValRef`
	3	D action is derived from any variable applied to the `Z/DisturbValRef` input; PI action from `X/ProcValAct`
ART2 / ModeS-witch2	0	Type: Fixed-setpoint controller
	4	Type: Master controller
	8	Type: Controller with HW backup; A/H = "1"

User interface: Tracking flag `YN/ActuatingValFollowUp`

No.	Flag	No.	Flag	No.	Flag	No.	Flag
1	M 968.0	17	M 970.0	33	M 972.0	49	M 974.0
2	M 968.1	18	M 970.1	34	M 972.1	50	M 974.1
3	M 968.2	19	M 970.2	35	M 972.2	51	M 974.2
4	M 968.3	20	M 970.3	36	M 972.3	52	M 974.3
5	M 968.4	21	M 970.4	37	M 972.4	53	M 974.4
6	M 968.5	22	M 970.5	38	M 972.5	54	M 974.5
7	M 968.6	23	M 970.6	39	M 972.6	55	M 974.6
8	M 968.7	24	M 970.7	40	M 972.7	56	M 974.7
9	M 969.0	25	M 971.0	41	M 973.0	57	M 975.0
10	M 969.1	26	M 971.1	42	M 973.1	58	M 975.1
11	M 969.2	27	M 971.2	43	M 973.2	59	M 975.2
12	M 969.3	28	M 971.3	44	M 973.3	60	M 975.3
13	M 969.4	29	M 971.4	45	M 973.4	61	M 975.4
14	M 969.5	30	M 971.5	46	M 973.5	62	M 975.5
15	M 969.6	31	M 971.6	47	M 973.6	63	M 975.6
16	M 969.7	32	M 971.7	48	M 973.7	64	M 975.7

In automatic mode, the actuating value `Y/ActuatingVal`is overwritten with the value `YNF/FollowUpValRef` as long as flag `YN/ActuatingValFollowUp` is set.

Optimization PID controllers

Setting the control parameters without being familiar with the plant behavior

In this case, the controller parameters for optimum control are not yet known. In order to nonetheless achieve reasonable control, the following adjustments need to be carried out:

Proportional coefficient:	`KP / Gain`	1	(minimum value)
Sampling time:	`TA / SampleTime`	1	
Adjustment time:	`TN / PostControlFactor`	+32767 s	(largest value; 0=> Tn = infinite!)
Set-up time:	`TV / PreControlFactor`	0 s	

Controller type	Description
PI controller	Set desired setpoint and reduce system deviation to zero in Manual operation mode
	Switch to automatic mode
	Increase `KP/Gain` slowly until the control loop starts to oscillate due to small setpoint changes
	Slightly decrease `KP/Gain` until the oscillations are eliminated
	Reduce `TN/PostControlFactor` until the control circuit starts to oscillate again
	Slightly increase `TN/PostControlFactor` until oscillation is eliminated

Controller type	Description
PID controller	Set desired setpoint and reduce system deviation to zero in Manual operation mode
	Switch to automatic mode
	Increase `KP/Gain` slowly until the control loop starts to oscillate due to small setpoint changes
	Switch `TV/PreControlFactor` from 0 to 1 s
	Increase `TV/PreControlFactor` until the oscillation is eliminated
	Slowly increase `KP/Gain` again until oscillation occurs again
	Repeat adjustment following steps stated above until oscillation can not be eliminated anymore
	Decrease `TV/PreControlFactor` and `KP/Gain` slightly until oscillations stop
	Reduce `TN/PostControlFactor` until the control circuit starts to oscillate again
	Slightly increase `TN/PostControlFactor` until the tendency to oscillation is eliminated

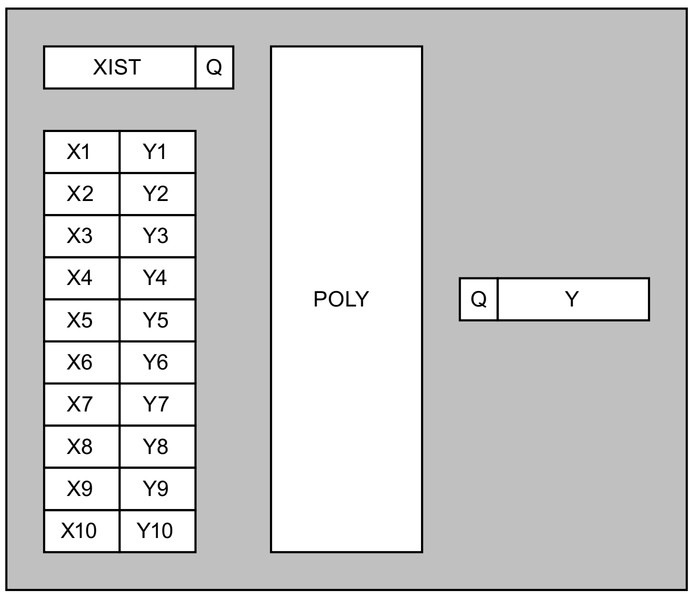
5.7.23 POLY - Polygon adjustment

Basics

The function enables the conversion of an input value via a polygon with up to 10 interpolation point pairs. The interpolation between the interpolation points is to be linear. 32 interrupt lines are available.

If the input value `X/ProcValRef` is outside of the interpolation point range (`X/ProcValRef < X1` or `X/ProcValRef > X10`), the output value `Y/OutputVal` is set to `Y1` or `Y10`.

- `X/ProcValRef` is transferred to the block via an interconnection.
- `Y/OutputVal` is available as a source for other blocks (in numerical formats REAL, INT, DINT for PCU V7).



Parameter sets for block POLY / DB 735

PCU parameter set before Version V7 (max. 32 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Y	NL	INT	RT	0	Output value
Factor	HI	INT	ENG	0	Y = Calculated value * factor
X	NL	QUEL	ENG	DB 0. DBW 0	Data source
X1	NL	INT	ENG	0	Interpolation point 1: X value
Y1	NL	INT	ENG	0	Interpolation point 1: Y-value
X2	NL	INT	ENG	0	Interpolation point 2: X value
Y2	NL	INT	ENG	0	Interpolation point 2: Y-value
X3	NL	INT	ENG	0	Interpolation point 3: X value
Y3	NL	INT	ENG	0	Interpolation point 3: Y-value

Name	Access	D.Type	A.Type	Value	Comment
X4	NL	INT	ENG	0	Interpolation point 4: X value
Y4	NL	INT	ENG	0	Interpolation point 4: Y-value
X5	NL	INT	ENG	0	Interpolation point 5: X value
Y5	NL	INT	ENG	0	Interpolation point 5: Y-value
X6	NL	INT	ENG	0	Interpolation point 6: X value
Y6	NL	INT	ENG	0	Interpolation point 6: Y-value
X7	NL	INT	ENG	0	Interpolation point 7: X value
Y7	NL	INT	ENG	0	Interpolation point 7: Y-value
X8	NL	INT	ENG	0	Interpolation point 8: X value
Y8	NL	INT	ENG	0	Interpolation point 8: Y-value
X9	NL	INT	ENG	0	Interpolation point 9: X value
Y9	NL	INT	ENG	0	Interpolation point 9: Y-value
X10	NL	INT	ENG	0	Interpolation point 10: X value
Y10	NL	INT	ENG	0	Interpolation point 10: Y-value

PCU Version V7 parameter set (max. 32 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
ProcValRef	NL	GREF	ENG	NULL	Reference to process value
S7-1500: ProcVal-Ref_ST	HR	WORD	SYS	W#16#0000	Signal status
ProcValAct	RD	REAL	RT	0.000	Process value
OutputVal	RD	REAL	RT	0.000	Output value
OutputFactor	NL	REAL	ENG	1.000	Output factor
OutputValInt	HR	INT	RT	0	Output Value (converted)
OutputVal-Dint	HR	DINT	RT	L#0	Output Value (converted)
X1	NL	REAL	ENG	0.000	Interpolation point 1: X value
Y1	NL	REAL	ENG	0.000	Interpolation point 1: Y-value
X2	NL	REAL	ENG	0.000	Interpolation point 2: X value
Y2	NL	REAL	ENG	0.000	Interpolation point 2: Y-value
X3	NL	REAL	ENG	0.000	Interpolation point 3: X value
Y3	NL	REAL	ENG	0.000	Interpolation point 3: Y-value
X4	NL	REAL	ENG	0.000	Interpolation point 4: X value
Y4	NL	REAL	ENG	0.000	Interpolation point 4: Y-value
X5	NL	REAL	ENG	0.000	Interpolation point 5: X value
Y5	NL	REAL	ENG	0.000	Interpolation point 5: Y-value
X6	NL	REAL	ENG	0.000	Interpolation point 6: X value
Y6	NL	REAL	ENG	0.000	Interpolation point 6: Y-value
X7	NL	REAL	ENG	0.000	Interpolation point 7: X value
Y7	NL	REAL	ENG	0.000	Interpolation point 7: Y-value
X8	NL	REAL	ENG	0.000	Interpolation point 8: X value
Y8	NL	REAL	ENG	0.000	Interpolation point 8: Y-value

Name	Access	D.Type	A.Type	Value	Comment
X9	NL	REAL	ENG	0.000	Interpolation point 9: X value
Y9	NL	REAL	ENG	0.000	Interpolation point 9: Y-value
X10	NL	REAL	ENG	0.000	Interpolation point 10: X value
Y10	NL	REAL	ENG	0.000	Interpolation point 10: Y-value

5.7.24 PULSE - Control block for pulse sequences

Basics of the "PULSE" class

The "PULSE" function allows the generation of binary pulse sequences.

Typical applications are:

- Flushing of various CIP valves / double seat valves
- Pulse for the injection of substances (injection on/off)
- Switching ON of outputs (one by one)
- Changing subsequent pulses depending on the selected mode (activate/deactivate specific outputs)

Implementation

For control of seatlifting in double seat valves, all outputs of a PULSE instance are connected to the seatlifting inputs of the SLB instances which are assigned to a cleaning train. This can be done either using the configured references or via routing in the user program.

Quantities and configuration

- The PULSE class consists of **max. 96 (S7-400) / 84 (S7-1500)** data records.
- An instance can control up to 24 pulse outputs/seat liftings. This is configured with the "`PulseCount`" attribute.
- In addition, pulse times and number of loop runs per triggering are defined in the parameter assignment.
- The assignment to the signal outputs of the PULSE instance is specified in the SLB parameter assignment with the input references (`'Top_PulseSigRef, Btm_PulseSigRef')` attribute).
- The PULSE class does not have its own visualization object.

Parameter sets for block PULSE / DB 2121

Parameter set PCU Version V7 (S7-400: 96 / S7-1500: 84)

Name	Access	D.Type	A.Type	Value	Comment
PulseType	NL	ENUM	ENG	<Simple pulses> <Overlapping pulses>	Pulse type
PulseCount	NL	INT	ENG	2	Number of pulses
CycleCount	NL	INT	ENG	1	Number of loops (0=continuous mode)
CycleCounterAct	HI	INT	RT	0	Loop counter
OffDelay	NL	REAL	ENG	0.000	Off delay (overlap)
EvalEnableSignals	NL	BOOL	ENG	FALSE	Evaluate individual enable signals
CmdActivate	NL	BOOL	RT	FALSE	Activate pulse group
CmdHoldResume	NL	BOOL	RT	FALSE	1=Hold pulses, 0=Resume pulses
CmdReset	NL	BOOL	RT	FALSE	Reset pulse group
FbkComplete	RD	BOOL	RT	FALSE	Pulse generation complete
ErrorInfo	RD	ENUM	RT	--	Error info
ErrAddInfo	RD	ENUM	RT	--	Additional error information
P01_Duration	NL	REAL	ENG	2.000	Pulse duration
P01_Enabled	HI	BOOL	RT	TRUE	Enable signal
P01_Interlock	HI	BOOL	RT	TRUE	Interlock
P01_InterlockRef	HI	GREF	ENG	(NULL)	Reference to interlock
S7-1500: P01_InterlockRef_ST	HR	WORD	SYS	W#16#0000	Signal status
P01_Signal	RD	BOOL	RT	FALSE	Pulse signal
P01_SignalRef	HI	PREF	ENG	(NULL)	Target reference to pulse signal
S7-1500: P02_InterlockRef_ST	HR	WORD	SYS	W#16#0000	Signal status
...					

Note

Configuration changes to pulse sequence runtime

- A change to the ENG attributes (particularly PulseType, PulseCount or CycleCount) on a running block instance can result in the premature interruption of the pulse sequence.

General functions

- The active outputs "Pxx_Signal" are activated sequentially in form of a pulse sequence.
- The pulse sequence can be executed in various operating modes:
  - Simple pulses
  - Overlapping pulses
- In the operating modes "Simple pulses" and "Overlapping pulses", the pulse sequence can be repeated in a loop.
- Any signal output can be enabled and interlocked separately.

Basic functions for all operating modes:

Enable function

Individual outputs of the pulse sequence can be skipped with the enable function, whereby the control as well as the ON time and interlock are suppressed.

- "EvalEnableSignals"=1 activates the enable function. Only the pulse outputs are activated with "Pxx_Enabled"=1, and the sequence of the pulse sequence can be manipulated in the user program.
- When an output is reached with "Pxx_Enabled"=0, it is skipped and the pulse sequence is resumed at the next output with "Pxx_Enabled"=1.
- "EvalEnableSignals"=0 deactivates the enable function. The pulse sequence runs completely from the first to the last pulse output.

Interlock of outputs

The reference inputs "Pxx_InterlockRef" are always evaluated, which means they do not have a separate enable signal.

- When "Pxx_Interlock"=0 while the "Pxx_Signal" output is active, this is switched off immediately and the associated On time stops.
- When "Pxx_Interlock" is active once again, the signal output is activated and the associated ON time runs to the end.

Starting / stopping / resetting the pulse sequence

- The pulse sequence is started with "CmdActivate"=1.
- The pulse sequence is stopped immediately and all outputs are deactivated with "CmdActivate"=0.
- The pulse sequence is stopped immediately and all outputs are deactivated with "CmdReset"=1. "CmdActivate"=1 and "CmdHoldResume"=1 is reset by the block.

Holding / resuming the pulse sequence

- The sequence of the pulse sequence can be interrupted or "frozen" with "CmdHoldResume"=1. The activated output remains active and the associated On time stops.
- When "CmdHoldResume"=0, the sequence continues, i.e. the associated On time runs to the end.

"Simple pulses" operating mode

- Starting at the first output with `"Pxx_Enabled"=1`, the outputs of the pulse sequence are activated and deactivated again one after the other.
- Each output `"Pxx_Signal"` is activated and deactivated again after expiration of the time `"Pxx_Duration"`.
- The succeeding output is activated immediately after the preceding one is switch off.
- `"CycleCount"` = 1...n = number of cycles; `"CycleCount"` = 0 = Continuous mode
- When the number of loop runs has been processed, pulse generation is stopped and `"FbkComplete"` is set to 1.

"Overlapping pulses" operating mode

- This operating mode is similar to the "Simple pulses" operating mode. Starting at the first output with `"Pxx_Enabled"=1`, the outputs of the pulse sequence are activated and deactivated again one after the other.
- However, each output `"Pxx_Signal"` is activated for the time interval `"Pxx_Duration"` + `"OffDelay"`.
- The succeeding output is activated immediately after `"Pxx_Duration"` of the preceding output. The result is an overlapping of the preceding and succeeding pulse for the time `"OffDelay"` in [sec].
- During the `"OffDelay"` of a `"Pxx_Signal"` output, a `"Pxx_Interlock"=0` immediately deactivates the associated output signal while the next output signal with `„Pxx_Enabled"=1` remains activated and the interlock of the previous output signal has no effect.

Notes on the call system

The block is disabled by default, because use in the projects cannot be generally assumed.

Note

PCU type S7-400: The user must call the "BmExecPulseFC" FC774" in a suitable user interface FB:

- **"BmUsr1000msFB" FB 1225 → Standard case**; a granularity of 1 sec for the pulse times should be sufficient for most applications.
- "BmUsr100msFB" FB 1224 → only for special cases - not suitable when using CPU 414 or CPU 416/417 < Version 07
  Notice: This applies to all active instances of the block, which means it is essential to ensure a sufficient cycle time reserve in time slices "TS01_CycleAct" ... "TS10_CycleAct" (see "ZyklMess" block). For this reason, the number of active instances must be reduced to the minimum number required. In case of doubt, contact SIMATIC Customer Support.
- The time slice chosen here must conform to the receiver of the PULSE signal outputs, i.e. the "SLB" class.

Note

PCU type S7-1500: The user must call the block in a suitable user interface FB:

- **"BmUsrCallIf1000msFC" FC 907 → Standard case**; a granularity of 1 sec for the pulse times should be sufficient for most applications.
- "BmUsrCallIf100msFC" FC 903 → Only for special cases - only recommended when CPU 1518 PN/DP V2.0 or higher is used!
  Notice: This applies to all active instances of the block. For this reason, a sufficient cycle time reserve should be ensured in the time slices "TS01_CycleAct" ... "TS10_CycleAct" (see "ZyklMess" block).
  For this reason, the number of active instances must be reduced to the minimum number required. In case of doubt, contact SIMATIC Customer Support.
- The time slice chosen here must conform to the receiver of the PULSE signal outputs, i.e. the "SLB" class.

Example:

- Call syntax STL: `CALL "BmPulseClassFB", "PULSE"`
- Call syntax SCL: `"PULSE" ();`

5.7.25 SENDPU - Send buffer - 1 to 6

Basics

- The messages are taken from the FIFO and registered in the send buffer of the according server IOS.
- The transmit buffers are permanently assigned to the FIFOs and thus to the servers:
  SENDPUs 1 to 3 (DB 674, 675, 676) are assigned to FIFOs 1 to 3 and server 1.
  SENDPUs 4 to 6 (DB 694, 695, 696) are assigned to FIFOs 4 to 6 and server 2.

Note

- This SENDPU parameterization is permanently saved in the system and may not be changed by the user.
- The following description is therefore used for diagnostics only.

Note

Not for PCU type S7-1500!

The SENDPU parameter assignment is not available for PCU type S7-1500!

Parameter sets for SENDPUx block

PCU parameter set

Name	Access	D.Type	A.Type	Value	Comment
QUITT_PC	NL	WORD	SYS	W#16#000D	Acknowledgment from IOS
ZUST	NL	INT	SYS	0	State of FB
TUES	NL	INT	SYS	16	Monitoring time target value in seconds
TUEI	RD	INT	SYS	16	Monitoring time actual value
TELE_NR	NL	WORD	SYS	W#16#000D	Consecutive telegram number
MAX_RETRY	NL	INT	SYS	3	Max. number of retransmissions
RETRY_CNT	RD	INT	SYS	1	Retransmission counter
RET_A8P	RD	INT	SYS	0	Return value Alarm_8P
ERR_A8P	RD	BOOL	SYS	FALSE	Alarm_8P error
ERR_A8P_Init	RD	BOOL	SYS	TRUE	Alarm_8P Init error
ERR_A8P_Len	RD	BOOL	SYS	FALSE	Alarm_8P error associated value length
STAT_A8P	RD	INT	SYS	0	Status Alarm_8P
SERVER	NL	INT	SYS	1	Registered server
DATALEN	RD	INT	SYS	140	Length of current data
EN_SWITCH	NL	BOOL	SYS	FALSE	Enable switch-over to backup server
MAIN_SRV	NL	INT	SYS	1	Main server
RES_SRV	NL	INT	SYS	2	Backup server
ACT_SRV	NL	USINT	SYS	1	current Server IOS
CHANNEL	NL	USINT	SYS	1	Transmission channel
Check_Len_A8P	HI	BOOL	SYS	FALSE	Start length check
Act_Len_A8P	HI	INT	SYS	430	Current length SFB35 in FB585
DbgCommProblems	HI	INT	SYS	0	Error counter: Communication problems
DbgNewstartErr	HI	INT	SYS	0	Error counter: Error during initial call
DbgMemAccessErr	HI	INT	SYS	0	Error counter: Memory access
DbgDBErr	HI	INT	SYS	0	Error counter: DI incorrect
DbgEVIDUsed	HI	INT	SYS	0	Error counter: EV_ID already used
DbgLoMem	HI	INT	SYS	0	Error counter: Insufficient RAM
DbgMsgIsBlocked	HI	INT	SYS	0	Error counter: Message blocked with EV_ID
DbgStillBusy	HI	INT	SYS	0	Error counter: Job not yet comple-ted
DbgPointerErr	HI	INT	SYS	0	Error counter: ANY-Pointer not permitted
DbgUnknownState	HI	INT	SYS	13	Error counter: Other states

5.7.26 Sequences / TeilAnl - sequence control

Basics

The 'SEQUENCES' system block enables simultaneous operation of up to 64 / 128 units (per PCU). In such a case, the commands and step enabling conditions stored in EOPs are processed recipe-controlled.

Note

Functionality of the SEQUENCES / unit block in the PCU versions:

- Up to PCU version **V4.x** the sequence control block is called **Sequence**
- From PCU version **V5.x** the sequence control block is called **SEQUENCES.**
- From PCU version **V7** the sequence control block is called **SEQUENCES** with support for max. **128 sequences.**

The texts/names of the following sections are to be assigned accordingly!

Integral components of the SEQUENCES block are up to 24 "Digital Function Modules" (DFM) plus runtime monitoring, which can be configured as time increment, up counter, down counter, limit value increment, setpoint increment, router or decoder or as a mask or setpoint/actual value cell. The respective setpoint or limit values are stored in the recipe lists.

Parameter sets for block SEQUENCES / DB 725

PCU parameter set before Version V7 (max. 64 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Status	HI	DWORD	RT, LOG	DW#16#00000182	Status flags
AtSync	HI	BOOL	RT	FALSE	Seq. stopped at synchronization
AtAlter	HI	BOOL	RT	FALSE	Seq. stopped at alternative
AtAlterEx	HI	BOOL	RT	FALSE	Seq. stopped at alternative exit
AtTrans	HI	BOOL	RT	FALSE	Seq. stopped at transition
UserFl	HI	BOOL	RT	FALSE	User flag for sequencer
OpReq	HI	BOOL	RT	FALSE	Operation request
OpStart	HI	BOOL	RT	FALSE	Operation start flag
OpStop	HI	BOOL	RT	FALSE	Operation stop flag
SeqRun	HI	BOOL	RT	FALSE	Seq. running
SeqStrtImp	HI	BOOL	RT	FALSE	Seq. start pulse
OpTimeErr	HI	BOOL	RT	FALSE	Error TUE
RecLoadErr	HI	BOOL	RT	FALSE	Recipe load error
MsgErrBuf	HI	BOOL	RT	FALSE	Error messages for seq.
MsgProcBuf	HI	BOOL	RT	FALSE	Process messages for seq.
MsgRCSBuf	HI	BOOL	RT	FALSE	RCS messages for seq.
StopAtSync	HI	BOOL	RT	FALSE	Blocking flag for synchronization
NotSeqHeld	HI	BOOL	RT	TRUE	Continue Release/Stop

Name	Access	D.Type	A.Type	Value	Comment
SeqPaused	HI	BOOL	RT	FALSE	Set sequence to pause
Comm_Mode	HI	BOOL	RT	FALSE	Commissioning mode for seq.
Enable	NL	BOOL	RT	TRUE	Permanent condition
Man_Moded	NL	BOOL	RT	FALSE	Manual / Auto mode
Occupied	HI	BOOL	RT	FALSE	Sequence occupied
MsgSystem	HI	BOOL	RT	FALSE	System message for seq.
MsgWarning	HI	BOOL	RT	FALSE	Warning message for seq.
MsgOperating	HI	BOOL	RT	TRUE	Operator message for seq.
Intern	HI	DWORD	RT	DW#16#A0000100	Internal flags
Tel15_Strt	HI	BOOL	RT	FALSE	Tele 15: send batch start
Tel15_End	HI	BOOL	RT	FALSE	Tele 15: send batch end
Tel15_Err	HI	BOOL	RT	FALSE	Tele 15: send batch error
Tel13_Ima	HI	BOOL	RT	FALSE	Tele 13: send image
RecChanged	HI	BOOL	RT	FALSE	Request recipe at next start
WaitForEOP	HI	BOOL	RT	FALSE	Wait for EOP processing
ReverseSupvCounting	HI	BOOL	ENG	FALSE	Revert the step monitoring time
Ctrl	HI	DWORD	RT	DW#16#00000000	Control flags
RelT_Supv	HI	BOOL	ENG	FALSE	Enable msg. monitoring time
Disable	HI	BOOL	ENG	FALSE	Disable editing
DisProtoc	HI	BOOL	ENG	FALSE	Disable logging
UserBit	HI	BOOL	RT	FALSE	User flag
RelTime	HI	BOOL	ENG	FALSE	Enable step time
LvSyncAlt	HI	BOOL	RT	FALSE	Leave sync. or alternative
R_TimeRel	HI	BOOL	ENG	FALSE	Release step time via CFC
AdaptBFTA	HI	BOOL	ENG	FALSE	Call adapt-FC
ReloadDFM	HI	BOOL	RT	FALSE	Reload setpoints
FixedRouteID	HI	BOOL	ENG	FALSE	Unit with fixed Route-ID
R_Holding	HI	BOOL	RT	FALSE	Requirement: Hold (int. state)
R_Held	HI	BOOL	RT	FALSE	Requirement: Hold
R_Restart	HI	BOOL	RT	FALSE	Request restart
R_Running	HI	BOOL	RT	FALSE	Requirement: Running
R_Pausing	HI	BOOL	RT	FALSE	Requirement: Pause (int. state)
R_Paused	HI	BOOL	RT	FALSE	Requirement: Pause
R_Abortin	HI	BOOL	RT	FALSE	Requirement: Abort (int. state)
R_Aborted	HI	BOOL	RT	FALSE	Requirement: Cancel
R_Stoppin	HI	BOOL	RT	FALSE	Requirement: Stopping (int. state)
R_Stopped	HI	BOOL	RT	FALSE	Requirement: Stop
R_Reset	HI	BOOL	RT	FALSE	Requirement: Inactive
R_Starting	HI	BOOL	RT	FALSE	Requirement: Start (int. state)
R_Rest_ng	HI	BOOL	RT	FALSE	Requirement: Restart (int. state)

Name	Access	D.Type	A.Type	Value	Comment
SeqStat	HI	USINT	RT, LOG	0	Automation state of sequencer
ReqLdState	HI	USINT	RT	0	Get machine status for recipe
NewStep	NL	INT	RT	0	Step new
Step	NL	INT	RT	0	Step old
ManGroup	NL	USINT	ENG, LOG	0	0 none, 1..64 manual group, >64 SeqManMode=1
OpState	HI	USINT	RT	0	EOP execution state
BA_Year	NL	USINT	RT, LOG	97	Year for order no./batch no.
BA_RecType	NL	USINT	RT, LOG	1	Recipe type
BA_RecNo	NL	INT	RT, LOG	0	Recipe number
BA_ONo	NL	INT	RT, LOG	0	Order number
BA_BNo	NL	INT	RT, LOG	0	Batch number
BA_Name0	HI	DWORD	RT	DW#16#10000000	Batch name length
BA_Name1	HI	DWORD	RT	DW#16#00000000	Batch name 0-3
BA_Name2	HI	DWORD	RT	DW#16#00000000	Batch name 4-7
BA_Name3	HI	DWORD	RT	DW#16#00000000	Batch name 8-11
BA_Name4	HI	DWORD	RT	DW#16#00000000	Batch name 12-15
BA_ID	NL	DWORD	RT	DW#16#00000000	Batch ID
Step_ID	NL	DWORD	RT	DW#16#00000000	Step ID
EOP_No	NL	INT	RT, LOG	0	Operation number * 10
Start_EOP	HI	DWORD	RT	DW#16#00000000	EOP start time
StTime_SP	NL	DWORD	RT, LOG	DW#16#00000000	EOP monitoring time start value [sec]
StTime_AV	NL	DWORD	RT, LOG	DW#16#00000000	EOP monitoring time actual value [sec]
Delay_SP	HI	INT	RT	10	Wait time target value (sec.)
Delay_AV	HI	INT	RT	0	Wait time actual value (sec.)
RecReq_SP	HI	USINT	RT	0	Rec.load-TUE target value (sec.)
RecReq_AV	HI	USINT	RT	0	Rec.load-TUE actual value (sec.)
TransNo	HI	INT	RT	0	EOPnr der Transition
SyncAltNo	HI	USINT	RT	0	Number Sync/Alternative
AltResult	HI	USINT	RT	0	Result of alternative
StepMode	HI	INT	RT	0	Step mode
EOPState	HI	USINT	RT, LOG	0	Status of equipment operation
Time_Rel	NL	STEP	ENG	U M108.1	Enable monitoring time: Request command

Name	Access	D.Type	A.Type	Value	Comment
RouteID	HI	INT	ENG	0	RCS: RouteID
Step_Int	HR	DWORD	RT, LOG	DW#16#00000000	Step

PCU version V7 parameter set (max. 128 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Status	HR	DWORD	RT, LOG	DW#16#00000082	Status flags
AtSync	HR	BOOL	RT	FALSE	Seq. stopped at synchronization
AtAlter	HR	BOOL	RT	FALSE	Seq. stopped at alternative
AtAlterEx	HR	BOOL	RT	FALSE	Seq. stopped at alternative exit
AtTrans	HR	BOOL	RT	FALSE	Seq. stopped at transition
MatCheckIsActive	HR	BOOL	RT	FALSE	Material check is active
UserFl	HI	BOOL	RT	FALSE	User flag for sequencer
OpReq	HI	BOOL	RT	FALSE	Operation request
OpStart	HI	BOOL	RT	FALSE	Operation start flag
OpStop	HI	BOOL	RT	FALSE	Operation stop flag
SeqRun	HI	BOOL	RT	FALSE	Seq. running
SeqStrtImp	HI	BOOL	RT	FALSE	Seq. start pulse
OpTimeErr	HI	BOOL	RT	FALSE	Error TUE
RecLoadErr	HI	BOOL	RT	FALSE	Recipe load error
MsgErrBuf	HI	BOOL	RT	FALSE	Error messages for seq.
MsgProcBuf	HI	BOOL	RT	FALSE	Process messages for seq.
MsgRCSBuf	HI	BOOL	RT	FALSE	RCS messages for seq.
StopAtSync	HI	BOOL	RT	FALSE	Blocking flag for synchronization
NotSeqHeld	HI	BOOL	RT	FALSE	Continue Release/Stop
SeqPaused	HI	BOOL	RT	FALSE	Set sequence to pause
NextStepCond	HR	BOOL	RT	FALSE	Value of the step condition
Comm_Mode	HI	BOOL	RT	FALSE	Commissioning mode for seq.
PermCond	RD	BOOL	RT	TRUE	Permanent condition
Man_Moded	HI	BOOL	RT	FALSE	Manual / Auto mode
NoteIsActive	HI	BOOL	RT	FALSE	Note is available
Occupied	HR	BOOL	RT	FALSE	Sequence occupied
MsgSystem	HI	BOOL	RT	FALSE	System message for seq.
MsgWarning	HI	BOOL	RT	FALSE	Warning message for seq.
MsgOperating	HI	BOOL	RT	TRUE	Operator message for seq.
Intern	HI	DWORD	RT	DW#16#80000000	Internal flags
Tel15_Strt	HI	BOOL	RT	FALSE	Tele 15: send batch start
Tel15_End	HI	BOOL	RT	FALSE	Tele 15: send batch end
Tel15_Err	HI	BOOL	RT	FALSE	Tele 15: send batch error
Tel13_Ima	HI	BOOL	RT	FALSE	Tele 13: send image

Name	Access	D.Type	A.Type	Value	Comment
RecChanged	HI	BOOL	RT	FALSE	Request recipe for next start
ReverseSupv Counting	HI	BOOL	ENG	FALSE	Revert the step monitoring time
Ctrl	HI	DWORD	RT	DW#16#00000000	Control flags
RelT_Supv	HI	BOOL	ENG	FALSE	Enable msg. monitoring time
Disable	HI	BOOL	ENG	FALSE	Disable editing
DisProtoc	HI	BOOL	ENG	FALSE	Disable logging
UserBit	HI	BOOL	RT, LOG	FALSE	User flag
RelTime	HI	BOOL	ENG	FALSE	Enable step time
LvSyncAlt	HI	BOOL	RT	FALSE	Leave sync. or alternative
ReloadDFM	HI	BOOL	RT	FALSE	Reload setpoints
FixedRouteID	HI	BOOL	ENG	FALSE	Unit with fixed Route-ID
SuppressOp StepChange	HI	BOOL	RT	FALSE	Prevent manual step change
ReqHolding	HI	BOOL	RT	FALSE	Requirement: Hold (int. state)
ReqHeld	HI	BOOL	RT	FALSE	Requirement: Hold
ReqRestart	HI	BOOL	RT	FALSE	Request restart
ReqRunning	HI	BOOL	RT	FALSE	Requirement: Running
ReqPausing	HI	BOOL	RT	FALSE	Requirement: Pause (int. state)
ReqPaused	HI	BOOL	RT	FALSE	Requirement: Pause
ReqAborting	HI	BOOL	RT	FALSE	Requirement: Abort (int. state)
ReqAborted	HI	BOOL	RT	FALSE	Requirement: Cancel
ReqStopping	HI	BOOL	RT	FALSE	Requirement: Stopping (int. state)
ReqStopped	HI	BOOL	RT	FALSE	Requirement: Stop
ReqReset	HI	BOOL	RT	FALSE	Requirement: Inactive
ReqStarting	HI	BOOL	RT	FALSE	Requirement: Start (int. state)
ReqRestarting	HI	BOOL	RT	FALSE	Requirement: Restart (int. state)
SeqStat	HI	USINT	RT, LOG	0	Automation state of sequencer
ReqLdState	HI	USINT	RT	0	Get machine status for recipe
NewStep	NL	INT	RT	0	Step new
Step	NL	INT	RT	0	Step old
Step_Int	HR	DWORD	RT, LOG	DW#16#00000000	Step
ManGroup	NL	USINT	ENG, LOG	0	0: none, 1..128: Manual group, >128: SeqManMode=1
OpState	HI	USINT	RT	0	EOP execution state
BA_Year	NL	USINT	RT, LOG	0	Year for order no./batch no.
BA_RecType	NL	USINT	RT, LOG	1	Recipe type
BA_RecNo	NL	INT	RT, LOG	0	Recipe number
BA_ONo	NL	INT	RT, LOG	0	Order number
BA_BNo	NL	INT	RT, LOG	0	Batch number
BA_Name	NL	STRING[32]	RT		Batch name

Name	Access	D.Type	A.Type	Value	Comment
BA_ID	NL	DWORD	RT	DW#16#00000000	Batch ID
ProductID	HI	DINT	RT	L#0	Product
Step_ID	NL	DWORD	RT	DW#16#00000000	Step ID
EOP_No	NL	INT	RT, LOG	0	Operation number * 10
MatCheckIsActive	HI	BOOL	RT	FALSE	Material check is active
MatCheckReset	HI	BOOL	RT	FALSE	Reset material compatibility check
MatCheckPreID	HI	DINT	RT	0	Predecessor for material check
MatCheckPostID	HI	DINT	RT	0	Successor for material check
MatCheckStatus	HR	USINT	RT	0	Material check status
Start_EOP	HI	DINT	RT	L#0	EOP start time
StTime_SP	NL	DINT	RT, LOG	L#0	EOP monitoring time start value [sec]
StTime_AV	NL	DINT	RT, LOG	L#0	EOP monitoring time actual value [sec]
Delay_SP	NL	INT	ENG	10	Wait time target value (sec.)
Delay_AV	HR	INT	RT	0	Wait time actual value (sec.)
RecReq_SP	HI	USINT	ENG	0	Rec.load-TUE target value (sec.)
RecReq_AV	HR	USINT	RT	0	Rec.load-TUE actual value (sec.)
TransNo	HR	INT	RT	0	EOP number of transition
SyncAltNo	HR	USINT	RT	0	Number Sync/Alternative
AltResult	HR	USINT	RT	0	Result of alternative
StepMode	HR	INT	RT	0	Step mode
EOPState	HR	USINT	RT, LOG	0	Status of equipment operation
Time_Rel	NL	STEP	ENG	U M108.1	Enable monitoring time: Request command
S7-1500: Time_Rel_ST	HR	WORD	SYS	W#16#0000	Signal status
RouteID	HI	INT	RT	0	RCS: RouteID
PermCondRef	NL	GREF	ENG, LOG	NULL	Source for the permanent condition
S7-1500: PermCondRef_ST	HR	WORD	SYS	W#16#0000	Signal status
EopFeatures	HR	DWORD	RT	DW#16#00000000	EOP features
EopNscBlrIDB	HR	INT	RT, LOG	0	EOP-NSC via BLR object: BLR-IDB
EopNscBlrInst	HR	INT	RT, LOG	0	EOP-NSC via BLR object: BLR instance

Note

Features of V7-PCU

- If the permanent condition „PermCondRef“ is assigned to the result bit „Result“ of a BLR object, the respective logic can be operated and monitored in the system overview application and in the unit block.
- To visualize the step condition via a BLR object, a corresponding configuration is necessary in the recipe editor (cf. "Configure EOP definition").

Sequence - Function description

Starting a sequence:

Starting a sequence can be carried out by the following methods:

- Recipe system (synchronization, alternative), start conditions: step=0, Automatic mode, AutomaticStepChange = 1
- SEQS operation start block, start condition step=0, Automatic mode

From the IOS by steering the step register; recipe, batch and order numbers may need to be set beforehand.

For all start methods, it is a prerequisite that the permanent condition be fulfilled.

Step operation:

From the IOS display, every step of a sequence can be activated independently of the step enabling conditions by setting the step from 0 to n (n=1...). If the selected step has an alternative or synchronization set by the operator, then the unit control does not run an EOP and column "EOP" in the process cell overview remains empty for the unit.

Manual mode:

(BTR indication H/A on IOS display):

The signal Manual mode indicates switching to the manual control level.

In Manual mode, the step enabling of the sequence is locked.

The signal Manual mode is given directly from the block SeqManMode "Manual signal distributor“. Therefore, each sequence is assigned to a manual group via the parameter HZUO/ManGroup.

Release/Stop - Control:

(BTR indication "+/-" on IOS display): By switching the rel./stop control bit to the stop state (BTR indication "-") the step enabling of a sequence can be blocked even if the step enabling conditions are fulfilled. Further, the participating synchronizations and alternatives are blocked.

Abort of a sequence:

The processing of a unit is aborted when a permanent condition is missing (DB/SeqPermCond = "0") or when step 0 is specified.

Monitoring time of unit TUET/EopMonTmError:

(ANZ indicator S, flashing when addressing the monitoring): The individual sequence steps are monitored over time.

At the end of the runtime of the step (M101.0=1) given via the recipe list with released runtime monitoring (flag M 101.7=1), the error indication (S) is displayed flashing; horn operation is activated (M 99.4) and the message "Error monitoring time start" is output if sequence messages are enabled.

The unit control ignores the exceeded runtime monitoring, if the control bit EnableMonTm (M101.7) in the EOP is reset.

Wait time or delay time is active TVERZ/EopDelayTmActive

When an EOP is started, a second time (the wait/delay time) is started. This time is the same for all steps and is given by parameter Delay_SP in the parameter assignment.

The flag TVERZ/EopDelayTmActive (M101.1) can be interpreted in the EOP:

Time running: M101.1 = 0

Time expired: M101.1 = 1

On a signal change PLUS/AutomaticStepChange from 0 to 1 (indication "-" to "+", from STOP to ENABLE), the time is retriggered.

Operation request BEDA/OperatorRequest

for a sequence indication B (flashing) on the IOS display:
By setting control bit BEDA/OperatorRequest (M101.6) in the EOP, a request to the operator can be signaled in the process cell overview (e.g. sample taking). The operator can acknowledge the request using the O-quit function key, which resets the BEDA/OperatorRequest control bit.

Report disable PSPR/DisableStepProt

- By setting control bit PSPR/DisableStepProt (M102.5) in the EOP, the entry of the processed EOP in the report can be disabled.

Additional unit On / Off ZGEA/AddDeviceOn

By linking the flag ZGEA/AddDeviceOn (M101.5) in the EOP or unit program with an ICM, higher-level output or the like, an additional device can be switched on and off in the process cell overview using two function keys (ON / OFF).

Processing operation:

After the unit is started, the data for processing the current step is read from the transferred control recipe and the specified EOP FC is called.

At the start of a step (start EOP), the setpoints for the DFMs are set.

When the next step condition is met and the processing of the EOP, automatic mode (A+) and sequence enable (+) are returned, the next step of the recipe list is processed. After processing all steps from the recipe list the system returns automatically to step 0 and the processing of the sequence is finished.

In the actual recipe all parameters can be changed during processing. The insertion and deletion of complete parts of recipes stays blocked, however, and is only released when the sequence has returned to step 0.

The user program for the unit (TA 1 ... TA 128) is called **before and after the EOP FC**:

S7-400: FB 1001 ... FB 1128

Flag "SeqFbCalledAfterEop" (=M104.4) is used to evaluate the call before or after the EOP FC.

SeqFbCalledAfterEop = 0: Call before the EOP processing

SeqFbCalledAfterEop = 1: Call after the EOP processing

Call	Functions
Before EOP:	Designing the continuous condition Calls for DFM operation Manipulations before EOP processing (e.g. with multiplex EOPs)
After EOP:	Manipulations after EOP processing Step manipulations (jump forward/backward)

S7-1500: FC 3001 ... FC 3128

- Alternative 1 (default/delivery condition): Sequencer data record #SeqData is used to evaluate the call before or after the EOP FC.
  #SeqData.xCallEnvBeforeEop = 1: Call before the EOP processing
  #SeqData.xCallEnvAfterEop = 1: Call after the EOP processing
- Alternative 2: Flag "SeqFbCalledAfterEop" (=M104.4) is used to evaluate the call before or after the EOP FC.
  SeqFbCalledAfterEop = 0: Call before the EOP processing
  SeqFbCalledAfterEop = 1: Call after the EOP processing

In step 0 only DFM modules that are programmed outside the EOPs are processed.

Manual group allocation of units

The actuators and unit controls for a process cell can be structured in manual groups. The classification is of any kind but will generally comply with the technological requirements. The assignment of a unit to the manual group is specified by the parameter `HZUO/ManGroup` of the dataset.

HZUO <V7 ManGroup V7	Meaning
0	No allocation of manual signals to the data set. In this case, the manual signal can be assigned by the user program. Example: Unit 12: By program `="SEQ".au[12].boHAND;`
1..64 / 1..128	SeqManMode bit comes after the status of the relevant manual group
>64 / >128	SeqManMode bit is always = 1 !

Activation of the manual groups in the user program

The manual group releases can be assigned in the user program via defined flags (flag = "1" means "manual").

The distribution to the other ICM and unit blocks of the allocated group is performed by the system.

Bit address per manual group								Byte
.0	.1	.2	.3	.4	.5	.6	.7	Flag
1	2	3	4	5	6	7	8	712
9	10	11	12	13	14	15	16	713
17	18	19	20	21	22	23	24	714
25	26	27	28	29	30	31	32	715
33	34	35	36	37	38	39	40	716
41	42	43	44	45	46	47	48	717
49	50	51	52	53	54	55	56	718
57	58	59	60	61	62	63	64	719
65	66	67	68	69	70	71	72	720 *)
73	74	75	76	77	78	79	80	721 *)
81	82	83	84	85	86	87	88	722 *)
89	90	91	92	93	94	95	96	723 *)
97	98	99	100	101	102	103	104	724 *)
105	106	107	108	109	110	111	112	725 *)
113	114	115	116	117	118	119	120	726 *)
121	122	123	124	125	126	127	128	727 *)

*) as of PCU V7

See also

Unit function block (Page 648)

Sequencer interface (Page 654)

5.7.27 SeqStart - batch job start in the PCU

Basics

A 'SeqStart' data set is available for each unit.

In the order system, the batches are created with one of the 4 start modes. After being enabled, the system enters the batch data in the SeqStart dataset of the appropriate start unit.

(For further details, refer to section Description of processing sequence (Page 698))

Note

Functionality of the batch order start control block SeqStart / CAS / BOS [DB718] in the PCU versions:

- Up to PCU Version **V4.x** → Name: **CAS** max. **64 units**
- Starting from PCU Version **V5.x** → Name: **SeqStart** max. **64 units**
- Starting from PCU Version **V7** → Name: **SeqStart** max. **128 units**
- For **PCU type S7-1500** → Name: **SeqStart (DE) / BOS (EN)** max. **128 units**

The number of data sets must be at least the same as that of the unit data sets.

Parameter sets for the SeqStart / DB 718 block

PCU parameter set before Version V7 (max. 64 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Jahr	NL	USINT	RT	0	Year for RType, JobNo, BatchNo
RTyp	NL	USINT	RT	0	Recipe type
RezNr	NL	INT	RT	0	Recipe number
AuftrNr	NL	INT	RT	0	Order number
ChargeNr	NL	INT	RT	0	Batch number
Teilanl	NL	INT	RT	0	Unit no
ModSt	NL	USINT	RT	0	Start mode
Zeit_hi	NL	INT	RT	0	Start time since 1.1.70
Zeit_lo	NL	INT	RT	0	Start time since 1.1.70
Nachr	NL	USINT	RT	1	Message CAS state
Schr_int	HR	USINT	RT	0	Internal step number
Zeit_CAD	HR	USINT	RT	0	Monitoring time arrival batch data
Startbed	HR	BOOL	RT	FALSE	Start condition met
AnwendSt	NL	BOOL	RT	TRUE	Start by user
FreiTA	HR	BOOL	RT	FALSE	Unit enable
StSper	NL	BOOL	RT	FALSE	Start lock by user
CADiO	RD	BOOL	RT	FALSE	Batch data OK
CADniO	RD	BOOL	RT	FALSE	Batch data faulty

Name	Access	D.Type	A.Type	Value	Comment
WartHa	NL	BOOL	RT	FALSE	Wait Manual by user
CASLoe	NL	BOOL	RT	FALSE	CAS entry delete by user
StaImp	NL	BOOL	RT	FALSE	Start pulse
FlInt	HR	BOOL	RT	FALSE	Internal edge
CADAnf	NL	BOOL	RT	FALSE	Batch data request by user
Anwendbi	HI	BOOL	RT	FALSE	User bit
CASAuftr	HR	BOOL	RT	FALSE	CAS from order control
StatCAS	HR	BOOL	RT	FALSE	Status request running
TrigBearb	HR	BOOL	RT	FALSE	CAS entry first time cycle
SendTeleUpdate	HI	BOOL	RT	FALSE	Send message update

PCU version V7 parameter set (max. 128 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
BatchYear	NL	USINT	RT	0	Year for RType, JobNo, BatchNo
RecCatNo	NL	USINT	RT	0	Recipe type
RecipeNo	NL	INT	RT	0	Recipe number
OrderNo	NL	INT	RT	0	Order number
BatchNo	NL	INT	RT	0	Batch number
SequenceNo	NL	INT	RT	0	Unit no
ProductID	HI	DINT	RT	L#0	Product
StartMode	NL	USINT	RT	0	Start mode
StartTime	NL	DINT	RT	L#0	Start time since 1.1.70
SeqStartState	NL	USINT	RT	1	Message CAS state
SeqStartFAS	HR	USINT	RT	0	Internal step number
MonTmBatchData	HR	USINT	RT	0	Monitoring time arrival batch data
StartCond	HR	BOOL	RT	FALSE	Start condition met
UserStart	NL	BOOL	RT	FALSE	Start by user
DeallocUnit	HR	BOOL	RT	FALSE	Unit enable
UserLock	NL	BOOL	RT	FALSE	Start lock by user
BatchDataOk	RD	BOOL	RT	FALSE	Batch data OK
BatchDataErr	RD	BOOL	RT	FALSE	Batch data faulty
ManualWait	NL	BOOL	RT	FALSE	Wait Manual by user
DeleteRecord	NL	BOOL	RT	FALSE	CAS entry delete by user
StartPulse	NL	BOOL	RT	FALSE	Start pulse
EdgeInternal	HR	BOOL	RT	FALSE	Internal edge
BatchDataReqUsr	NL	BOOL	RT	FALSE	Batch data request by user
UserFlag	HI	BOOL	RT	FALSE	User bit
SeqStartOrderSys	HR	BOOL	RT	FALSE	CAS from order control
StatusUpdateAc-tive	HR	BOOL	RT	FALSE	Status request running
FirstExec	HR	BOOL	RT	FALSE	CAS entry first time cycle

Name	Access	D.Type	A.Type	Value	Comment
SendUpdateTele	HI	BOOL	RT	FALSE	Send message update
BatchName	HI	STRING[32]	RT		Batch name

Start mode: Four different start modes are available

The start mode is set in the order system when defining the order type in "Options / Start mode".

(For further details, refer to section "History (Page 712)")

ModSt / StartMode	Description
0	Order data set free
1	This start mode can only be called from the AS user program. If the unit is not free, there is an IOS error message (message class = "Warning", message text = "Batch start not possible, unit occupied")
2	Start mode = "soon as possible" the batches start as soon as the unit is free
3	Start mode = "by time..." if the unit is not free when the time elapses, there is an IOS error message
4	Start mode = "by event" • Event-dependent start by user program; if SEQ not free, send error message • The batch order start is processed by function block CAS_FB / BmCasFB / BmBosFB [FB 705]. During processing of the CAS, the current data record is stored in the CAS_UDT (UDT 718) / BmBosUDT. • The function block calls user block CAS_USR_FB / BmUsrCasFB / BmUsrBosFB [FB 1205]. Based on the transferred data record number, the user can determine the unit for which data is available. • The unit is started once a positive signal is received from start flag (boFreeMode4).

If the unit is started, a telegram with start mode 0 is sent to the IOS. The CAS entry is then free again.

At the start, order number, batch number, recipe type and recipe number are copied from the order data set to the unit to be started.

5.7.28 SLB - Seat lifting for double seat valves

Basics of the "SLB" class

You use the "Seat Lifting Block (SLB)" function to control double seat valves (DSV).

Typical applications are:

- Separate activation of the main valve and the seat liftings
- Interlocks for main valve and seat liftings
- Free I/O assignment with configurable end position monitoring

- External mode for synchronized purging of the double seat valves in groups
- Internal mode for individual purging

Implementation

A double seat valve (DSV) is composed of a main valve and 2 seat liftings (top, bottom).

In BRAUMAT/SISTAR the main valve is controlled with an ICM instance while the seat liftings are controlled together with an SLB (Seat Lifting Block) referenced as an AddOn block.

- The connection of an SLB instance to the ICM main valve takes place with an object reference in the SLB data set.
- The SLB function block transfers the state of the main valve (open/closed, fault, manual/automatic, etc.) to the seat liftings.
- The SLB function controls the seat liftings.

Quantities and configuration

- Each double seat valve needs an ICM and an SLB instance.
- The SLB class comprises **max. 436 (S7-400) / 342 (S7-1500)** data records.
- The configuration engineer specifies the assignment to the ICM instance in the SLB parameter assignment with the `'PartnerIcm'` attribute.
- The assignment of the SLB instance to the ICM group/instance can be random, **but a strict one-to-one assignment must be observed which means the same ICM must not be referenced in multiple SLBs.**
- The interconnection as well as the definition of the pulse/pause and monitoring functions for the seat liftings takes place in the SLB parameter assignment.
- AS configuration:
  In the den EOP user blocks, the user configures the interlocks and the activation of the main valve (ICM data set) and the seat liftings (SLB data set).
- IOS configuration:
  The standard ICM control is used for visualization in process picture generation, and it is interconnected to the **SLB instance** (instead of the ICM instance).
  With that, the state and the properties of the main valve and, in a separate tab, also the seat liftings are displayed and operated in the ICM OCX during runtime.

Parameter sets for block SLB / DB 2120

Parameter set PCU Version V7 (S7-400: 436 / S7-1500: 342)

Name	Access	D.Type	A.Type	Value	Comment
PartnerIcm	NL	OREF	ENG, LOG	(NULL)	Partner ICM instance
S7-1500: PartnerIcm_ST	HR	WORD	SYS	W#16#0000	Signal status
PupaMode	NL	ENUM	ENG	Extern	Pulse generation type

Name	Ac-cess	D.Type	A.Type	Value	Comment
ExclusiveLift	NL	BOOL	ENG	TRUE	1=Exclusive seatlifting only
NoLiftOnOpenIcm	NL	BOOL	ENG	TRUE	1=No seatlifting when ICM is open
NoLiftOnIcmError	NL	BOOL	ENG	TRUE	1=No seatlifting when ICM is in error state
MonTimeTop	NL	REAL	ENG, LOG	0.000	Monitoring time feedback top seat [sec]
MonTimeBtm	NL	REAL	ENG, LOG	0.000	Monitoring time feedback bottom seat [sec]
Top_PulseSigRef	NL	GREF	ENG	(NULL)	Reference to external pulse signal for top seat
S7-1500: Top_PulseSigRef_ST	HR	WORD	SYS	W#16#0000	Signal status
Top_PulseSigAct	HI	BOOL	RT	FALSE	External pulse signal for top seat
Top_CmdActivateRef	NL	GREF	ENG	(NULL)	Reference to command 'activate pulse operation' for top seat
S7-1500: Top_CmdActivateRef_ST	HR	WORD	SYS	W#16#0000	Signal status
Top_CmdActivateAct	HI	BOOL	RT	FALSE	Command 'activate pulse operation' for top seat
Top_FbkRef	NL	GREF	ENG	(NULL)	Reference to feedback 'top seat is lifted'
S7-1500: Top_FbkRef_ST	HR	WORD	SYS	W#16#0000	Signal status
Top_CtrlRef	NL	PREF	ENG	(NULL)	Reference to control output top seat
S7-1500: Top_CtrlRef_ST	HR	WORD	SYS	W#16#0000	Signal status
Top_InterlockRef	NL	GREF	ENG	(NULL)	Reference to interlock top seat
S7-1500: Top_InterlockRef_ST	HR	WORD	SYS	W#16#0000	Signal status
Top_Error	RD	BOOL	RT	FALSE	Feedback error top seat
Top_Interlock	RD	BOOL	RT	TRUE	Interlock top seat
Top_OnFbk	RD	BOOL	RT	FALSE	Feedback 'top seat is lifted'
Top_OnCtrl	RD	BOOL	RT	FALSE	Top seat lifting is active
Top_AvMonTime	HR	REAL	RT, LOG	0.000	Actual value for feedback monitoring top seat [s]
Btm_PulseSigRef	NL	GREF	ENG	(NULL)	Reference to external pulse signal for bottom seat
S7-1500: Btm_PulseSigRef_ST	HR	WORD	SYS	W#16#0000	Signal status

Name	Access	D.Type	A.Type	Value	Comment
Btm_PulseSigAct	HI	BOOL	RT	FALSE	External pulse signal for bottom seat
Btm_CmdActivateRef	NL	GREF	ENG	(NULL)	Reference to command 'activate pulse operation' for bottom seat
S7-1500: Btm_CmdActivateRef_ST	HR	WORD	SYS	W#16#0000	Signal status
Btm_CmdActivateAct	HI	BOOL	RT	FALSE	Command 'activate pulse operation' for bottom seat
Btm_FbkRef	NL	GREF	ENG	(NULL)	Reference to feedback 'bottom seat is lifted'
S7-1500: Btm_FbkRef_ST	HR	WORD	SYS	W#16#0000	Signal status
Btm_CtrlRef	NL	PREF	ENG	(NULL)	Reference to control output bottom seat
S7-1500: Btm_CtrlRef_ST	HR	WORD	SYS	W#16#0000	Signal status
Btm_InterlockRef	NL	GREF	ENG	(NULL)	Reference to interlock bottom seat
S7-1500: Btm_InterlockRef_ST	HR	WORD	SYS	W#16#0000	Signal status
Btm_Error	RD	BOOL	RT	FALSE	Feedback error bottom seat
Btm_Interlock	RD	BOOL	RT	TRUE	Interlock bottom seat
Btm_OnFbk	RD	BOOL	RT	FALSE	Feedback bottom seat is lifted
Btm_OnCtrl	RD	BOOL	RT	FALSE	Bottom seat lifting is active
Btm_AvMonTime	HR	REAL	RT, LOG	0.000	Actual value for feedback monitoring bottom seat [s]
OpAckError	HI	BOOL	RT	FALSE	Acknowledge feedback error (operator)
OpCmdLiftTop	HI	BOOL	RT	FALSE	Lift top seat (operator)
OpCmdLiftBtm	HI	BOOL	RT	FALSE	Lift bottom seat (operator)
IcmModeValid	HR	BOOL	RT	FALSE	(Internal)
IcmMode	HR	ENUM	RT	Automatic	ICM operating mode
Status	HR	DWORD	RT, LOG	DW#16#0F10100C	(Internal)
ErrorInfo	RD	ENUM	RT	--	Error info
AddInfo	RD	ENUM	RT	Partner ICM is not configured	Additional sequence information

Configuration notes

Request, feedback, interlock and control are references that can be used as an option. We highly recommend their use to significantly increase transparency of the configuration.

- If a reference is empty (process value "(NULL)"), the user program itself must ensure the required signal routing.

Status transfer ICM ↔ SLB

1. ICM to SLB expansion

The SLB expansion takes into account the following attributes of the main valve:

ICM attribute (UDT)	Description	Effect on SLB
xManualMode	ICM is in manual mode	SLB accepts operating mode
xFbk0	Feedback off/closed	Seat lifting enable only with closed main valve "FeatureBit" → NoLiftOnOpenIcm
xErrMem	ICM error memory	Seat lifting enable only with error-free main valve "FeatureBit" → NoLiftOnIcmError
xDisabled	ICM is disabled	Seat lifting enable only with active main valve
xMaintActive xAffected-ByMaintActive	ICM is in maintenance or is affec-ted by it	Seat lifting enable only with process-ac-tive main valve

2. SLB expansion to ICM

The SLB expansion can influence the main valve with functional characteristics:

ICM attribute (UDT)	Description
xFtrOnSlErrorSafePos	Functional characteristic: Bring ICM in safety position on seat lifting feed-back error
xFtrOnSlActiveSafePos	Functional characteristic: Bring ICM in safety position when seat lifting is in progress
xInfSlIsActive	Information of the SL block: Seat lifting is active
xInfSlFbkError	Information of the SL block: A feedback error is pending
xInfSlLockActive	At least one functional characteristic in combination with runtime informa-tion of the SLB expansion prevent activation

SLB functions

Control logic

- The seat liftings top and bottom can be requested separately from each other, activation takes place exclusively upon request.
  "FeatureBit" → ExclusiveLift
- If required, both seat liftings can only be enabled when the main valve is closed.
  "FeatureBit" → NoLiftOnOpenIcm

- When the main valve is in an error or maintenance state, seat lifting is not possible.
- "FeatureBit" → NoLiftOnIcmError

Activation pulses

The activation and feedback signals can be configured as output or input references or they can be activated in the user program.

- Configuration: (PREF) Top_CtrlRef, Btm_CtrlRef, (GREF) Top_FbkRef, Btm_FbkRef
- UDT: xCtrlLiftTop, xCtrlLiftBtm, xFbkLiftTop, xFbkLiftBtm

The activation signals are pulsed. Pulsing can take place internally or be specified externally. For internal operation the pulse-pause periods can be configured separately for the top and bottom seat liftings.

- Configuration: Top_PulseDuration, TopPauseDuration, Btm_PulseDuartion, BtmPauseDuration in [sec]

Operation interlock

A missing enable operation of the ICM has a direct effect on the SLB expansion. The function release of the SLB expansion can be further restricted for each seat lifting with an existing ICM enable operation (internal AND logic operation).

- Configuration: Top_InterlockRef, Btm_InterlockRef
- UDT: xInfoInterlockTop, xInfoInterlockBtm

Double batch reference

The double seat valve, here considered to be an aggregate of main valve and SLB expansion, can be used by two batches in process mode. In case of an error, the batches involved can be identified by the double entries in the message archive and in the change log. For this, the ICM data record was expanded to include a second unit reference. The batch references can be found in the respective units. The following premise applies here:

- There is no batch without unit.

Seat lifting in manual mode

In manual mode, pulse control is completely ignored. The seats can be lifted individually and directly in the "Seat lifting" tab of the ICM user interface.

Seat lifting in automatic mode

In automatic mode, activation is exclusively pulse driven. Activation and origin of the pulses can either be configured as reference or activated in the user program:

<table>
<tr><th>Pulse mode</th><th>Activation</th><th>Pulse supply</th></tr>
<tr><td>PupaMode=Internal</td><td rowspan="2">Configuration:
Top_PulseActivateRef
Btm_PulseActivateRef
UDT:
xCmdActivateTopAut
xCmdActivateBtmAut</td><td>(internal)</td></tr>
<tr><td>PupaMode=External</td><td>Configuration:
Top_PulseSigRef
Btm_PulseSigRef
UDT:
xPulseSigTopAut
xPulseSigBtmAut</td></tr>
</table>

Notes on the call system

The block is disabled by default, because use in the projects cannot be generally assumed.

Note

PCU type S7-400: The user must call the "BmExecSLBFC" FC773" in a suitable user interface FB:

- **"BmUsr1000msFB" FB 1225 → standard case** - a granularity of 1 sec for the pulse times should be sufficient for most applications.
- "BmUsr100msFB" FB 1224 → only for special cases - not suitable when using CPU 414 or CPU 416/417 < Version 07
  Notice: This applies to all active instances of the block, which means it is essential to ensure a sufficient cycle time reserve in time slices "TS01_CycleAct" ... "TS10_CycleAct" (see "ZyklMess" block). For this reason, the number of active instances must be reduced to the minimum number required. In case of doubt, contact SIMATIC Customer Support.
- The time slice selected here must match the sender of the seat lifting signals of the "PULSE" class.

Note

PCU type S7-1500: The user must call the block in a suitable user interface FB:

- **"BmUsrCallIf1000msFC" FC 907 → Standard case**; a granularity of 1 sec for the pulse times should be sufficient for most applications.
- "BmUsrCallIf100msFC" FC 903 → Only for special cases - only recommended when CPU 1518 PN/DP V2.0 or higher is used!
  Notice: This applies to all active instances of the block. For this reason, a sufficient cycle time reserve should be ensured in the time slices "TS01_CycleAct" ... "TS10_CycleAct" (see "ZyklMess" block).
  For this reason, the number of active instances must be reduced to the minimum number required. In case of doubt, contact SIMATIC Customer Support.
- When a reference to external signal outputs of "PULSE" class exists, the time slice selected here must conform to its call cycle.

Example:

- Call syntax STL: `CALL "BmSlbClassFB", "SLB"`
- Call syntax SCL: `"SLB" ();`

See also

ICM - Individual Control Modules (ICM1 ICM2 ICM3 ICM4) (Page 269)

5.7.29 DIS_MSG - PCU message block

General

The block renders it possible to lock out any message initiated by the PCU selectively. The respective message telegrams are then not even entered into the event buffer for the IOS

Parameter sets for block DISABLE / DB 701

PCU parameter set before Version V7 (1 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
MsgLock01	NL	BOOL	ENG	FALSE	Message lock user fault incoming
MsgLock02	NL	BOOL	ENG	FALSE	Message lock user fault outgoing
MsgLock03	NL	BOOL	ENG	FALSE	Message lock user message incoming
MsgLock04	NL	BOOL	ENG	FALSE	Message lock user message outgoing
MsgLock05	NL	BOOL	ENG	FALSE	Message lock ICM1 fault incoming
MsgLock06	NL	BOOL	ENG	FALSE	Message lock ICM1 fault outgoing
MsgLock07	NL	BOOL	ENG	FALSE	Message lock AIN fault incoming
MsgLock08	NL	BOOL	ENG	FALSE	Message lock AIN fault outgoing
MsgLock09	NL	BOOL	ENG	FALSE	Message lock EOP-TUE fault
MsgLock10	NL	BOOL	ENG	FALSE	Message lock SEQS start
MsgLock11	NL	BOOL	ENG	FALSE	Message lock SEQS parametrization error
MsgLock12	NL	BOOL	ENG	FALSE	Message lock PID
MsgLock13	NL	BOOL	ENG	FALSE	Message lock sequence operation messag-es
MsgLock14	NL	BOOL	ENG	FALSE	Message lock ICM2 fault incoming
MsgLock15	NL	BOOL	ENG	FALSE	Message lock ICM2 fault outgoing
MsgLock16	NL	BOOL	ENG	FALSE	Message lock 3-PU
MsgLock17	NL	BOOL	ENG	FALSE	Message lock user warning incoming
MsgLock18	NL	BOOL	ENG	FALSE	Message lock user warning outgoing
MsgLock19	NL	BOOL	ENG	FALSE	Message lock user operator control mes-sage incoming
MsgLock20	NL	BOOL	ENG	FALSE	Message lock user operator control mes-sage outgoing
MsgLock21	NL	BOOL	ENG	FALSE	Message lock ICM3 fault incoming
MsgLock22	NL	BOOL	ENG	FALSE	Message lock ICM3 fault outgoing
MsgLock23	NL	BOOL	ENG	FALSE	Message lock ICM4 fault incoming
MsgLock24	NL	BOOL	ENG	FALSE	Message lock ICM4 fault outgoing

PCU Version V7 parameter set (1 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
MsgLock01	NL	BOOL	ENG	FALSE	Message lock user fault incoming
MsgLock02	NL	BOOL	ENG	FALSE	Message lock user fault outgoing
MsgLock03	NL	BOOL	ENG	FALSE	Message lock user message incoming
MsgLock04	NL	BOOL	ENG	FALSE	Message lock user message outgoing
MsgLock05	NL	BOOL	ENG	FALSE	Message lock ICM1 fault incoming
MsgLock06	NL	BOOL	ENG	FALSE	Message lock ICM1 fault outgoing
MsgLock07	NL	BOOL	ENG	FALSE	Message lock AIN fault incoming
MsgLock08	NL	BOOL	ENG	FALSE	Message lock AIN fault outgoing
MsgLock09	NL	BOOL	ENG	FALSE	Message lock EOP-TUE fault
MsgLock10	NL	BOOL	ENG	FALSE	Message lock SEQS start

Name	Access	D.Type	A.Type	Value	Comment
MsgLock11	NL	BOOL	ENG	FALSE	Message lock SEQS parametrization error
MsgLock12	NL	BOOL	ENG	FALSE	Message lock PID
MsgLock13	NL	BOOL	ENG	FALSE	Message lock sequence operation messag-es
MsgLock14	NL	BOOL	ENG	FALSE	Message lock ICM2 fault incoming
MsgLock15	NL	BOOL	ENG	FALSE	Message lock ICM2 fault outgoing
MsgLock16	NL	BOOL	ENG	FALSE	Message lock THRESTEP
MsgLock17	NL	BOOL	ENG	FALSE	Message lock user warning incoming
MsgLock18	NL	BOOL	ENG	FALSE	Message lock user warning outgoing
MsgLock19	NL	BOOL	ENG	FALSE	Message lock user operator control mes-sage incoming
MsgLock20	NL	BOOL	ENG	FALSE	Message lock user operator control mes-sage outgoing
MsgLock21	NL	BOOL	ENG	FALSE	Message lock ICM3 fault incoming
MsgLock22	NL	BOOL	ENG	FALSE	Message lock ICM3 fault outgoing
MsgLock23	NL	BOOL	ENG	FALSE	Message lock ICM4 fault incoming
MsgLock24	NL	BOOL	ENG	FALSE	Message lock ICM4 fault outgoing
MsgLock25	NL	BOOL	ENG	FALSE	Message lock tank block
MsgLock26	NL	BOOL	ENG	FALSE	Message lock pipe block

As a general rule:

- Parameter: 0/1 = Release/block

5.7.30 TANK - Tank status

5.7.30.1 TANK - Tank status

Area of application and operating principle

This function block manages the tank and container statuses and provides runtime monitoring with interfaces and messages for process and quality status for this purpose, including an operating and monitoring interface. The block can be used for various types of containers, for example silos and buffer tanks.

- The process status allows the handling of up to 255 tank states (10 at delivery) and can be operated using the OS faceplate and / or control inputs.
- The quality state allows the handling of up to 255 quality states (4 at delivery) and can be operated using the OS faceplate and / or control inputs.
- Three separate timers are available for monitoring the times remaining for material quality, tank cleaning and sterility.
- The material is selected in the OS faceplate on the basis of the materials defined in the material configuration dialog.

Parameter sets for block TANK / DB 2100

The instance attributes, such as tank group, tank type, limit value, unit allocation and references to measured values are configured via the parameterization (block class TANK <Tank state>).

Linking in the process data

To simplify the configuration, the tank block has a series of optional references. If a reference is not used, the user program needs to make the runtime data are available. The reference type describes whether the attribute is read ("GET", "OBJECT") or written ("PUT") . If references for measured values (e.g. temperature, pressure, fill level) are interconnected with absolute addresses, remember that in the interconnection dialog the data type is specified that actually exists at the destination (e.g. REAL instead of DINT).

Reference type	Reference attribute	Runtime attribute	Variable in the UDT
PUT	TankStatusOutRef	TankStatusAct	`iTankStatusAct`
GET	FullDetectorRef	FullDetectorSignal	`xFullDetector`
GET	EmptyDetectorRef	EmptyDetectorSignal	`xEmptyDetector`
GET	Tx_ValueRef	Tx_ValueAct Tx_IsPresent	`arrTempValues[1..8]` `arrTempSensorIsPresent[1..8]`
GET	Tx_IsValidRef	Tx_IsValidAct	`arrTempValueIsValid[1..8]`
OBJECT (ICM-Instance)	TankValveRef	n.a.	n.a.
GET	PressureRef	PressureAct	`rPressureAct`
GET	QuantityRef	QuantityAct	`rQuantityAct`
GET	MaterialIDRef	MaterialIDPv	`diMatIDPv`
GET	QuantitySpRef	QuantitySpPv	`rQuantitySpPv`

"x" = index for temperature sensor 1...8

Note

User patching of the measured temperature values

If the user program handles the patching of the temperature measured values, i.e. the configured references Tx_IsValidRef = (NULL), their validity must be signaled explicitly. In the UDT of the tank block the following bits are intended for this:

- `arrTempSensorIsPresent[1..8] : BOOL     // Temperature sensor 1..8 is present`

The result of this is that if temperature sensors do not exist the reference attribute Tx_IsValidRef = (NULL) **and** the user patching bit `arrTempSensorIsPresent[1..8]` = FALSE must be set.

In addition to this, the validity of the temperature measured values (sensor = "wet") must be signaled explicitly. This can be done either using the configured references Tx_IsValidRef <> (NULL) **or** using the following user patching bits:

- `arrTempValueIsValid[1..8] : BOOL       // Temperature value 1..8 is valid`

Parameter set (max. 128 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Sequence	NL	ENUM	ENG, LOG	<0>	Assigned unit
RCSLocation	NL	ENUM	ENG	<0>	RCS location
TankGroup	NL	ENUM	ENG, LOG	TANK-GROUP 1	Tank group
TankType	NL	ENUM	ENG, LOG	TANK-TYPE 1	Tank type
FireInventoryOnChange	NL	BOOL	ENG	TRUE	Dispatch datagram on status change
FireInventoryCyclic	NL	BOOL	ENG	FALSE	Dispatch datagram cyclically
InventoryQuantityHyst	NL	REAL	ENG	0.000	Inventory datagram: Hysteresis for quantity change
InventoryCycleTime	NL	DINT	ENG	60	Inventory datagram: cycle time
TankStatusAct	HR	ENUM	RT, LOG	(Undefined)	Tank status
TankStatusOutRef	HI	PREF	ENG	(NULL)	Target reference for tank status
S7-1500: TankStatusOutRef_ST	HR	WORD	SYS	W#16#0000	Signal status
QualityStatusAct	HR	ENUM	RT, LOG	(Undefined)	Quality status
MaterialAct	HR	DINT	RT, LOG	0	Material ID
Status1	HR	DWORD	RT, LOG	DW#16#00000000	(Internal)
Status2	HR	DWORD	RT, LOG	DW#16#0B000200	(Internal)
Manual	HI	BOOL	RT	FALSE	1=Manual mode, 0=Automatic
MaterialManual	HI	BOOL	RT	FALSE	1=Manual mode for material modification
OpRequest	HI	BOOL	RT	FALSE	Acknowledge operator request
NoteIsActive	HR	BOOL	RT	FALSE	Note is available
QualityTimeMon	HI	BOOL	RT	FALSE	Monitor quality time
CleaningTimeMon	HI	BOOL	RT	FALSE	Monitor cleaning time
SterileTimeMon	HI	BOOL	RT	FALSE	Monitor sterile time
Reset	HI	BOOL	RT	FALSE	1=Reset quantity movements
SetMaterial	HI	BOOL	RT	FALSE	Apply material ID
RstMaterial	HI	BOOL	RT	FALSE	Reset material ID in manual mode
MsgLock	HI	BOOL	RT	FALSE	Message lock for commissioning
FullDetectorRef	NL	GREF	ENG	(NULL)	Full detector
FullDetectorSignal	HI	BOOL	RT	FALSE	Ext. signal (bitmask 1)

Name	Access	D.Type	A.Type	Value	Comment
EmptyDetectorRef	NL	GREF	ENG	(NULL)	Empty detector
S7-1500: EmptyDetectorRef_ST	HR	WORD	SYS	W#16#0000	Signal status
EmptyDetectorSignal	HI	BOOL	RT	FALSE	Ext. signal (bitmask 2)
SetExtInpSignal3	HI	BOOL	RT	FALSE	Ext. signal (bitmask 4)
SetExtInpSignal4	HI	BOOL	RT	FALSE	Ext. signal (bitmask 8)
TankStatusMask1	HI	WORD	ENG	W#16#0001	Bitmask for external TankStatus1
TankStatusMask2	HI	WORD	ENG	W#16#0002	Bitmask for external TankStatus2
TankStatusMask3	HI	WORD	ENG	W#16#0000	Bitmask for external TankStatus3
TankStatusMask4	HI	WORD	ENG	W#16#0000	Bitmask for external TankStatus4
ExtTankStatus1	HI	ENUM	ENG	Filled	Tank status in case Bitmask1 is valid
ExtTankStatus2	HI	ENUM	ENG	Emptied	Tank status in case Bitmask2 is valid
ExtTankStatus3	HI	ENUM	ENG	(Undefined)	Tank status in case Bitmask3 is valid
ExtTankStatus4	HI	ENUM	ENG	(Undefined)	Tank status in case Bitmask4 is valid
MonTankStatus1	NL	ENUM	ENG	Filled	Tank status comparison value 1 (Operator/quality time)
MonTankStatus2	NL	ENUM	ENG	CO2 preloading	Tank status comparison value 2 (Operator/quality time)
ExtQualStatus1	HI	ENUM	ENG	Not tested	Quality state 1 if ExtQualStatus1 is set
ExtQualStatus2	HI	ENUM	ENG	Permitted	Quality state 2 if ExtQualStatus2 is set
ExtQualStatus3	HI	ENUM	ENG	Not permitted	Quality state 3 if ExtQualStatus3 is set
ExtQualStatus4	HI	ENUM	ENG	Irrelevant	Quality state 4 if ExtQualStatus4 is set
SetExtQualStatus1	HI	BOOL	RT	FALSE	Ext. signal for quality status change on ExtQualStatus1
SetExtQualStatus2	HI	BOOL	RT	FALSE	Ext. signal for quality status change on ExtQualStatus2
SetExtQualStatus3	HI	BOOL	RT	FALSE	Ext. signal for quality status change on ExtQualStatus3
SetExtQualStatus4	HI	BOOL	RT	FALSE	Ext. signal for quality status change on ExtQualStatus4
UnQualityStatus	NL	ENUM	ENG	(Undefined)	Quality state after monitoring time expired

Name	Ac-cess	D.Type	A.Type	Value	Comment
CleaningStatus	NL	ENUM	ENG	Clean	Tank status when cleaning time starts
UnCleaningStatus	NL	ENUM	ENG	(Undefined)	Tank status after cleaning time expired
SterileStatus	NL	ENUM	ENG	Sterile	Tank status when sterile time starts
UnSterileStatus	NL	ENUM	ENG	(Undefined)	Tank status after sterile time expired
TemperatureUom	NL	ENUM	ENG, LOG	°C	Unit of measurement for temperature
TemperatureAvg	HR	REAL	RT, LOG	0.000	Average temperature
T1_ValueRef	NL	GREF	ENG	(NULL)	Reference to temperature value
S7-1500: T1_ValueRef_ST	HR	WORD	SYS	W#16#0000	Signal status
T1_ValueAct	HI	REAL	RT, LOG	0.000	Temperature value
T1_IsPresent	NL	BOOL	RT	TRUE	Temperature sensor is available
T1_IsValidRef	NL	GREF	ENG	(NULL)	Reference to condition 'temperature value is valid'
S7-1500: T1_IsValidRef_ST	HR	WORD	SYS	W#16#0000	Signal status
T1_IsValidAct	NL	BOOL	RT	FALSE	Temperature value is valid
...					
T8_ValueRef	HI	GREF	ENG	(NULL)	Reference to temperature value
S7-1500: T8_ValueRef_ST	HR	WORD	SYS	W#16#0000	Signal status
T8_ValueAct	HI	REAL	RT, LOG	0.000	Temperature value
T8_IsPresent	HI	BOOL	RT	FALSE	Temperature sensor is available
T8_IsValidRef	HI	GREF	ENG	(NULL)	Reference to condition 'temperature value is valid'
S7-1500: T8_IsValidRef_ST	HR	WORD	SYS	W#16#0000	Signal status
T8_IsValidAct	HI	BOOL	RT	FALSE	Temperature value is valid
TankValveRef	NL	OREF	ENG	(NULL)	Object reference to tank valve
S7-1500: TankValueRef_ST	HR	WORD	SYS	W#16#0000	Signal status
PressureUom	NL	ENUM	ENG, LOG	mbar	Unit of measurement for pressure
PressureRef	NL	GREF	ENG	(NULL)	Reference to pressure process value
S7-1500: PessureRef_ST	HR	WORD	SYS	W#16#0000	Signal status
PressureAct	HI	REAL	RT, LOG	0.000	Pressure
QuantityUom	NL	ENUM	ENG, LOG	hl	Unit of measurement quantity
QuantityRef	NL	GREF	ENG	(NULL)	Reference to quantity process value

Name	Ac-cess	D.Type	A.Type	Value	Comment
S7-1500: QuantityRef_ST	HR	WORD	SYS	W#16#0000	Signal status
QuantityAct	HI	REAL	RT, LOG	0.000	Quantity
QualityTimeSp	NL	DINT	ENG, LOG	3600	Quality time
CleaningTimeSp	NL	DINT	ENG, LOG	3600	Cleaning time
SterileTimeSp	NL	DINT	ENG, LOG	3600	Sterile time
TankStatusOp	HI	ENUM	RT, LOG	(Undefiniert)	Tank status in manual mode
TankStatusPv	HI	ENUM	RT	(Undefiniert)	Tank status in automatic mode
QualityStatusOp	HI	ENUM	RT, LOG	(Undefiniert)	Quality status in manual mode
MaterialIDOp	HI	DINT	RT, LOG	0	Material ID manual mode
MaterialIDRef	NL	GREF	ENG	(NULL)	Reference to material ID in automatic mode
S7-1500: MaterialIDRef_ST	HR	WORD	SYS	W#16#0000	Signal status
MaterialIDPv	HI	DINT	RT	0	Material ID automatic mode
QuantitySpOp	HI	REAL	RT, LOG	0.000	Quantity change manual mode
QuantitySpRef	NL	GREF	ENG	(NULL)	Reference to quantity change in automatic mode
S7-1500: QuantitySpRef_ST	HR	WORD	SYS	W#16#0000	Signal status
QuantitySpPv	HI	REAL	RT	0.000	Quantity change automatic mode
QuantityTotal	HR	REAL	RT, LOG	0.000	Sum of quantity movements
QuantityMovements	HR	INT	RT, LOG	0	Number of quantity movements
BatchCounter	HI	INT	RT, LOG	0	Batch counter
QualityTimeAct	HR	DINT	RT, LOG	0	Quality time
CleaningTimeAct	HR	DINT	RT, LOG	0	Cleaning time
SterileTimeAct	HR	DINT	RT, LOG	0	Sterile time
QualityRmTimeAct	HR	DINT	RT, LOG	3600	Remaining quality time
CleaningRmTimeAct	HR	DINT	RT, LOG	3600	Remaining cleaning time
SterileRmTimeAct	HR	DINT	RT, LOG	3600	Remaining sterile time

The text parameters of the tank are defined through the following enumerations and are configured at the corresponding block attributes:

Description / Purpose	Text file	Block attribute (type ENUM)
Definition tank name and assignment to the Location ID	...\texte.x\rcs_Loc.txt	`RCSLocation`
Definition of the tank type	...\texte.x\TankType.txt	`TankType`
Definition of the tank group	...\texte.x\TankGroup.txt	`TankGroup`
Assignment to unit	...\PCU.nnn\texte.x\SEQUENCE.txt	`Sequence`
Definition of the tank state, cleaning state	...\texte.x\TankStatus.txt	`ExtTankStatus1 … ExtTankStatus4, MonTankStatus1, MonTankstatus2, CleaningStatus, UnCleaningStatus, SterileSt, UnSterileStatus, TankStatusOp, TankStatusPv`
Definition of the quality status	...\texte.x\QualityStatus.txt	`ExtQualStatus1 … ExtQualStatus4, UnQualityStatus, QualityStatusOp`
Definition of the units (UnitOfMeasurement)	...\texte.x\TankUnits.txt	`TemperatureUnit, PressureUnit, QuantityUnit`

5.7.30.2 TANK - functions

Basics

The input/output signals described below are available essentially for display in the TankCtrl OCX (see also section TankCtrl Control (Page 497)):

Block attribute	Function	Application

`FullDetectorRef`
`FullDetectorSignal`
`EmptyDetectorRef`
`EmptyDetectorSignal`
	Display of full and empty detectors
	TankCtrl OCX, 'General' tab
		Interconnection to relevant sensor or patching from the user program

`Tx_ValueRef`
`Tx_ValueAct`
`Tx_IsPresent`
`Tx_IsValidRef`
`Tx_IsValidAct`
	Display of up to 8 temperature sensors
	TankCtrl OCX, 'General' tab
		Interconnection to relevant sensor or patching from the user program. Values are only displayed for active and valid/wetted temperature sensors.

`TemperatureUoM`		Unit parameter assignment

Block attribute	Function	Application
TemperatureAvg	Display of the average temperature of the valid/wet temperature sensors TankCtrl OCX, 'General' tab	calculated by the block
PressureRef PressureAct	Indication of the pressure in the tank TankCtrl OCX, 'General' tab	Interconnection to pressure sensor or patching from the user program
PressureUom		Unit parameter assignment
QuantityRef QuantityAct	Indication of the fill quantity in the tank TankCtrl OCX, 'General' tab	Interconnection to sensor or patching from the user program
QuantityUom		Unit parameter assignment
TankValveRef	Display of the outlet valve icon TankCtr OCX, 'General' tab	If there is an interconnection to the ICM instance, the corresponding ICM icon is shown here
BatchCounter	Display of the number of batches in the tank TankCtrl OCX, 'Batch' tab	The block parameter must be set by the user program

Tank status output TankStatusAct, TankStatusOutRef

The Tank block provides a tank status management with different statuses(default = 13) at output TankStatusAct. With the corresponding "PUT" reference, the status can also be written actively to any destination parameter. The state definitions are contained in the enumeration "TankStatus". The following states are defined and preset here:

Table 5-7 Preset content in the text file 'TankStatus.txt'

Value	Display name
0	(Undefined)
1	Clean
2	Sterile
3	Fill
4	Filled
5	CO2 preloading
6	Empty
7	CO2 preloaded
8	Emptied
9	CIP
10	Ready for filling
11	Ready for emptying
12	Ready for cleaning
...	...

Process status, activation

The process status can be activated in the following ways:

- In manual mode via the OS faceplate
- In automatic mode, by assigning various block attributes in the user program

AUTO mode and `TankStatusPv > 0`:

The setpoint `TankStatusPv` is transferred to the process status `TankStatusAct` .

AUTO mode and `TankStatusPv = 0`:

If the setpoint `TankStatusPv = 0` (undefined), the output `TankStatusAct` retains the last valid value and the process status can be influenced by the four following block attributes. These four signal bits can be enabled by the user program, but the following preferred allocation should be observed:

- `FullDetector<Ref/Signal>` = Tank HighLevel probe
- `EmptyDetector<Ref/Signal>` = Tank LowLevel probe
- `SetExtInExtInpSig3` = Freely interconnectable
- `SetExtInExtInpSig4` = Freely interconnectable

To evaluate these signal bits, the following four bit masks of type INT are available, in which only bits 0 - 3 are evaluated. The signal bits are linked with the 4 bit masks as follows logically AND

Mask \ Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TankStatusMask1	-	-	-	-	-	-	-	-	-	-	-	-	1/0	1/0	1/0	1/0
TankStatusMask2	-	-	-	-	-	-	-	-	-	-	-	-	1/0	1/0	1/0	1/0
TankStatusMask3	-	-	-	-	-	-	-	-	-	-	-	-	1/0	1/0	1/0	1/0
TankStatusMask4	-	-	-	-	-	-	-	-	-	-	-	-	1/0	1/0	1/0	1/0
Signal bit													ExtInpSig4	ExtInpSig3	EmptyDet.	FullDet.

(The mask bits 4 to 15 are not evaluated)

If the status of the 4 input signals corresponds to one of the bit masks the status set at the assigned input `ExtTankStatus1 … ExtTankStatus4` is adopted as `TankStatusAct`. If more than one event is true, the status with the highest priority at the output `TankStatusAct` is imported:

- `ExtTankStatus4` = highest priority
- …
- `ExtTankStatus1` = lowest priority

To ensure a bumpless automatic / manual switchover, the `TankStatusOp` control value is tracked to the current status in automatic mode.

Tank status, runtime monitoring "Clean" and "Sterile"

Two monitoring timers (cleanliness and sterility timer) are available for influencing the tank status. The following conditions apply for the handling of the two timers:

- The timer setpoints `CleaningTimeSp` and `SterileTimeSp` can be specified in the OS faceplate in the format (Number of days, hr, min) and the `CleaningTimeMon` and `SterileTimeMon` enables can be set.
- The timer remaining times `CleaningRmTimeAct` and `SterileRmTimeAct` can be followed at the OS faceplate.
- Start condition "Cleaning timer":
  If the tank status `TankStatusAct` corresponds to the value set for `CleaningStatus` (default = 1 = "Clean") and the cleaning timer is enabled (`CleaningTimeMon =1`), it is started. The timer is reset automatically when the starting condition is no longer fulfilled.
- Start condition "Sterile timer":
  If `TankStatusAct` corresponds to the value set for `SterileStatus` (default = 2 = "Sterile") and the sterile timer has been enabled (`SterileTimeMon =1`), it is started. The timer is reset automatically when the starting condition is no longer fulfilled.
- After the configured time has expired, the respective status is set to the value that is configured at the input parameters `UnCleaningStatus` or `UnSterileStatus` .
- If messages are enabled `MsgLock=1`, an alarm message is generated when one of the timers `CleaningTimexx` or `SterileTimexx` elapses or the tank status `TankStatusAct` changes.

Quality state, output **`QualityStatusAct`**

The block provides a quality status management with different states (default = 5). The state definitions are contained in the enumeration "`QualityStatus`". The following states are defined and preset here:

Value	Display name
0	Undefined
1	Not_tested
2	Permitted
3	Not_approved
4	Irrelevant
...	...

The quality status can be influenced both by the OS faceplate and the block inputs (no preferred operating mode). The operator can set the quality status in manual operation by selecting a value from the list in the OS faceplate. If the edge of the control input `InExtQualStatus1 … 4` is positive at the block in the AS, the correspondingly configured parameter value of the assigned input `ExtQualStatus1 … 4` is imported. In both cases the corresponding status is displayed at the output `QualityStatusAct` .

Quality status, operator prompt

If the current process status at the output `TankStatusAct` matches one of the parameter inputs `MonTankStatus1` or `MonTankStatus2` , an operator prompt message "Take sample" is initiated and the quality status at the output `QualityStatusAct` is set to "1" (in this case: "Not_tested"). The operator prompt is displayed at the output `OpRequest=1` and is reset when the quality status "1" is exited.

Quality state, runtime monitoring "permitted"

- The timer setpoint `QualityTimeSp` can be specified in the OS faceplate in the format (Number of days, hr, min) and the `QualityTimeMon` enable can be set.
- If the quality status "2" (in this case: "Permitted") is selected and the current process status of the tank corresponds to one of the set values `MonTankStatus1` or `MonTankStatus2`, the quality timer is started.
- The timer remaining time `QualityRmTimeAct` can be followed at the OS faceplate.
- After the configured time has expired, the quality status at the output `QualityStateAct` is set to the value `UnQualityStatus` (default = 0 = "Undefined").
- The timer is reset automatically when one of the status conditions for starting the timer is no longer fulfilled.
- If messages are enabled `MsgLock=1`, an alarm message is generated when the timer `QualityTimexx` elapses or the quality status `QualityStateAct` changes.

Material movements:

The material current transferred or held in the tank, including a quantity, can be addressed as follows:

- In manual mode via the OS faceplate
- In automatic mode, by assigning various block attributes in the user program
- The auto/manual switchover is performed in the OS faceplate.
- Material correction quantity in automatic mode (`MaterialManual = 0`):
  - If the attribute `SetMaterial` is set (pos. signal edge), the value `MaterialIDPv` is transferred to the output `MaterialAct` and displayed in the faceplate.
  - The quantity `QuantitySPPv` is added to the totalizer `QuantityTotal`. The counter `QuantityMovements` is increased by 1. Instead of addressing the direct block attributes in the user program, the respective reference attributes `MaterialIDRef` and `QuantitySPRef` can also be connected.

- Material correction quantity manual mode (`MaterialManual = 1`):
  In the OS faceplate, a manual material correction can be entered for the tank. The material is selected from a list box. A correction quantity can also be specified. The material is applied with the save button, the correction quantity is added to the `QuantityTotal` parameter and the `QuantityMovements` parameter is incremented by 1. As long as the data has not yet been saved for manual entry, the last material input value can be restored again with the "Back" button (`RstMaterial` parameter).
- Reset via `Reset` input
  With the positive edge, the `QuantityTotal` and `QuantityMovements` parameters are reset.

Tank inventory - Sending inventory telegrams to IOS

The inventory data of the TANK instances used in the project can be transferred cyclically and event-driven to the IOS servers for different purposes. This online data is kept in the memory as "Tankinventory" and can be used by the following functions:

- Order system → Here you can select tanks according to different filter criteria and transferred as batch parameters (see section 'Tank selection' dialog (Page 744))
- "SQL Adapter" option → with enabled "tank history", the engineering data as well as the runtime data of the tank inventory are transferred to the respective SQL tables.

All active TANK instances are transferred completely during the PCU server startup.

The transfer takes place according to the inventory telegrams configured in the TANK instances. The following trigger options can be configured for each instance:

- `FireInventoryCyclic = true / false` → enable cyclic transfer
  `InventoryCycleTime = <...>` → cyclic transfer / configurable cycle time
- `FireInventoryOnChange = true / false` → status changes (tank status, quality status, material) and ...
  `InventoryQuantityHyst = <...>` → change quantity by means of configurable hysteresis
- `'TANK'.u.xFireInventoryNow = false /true` → application-based triggering (tank instance is derived from configurable assignment in the `'Sequence'` attribute)

NOTICE

Configuring the inventory telegrams

- According to the setting of these attributes, the inventory telegrams are sent to the SQL DB host and entered in the "`sistar_rt_tank_history`" table.
- They also determine the update rates in the TANK selection dialog of the BRAUMAT order system for "Tank" or "Tank location" parameter type (PCU as of V7).
- The user/configuration engineer has to pay special attention to a sensible and necessary number of update telegrams during configuration, especially for the attributes "`InventoryQuantityHyst`", "`FireInventoryOnChange`" and "`InventoryCycleTime`".
- An incorrect configuration or one that has not been adjusted, can increase the data volume in the SQL table significantly and have a negative effect on the entire coupling performance.
- **The telegram type 22 used for the "Tank inventory telegrams" must also be entered in the system configuration / PCU settings / FIFOs / FIFO 1+4!**

5.7.31 TIMER - switch-on delay/impulse

General

This block makes 2 x 512 additional timers available that can be used as on/off delay or pulse.

The start input and the two outputs for positive and negative pulse to the flag interface are routed according to the tables further below in this section.

Note

Activation of timer group 2 (only necessary for PCU < V7)

Group 2 timers are disabled as default in earlier PCU versions. In the project supplied for the PCU V6, although the corresponding parameter assignment is already included in the "SCHEDULE / DB720" block, processing of the block for "TIMER02 / DB745" needs to be enabled.

- Set the "Disabled" flag of the corresponding SCHEDULE datasets to "FALSE".
  TIMER_02 → dataset 58, 59 and 60: Parameter Disable = "FALSE"
- Please ensure that all three data sets have been enabled.

Parameter sets for the TIMER1, 2 / DB 724, 745 block

PCU Version before V7 parameter set (max. 2 x 512 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
TimeValPos	NL	INT	ENG	5	Positive pulse: Time in seconds
TimeValNeg	NL	INT	ENG	5	Negative pulse: Time in seconds
TimeTyp	NL	BOOL	ENG	FALSE	TIMER 0=SE, 1=SI
In	NL	BOOL	RT	FALSE	Start input
OutPos	NL	BOOL	RT	FALSE	Output pos. impulse
OutNeg	NL	BOOL	RT	TRUE	Output neg. impulse
NoteIsActive	HR	BOOL	RT	FALSE	Note is available
TimeVal	NL	INT	RT	0	Current time value
TypFlag	HR	BOOL	SYS	FALSE	Copy of type
Sim	HI	BOOL	ENG	FALSE	Simulate input
InvNegOut	HI	BOOL	ENG	FALSE	Invert negative output
InvPosOut	HI	BOOL	ENG	FALSE	Invert positive output
STAT	HR	WORD	RT, LOG	W#16#1800	Status

PCU Version V7 parameter set (max. 2 x 512 per PCU)

Common attributes

Name	Access	D.Type	A.Type	Value	Comment
TimerType	NL	ENUM	ENG	...	Timer type
... Attributes according to TimerType=0/1 see below ...					
In	NL	BOOL	RT	FALSE	Start input
NoteIsActive	HR	BOOL	RT	FALSE	Note is available
TimeVal	NL	INT	RT	0	Actual time value
Sim	HI	BOOL	ENG	FALSE	Simulate input
InvNegOut	HI	BOOL	ENG	FALSE	Invert negative output
InvPosOut	HI	BOOL	ENG	FALSE	Invert positive output
STAT	HR	WORD	RT, LOG	W#16 #580E	Status
RefIn	NL	GREF	ENG	NULL	Reference to start input
S7-1500: RefIn_ST	HR	WORD	SYS	W#16#00 00	Signal status
RefOutPos	NL	PREF	ENG	NULL	Reference to positive output
S7-1500: RefOutPos_ST	HR	WORD	SYS	W#16#00 00	Signal status
RefOutNeg	NL	PREF	ENG	NULL	Reference to negative output
S7-1500: RefOutNeg_ST	HR	WORD	SYS	W#16#00 00	Signal status

Attributes of the variant "On/off delay" (timer type = 0)

Name	Access	D.Type	A.Type	Value	Comment
DelayPos	NL	INT	ENG	0	Delay time for positive output signal [sec]
DelayNeg	NL	INT	ENG	0	Delay time for negative output signal [sec]
OutPosDe-layed	RD	BOOL	RT	0	Delayed on output
OutNegDe-layed	RD	BOOL	RT	0	Delayed off output

Attributes of the variant "Pulse" (timer type = 1)

Name	Access	D.Type	A.Type	Value	Comment
PulsePos	NL	INT	ENG	0	Positive pulse
PulseNeg	NL	INT	ENG	0	Negative pulse
OutPulsePos	RD	BOOL	RT	0	Positive output pulse
OutPulseNeg	RD	BOOL	RT	0	Negative output pulse

Note

Features of V7-PCU

- Alternatively, the digital input and the two digital output signals can be connected via references (A.Type = GREF/PREF).
- If the respective references are interconnected, standard routing to flags no longer takes place.

Flag User interface to block TIMER 1

Table 5-8 Timer1 - 1 ...256

Bit address								In	OutPos	OutNeg
0	1	2	3	4	5	6	7	(MB)	(MB)	(MB)
1	2	3	4	5	6	7	8	1240	1304	1368
9	10	11	12	13	14	15	16	1241	1305	1369
17	18	19	20	21	22	23	24	1242	1306	1370
25	26	27	28	29	30	31	32	1243	1307	1371
33	34	35	36	37	38	39	40	1244	1308	1372
41	42	43	44	45	46	47	48	1245	1309	1373
49	50	51	52	53	54	55	56	1246	1310	1374
57	58	59	60	61	62	63	64	1247	1311	1375
65	66	67	68	69	70	71	72	1248	1312	1376
73	74	75	76	77	78	79	80	1249	1313	1377
81	82	83	84	85	86	87	88	1250	1314	1378
89	90	91	92	93	94	95	96	1251	1315	1379

Bit address								In	OutPos	OutNeg
97	98	99	100	101	102	103	104	1252	1316	1380
105	106	107	108	109	110	111	112	1253	1317	1381
113	114	115	116	117	118	119	120	1254	1318	1382
121	122	123	124	125	126	127	128	1255	1319	1383
129	130	131	132	133	134	135	136	1256	1320	1384
137	138	139	140	141	142	143	144	1257	1321	1385
145	146	147	148	149	150	151	152	1258	1322	1386
153	154	155	156	157	158	159	160	1259	1323	1387
161	162	163	164	165	166	167	168	1260	1324	1388
169	170	171	172	173	174	175	176	1261	1325	1389
177	178	179	180	181	182	183	184	1262	1326	1390
185	186	187	188	189	190	191	192	1263	1327	1391
193	194	195	196	197	198	199	200	1264	1328	1392
201	202	203	204	205	206	207	208	1265	1329	1393
209	210	211	212	213	214	215	216	1266	1330	1394
217	218	219	220	221	222	223	224	1267	1331	1395
225	226	227	228	229	230	231	232	1268	1332	1396
233	234	235	236	237	238	239	240	1269	1333	1397
241	242	243	244	245	246	247	248	1270	1334	1398
249	250	251	252	253	254	255	256	1271	1335	1399

Table 5-9 Timer1 - 257 ...512

Bit address								In	OutPos	OutNeg
0	1	2	3	4	5	6	7	(MB)	(MB)	(MB)
257	258	259	260	261	262	263	264	1272	1336	1400
265	266	267	268	269	270	271	272	1273	1337	1401
273	274	275	276	277	278	279	280	1274	1338	1402
281	282	283	284	285	286	287	288	1275	1339	1403
289	290	291	292	293	294	295	296	1276	1340	1404
297	298	299	300	301	302	303	304	1277	1341	1405
305	306	307	308	309	310	311	312	1278	1342	1406
313	314	315	316	317	318	319	320	1279	1343	1407
321	322	323	324	325	326	327	328	1280	1344	1408
329	330	331	332	333	334	335	336	1281	1345	1409
337	338	339	340	341	342	343	344	1282	1346	1410
345	346	347	348	349	350	351	352	1283	1347	1411
353	354	355	356	357	358	359	360	1284	1348	1412
361	362	363	364	365	366	367	368	1285	1349	1413
369	370	371	372	373	374	375	376	1286	1350	1414
377	378	379	380	381	382	383	384	1287	1351	1415
385	386	387	388	389	390	391	392	1288	1352	1416

Bit address								In	OutPos	OutNeg
393	394	395	396	397	398	399	400	1289	1353	1417
401	402	403	404	405	406	407	408	1290	1354	1418
409	410	411	412	413	414	415	416	1291	1355	1419
417	418	419	420	421	422	423	424	1292	1356	1420
425	426	426	428	429	430	431	432	1293	1357	1421
433	434	435	436	437	438	439	440	1294	1358	1422
441	442	443	444	445	446	447	448	1295	1359	1423
449	450	451	452	453	454	455	456	1296	1360	1424
457	458	459	460	461	462	463	464	1297	1361	1425
465	466	467	468	469	470	471	472	1298	1362	1426
473	474	475	476	477	478	479	480	1299	1363	1427
481	482	483	484	485	486	487	488	1300	1364	1428
489	490	491	492	493	494	495	496	1301	1365	1429
497	498	499	500	501	502	503	504	1302	1366	1430
505	506	507	508	509	510	511	512	1303	1367	1431

Flag User interface to block TIMER 2

Table 5-10 Timer2 - 1 ... 256

Bit address								In	OutPos	OutNeg
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**(MB)**	**(MB)**	**(MB)**
1	2	3	4	5	6	7	8	1432	1496	1560
9	10	11	12	13	14	15	16	1433	1497	1561
17	18	19	20	21	22	23	24	1434	1498	1562
25	26	27	28	29	30	31	32	1435	1499	1563
33	34	35	36	37	38	39	40	1436	1500	1564
41	42	43	44	45	46	47	48	1437	1501	1565
49	50	51	52	53	54	55	56	1438	1502	1566
57	58	59	60	61	62	63	64	1439	1503	1567
65	66	67	68	69	70	71	72	1440	1504	1568
73	74	75	76	77	78	79	80	1441	1505	1569
81	82	83	84	85	86	87	88	1442	1506	1570
89	90	91	92	93	94	95	96	1443	1507	1571
97	98	99	100	101	102	103	104	1444	1508	1572
105	106	107	108	109	110	111	112	1445	1509	1573
113	114	115	116	117	118	119	120	1446	1510	1574
121	122	123	124	125	126	127	128	1447	1511	1575
129	130	131	132	133	134	135	136	1448	1512	1576
137	138	139	140	141	142	143	144	1449	1513	1577
145	146	147	148	149	150	151	152	1450	1514	1578
153	154	155	156	157	158	159	160	1451	1515	1579
161	162	163	164	165	166	167	168	1452	1516	1580

Bit address								In	OutPos	OutNeg
169	170	171	172	173	174	175	176	1453	1517	1581
177	178	179	180	181	182	183	184	1454	1518	1582
185	186	187	188	189	190	191	192	1455	1519	1583
193	194	195	196	197	198	199	200	1456	1520	1584
201	202	203	204	205	206	207	208	1457	1521	1585
209	210	211	212	213	214	215	216	1458	1522	1586
217	218	219	220	221	222	223	224	1459	1523	1587
225	226	227	228	229	230	231	232	1460	1524	1588
233	234	235	236	237	238	239	240	1461	1525	1589
241	242	243	244	245	246	247	248	1462	1526	1590
249	250	251	252	253	254	255	256	1463	1527	1591

Table 5-11 Timer2 - 257 ... 512

Bit address								In	OutPos	OutNeg
0	1	2	3	4	5	6	7	(MB)	(MB)	(MB)
257	258	259	260	261	262	263	264	1464	1528	1592
265	266	267	268	269	270	271	272	1465	1529	1593
273	274	275	276	277	278	279	280	1466	1530	1594
281	282	283	284	285	286	287	288	1467	1531	1595
289	290	291	292	293	294	295	296	1468	1532	1596
297	298	299	300	301	302	303	304	1469	1533	1597
305	306	307	308	309	310	311	312	1470	1534	1598
313	314	315	316	317	318	319	320	1471	1535	1599
321	322	323	324	325	326	327	328	1472	1536	1600
329	330	331	332	333	334	335	336	1473	1537	1601
337	338	339	340	341	342	343	344	1474	1538	1602
345	346	347	348	349	350	351	352	1475	1539	1603
353	354	355	356	357	358	359	360	1476	1540	1604
361	362	363	364	365	366	367	368	1477	1541	1605
369	370	371	372	373	374	375	376	1478	1542	1606
377	378	379	380	381	382	383	384	1479	1543	1607
385	386	387	388	389	390	391	392	1480	1544	1608
393	394	395	396	397	398	399	400	1481	1545	1609
401	402	403	404	405	406	407	408	1482	1546	1610
409	410	411	412	413	414	415	416	1483	1547	1611
417	418	419	420	421	422	423	424	1484	1548	1612
425	426	426	428	429	430	431	432	1485	1549	1613
433	434	435	436	437	438	439	440	1486	1550	1614
441	442	443	444	445	446	447	448	1487	1551	1615
449	450	451	452	453	454	455	456	1488	1552	1616
457	458	459	460	461	462	463	464	1489	1553	1617

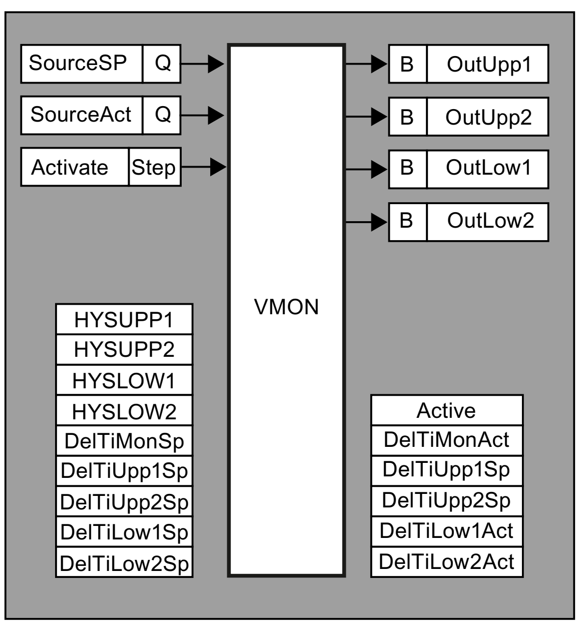
Bit address								In	OutPos	OutNeg
465	466	467	468	469	470	471	472	1490	1554	1618
473	474	475	476	477	478	479	480	1491	1555	1619
481	482	483	484	485	486	487	488	1492	1556	1620
489	490	491	492	493	494	495	496	1493	1557	1621
497	498	499	500	501	502	503	504	1494	1558	1622
505	506	507	508	509	510	511	512	1495	1559	1623

5.7.32 VMON – Value monitoring

Basics

The function block VMON (value monitoring) can carry out up to 256 comparisons (setpoint and actual value or actual value and limit value). The user can define two offset values for the upper limit and two offset values for the lower limit.

The comparative target and actual values are taken by combination from another element (FIXV, AIN, MULT, PID, POLY, SEQU).



Per actual value, two hysteresis boundaries can be defined for the upper and lower band. For these, a delay time (in sec) can be parameterized in each case. For the activation of the check, a delay time (sec) keeps on being possible.

The ACTIVATE input signal releases the value monitoring and the monitoring delay DelTiMonSp, DelTiMonAct is started.

When the activation time has elapsed and the actual value has crossed a boundary value for the parameterized period, the corresponding flag bit (DelTixxxAct) is set in the data record. The user may evaluate these bits.

Depending on the input signal "NoSpVal", either an actual/setpoint or actual/limit comparison is performed, which executes the following formulas:

Mode 1: Actual/setpoint comparison (NoSpVal = FALSE):

If (SourceAct > SourceSP + HysUpp1) and delay time expired → OutUpp1 = TRUE

If (SourceAct > SourceSP + HysUpp2) and delay time expired → OutUpp2 = TRUE

If (SourceAct < SourceSP + HysLow1) and delay time expired → OutLow1 = TRUE

If (SourceAct < SourceSP + HysLow2) and delay time expired → Outlow2 = TRUE

Mode 2: Actual/limit value comparison (NoSpVal = TRUE):

If (SourceAct > HysUpp1) and delay time expired → OutUpp1 = TRUE

If (SourceAct > HysUpp2) and delay time expired → OutUpp2 = TRUE

If (SourceAct < HysLow1) and delay time expired → OutLow1 = TRUE

If (SourceAct < HysLow2) and delay time expired → OutLow2 = TRUE

Activation of VMON:

Note

Activation of VMON for PCU type S7-1500

When PCU type S7-1500 is used, **no activation of the VMON class** is required because the call already takes place in the AS runtime system.

For PCU type S7-400, the block is not activated by default.

- To activate it, the VMON block (FB750) along with its instance data block (DB750) must be called in the Scheduler.
- It can be activated using the parameter assignment of the class "Schedule <Schedule List>" shown in the following tables.
- The call should be in time slice 9 or 10.

The following section describes the parameter assignment for the particular PCU version:

Table 5-12 Example PCU < V7 / time disc=9 → parameter set "SCHEDULE / DB720 <54>"

Name	D.Type	Value	Comment
TimeSlice	USINT	9	Time slice no.
Disable	BOOL	FALSE	Delete dataset **Notice:** Set Disable = FALSE - **as the last step after the other parameters**
IsFC	BOOL	FALSE	Call FC (function), not FB → enter with diagnostics tool "DB Editor" in DB720/ DBX551.1

Name	D.Type	Value	Comment
FB_FC	INT	750	FB/FC no. → enter with diagnostics tool "DB Editor" in DB720/ DBW552
Datablock	INT	750	DB with datasets, 0=no DB → enter with diagnostics tool "DB Editor" in DB720/ DBW554
Parameter	WORD	W#16#0000	Optional parameters → not used with VMON
UsedTime	INT	0	Indication of the current runtime (ms)

Table 5-13 Example PCU V7 / time disc=9 → parameter set "SCHEDULE / DB720 <135>"

Name	D.Type	Value	Comment
TimeSlice	USINT	9	Time slice number
ExecType	ENUM	Tec-FB, u-copy	Execution type **Notice:** Set type **<>** (disable) **only as the last step after the other parameters**
Function	INT	750	Function number
DataBlockFirst	INT	750	(Instance) data block
DataBlockLast	INT	0	(Instance) data block
Parameter	WORD	W#16#0000	Optional parameter
UsedTime	INT	0	Time used (ms)

With that configuration, FB750 is called within 1 sec. cycle.

Parameter sets for block VOMN / DB 750

Table 5-14 PCU parameter set before Version V7 (max. 256 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Setp	NL	INT	ENG	107	Setpoint low word
Setp_DINT	NL	DINT	ENG	107	Setpoint double integer
Actual	NL	INT	RT	107	Actual value low word
Actual_DINT	NL	DINT	RT	107	Actual value double integer
HYSUPP1	NL	INT	ENG	10	Hysteresis upper limit 1
HYSUPP2	NL	INT	ENG	20	Hysteresis upper limit 2
HYSLOW1	NL	INT	ENG	10	Hysteresis lower limit 1
HYSLOW2	NL	INT	ENG	20	Hysteresis lower limit 2
DelTiMonSp	NL	INT	ENG	30	Target value delay time for monitoring
DelTiMonAct	NL	INT	RT	30	Actual value delay time for monitoring
DelTiUpp1Sp	NL	INT	ENG	10	Target value delay time for upper limit 1
DelTiUpp1Act	NL	INT	RT	0	Actual value delay time for upper limit 1
DelTiUpp2Sp	NL	INT	ENG	10	Target value delay time for upper limit 2

Name	Access	D.Type	A.Type	Value	Comment
DelTiUpp2Act	NL	INT	RT	0	Actual value delay time for upper limit 2
DelTiLow1Sp	NL	INT	ENG	10	Target value delay time for lower limit 1
DelTiLow1Act	NL	INT	RT	0	Actual value delay time for lower limit 1
DelTiLow2Sp	NL	INT	ENG	10	Target value delay time for lower limit 2
DelTiLow2Act	NL	INT	RT	0	Actual value delay time for lower limit 2
OutUpp1	NL	BOOL	RT	FALSE	Actual value > Upper limit 1
OutUpp2	NL	BOOL	RT	FALSE	Actual value > Upper limit 2
OutLow1	NL	BOOL	RT	FALSE	Actual value < Lower limit 1
OutLow2	NL	BOOL	RT	FALSE	Actual value < Lower limit 2
Active	NL	BOOL	RT	TRUE	Monitoring enabled
SourceSP	NL	GREF	ENG	ANA, 1,ANA	Setpoint source
SourceAct	NL	GREF	ENG	ANA, 1,ANA	Source of the actual value
Activate	NL	STEP	ENG	U M102.0	Enable monitoring
NoSpVal	NL	BOOL	ENG	FALSE	Compare actual value to: 0=Setpoint, 1=Limit
Error	HI	BOOL	RT	FALSE	Parametrization error
ErrCode	HI	INT	RT	0	Error number

PCU Version V7 parameter set (max. 256 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
Setpoint	NL	REAL	ENG	0.000	Setpoint
ActualValue	NL	REAL	RT	0.000	Actual value
HYSUPP1	NL	REAL	ENG	10.000	Hysteresis upper limit 1
HYSUPP2	NL	REAL	ENG	20.000	Hysteresis upper limit 2
HYSLOW1	NL	REAL	ENG	10.000	Hysteresis lower limit 1
HYSLOW2	NL	REAL	ENG	20.000	Hysteresis lower limit 2
DelTiMonSp	NL	INT	ENG	30	Target value delay time for monitoring
DelTiMonAct	NL	INT	RT	30	Actual value delay time for monitoring
DelTiUpp1Sp	NL	INT	ENG	10	Target value delay time for upper limit 1
DelTiUpp1Act	NL	INT	RT	10	Actual value delay time for upper limit 1
DelTiUpp2Sp	NL	INT	ENG	10	Target value delay time for upper limit 2
DelTiUpp2Act	NL	INT	RT	10	Actual value delay time for upper limit 2
DelTiLow1Sp	NL	INT	ENG	10	Target value delay time for lower limit 1
DelTiLow1Act	NL	INT	RT	10	Actual value delay time for lower limit 1
DelTiLow2Sp	NL	INT	ENG	10	Target value delay time for lower limit 2
DelTiLow2Act	NL	INT	RT	10	Actual value delay time for lower limit 2
OutUpp1	NL	BOOL	RT	FALSE	Actual value > Upper limit 1
OutUpp2	NL	BOOL	RT	FALSE	Actual value > Upper limit 2
OutLow1	NL	BOOL	RT	FALSE	Actual value < Lower limit 1
OutLow2	NL	BOOL	RT	FALSE	Actual value < Lower limit 2

Name	Access	D.Type	A.Type	Value	Comment
Active	NL	BOOL	RT	FALSE	Monitoring enabled
SourceSP	NL	GREF	ENG	(NULL)	Setpoint source
S7-1500: SourceSP_ST	HR	WORD	SYS	W#16#0 000	Signal status
SourceAV	NL	GREF	ENG	(NULL)	Source of the actual value
S7-1500: SourceAV_ST	HR	WORD	SYS	W#16#0 000	Signal status
Activate	NL	STEP	ENG	(NULL)	Enable monitoring
S7-1500: Activate_ST	HR	WORD	SYS	W#16#0 000	Signal status
NoSpVal	NL	ENUM	ENG	Setpoint	Compare actual value to: 0=Setpoint, 1=Limit
Error	HI	BOOL	RT	TRUE	Parametrization error
ErrCode	HI	INT	RT	-1	Error number

5.7.33 ZYKLMESS - Measuring of cycle time

General

With this block, the time values of the OB1 cycle as well as for the distinct time slices of the time slice distributor are entered in OB35.

On enabling the block, measuring continues as long as the block is disabled again.

Note

Operation during the configuration phase under "PLCSim"

The measurement of the CPU cycle times and time slice loads under "PLCSim" leads to false results.

Measured values must always, therefore, be evaluated on the basis of real AS HW.

Parameter sets for block CyclMESS / DB 700

PCU parameter set (1 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
EnableMeasuring	NL	BOOL	ENG	TRUE	Measure: 1=Enable; 0=Disable
ResetMaxValues	NL	BOOL	RT	FALSE	Reset
OB1CycleAct	RD	INT	SYS	3	OB1 current cycle time
OB1CycleAvg	HI	INT	SYS	3	OB1 average cycle time
OB1CycleMax	HI	INT	SYS	17	OB1 maximum cycle time

Name	Access	D.Type	A.Type	Value	Comment
TS01_Overflow	HI	BOOL	SYS	TRUE	Display time disk overrun ZS1
TS02_Overflow	HI	BOOL	SYS	FALSE	Display time disk overrun ZS2
TS03_Overflow	HI	BOOL	SYS	FALSE	Display time disk overrun ZS3
TS04_Overflow	HI	BOOL	SYS	FALSE	Display time disk overrun ZS4
TS05_Overflow	HI	BOOL	SYS	FALSE	Display time disk overrun ZS5
TS06_Overflow	HI	BOOL	SYS	FALSE	Display bit time disk overrun ZS6
TS07_Overflow	HI	BOOL	SYS	FALSE	Display bit time disk overrun ZS7
TS08_Overflow	HI	BOOL	SYS	TRUE	Display bit time disk overrun ZS8
TS09_Overflow	HI	BOOL	SYS	FALSE	Display bit time disk overrun ZS9
TS10_Overflow	HI	BOOL	SYS	FALSE	Display bit time disk overrun ZS10
TS01_CycleAct	RD	INT	SYS	20	OB35 Time disc 1 current cycle time
TS02_CycleAct	RD	INT	SYS	12	OB35 Time disc 2 current cycle time
TS03_CycleAct	RD	INT	SYS	14	OB35 Time disc 3 current cycle time
TS04_CycleAct	RD	INT	SYS	36	OB35 Time disc 4 current cycle time
TS05_CycleAct	RD	INT	SYS	15	OB35 Time disc 5 current cycle time
TS06_CycleAct	RD	INT	SYS	6	OB35 Time disc 6 current cycle time
TS07_CycleAct	RD	INT	SYS	18	OB35 Time disc 7 current cycle time
TS08_CycleAct	RD	INT	SYS	12	OB35 Time disc 8 current cycle time
TS09_CycleAct	RD	INT	SYS	12	OB35 Time disc 9 current cycle time
TS10_CycleAct	RD	INT	SYS	25	OB35 Time disc 10 current cycle time
TS01_CycleMax	HI	INT	SYS	78	OB35 Time disc 1 maximum cycle time
TS02_CycleMax	HI	INT	SYS	20	OB35 Time disc 2 maximum cycle time
TS03_CycleMax	HI	INT	SYS	15	OB35 Time disc 3 maximum cycle time
TS04_CycleMax	HI	INT	SYS	38	OB35 Time disc 4 maximum cycle time
TS05_CycleMax	HI	INT	SYS	17	OB35 Time disc 5 maximum cycle time
TS06_CycleMax	HI	INT	SYS	7	OB35 Time disc 6 maximum cycle time
TS07_CycleMax	HI	INT	SYS	20	OB35 Time disc 7 maximum cycle time
TS08_CycleMax	HI	INT	SYS	78	OB35 Time disc 8 maximum cycle time
TS09_CycleMax	HI	INT	SYS	51	OB35 Time disc 9 maximum cycle time
TS10_CycleMax	HI	INT	SYS	27	OB35 Time disc 10 maximum cycle time

5.8 User blocks

5.8.1 User- and Option blocks

Besides of the description of the System blocks, USER- and/or PCU-specific descriptions of blocks can be created. It isn't important if all descriptions are distributed in one or several files. However, the files must have the extension "*.pcu" and should normally be in the ...\PCU.XXX folder, otherwise they will not be considered during the reading process.

Global "User Param.pcu" file for Simatic S7:

S7 User Param files can be stored as global definitions in the language-specific "**...\param.pcu \usrfiles.xxx**" folders instead of inside the individual PCU-specific folders.

The following steps have to be considered:

1. Activate appropriate INI switches
   File: "om.ini"
   ```
   [Config]

   ;Enable/disable global user parameterization files (default:1)

   GlobalUserParamFiles=1
   ```
2.
   Create specific folders:
   ...\param.pcu\usrfiles.000 German
   ...\param.pcu\usrfiles.001 English
   ...\param.pcu\usrfiles.002 Spanish
   ...
3. Store User-Files in the according folders
   ...\param.pcu\usrfiles.000\xyz.pcu
   ...

5.8.2 Create user description

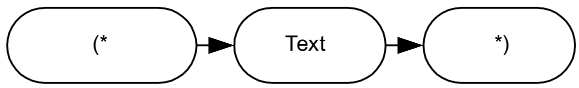
Basics

For the description of a user block a syntax must be kept to..

Structure of the description:

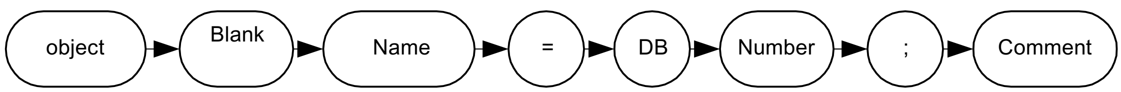
- Line for the class description
- General data
- Block head data
- Description of a data record
- Description of an attribute

Comment



The comment needs to be framed by keys "(*" and "*)". The comment may be a maximum of 44 characters long.

Class description



- object: Keyword
- Name: Name of the block / Class (maximum of 10 characters)
- DB: Keyword
- Number: Number of the data block

Example

object Sequence=DB725; (*Sequencer Control*)

General data

The general data should begin directly after the class description in the following line. Each line has to be finished with a semicolon (" ; ").

The following general attributes need to be allocated:

- CID=
  Indication of an explicit class ID (1000 – 2000)
- max=
  Max. number of data records
- offset=
  Byte address of the first data set
- length=
  data set length in Byte
- number=
  Indication of a fixed amount of data sets or address indication
- filename=
  Optional indication of a file name for the names of the instances. If the input is missing, the names are expected in a file which has the name of the block.
- classtype=
  Indication of the class type
- version=
  Indication of the version

Type of classes

For the display in the configuration the block can be assigned to a class type. The following types are delivered by the System:

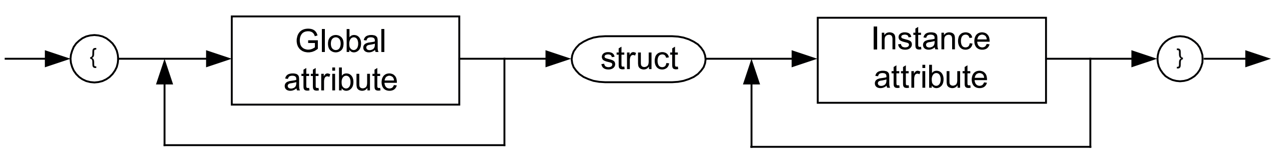
- 0: Display of all types
- 1: Technology
- 2: Periphery
- 3: System
- 4: Route Control
- 5: Weighers
- 6: Cross coupling

The type of classes can be extended by the user types.

File: `<proj-path>\texte.x\classtype.txt`

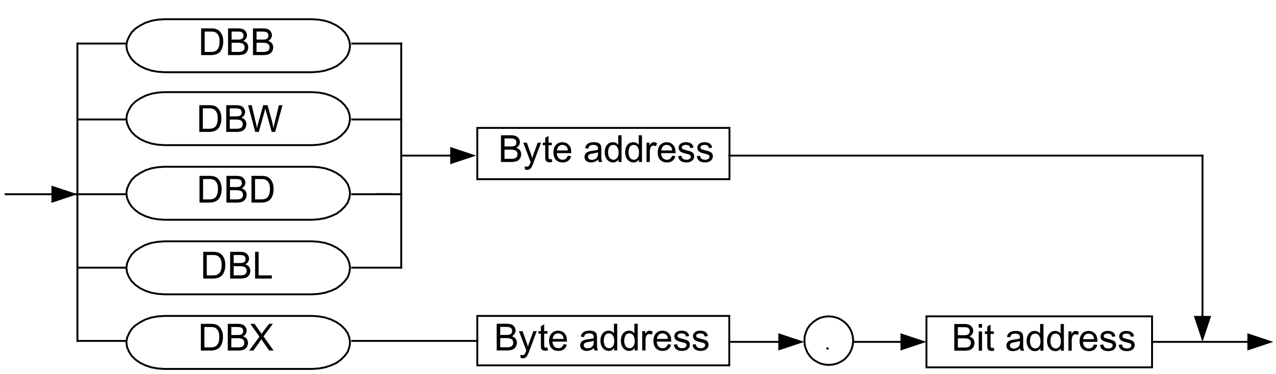
The file has to be edited via the text configuration.

Data block description



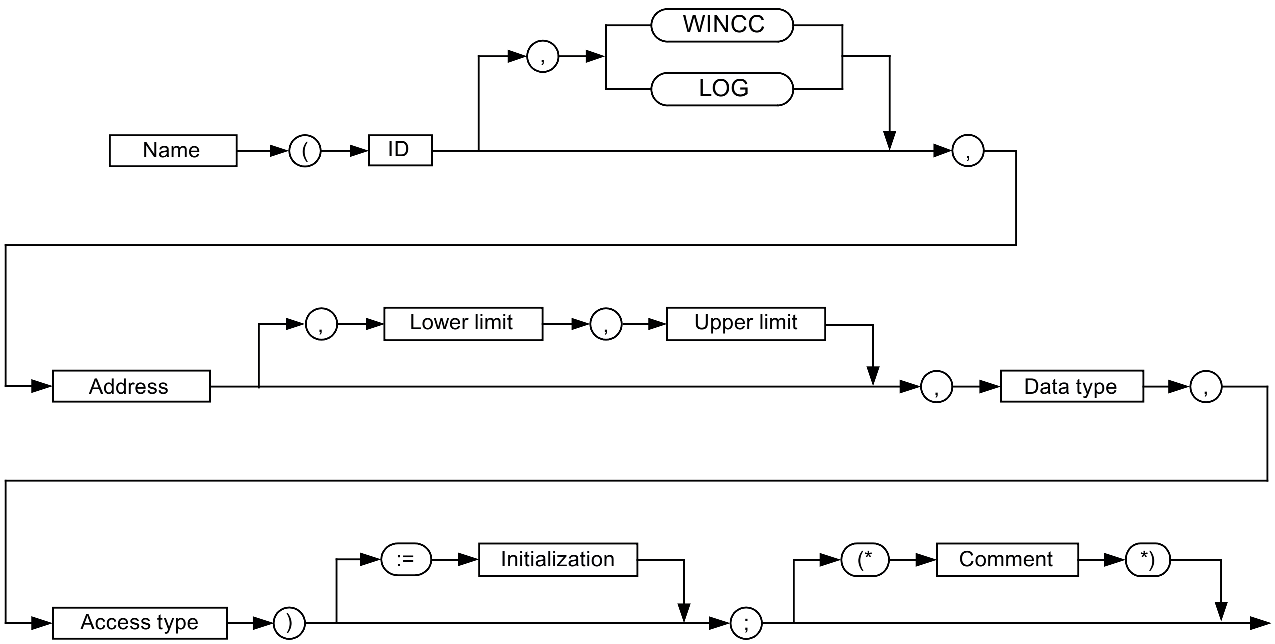
The data block description is started via a curly bracket. After the bracket there are the global attribute descriptions line per line. The description of a data record is started with the keyword 'struct'. After that the attribute descriptions of the data record follow. The data block description is closed with a curly bracket.

Address indication



The data record description can include a maximum of 200 attributes.

Attribute description



The sections of the attribute description are separated by commas.

- Name
  The name of the attribute can be a maximum of 10 characters long. It may not include any special characters.
- Attribute ID
  Within the block an explicit ID must be assigned to each attribute.
- Address of the attribute
- Low limit
  Minimum input value in the parameter assignment. This value is optional. If a low limit is specified, a high limit must be specified as well.
  If no limits are specified, the min/max values for the data type apply.
- Upper limit
  Maximum input value in the parameter assignment.

- Data type

Data type PCU < V7	Data type PCU V7	Description
B1	BOOL	Bit value
HEXA8	BYTE	Hexa number 8 bit
HEXA	WORD	Hexa number 16 bit
HEXA32	DWORD	Hexa number 32 bit
---	LWORD	Hexa number 64 bit
CHAR	CHAR	ASCII characters
I8	USINT	'Unsigned short integer' 8 bits
I16	INT	Integer number 16 bit
I32	DINT	Double integer number 32 bit
S7R	REAL	Floating-point number 32 bit
STEP	STEPREF	Logic command in the notation of STL The following are supported: A <I, Q, M, DBnnn.DBX> x.yAN <I, Q, M, DBnnn.DBX> x.y
QUEL	QUEL	Address of a data source Class/attribute (e.g. "ICM0,12, Stat") **Note:** You can only use flag bits in the range from 0.0 to 2047.7 for source connection
---	GREF, PREF	Address of a data source generally replaces QUEL from PCU Version V7 (G=Get, P=Put) Default = (NULL)
---	OREF	Reference to object instance
---	ENUM_G	Enumeration of global text file (project)
---	ENUM_P	Enumeration of PCU text file
---	ENUM_S	Enumeration of global text file (system)
---	ENU_SEQ	Enumeration for unit references of PCU text file (1...128 : "Sequence.txt", 129...255 : "GRUP_TA.txt")
---	DT	Date and time
---	STRING	Character string with length <n> (max. 254)
---	BLOB	Binary data of the length <n> bytes (max. 1024)

- Access type
  RD: The value can-only be read via the parameter assignment.
  HI: The value is in the 'Hidden attributes' areas
  HR: The value is read-only and in the 'Hidden attributes' area

Example

object Karlheinz=DB10;(*testKH*)	
	CID=1920;
	max=128;
	offset=100;

	length=100;		
	number=DBW8;		
	filename=KHtest;		
	version=1;		
{			
	Offset	(3 ,DBW2,INT)=100;	(*Offset to 1st data record*)
	Satz_Lae	(2,DBW4,INT)=100;	(*Record length*)
	SILO_Anz	(1,DBW8,INT,RD);	(*Number of silos*)
struct			
	Bit1	(1,DBX1.0,BOOL)=0;	(*Bit value*)
	Int8	(2,DBB2,USINT);	(*'Unsigned short integer' 8 bits*)
	Int16	(3,DBW3,INT);	("Integer number 16 bit")
	Int32	(4,DBD5,DINT);	("Double integer number 32 bit")
	STEP7	(5,DBW10,STEP);	(*S7 command*)
	QUELLE	(6,DBD20,QUEL);	(*Source reference*)
	HEXA_8	(7,DBB12,BYTE);	(Hexa number 8 bit*)
	HEXA_16	(8,DBW13,WORD);	(*Hexa number 16 bit*)
	HEXA_32	(9,DBD15,DWORD);	(*Hexa number 32 bit*)
}			

Note

F1 Help in the 'Parameter Assignment' application

A help file can be created for each PLC-block, that can be displayed by pressing the F1 key in the parameter assignment.

The file must be stored in the folder \Help.x. The name of the file must be identical with the block name. An underscore character "_" must be prefixed to the file name.

See also

Parameter types (Page 204)

5.8.3 Diagnosis with erroneous description files

Basic principles

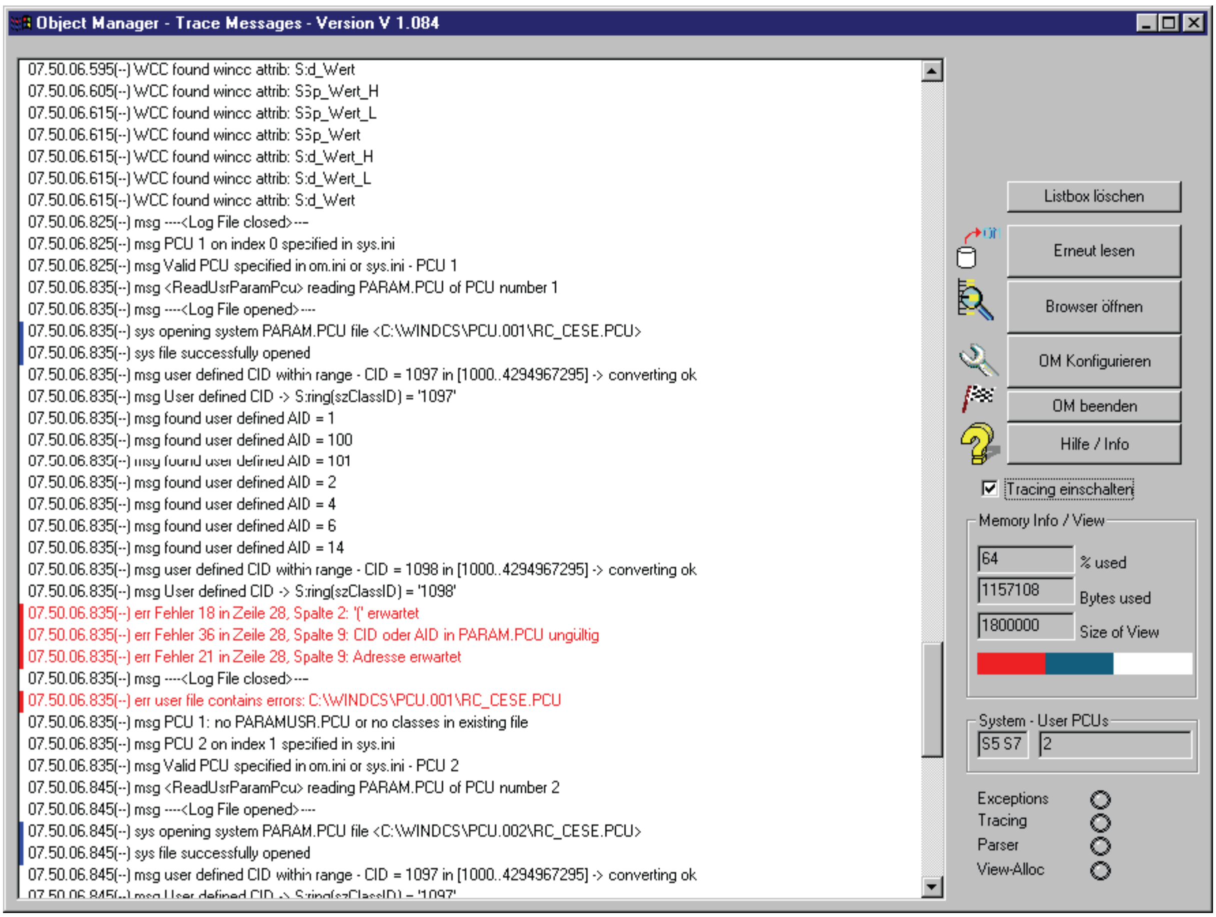
If faults occur with reading in, the total block is rejected by OM. In order to receive a diagnosis for the IOS, the OM diagnosis window can be overlaid. The text in the window is only in German.

File: `'<local-proj-path>\sys\om.ini'`

`[Config]`

`bTracingDialog=1`

Once a change has been made to this setting, all applications must be closed and the PCU server restarted.



Functions of the diagnosis window

- Listbox Delete
  With this button all inputs in the list are deleted.
- Read in again
  With the button all descriptions are read in again.
- Open Browser
  With this button a further dialog is opened in which all classes (= Blocks) and their attributes can be displayed.
- Configure OM
  With this button a further dialog is opened in which adjustments for the diagnosis could be made.

- Close OM
  With this button the diagnosis window is closed.
- Help/Info
  With this button a dialog with important information to the Object Manager is opened.
- Switch on Tracing
  With this connector the trace function can be switched on and off.
- All actions of the OM are displayed in the list. Via the dialog "Configure OM" the display can be restricted to errors and warnings.

Initial memory capacity of the OM

The Object Manager is only started once for each IOS and has the necessary block descriptions for all applications. One fixed memory location is used for this central function. This memory location can't be enlarged. Therefore it is started from a presetting quantity of this memory location. If the consumption of the memory should be increased due to additional description files over this limit, the presetting will be changed consequently.

5.8.4 User description in CSV files

Basics

Besides the normal system classes in which exactly one data record with fixed length is reserved in the data block for each instance, the method of working with instance DBs is also available. Here, each instance corresponds to exactly one data block with fixed size.

Each *.PCU file is allocated a *.CSV file, The CVS file contains one line per instance. The object manager converts this information into system objects in the usual way.

Class description

Object <Name> = `DB_CFC`; (* ... Comment ...*)

Example of object class XCDB75:

```
object XCDB75x=DB_CFC; (*QK: Puffer*)
CID=19999;
max=1;
offset=0;
length=30;
number=1;
classtype=4;
{
struct
iGetState (1040, DBW16,I16);(*Get State*)
iPutState (1050, DBW156,I16); (*Put State*)
boLen (100, DBX798.0, B1); (*Check length*)
bTimer (1080, DBB799, BYTE); (*Check length Timer*)
iMaxPut1_Len (1140, DBW800,I16); (*Max Length Put 1*)
iMaxPut2_Len (1160, DBW802,I16); (*Max Length Put 2*)
iMaxGet1_Len (1180, DBW804,I16); (*Max Length Get 1*)
iMaxGet2_Len (1190, DBW806,I16); (*Max Length Get 1*)
}
```

CSV syntax

<Name>; <Instance name>; <Object name from *.PCU>; <DB number>; <Identifier>

Example CSV file for object class XCDB75:

```
XCDB75x;XCDB75x 1;XCDB75x;DB 751;Instanz-DB
XCDB75x;XCDB75x 2;XCDB75x;DB 752;Instanz-DB
XCDB75x;XCDB75x 3;XCDB75x;DB 753;Instanz-DB
XCDB75x;XCDB75x 4;XCDB75x;DB 754;Instanz-DB
XCDB75x;XCDB75x 5;XCDB75x;DB 755;Instanz-DB
...
XCDB75x;XCDB75x31;XCDB75x;DB 781;Instanz-DB
```

5.9 Excel import/export for external parameterization

5.9.1 Installation and functionality

Note

- For use of the functionality described here, the Excel workbooks listed below are supplied in the '<sys-path>\!BM_Proj\excel\...' sample project. Copy these into a subfolder of their own within the project folder, e.g. "<proj-path>\EXCEL".
- It is recommended that the current Excel worksheets from the system installation are always used. This is the only way to guarantee correct parameterization in conjunction with new object classes or attributes.
- Insofar as configuration data has to be transferred from existing Excel worksheets from previous system versions, these have to be transferred into the new Excel worksheets by "copy/paste" before being imported.

File	Function
macro.xls	Excel file that includes the Excel macros listed in the following. With that, the user can trigger different import/export functions.
Param_S7_400_V6_PCU001_0.xls	Excel workbook for PCU type S7-400 versions before V7 with 30 worksheets including cover sheet. The basic process cell configuration (technological objects of the equipment) is covered here. There is a table for each of the following block classes or types of information: Sequences, bit memory, FIXV, AOUT, SEQS, DFM0-3, THREESTEP, ICM1-4, MVC, ALARM, AIN, MULT, PID, POLY, SEQU, SPEVAL, TIMER01/02, VMON, XC-JOB, EPE.INI, EPAR.INI, SEQUENCE.INI.
Param_S7_400_V7_PCU001_0.xls	Excel workbook for PCU type S7-400 versions V7 and higher with 48 worksheets including cover sheet. The basic process cell configuration (technological objects of the equipment) is covered here. One sheet each exists for the following block classes and information types (new classes in bold type): EOP-DFM assignment, flag, ANA, AOUT, SEQS, **BLR1-9**, **DFM0-8**, THRESTEP, ICM1-4, **SLB**, **PULSE**, **LINE**, MVC, MSG, AIN, MULT, PID, POLY, SEQUENCE, SPEVAL, **TANK**, TIMER01/02, VMON, XC-JOB, EPE.INI, EPAR.INI, SEQUENCE.INI
Param_S7_400_V6_PCU001_1.xls	Version PCU type S7-400 versions before V7 for English language
Param_S7_400_V7_PCU001_1.xls	Version PCU type S7-400 versions V7 and higher for English language

File	Function
Param_S7_1500_V7_PCU001_0.xls	Excel workbook for PCU type S7-1500 with 47 worksheets including cover sheet. The basic process cell configuration (technological objects of the equipment) is covered here. One sheet each exists for the following block classes and information types (new classes in bold type): EOP-DFM assignment, flag, ANA, AOUT, SEQS, BLR1-9, DFM0-8, THREESTEP, ICM1-4, SLB, PULSE, LINE, MVC, MSG, AIN, MULT, PID, POLY, SEQUENCE, SPEVAL, TANK, TIMER01/02, VMON, EPE.INI, EPAR.INI, SEQUENCE.INI
Param_S7_1500_V7_PCU001_1.xls	Version PCU type S7-1500 for English language

The Excel workbook "Macro.xls" contains the macros described in the following table:

Macro	Function
CompareAll	All the tables are compared with the offline configuration of the active project one after the other. The results of the comparison are written to a log file. All the errors and warnings are entered in this
CompareOneTable	The current table is compared with the offline configuration of the active project. The results of the comparison are written to a log file. All the errors and warnings are entered in this
ExportAll	All the table contents are exported to the offline configuration of the active project one after the other (see below for details)
ExportOneTable	The current table (=object class) is exported to the offline configuration of the active project. At the end, all the errors and warnings are entered in a log file (file “Export.txt” in the current directory)
ImportAll	All the tables from the offline configuration of the active project are imported one after the other (see below for details)
ImportOneTable	The current table is imported from the offline configuration of the active project. In the end a log file is opened. All the errors and warnings are entered in this. Then the current table is completed with the imported data

In order to run these macros, the System-Dll "Export.dll" is required, which was installed during the system installation. It implements the actual import/export functions of the configuration data from/to the (binary) offline blocks or INI files of the system. All the above-mentioned macros access this DLL.

5.9.2 Application

Requirement

- Before starting the offline engineering with MS Excel, it must be ensured that the offline DBs in the project/PCU directory `<proj-path>\pcu.nnn\DB\...` have a current version that corresponds to the actual PCU version. This can be done with the application "Block transfer" / "Save" function (PCU→IOS) for all data blocks (type DB *).
- From the delivery state, an Excel workbook corresponding to the PCU type must be created for each PCU and assigned to the associated PCU No. in the cover sheet.
- It is recommended that an "Import All" operation always be performed first and that this version be stored separately. This prevents the configuration from being overwritten by undefined versions with the next (perhaps inadvertent) export.

Note

Closing of applications with high GDI resource use

- To prevent error messages when importing or exporting the parameter assignment, all unneeded applications should be closed.
- In particular, applications with increased GDI resource use such as a process diagrams, STEP 7 V5.x or STEP 7 TIA Portal should be closed.

Procedure

The following procedure is recommended for application:

- Launch MS-Excel
- First of all, open the macro file “Macro.xls” and activate Macros. The MS Excel security setting should be "medium" or "low". The action for this differs slightly in the supported versions of MS Excel and can be obtained from the associated documentation.
- Open the PCU-specific workbook.
- Multiple PCUs/workbooks can also be opened simultaneously. All the following operations, in particular the macro calls always relate to the currently selected workbook or PCU.
- The standard way to call macros is via menu "Options" or "View" / Macro→Macros. However, the user can also define call buttons for the most frequently used functions (refer to MS-Excel documentation)
- Import the offline configuration to Excel; one Excel worksheet is filled in for each object class.
- Configure and edit the tables with all Excel tools and options
- Compare the configured data with the (offline) configuration in the project path
- Export the configured tables to the offline DBs and INI files
- After conclusion of configuring and after export of the respective classes, the associated offline DBs must be loaded to the respective PCU.

5.9.3 Table structure

Basic table structure:

- In the relevant worksheet for a class, the table header stretches from line 1 through line 7 (inclusive).
- Line 9 is a header line and is not evaluated as part of the import/export process.
- The instance area for this class follows on subsequently (i.e. as of line 10).

Table 5-15 Table Header:

	A	B	C	D	E	F	G	H
1	**PCU**	1						
2	**Class**	AIN						
3	**Amount**	1						
4	**Max**	256						
5	**Export**	E	DB,N,A					
6								
7	**Def**	No	NAME	DigX_ ANF	DigX_ END	XANF	XEND	UNTGR
8								
9	**Filter**	*AIN no.*	*Name*	*Xxx*				
10	**E**	*1*	*AIN1*	*0*	*100*			

The bold entries are keywords for the export function. They must be entered at the positions shown here in the sheet.

Table 5-16 Description of the header keywords:

Keyword	Meaning
`Class`	Class or technological block name that is being handled with this worksheet; must correspond exactly to the class name in the "param.pcu" file
`Amount`	Import: Shows the number of imported instances; only the area that is actually configured in the block is imported up to the max. line count in the table; if there are "surplus" data sets, corresponding warnings are issued in the log file window. Export: Target number of instances/lines to be exported; this number is applied to block parameterization, i.e. it is increased/reduced as necessary; if Amount is larger than the filled number of lines, default values are assigned to parameterization data sets. Therefore be careful at modification of the former imported value!
`Max`	Additional and more priority possibility for limitation of function import/export (next to Amount); should correspond normally to the max. value from the "param.pcu" file

Keyword	Meaning
`Export`	"E" shows which lines should be exported. Only those lines are exported which have the character specified in cell B5 in the cell area of the instances in the "Filter" column. The character sequence "DB, N, A" in the cell to the right indicates which attribute groups are to be exported. If the cell is empty, all attribute groups are exported. Otherwise, those groups that are separated by a “,“ are exported. The following attribute groups are possible: A: Export the new attributes: Address/Location/Description/Image DB: Export the DB-related attributes N: Export the name
`Def`	Definition of the class attributes; the attributes which follow in the cells to the right should be repeated in the instance area, i.e. as of line 9. Here they will serve as column headers.

Definition line

- A definition line must follow the head of the table. It must be possible to assign each cell an attribute of this class in accordance with the global system class definition in file "param.pcu". The entries are case sensitive.
- If this is not the case, a fault message occurs. If a cell and with that the entire column is not taken into account, an “_” underline must explicitly be entered.
- The keyword “Def” must be entered in the cell A7. The keyword “No” must be entered in the cell B7. With two-stage addressing, B7/C7 is used with an extended token (e.g., for EntityClass NoTemplate/NoEntity). Continue as you wish for the further sequence of attributes.
- You do not have to position the definition line as the 7th line; the definition line must however exist before the data sets.
- Entries to assist the user can be entered in line 9 (this line number is not compulsory either). This serves as a comment line.
- The keyword “Name” is entered as the name of the data set.

Instance data area

- The first instance data column is the filter column. The entry here corresponds to the export definition in the table header (field B5, see above). Only instance lines which contain the characters within the export definition are taken into account during the import/export process.

Note

Reading/Writing to individual data sets

To enable only specific instance data sets to be read or written to, you must ensure that **only these** have the corresponding entries set in the "Filter" column. The instance data sets with an empty field in the "Filter" column are then ignored during the export process.

- The attribute designations then follow in the cells to the right. These simply serve as column headers and do not need to match the definition line attributes.

Additional columns

After the B columns, columns that cannot be exported can also be added to the table. The condition for this is that the attribute does not exist in the "param.pcu"-file and doesn't represent any other keyword.

Special features of specific object classes

- For some classes the header attributes are restricted as follows:

Class	PCU	Class	Amount	Max.	Export
Entity_Group	--	3	3	--	3
Entity	--	3	--	--	3
MasterTagList	--	3	--	--	3
Entity_Template	--	3	3	--	3
Template_Parameter	--	3	--	--	3

- For the units/sequences of the PCU < V7, the header attributes are expanded as follows:

Line	A	B	C
1	PCU	1	
2	**Class_RecV3**	**Plant sections**	
3	**Class_RecV5**	**Sequences**	
4	Amount	64	
5	Max.	64	
6	Export	E	DB,N,A
7			
8	**Def_RecV3**	No	NAME
9	**Def_RecV5**	No	NAME
10			

This enables access to RecipeV3 unit data records and to RecipeV5 sequence data records (Def_RecV3/V5 operates dynamically depending on the recipe version). The following lines move accordingly.

Note

Excel spreadsheets provided in the delivery state

The selected columns in the supplied standard templates correspond to the attributes of the respective classes or text files relevant for the process cell configuration. They will not be described further here because they represent "general engineering know-how", which is already described in other documents. For better orientation and overview, however, comment fields with information on the respective connected data element of the configuration have been inserted in the definition cells.

Excel spreadsheets for user classes

If configuration is also to be performed for user-defined object classes, we recommend creating a new Excel spreadsheet for each PCU. A worksheet for each user object class must be created within this spreadsheet. It is recommended that you observe the following procedure when creating them. This requires that the user classes be available in the parameter assignment (see section User- and Option blocks (Page 389)).

1. Copy the cover sheet and a class worksheet from an existing Excel file into the new PCU Excel file.
2. Next, adapt the name of the worksheet for the first user class, including the header data for this class, in accordance with the definitions in the "*.pcu" class file.
   - `PCU` → <PCU-No.>
   - `Class` → <`Object` name>
   - `Amount` → 1 (the current number of instances is entered on import)
   - max → `<max` number of instances of this class>
   - `Export` → E
   - `Def..` → No, NAME, ...
     The required user class parameters must be entered into the adjacent fields on the right (description as per key word `struct ...`)
   - Line 8 normally remains empty
   - The instance data area header line follows on from line 9, beginning with the "Filter" column.
   - Underneath it, there must be as many lines as the maximum number of instances to be imported or exported. The character "E" must be entered in the first field ("Filter" column)
3. Now test the function for this object class with the macro "ImportOneTable". This macro reads the available number of user class instances from the offline block container (current number is in the field next to `Amount`).
4. Check whether the data that has been read matches up with the same instances in the "Parameterization" application.
5. Copy the finished worksheet into new worksheets for each of the other user classes and rename them accordingly. In addition, the header data and the column headers must also be adapted as appropriate.
6. Copy this Excel file for each additional PCU into a new file and adapt the PCU-No. in the header data accordingly in each case.

5.9.4 Keywords

Other keywords are recognized alongside the attributes of the param.pcu.

Module	Keyword	Export to
All	Name	Name of the data set; is exported in the text file of the module (see param.pcu)
All	No	Number of the data set; must be in the first exportable column; for SW.INI/EPAR.INI DFM Group.DFM Number must be specified
All apart from EXT_CLASS	Addr	Text for address, stored in PCU.x\Address.L displayed in faceplate on the "About" tab
All apart from EXT_CLASS	Desc	Text for description, stored in PCU.x\Desc.L displayed in faceplate on the "About" tab
All apart from EXT_CLASS	Image	Path for process picture, stored in PCU.x\Image.L displayed in faceplate on the "About" tab
All apart from EXT_CLASS	Loc	Text for location, stored in PCU.x\Location.L displayed in faceplate on the "About" tab
ENTITY	BaseAdrByte	Byte of the basic address for relative addressing
ENTITY	BaseAdrDB	DB of the basic address for relative addressing
ENTITY	BaseAdrPCU	PCU of the basic address for relative addressing
ENTITY	ID	ID of the entity (globally understandable!)
ENTITY	NoEntity	Entity number (relative to type)
ENTITY	NoEnti-tyGroup	Number of entity type
ENTITY	NoTemplate	Number of template to be assigned
Entity_Group	EntityAmount	Number of entities of this type
Entity_Group	Name	See above
Entity_Group	NoEnti-tyGroup	Number of entity type
Entity_Template	Name	Name of template
Entity_Template	NoTemplate	Template number
Entity_Template	ParameterA-mount	Amount of parameters of a template
ICM1 ... ICM4	Entity	Assignment of ICM to entity; several values can be entered, sepa-rated by a comma
MasterTagList	EntityID	Globally understandable entity ID
MasterTagList	Tag	Understandable instance name for TAG to be assigned
MSG	Incoming	Meldkom.txt
MSG	Outgoing	Meldgeh.txt
Sp.Value	Comment	Comment
Sp.Value	Dim	Dimension
Sp.Value	Type	Type of special variable
Sp.Value	Value	Value of special variable
SW.INI/ EPAR.INI	Comment	Comment
SW.INI/ EPAR.INI	Dec	Decimal point of the set value 0 - 7

Module	Keyword	Export to
SW.INI/ EPAR.INI	Dim	Dimensions
SW.INI/ EPAR.INI	File	File name; only necessary and useful for TEXT type
SW.INI/ EPAR.INI	Max	Maximum input value (without decimal point)
SW.INI/ EPAR.INI	Min	Minimum input value (without decimal point)
SW.INI/ EPAR.INI	Name	Is ignored (in the DFMs)
SW.INI/ EPAR.INI	Type	Type of set value: SW, TEXT, 16Bit, 32Bit
GOP.INI/ EPE.INI	Age	2-255
GOP.INI/ EPE.INI	AT-TID_GOP_COMMENT	Comment for GOP/EPE
GOP.INI/ EPE.INI	DFM1 ... DFM24	Set the setpoints; the DFM can be specified as an absolute number or as a DFM <group>.<instance>; an "e" after the DFM indicates that the setpoint cannot be edited 001 ... 255: DFM 0.1 ... DFM 0.255 257 ... 511: DFM 1.1 ... DFM 1.255 513 ... 767: DFM 2.1 ... DFM 2.255 769...1023: DFM 3.1 ... DFM 3.255 1025...1279: DFM 4.1 ... DFM 4.255 1281...1635: DFM 5.1 ... DFM 5.255 1537...1791: DFM 6.1 ... DFM 6.255 1793...2047: DFM 7.1 ... DFM 7.255 2049...2303: DFM 8.1 ... DFM 8.255
GOP.INI/ EPE.INI	JumpName	String MaxLen 8
GOP.INI/ EPE.INI	Label	1-255
GOP.INI/ EPE.INI	Name	Is exported in the gop.txt/epe.txt
GOP.INI/ EPE.INI	OpReq	Text for operator request, exported in gop.ini
GOP.INI/ EPE.INI	OpReqFlags	0..configuration OPReq
GOP.INI/ EPE.INI	Unit	Plant section assignment, "Kzuo"; if the cell is empty then nothing is exported for this Gop
TEILANL.INI/ SEQUENCE.INI	Name	Is ignored (is in teilanl.txt/sequence.txt)
TEILANL.INI/ SEQUENCE.INI	DFM1 ... DFM20	The DFMs that are displayed in step 0.
TEILANL.INI/ SEQUENCE.INI	Name	Export in Teilanl.ini as "Comment="

Module	Keyword	Export to
TEILANL.INI/ SEQUENCE.INI	RecType	Separated with commas, the list of authorized recipe types for the start of a recipe
Template_Parameter	Byte_Bit	Dbyte/Dword/DDWord/Bit
Template_Parameter	DB	DB with absolute address (0 otherwise)
Template_Parameter	DecimalPoint	Number of decimal points
Template_Parameter	Dimension	Dimension
Template_Parameter	File	Display text
Template_Parameter	IsStatus	1 if the parameter is the status parameter, 0 otherwise
Template_Parameter	Max	Maximum
Template_Parameter	Min	Minimum
Template_Parameter	Name	Name of parameter
Template_Parameter	NoParameter	Number of parameter
Template_Parameter	PCU	PCU with absolute address (0 otherwise)
Template_Parameter	Type	Type of parameter SW/TEXT/HEX/BIN

Process pictures 6

6.1 Image construction

6.1.1 Functions and properties

6.1.1.1 General

Basics:

The application Image construction provides the following configuration functions:

- Definition of the process visualization interface
- Administration of static background images
- Connect process variables
- Create executable process diagrams
- While processing the same file by several users (IOS) simultaneously, the consistency of data is guaranteed when the last storing attains validity in each case.

Multiclient function:

Note the following points when operating a client in a multiclient configuration:

- When you "Open" an image file (see "File - Menu (Page 410)"), you determine the destination area server to initialize the "open" file dialog by means of the area selection dialog.
- At the same time the "default area" for the actual image design session is determined. The variables and faceplates for the process picture are not restricted to this area but variables/ faceplates of different areas can be configured in the same image.
- The dialogs ‘Specify data source’ of variables and for interconnecting the faceplates were extended by the 'AREA' parameter. This information is only shown for multiclients!
- File references within object properties (*.txt, *.bmp) are configured via the Windows standard ‘Open file dialog’. This also allows searching over network paths. A selected network path is indeed accepted first of all, yet during the saving of the picture this is replaced by the path of the server of the referenced AREA (for example related PCU). By opening the object again, the current path reference then appears.

Client-specific folders on IOS server

It is possible to create a subfolder (IOS.XXX) for each client on the server in which client-specific settings and process pictures can be stored.

Example:

...<proj-path>\IOS.102\etc\kpos001.ini → Area definitions for process cell overview

...<proj-path>\IOS.102\texte.0\bereich.txt→ Area names

...<proj-path>\IOS.102\texte.0\Panel.ini → Panel faceplate application bar

...<proj-path>\IOS.102\texte.0\Paneltree.ini → Panel faceplate navigation bar

...<proj-path>\IOS.102\bilder\xyz.bik → Process pictures

...<proj-path>\IOS.102\bilder\xyz.bmp → Background pictures

As soon as a client folder is created on the server, the relevant information is read from it (e.g. Open image dialog in LZSYS or "Process cell" menu in SEQCtrl). If no client folder exists, the default access path is used. This can also be used, for example, to manage/save specific AREA-related client images on the server.

Object size and positioning

When an object is selected in the process picture, its position and size are displayed in the status bar.

Use the shortcut Ctrl+arrow key to reduce or enlarge the symbol pixel-by-pixel when the size of the symbol can be changed:

- Ctrl+left → horizontal smaller
- Ctrl+right → horizontal larger
- Ctrl+up → vertical smaller
- Ctrl+down → vertical larger

6.1.1.2 File - Menu

New	Create a new document.	
	Keyboard:	CTRL+N
Open...	Opens an existing document	
	Keyboard:	CTRL+O
Save	Saves an open document under same file name.	
	Keyboard:	CTRL+S
Save As...	Saves an open document under a different target designation.	
Activate	Switch into the runtime system.	
	Keyboard:	F5
Print ...	Print the active document	
	Keyboard:	CTRL+P
Printer setup...	Determine the printer and the printing options	
File 1, 2, 3, 4	Selection option for the last four processed files.	

6.1.1.3 Edit Menu

Back	Undoes the last change	
	Keyboard:	CTRL+Z
Redo	Restores the last reset.	
	Keyboard:	CTRL+Y
Cut	Deletes the selected objects and copies them to the clipboard	
	Keyboard:	CTRL+X
Copy	Copies the selected objects to the clipboard.	
	Keyboard:	CTRL+C
Insert	Inserts the clipboard objects at the cursor position (possible several times)	
	Keyboard:	CTRL+V
Duplicate	Duplicates the selected objects	
	Keyboard:	CTRL+D
Delete selection	Deletes the selected items	
	Keyboard:	DEL
Select all	Selects the entire list of variables	
	Keyboard:	CTRL+A
Align	Aligning the objects of a selected block: left, right, top, bottom Center horizontally, center vertically, distribute horizontally, distribute vertically	
Variable search	on the screen	Opens search dialog for objects of the open image
	global	Opens search dialog for objects of all images
Align variable information	The selected variable information is aligned to the accompanying objects	

See also

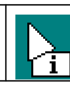
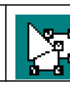
Variable search (Page 509)

6.1.1.4 Background image - Menu

Loading	Load a previously created background image.
Inverting	Inverting the active background image. This is done only temporarily on the screen, in order to relocate seemingly "lost" image objects. The bitmap file of the background image remains untouched.

6.1.1.5 Variable - Menu

Selection	The cursor changes to the status:	
	In this status objects can be selected.	

Information	The cursor changes to the status: 
	By clicking on the required object, an information box about this object appears • The position and size of the information box can be adapted with the mouse. • A connection line between the variable object and information box can be shown/hidden. • After saving the image with information boxes showing, these are also shown in the application process diagrams The information boxes can also be shown/ hidden there (menu "Process diagrams\Tag information").
	Changing the properties of the information box: • Right click/Properties on the information box shows the current setting • ">>" button opens a list box with the possible individual attributes **per variable instance**. Several attributes can be selected one after the other. The number and order are shown in the text box. • The "Font" button opens the standard Font dialog, a change **applies to all information boxes** • Default button: Sets the "$Default" info boxes **per variable type** (applies to all instances of this type) In the "Replace with" field attributes can be deleted. "OK" twice saves this setting in the file "...\Bilder\Image.ini". The basic status (= ALL POSSIBLE ATTRIBUTES OF THIS TYPE) can be restored by deleting all entries in the "Replace with" box and "OK" twice. • The selection box "Draw connection lines" displays a connection line
	Hiding the information window Double-click the corresponding object (not the information window) with the left mouse button to execute.
Levels	The cursor changes to the status: 
	The current layer and sequence information is shown for all images: • Lay: <n> identifies the current layer allocation • Pos:<n> identifies the current position within the image file
Integer/Real	Connect an integer variable or real variable
Hexadecimal	Connect a hexadecimal variable
Bit	Connect a bit variable
ICM	Connect an individual control element
Static text	Insert static text
Text	Connect process value display by text outputs
Static picture	Insert static image as bitmap
Line	Connect pipeline-type object
Bar	Process value display in the form of bars (horizontal or vertical)
Sound	Connect a variable with *.wav file
Control	Faceplate operation dialog
Change process diagram	Define order of change to different process diagrams
Exec FB/FC	Connect executable function block
Exec-IOS	Connect executable PC file

6.1.1.6 Tools - Menu

Basics

The Tools menu offers the following commands:

MS Paint

When supplied, the bitmap editing tool MS Paint integrated in Windows is configured for editing the background picture. When opening this menu item, the background picture belonging to this process diagram is automatically loaded.

This allocation can be changed via 'Setup tools'.

Setting up tools

This menu item is used to link programs into the **Tools** menu. Via a dialog box Windows programs can be linked in the **Tools** menu and can also be started from there.

Setup tools dialog

Table 6-1 The following dialog elements are available:

Menu text	Here the desired text is entered which should appear in the **Tools** menu later. A **&** underlines the following letter in the **Tools** menu. This enables the key combination ALT - and the chosen letter to start the linked program.
Add	This command is used in order to link a new program. By clicking this option the **Open file** dialog box is displayed. Here the desired program is selected and confirmed with **OK**.
Delete	Removes programs which have already been linked from the menu **Tools.**
Path	Displays the path of the selected program. **Note**: The path cannot be changed.
Drawing program	If this option is selected, the background picture will be loaded automatically in the **Tools** menu when the connected program is called.
Button "?"	Shows the help text for **Setup tools dialog box**.
Cancel	Close the dialog box without accepting the changes.
OK	Closes the dialog box and accepts changes.

Process diagram conversion

Process diagrams can be scaled for other screen resolutions via this menu item. A description of this can be found in the section Picture conversion (Page 415).

Edit DBF Files

Opens a dialog window to edit the dbf files for the connection type "Indirect addressing". Indirect addressing of the images and the configuration of the table content are described in the section ‘Specify data sources’ (dialog) (Page 429).

Create TAGs for process data logging

Opens the dialog for selecting the process diagrams for process data logging You will find a description of this in the section Configuring data recording for Replay mode (Page 519).

6.1.1.7 View - Menu

The view menu offers the following commands:

Toolbar	Displays or hides the toolbar. There are some symbols on the toolbar for some of the most frequent commands of the Image design including Open, Save, Print.
Object bar	Displays or hides the object bar. The object bar has icons for creating new picture objects: integer/real variables, hexadecimal variables, bit variables, ICM variables, text variables, static text, static pictures, line variables, bar variables, sound output, control OCX, picture changes, open FB or FC, run IOS applications, invert background.
Alignment bar	Displays or hides the alignment bar. The alignment bar contains icons for aligning the objects of a selected block (left, right, top, bottom, center horizontally, center vertically, justify horizontally, justify vertically.
Search bar	Displays or hides the search bar. The search bar contains icons for calling the tag search in the picture, global tag search, and for displaying statistics of all objects used in the picture.
Status line	Displays or hides the status bar in the display window at the bottom. • This is where all the actions are displayed, which are executed if a menu item is selected or a button in the toolbar is pressed, etc. • The left part of the status bar shows additional information on selected menu items (mouse or keyboard) or icons before they are executed. • The center area shows information about the layer, position, number of variables, number of control elements • The right area of the status line displays which of the following keys are activated: SHIFT: The Caps lock key is pressed. NUM: The Num lock key is pressed. SCROLL LOCK: The scroll lock key is pressed.
Layer view	Displays the edit bar for the layer setting of the picture editing. This can be permanently aligned to the left or lower edge or moved as an independent window anywhere on the screen. The last position is saved. You will find a description of the layer function in the section Layer view (Page 419).

6.1.1.8 Picture conversion

Scale process diagrams for other screen resolutions

In the 'Tools → Process diagram conversion' menu there is a dialog available for scaling, i.e. adjustment of one or more process diagrams to new screen resolutions.

Note

Settings in '...<proj-path>\sys\biko.ini' file

The following setting is required to activate the tool:

```
[Options]
...
BikoSizer=1
```

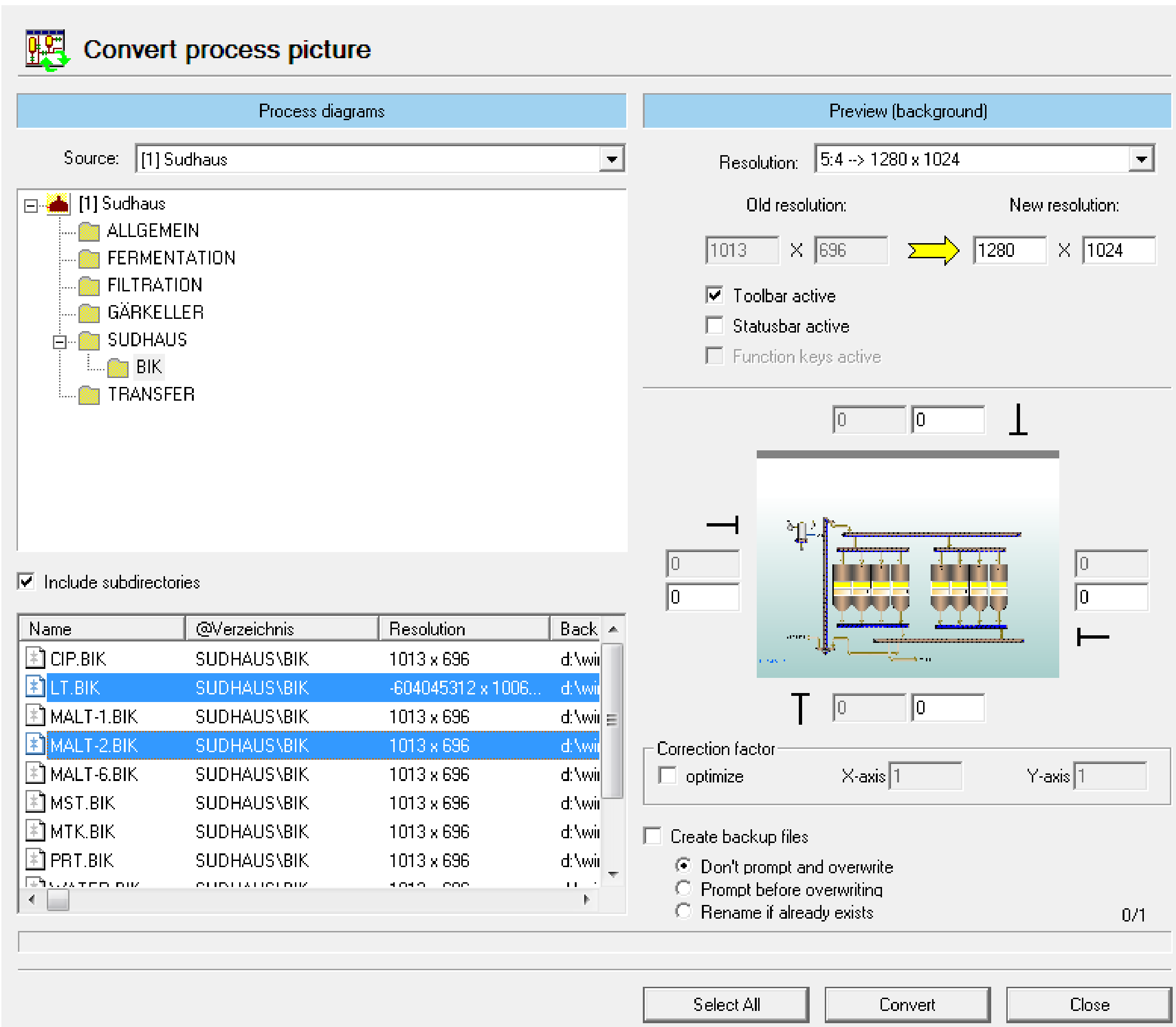
Scaling functions

The following scaling functions are available:

- The background image view is scaled to the new image resolution
  (the image file resolution itself remains unchanged).
- The positioning of the image objects is recalculated based on the image resolutions (Old/New).
- Additional correction factors for x-axis (horizontal) and y-axis (vertical) can be specified for the positioning.
- Empty areas (offset) can be provided for the four page sides.

Control dialog

The control dialog offers the following display and control options:



Selecting the process diagrams to be converted:

- "Source"
  The AREA and thereby the server from which the image directory is read can be selected in the list box.
- The tree view on the bottom displays the folder structure of the associated folder "`<proj-path>\bilder\`"
- 'Take subdirectories into account'
  The table view on the bottom shows all image files (*.bik) of the directory selected with or without subdirectories

- Table of the image files found with the columns:
  - Name
  - Directory
  - Aspect ratio
    This concerns the actual values of the last conversion. These are saved in the '*.bik' file, therefore with the first conversion the aspect ratio of the background image is displayed.
- The image files to be converted can now be selected in the table. Multiple selections are possible via the CTRL and SHIFT keys
- All images in the table are selected if the 'Select all' command is used.

Conversion settings and preview:

- 'Resolution'
  The new aspect ratio connected with the resolution can be determined in the listbox (pre-setting: 5:4 → 1280 x 1024). The monitor resolutions currently known are provided.
- The fields for the current as well as the new aspect ratio follow below this. The new aspect ratio can be changed by editing the x/y pixel values directly.
- 'Toolbar active'
  The space requirement for the toolbar of 'LzSys' is automatically taken into account in the process diagram (gray bar on top in the preview image)
- 'Status bar active'
  The space requirement for the status bar of 'LzSys' is automatically taken into account in the process diagram (gray bar on bottom in the preview image)
- 'Function keys active' (can only be selected if the statusbar is activated)
  The additional space requirement for the function keys of 'LzSys' is automatically taken into account in the process diagram (double gray bar on bottom in the preview image)
- Preview image
  Shows the image preview with the settings currently selected
- A pixel offset value can be defined via both text fields on each image side (light gray bar on the relevant side). Empty areas for further image objects to be added manually are created for this in the converted image.
  - The fields that cannot be edited (above / left) contain the last valid offset value.
    The offset values are saved in the '*.bik' file and therefore the fields are still empty with the first conversion.
  - The editable fields (below / right) contain the correction values (+ / - pixel information) for the subsequent conversion
  - This way offset values from previous conversion cycles can also be withdrawn again in whole or in part

- 'Correction factor'
  - 'Optimization'
    Enables both fields for the correction factor
  - X-axis / Y-axis
    The factor causes additional relocation of the display objects contained in the image to the relevant axis in relation to the image aspect ratios Old to New
  - Without optimization or where the factor = 1.0 the image objects are merely moved through the image aspect ratios Old → New
- Change PCU number
  - 'Change'
    Enable both fields for old value and new value
  - Old value, new value
    All object references with PCU number (old value) are changed to PCU number (new value).
    Notice: Pay attention to selected images!
- 'Create backup files'
  Upon activation the file to be converted is saved in a newly created folder 'Backup of <source folder>'. The folder is located at the same directory level as the source folder itself. The following options is available for the backup file:
  - 'No query before overwrite'
    The backup is made to the '<source file>.bak' file. If present then this will be overwritten
  - 'Query before overwrite'
    The backup is made to the '<source file>.bak' file. If present this will be overwritten following a request, otherwise the procedure is terminated
  - 'Rename if file already exists'
    The backup is made to the '<source file>.bak' file. If already present the backup is made to a new <source file>.nnn file (nnn is a figure which is incremented and begins with 001)

Starting the conversion and terminating the dialog

- All images in the table are selected if the 'Select all' command is used
- 'Convert'
  Starts the conversion / backup of the image files selected. Files which have already been converted are marked in the table with green checkmarks.
- 'Close'
  The conversion dialog is terminated

6.1.1.9 Layer view

Basics

The layer display makes it easier to configure the process diagrams by allocating the large number of picture objects to particular layers according to certain allocation criteria. During the picture construction, therefore, groups of picture objects can be displayed or hidden via the layers or they can be disabled for further editing.

- If a layer is hidden using the eye icon, the picture objects are hidden.
- If a layer is disabled by the lock icon, visible picture objects, for example, can no longer be selected or edited.
- The edit bar for the layer settings can be displayed via the menu View → Tools.
- The standard for new or existing process diagrams from previous versions is a layer, which can be expanded with other layers.
- Although a sensible number of layers is certainly less than 16, there is no limit to the number.

Functions of layer editing bar

To check the current layer and position allocations of the objects, the layer information display should be enabled (menu Variables→Layers).

The following functions are available in the edit bar via the context menu (right click with the mouse) or with a left click on the icon:

Command in the context menu	Functional description
Select all objects of the layer	All objects of the layer under the cursor are selected.
Add layer	Inserts an additional layer and activates it.
Remove layer	Deletes the selected layer and all associated objects. A confirmation prompt is issued before deleting.
Rename layer	Opens the dialog to change the name of the layer under the cursor
Activate layer or left click on the layers button	1. The layer under the cursor is activated (layer button is selected) 2. The layer is set to the "displayed" state (eye icon = visible) 3. The layer is set to the "enabled" state (lock icon = open) **Newly added objects are always allocated to the activated layer**
Show/hide layers or left click on the eye icon	• Hides the layer under the cursor → Objects invisible (eye icon = strike through) • Shows the layer under the cursor → Objects visible (eye icon = visible) **Selecting, moving, copying, deleting objects is only possible for displayed layers**
Enable/disable layer or left click on the lock icon	• Enables the layer under the cursor → Objects visible (lock icon = open) • Disables the layer under the cursor → Objects visible (lock icon = closed) **Selecting, moving, copying, deleting objects is only possible for enabled layers**

Command in the context menu	Functional description
Hide all layers	Hides all layers except the active layer → Objects invisible (eye icon = strike through)
Show all layers	Displays all layers → Objects visible (eye icon = visible)
Disable all layers	Disables all layers except the active layer → Objects visible + cannot be selected (lock icon = closed)
Enable all layers	Enables all layers → Objects visible + can be selected (lock icon = open)

Functions in the working area

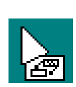
To check the current layer allocations of the objects, the layer information display should be enabled (Variables→Layers menu).

The following functions are available in the context menu (right click) on an object in the working area:

<table>
<tr><th>Command in the context menu</th><th colspan="2">Functional description</th></tr>
<tr><td>Layer settings [Lay:<x> Pos:<y>]</td><td colspan="2">Lay:<x> = current layer; Pos:<y>= current position
Allocates a new layer to the object under the cursor.</td></tr>
<tr><td rowspan="3">Layout</td><td colspan="2">Positioning of the selected object within the picture file</td></tr>
<tr><td>Forwards (= last position)</td><td>CTRL+Plus</td></tr>
<tr><td>Backwards (= first position)</td><td>CTRL+Minus</td></tr>
<tr><td>Object name</td><td colspan="2">Allows the entry/modification of the object name (important in conjunction with indirect addressing)</td></tr>
<tr><td>Properties</td><td colspan="2">Opens the configuration dialog of the object</td></tr>
<tr><td>Data source</td><td colspan="2">Opens the "Specify data source" dialog of the object
• with standard variables with source type and source information
• with controls with source type and reference variable for indirect addressing</td></tr>
<tr><td>Copy</td><td colspan="2">Copies the object under the cursor to the clipboard.
• The object can then be inserted at any desired location (context menu 'Insert').</td></tr>
<tr><td>Delete</td><td colspan="2">Deletes the object under the cursor</td></tr>
</table>

6.1.2 Standard variables

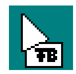
6.1.2.1 Image alternation

The cursor changes to the status: 

You can now set the cursor on the required position, press and hold down the left mouse button and move the mouse. A rectangle appears, which represents the area in which the picture change can be activated later. After releasing the left mouse key the dialog box **Specify image change** appears.

- Image change path: → the required process picture (*.bik file) can be selected
- Keyboard shortcut: → It is also possible to define a keyboard shortcut for this picture change

6.1.2.2 Exec-FB

The cursor changes to the status: 
Set the cursor on the required position, press and hold down the left mouse button and move the mouse. A rectangle appears, which represents the area in which the function block can be activated later. After releasing the left mouse key the dialog box **Specify executable function block** appears with the following input boxes.

- AREA → Selection only possible with multiclient, otherwise "(Local area)"
- PCU no.
- Block type → FB or FC are possible
- Parameter → This value is available at the start of the block in Accu1 **(→ applies only to PCU type S7-400)**
- Block number
- Keyboard shortcut → It is also possible to define a keyboard shortcut for this block call

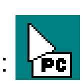
---

Note

Covering of other objects

- The button for Exec_FB covers objects of "Controls" type underneath.
- For objects of tag type, etc., the button is still transparent.
- Attention should be paid to this when configuring the size and position of the rectangle.

6.1.2.3 Exec-IOS

The cursor changes appearance to the following: 

Set the cursor on the required position, press and hold down the left mouse button and move the mouse. A rectangle appears, which represents the area in which the PC program can be activated later. After releasing the left mouse key the dialog box **Specify executable PC program** appears with the following input boxes.

- Command line → the required program can be entered directly. Using "Browse", the required program (*.exe) can be selected in an open file dialog
- Keyboard shortcut → It is also possible to define a keyboard shortcut for this program call
- Operating level → The program call can only be executed starting at the specified operating level (default = 1)
- End process diagrams → By calling the program "LzSys" is ended

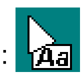
---

Note

Covering of other objects

- The button for Exec_IOS covers objects of "Controls" type underneath.
- For objects of tag type, etc., the button is still transparent.
- Attention should be paid to this when configuring the size and position of the rectangle.

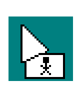
6.1.2.4 Static text

The cursor changes appearance to the following: 

Set the cursor to the background, left-click. The **Specify static text** dialog appears with the following input boxes:

- Text →
  Special characters (e.g. \n \t \r \\ ) in static text fields are no longer interpreted as control characters but displayed as clear text. However the quotation marks characters (" ") must not be used in text strings.
- Text width →
  Maximum number of characters that can be displayed The width of the text field is based on a standard character width according to the selected font. For specific text contents, the frame width may have to be adjusted after "Calculated size" is deactivated (see below).
- Alignment → Left aligned/Centered/Right aligned
- Text color → Color of the text characters (selection from a list or defined individually)
- Background color → Background color of the text line (selection from a list or defined individually
- Border width → line width (0 ...10 pixels)
- Border color → Line color (selection from a list or defined individually
- Clip other objects → The static text is covered by other objects.
- Calculated size → deactivated: The text field frame can be changed later (default = active).

6.1.2.5 Static picture

The cursor changes to the status: 

Position the cursor in the screen and left-click.

The dialog window "Configure graphic object" opens. The desired bitmap is selected here and the settings are made for the display.

Image selection:

Clicking "..." opens the file selection dialog in the folder "<proj-path>\symbole\...".

The following file formats are supported:

- Bitmap (*.bmp)
- EMF (*.emf)

- JPG (*.jpg)
- PNG (*.png)

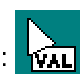
Preview

The picture prescribed in the picture selection is displayed here.

Layout

- Original size
  - If selected, the picture is displayed with the prescribed pixel dimensions, zooming is not possible
  - If not selected, the picture can be stretched after leaving the dialog and selecting the blue picture frame with the mouse.
- Transparent
  The bitmap is displayed in transparent mode, i.e. white pixels are shown transparently.

6.1.2.6 Value

The cursor changes appearance to the following: 

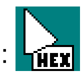
Set the cursor on the desired variable position and press the left mouse button. The **Integer Dialog Box** appears with the following input boxes:

- Type → Integer or real values can be selected as data types.
- Decimal point position →
  Specifies the decimal point position (from the right) both for integers and for REAL values. If no decimal point place is requested, the default value 0 must be kept. Valid input values are within the range of 0-10.
- Number of characters →
  Defines the decimal character count of the variable type.
  There are always only as many places shown as specified here. The width of the text field is based on a standard character width according to the selected font. For specific values, the frame width may have to be adjusted after "Calculated size" is deactivated (see below).
- Top entry limit → Since this is a tag type that can be controlled by the operator, the top entry limit can be specified.
- Bottom entry limit → If an input value may not fall below a certain limit, this can be specified here.
- Alignment (Left aligned/Centered/Right aligned) → default = Right aligned
- Entry text: →
  The text entered here will be displayed when working with tags in the runtime system. The combo box allows you to store different input texts and to select them again and again. An input text which is entered as new is entered immediately in the combo box, but it is only stored if the OK key is pressed and the dialog is closed without any error messages.
- Text color → Color of the text characters Either predefined colors or your own colors selected with "define" are used.
- Background color → Background color of the text line

- Dialog position → The position of the control dialog in the runtime system can be set (default = top left)
- Calculated size → deactivated: The text field frame can be changed later (default = active).

Once the properties have been specified, the system branches to the Specify data source dialog (refer to the section further below).

6.1.2.7 Hexadecimal

The cursor changes appearance to the following: 

Move the cursor to the required position and click the left mouse button. The **Hexadecimal** dialog box opens with the following input boxes:

- Number of characters → Maximum number of characters that can be displayed The width of the text field is based on a standard character width according to the selected font. For specific values, the frame width may have to be adjusted after "Calculated size" is deactivated (see below).
- Top entry limit → Since this is a tag type that can be controlled by the operator, the top entry limit can be specified.
- Bottom entry limit → If an input value may not fall below a certain limit, this can be specified here.
- Alignment (Left aligned/Centered/Right aligned) → default = Right aligned
- Entry text: →
  The text entered here will be displayed when working with tags in the runtime system. The combo box allows you to store different input texts and to select them again and again. An input text which is entered as new is entered immediately in the combo box, but it is only stored if the OK key is pressed and the dialog is closed without any error messages.
- Text color → Color of the text characters Either predefined colors or your own colors selected with "define" are used.
- Background color → Background color of the text line
- Dialog position → The position of the control dialog in the runtime system can be set (default = top left)
- Calculated size → deactivated: The text field frame can be changed later (default = active).

Once the properties have been specified, the system branches to the Specify data source dialog (refer to the section further below).

6.1.2.8 Bit

The cursor changes to the status: 

Move the cursor to the required position and click the left mouse button. The **Specify bit variable** dialog opens with the following input boxes:

- In the combo box 'Bit symbols', the available symbols are entered.
  The image construction anticipates the symbols in the folder <proj-path>\symbole\bit and accepts only the files which are included here. There is a number of predefined norm symbols (for example tank full/empty). It is possible to include any further, self-defined symbols in this folder. Image construction is now informed of the requested symbol palette by clicking a file. The possible representations of the selected bit are contained in a symbol palette.
- Entry text: →
  The text entered here will be displayed when working with tags in the runtime system. The combo box allows you to store different input texts and to select them again and again. An input text which is entered as new is entered immediately in the combo box, but it is only stored if the OK key is pressed and the dialog is closed without any error messages.
- Dialog position → The position of the control dialog in the runtime system can be set (default = top left)
- flash → The flash attribute can be activated separately for each status

Table 6-2 State assignment:

BIT	State
1st Symbol	Off
2nd Symbol	On

Once the properties have been specified, the system branches to the Specify data source dialog (refer to the section further below).

6.1.2.9 ICM

The cursor changes to the status: 

Set the cursor on the requested variable position and press the left mouse button. The **Specify ICM variable** dialog appears with the following input boxes:

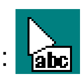
- In the combo box 'ICM symbols', the available symbols are listed.
  The image construction anticipates the symbols in the folder <proj-path>\symbole\esg and accepts only the files which are included here. This directory contains numerous predefined standard icons such as valves, motors etc. It is possible to add further self-defined icons to this directory. Image construction is now informed of the requested symbol palette by clicking a file. The eight possible representations of the selected individual control module are contained in a symbol palette.
- Entry text: →
  The text entered here will be displayed when working with tags in the runtime system. The combo box allows you to store different input texts and to select them again and again. An input text which is entered as new is entered immediately in the combo box, but it is only stored if the OK key is pressed and the dialog is closed without any error messages.
- Dialog position → The position of the control dialog in the runtime system can be set (default = top left)
- flash → The flash attribute can be activated separately for each status

Table 6-3 State assignment:

ICM	State
1. Symbol	Off
2. Symbol	Drive off
3. Symbol	Drive in
4. Symbol	On
5. Symbol	Error off
6. Symbol	Error drive off
7. Symbol	Error Drive in
8. Symbol	Error On

Once the properties have been specified, the system branches to the Specify data source dialog (refer to the section further below).

6.1.2.10 Text

The cursor changes appearance to the following: 

Set the cursor on the required position and press the left mouse button. The dialog 'Specify text variable' appears. In this dialog the following adjustments can be made:

- Display
  Selection of the text object type with which the visualization object is to be connected. The following types can be selected:
  - Text - Text file selectable (PCU-related or global); interconnection to any data object
  - Material - Text file fixed "Material.txt" (global); interconnection preferably to DFM (recipe setpoint)
  - Tank - Text file fixed "Tank.txt" (PCU-related); interconnection preferably to DFM (recipe setpoint)
  - Tank location - Text file fixed "rcs_loc.txt" (global); interconnection preferably to DFM (recipe setpoint)
  - Text (Bitfield) - Text file selectable (PCU-related or global); interconnection preferably to DFM (recipe setpoint)
- Offset → from which text line the text is displayed
- Number → Number of text lines which should appear in the picture
- Number of characters →
  Maximum number of characters that can be displayed The width of the text field is based on a standard character width according to the selected font. For specific text contents, the frame width may have to be adjusted after "Calculated size" is deactivated (see below).
- Reduction factor → Divider between data source and text file
- Alignment (Left aligned/Centered/Right aligned) → default = Left aligned

- Entry text: →
  The text entered here will be displayed when working with tags in the runtime system. The combo box allows you to store different input texts and to select them again and again. An input text which is entered as new is entered immediately in the combo box, but it is only stored if the OK key is pressed and the dialog is closed without any error messages.
- Text file → Text file which contains the requested text
- Text color → Color of the text characters (in each case for default and process value = 0)
- Background color → Background color of the text line (in each case for default and process value = 0)
- Font → The Windows font dialog opens.
- Dialog position → The position of the control dialog in the runtime system can be set (default = top left)
- Calculated size (inactive/active) → default = active. In case of "inactive", the text field frame can be changed later.

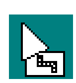
With the OK button these adjustments are accepted and the system will branch into the dialog 'Specify data source' in which the requested variable can be connected.

The value of the connected variable at runtime corresponds to the following equation:

- Line number = variable value/Reduction factor + Text offset
- where a specified 0 reduction factor is interpreted as 1.

6.1.2.11 Line

The line type must be specified in a defined background color in the background image or in the static picture object. The line object changes all the pixels within the position border that comply exactly with the background color to a target color. This occurs independently of the value of a connected variable.

The cursor changes to the status: 

A position border is initially drawn over the line type by pick & place, left-clicking, and dragging the mouse.

The **Specify line** dialog appears with the following input boxes:

- In this dialog eight colors (eight states of the line) can be assigned to the line. Either predefined colors or your own colors selected with "define" are used.
- The background color of the line type can be defined using the "define" color selection and the "Pipette" function.
- The target color is selected independently of the value of the connected variables. The eight statuses of the line are determined from the 4 bits of the low byte of the interconnected variable; in other words value 1 = color 0 , value 2 equals color 1, ... , value 8 ("0000 1000") = color 7.
- Once the properties have been specified, the system branches to the Specify data source dialog (see below).

6.1.2.12 Bar

The bar type must be specified in a defined base color in the background image or in the static picture object. The bar object changes all the pixels within the position border that comply exactly with the background color to a target color. This occurs independently of the value of a connected variable and certain limit values.

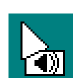
The cursor changes to the status: 

A position border is initially drawn over the bar type via pick & place, left click, and dragging the mouse.

The **Specify bar** dialog appears with the following input boxes:

- In this dialog, lower and upper limits, limit value1 and limit value2, as well as base color, color1, color2 and background color of the bar can be specified.
- The background color of the bar form can be defined using the "define" color selection and the "Pipette" function.
- For the basic color, color 1 and color 2 flashing attributes can be defined.
- Direction → Defines one of the four possible directions in which the bar is filled with accumulating variable values (vertically from below, horizontally from the left, vertically from the top, horizontally from the right)
- The bar is shown in color 1 if lower limit < variable < limit value 1
- The bar is shown in the basic color if limit value 1 < variable < limit value 2
- The bar is shown in color 2 if limit value 2 < variable < upper limit.
- Once the properties have been specified, the system branches to the Specify data source dialog (see below).

6.1.2.13 Sound

The cursor changes to the status: 

Move the cursor to the requested position and press the left mouse button. The **Specify Wave-Files** dialog appears with the following input boxes:

- In the combo box Wave-Files, the available sounds are listed.
  These are the default files included in the `<proj-path>\sound` directory. This directory contains several predefined sounds.
- Using "...", a different wave file can be selected in an open file dialog.
- If you click the loudspeaker icon, the sound is played and can be checked.
- In the combo box Search symbol, the available sound symbols are listed.
  The image construction expects the sound symbols in the folder `<proj-path>\symbole\sound` and accepts only the files it contains. This directory contains several sample icons. It is possible to add other self-defined symbols to this folder.
- The Symbol file box shows two possible representations of the selected sound icon.

Once the properties have been specified, the system branches to the Specify data source dialog (refer to the section further below).

6.1.2.14 'Specify data sources' (dialog)

Basic principles

This dialog determines all data needed by the system in order to store an explicit assignment of standard variables of the types Integer, Hexadecimal, Bit, ICM, Text, Line and Bar to the PCU data elements.

Dialog properties

The variable which has already been specified is connected to the indicated source in the runtime system and receives the current process value from it.

Image construction supports the concept of the object-oriented configuration and allows three different possibilities for the specification of an S7 data source. Once a source indication is specified, it is stored by the system and is changeable and can be called up any time by a simple click on the requested variable.

In general the name with the highest abstraction level is stored. The requested addressing mode is to be chosen. As the default setting 'object name' is preset. There are the following choices:

- Direct object address
- Direct addressing
- Indirect addressing

The dialog changes depending on the selected type of source. The individual input fields can be jumped between via the tab key.

Single updating

A variable defined with this feature is updated only once when building up the process image. After that it doesn't happen a second time. An operation of this variable won't be sent to the PCU, it is only displayed in the IOS.

Operation level

Here the operation level of the variables is entered (see chapter General (Page 154)).

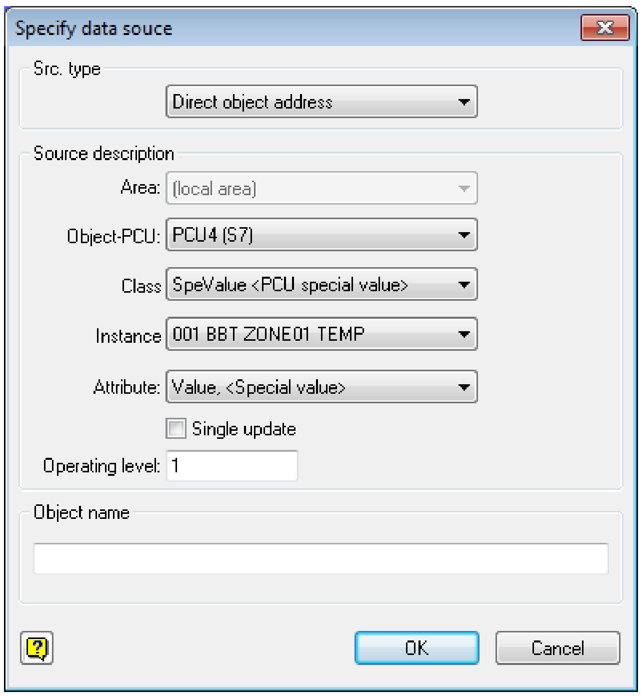
Special features:

- Operation level = -1: a direct operation is possible, i.e. values can be entered or switch-overs can be carried out without any further dialog window.
- Operation level = 0 or <empty>: there are no possible inputs or switch-overs, i.e. only monitoring is possible.
- Operation level = 1...255: Operation or value inputs via dialog windows can be protected by password.
- Operation level = <Name of the function level> : the assigned operation level ("User administration / Function levels") is replaced

Direct object address

This type of connection enables source indications to be specified by selecting object class/ instance attribute.

A specific data element in the memory area of the PCU is described by these details. However, internally all source indications are stored as 'Direct addresses', which are hidden to the user of the system.

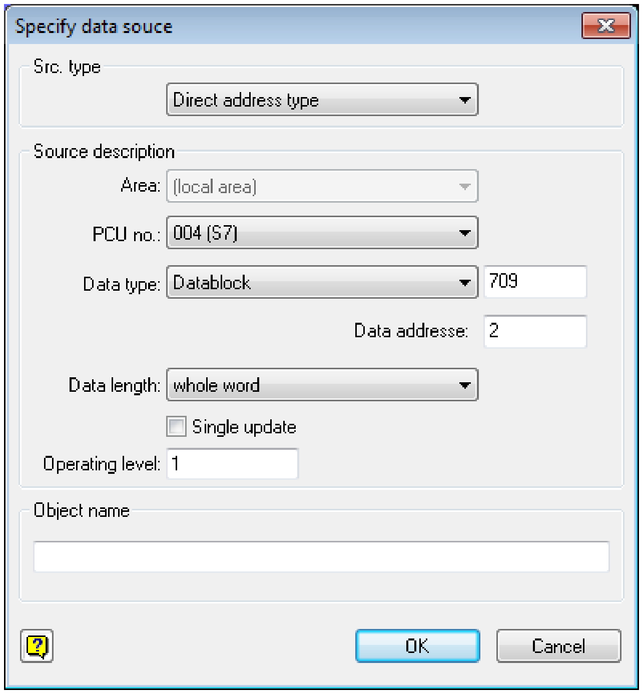


Available selection fields for object addressing:

- Object - PCU: PCU to which the connection is to be made.
- Class - Object class / block type
- Instance - Instance of the block type
- Attribute - Attribute / value of the selected instance

Direct addressing

This type of connection enables the connection of a variable by indication of a direct PCU address.



The following selection boxes and fields are available for inputting a valid direct address for the connected PCU:

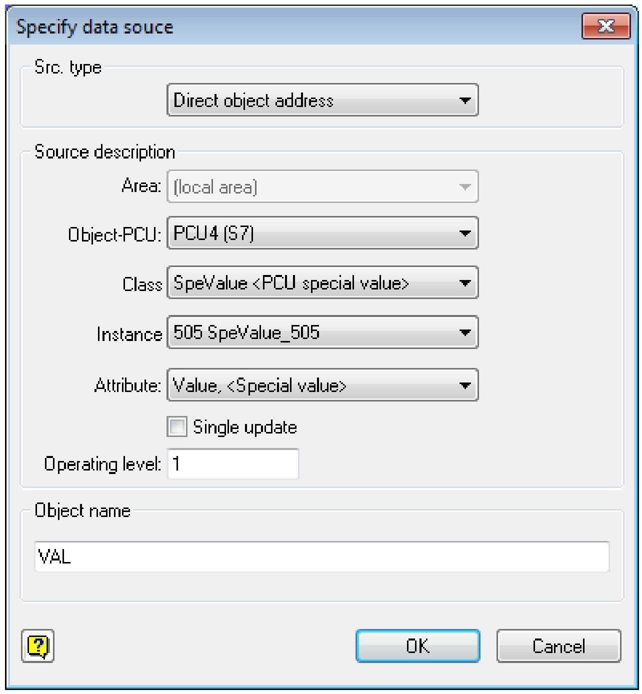
- PCU no.
  Specifies the number of the PCU to which a connection should be established. Here the requested PCU is to be chosen.
- Data type:
  The desired block type is to be selected from the displayed combo box.
  It has to be considered that no block number may be indicated when block types 'flag word/byte', 'input word/byte' or 'output word/byte', are selected.
- Block number:
  If a block type was selected, the necessary block number must be entered here.
- Data address:
  It indicates the data byte number within the entered memory area which was indicated by the block number.
- Data length:
  It specifies the range, whether an individual bit, byte, word or double word of the variable is connected.

Indirect addressing

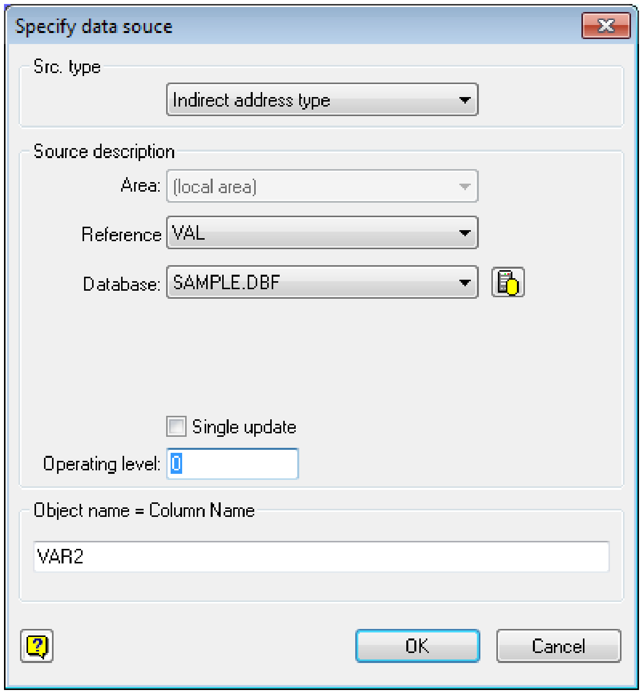
In order to be able to observe pumping processes in a process diagram, particular picture objects should be dynamically connected, depending on the tank or product path currently being used, to the physical PCU data elements. The representation of the variable is fixed here, the data sources have to be connected dynamically, however, depending on the reference variables.

To configure the indirect addressing of a group of picture objects, the following steps are necessary, whereby step 1 (reference variable) and step 3 (address table) apply together to the entire group:

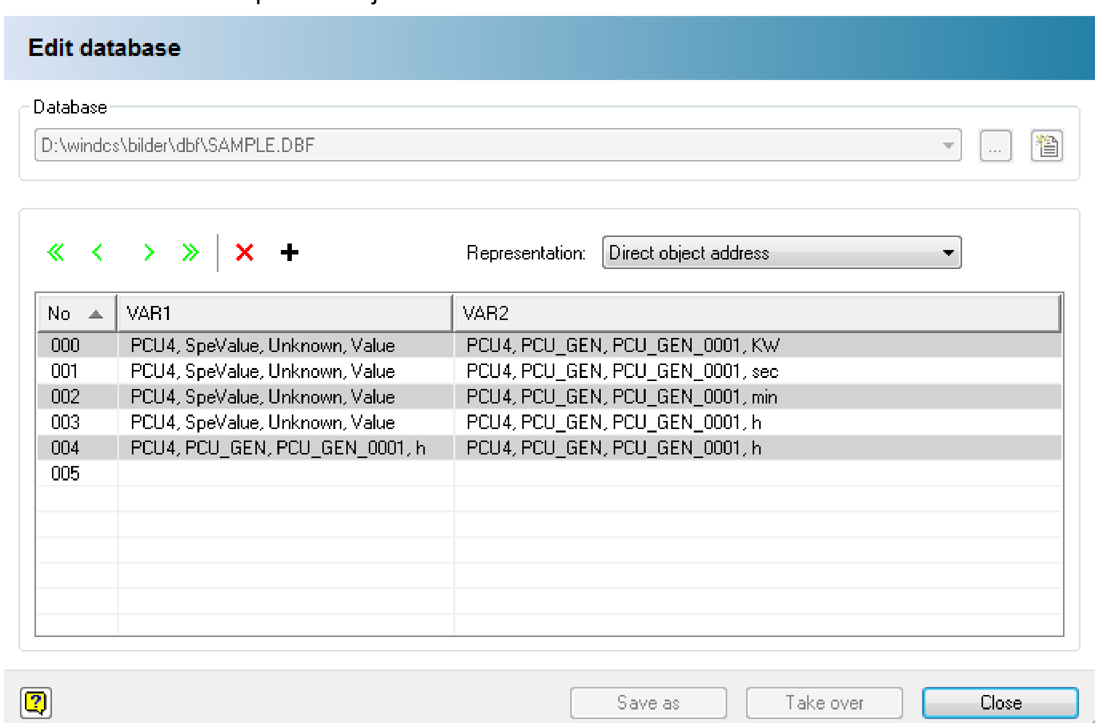
1. The reference variable is defined with the connection type 'Direct object addressing' and the attribute 'one-off updating'. A unique object name has to be issued for this.



2. The connection type "indirect addressing" is selected for the referencing picture objects / variables. In the field 'Reference variable', the object name of the previously defined reference variables has to be selected. In addition, the database file (*.DBF) with the address information has to be selected, and the object name of the picture object defined. The column in the table is referenced via this object name.



3. The button next to the selected database table takes you to the editor widow for the address information of the picture objects.



Structure of the table and editor functions

- The dbf-file to be edited is selected in the "Database" field. The list box contains the dbf files already disclosed. An existing dbf file not included in the list box can be opened via the '...' button.
- A new dbf file can be set up and added to the list.
- Columns or lines can be added or deleted in the table window. To do this, select the column/ line and then click '+' or 'x'.

Column "No." - cannot be edited, contains the line index. The corresponding lines are selected for the duration via the reference variables. The address information for the picture objects referenced in the columns is calculated from this.

The following applies to all other columns:

- The column name must correspond to the object name of the referenced picture objects (in the example "VAR1", "VAR2"). The column name can be edited via the context menu "Rename column".
- The field content per line contains the address information for the picture objects referenced. The field content can be edited via the button "..." in the selected field.

Note

The structure of the field content of a column varies depending on the type of picture object referenced, i.e. the field content is interested by the picture object itself and is not known in the editor dialog. Consequently, the project manager has to select one of the following settings via the "Display" selection box before editing field content:

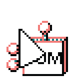
- For standard variables (VAL, HEX, Bit, ICM) → "Direct object addressing" or "Direct addressing"
- For controls (Analog, Digital, ICM, UNIT, ... → "Control instance addressing"

6.1.3 Controls

6.1.3.1 Overview of the controls

Basics

With the "Variables / Control" menu or by clicking on the corresponding icon in the toolbar the cursor changes to the status:



- Move the cursor to the required position in the process diagram and press the left mouse button.
- A dialog window appears with the controls defined in the system.
- After selecting and confirming with the "'OK" button, the control is placed on the process diagram.

The following control elements are available:

Name	Description
BRAUMAT/SISTAR Analog2 Control	Analog values 'Extended'
BRAUMAT/SISTAR Controller Control	Controller (PID, 3-point)
BRAUMAT/SISTAR Date-Time Control	Date / clock display (digital or analog)
BRAUMAT/SISTAR Digital Control	Digital values
BRAUMAT/SISTAR Entity Control	Function units / data grouping
BRAUMAT/SISTAR ICM Control	Individual control elements (valves, motors)
BRAUMAT/SISTAR Image Navigator Control	Process diagram change
BRAUMAT/SISTAR LineCtrl Control	Line status administration
BRAUMAT/SISTAR Message View Control	Messages
BRAUMAT/SISTAR Panel Control	Picture navigation / Application call
BRAUMAT/SISTAR Route Control	Route control
BRAUMAT/SISTAR Smart Unit Control	Unit information 'compact'
BRAUMAT/SISTAR Statusbyte Control	Status display
BRAUMAT/SISTAR TankCtrl Control	Tank status administration
BRAUMAT/SISTAR Unit Control Control	Unit operation

Global configuration data in "Controls.ini"

Certain initial values and display properties of the control elements are maintained globally in the "`"<proj-path>\sys\controls.ini"`" file.

Note

The IOS client stations always read this configuration file from the active standard server. Changes must therefore always be synchronized on both servers.

The following entries relate to properties in the "Extended" tab of the OCX design mode and can be displayed there and influenced individually for each instance.

- The file can be edited with the Windows Notepad.
- Changes to the settings become active the next time the "Process diagrams" or "Process diagram design" applications are started.
- There is a separate section in "`"Controls.ini"`" for each type:
  e.g. `[ANALOG], [DIGITAL], [CTRL3], [ICM3], ...`

The keys that follow allow division into the following categories in terms of the use in the control element:

1. Default for instance-specific settings
   The following keys are adopted as the initial value when adding a new OCX instance; this however remains instance-specific:
   “View” section: `FixedSize=, Transparent=, ToolTip=, DecimalPoint=`
   "Color settings" section: `ColorBorderWith=`
   “Control Dialog” section: `DlgModal=, DlgTopMost=, DlgPositionFixed=, DlgPositionX=, DlgPositionY=`
2. Central specification of global settings
   The following keys are used as the central specification for icon signaling in all OCX instances of this type when "global settings" is activated.
   “Signaling” section: `SignalManualMode=, SignalSimulation=, SignalNote=, Signalxxxxx=`
   "Color settings" section: `Colorxxxx=, ColorBackGround=, ColorBorder=`

3. Release of a button for program execution in the control elements `[ANALOG], [DIGITAL], [CTRL3], [ICM3], [LINECTRL], [TANKCTRL]`
   The button appears in the "Info" tab of the relevant control element. The configured program "exe/cmd/bat" is executed when clicked and receives the PCU, CID, instance no. and instance name. This allows actions suitable for the element to be called.

   Note

   The configured programs *.exe, *.cmd, *.bat and all the programs called there must exist on the project path `"<proj-path>\sys\..."` locally on every IOS.

   Example of an AboutCmd entry in the file `"Controls.ini"` (corresponds to the status when supplied):
   ```
   ;Path of program to be executed, if the "CMD"-Button of the About
   TAB is pressed
   AboutCmd=\\localhost\bm_proj\sys\about.cmd
   ;Text to be displayed at the "CMD"-Button
   AboutCmdLabel=CMD
   ```
   Example of a command file `"about.cmd"` (corresponds to the status when supplied).
   ```
   @echo off
   if %1:==: goto NOSPR
   if %2:==: goto NOSPR
   if %3:==: goto NOSPR
   if %4:==: goto NOSPR
   @echo call user.exe with
   @echo PCU-No:%1, ClassID:%2, InstID:%3, Name:%4
   pause
   goto ENDE
   :NOSPR
   @echo parameter error !
   pause
   :ENDE
   ```
4. Central coloring of the hand symbol in various control elements
   ```
   [Global]
   ;Use following manual bitmap for state picture
   ;0=gray color, 1=blue color, 2=red color
   ManualColor=2
   ```

6.1.3.2 Analog control

Basics

The control is used for monitoring and simulating the analog input and output values. In draft mode, the following tabs are available for connecting an analog variable and for setting the representation in the process diagram (=control dialog):

General tab

- Source type:
  The following source types are available:
  - Analog output value
  - Analog input value
  - Direct object addressing:
    Any block class may be selected with this source type. In addition the block instance as well as the desired attribute word is selected here.
  - Direct addressing
    With this source type a direct PCU address is connected. The selection options according to the following table are valid here:

Data type	Further specifications	Data format
Data block	DB number Index (byte address)	Bit 0 …7 (DBX) Byte (DBB) Word (DBW) Double word (DBD)
Flag	Index (byte address)	Bit 0 …7 (M) Byte (MB) Word (MW) Double word (MD)
Input	Index (byte address)	Bit 0 …7 (I) Byte (IB) Word (IW) Double word (ID)
Output	Index (byte address)	Bit 0 …7 (Q) Byte (QB) Word (QW) Double word (QD)

- Source:
  The individual variable may be selected with the following combo boxes:
  - Area → Area of the variables to be interconnected
  - PCU → PCU of the selected AREA
  - Instance → Object instance in the selected PCU
- Symbol file:
  The symbol of the faceplate can be selected here

Extended tab

Display:

- Original size
  The size of the icon cannot be changed.
- Display of values
  Instead of the icon, the current value of the connected variable is shown.
- Transparent
  The icon is displayed transparently.
- Tooltip
  Information related to the interconnected variable is displayed by touching the icon with the mouse pointer.
- The display attributes of the variable value of the interconnected data point (type, unit, decimal point, number of characters, upper limit, lower limit) can be enabled or disabled depending on the data source.
- With direct object addressing plus the direct addressing the 'Lower /Upper limit' fields are activated. Correct values should always be entered here. Only then are the functions in the open control of the "General" tab (bar and simulation function) and "Trend" tab (trend curve) available.
  - The desired Lower/Upper limit must be entered for word byte.
  - For bit values the lower limit entered should = 0 and the upper limit should = 1.
  - Otherwise the current value of the connected variables can only be seen in the control symbol.

Signaling:

- Global settings
  Switchover between global ("`Controls.ini`") and instance-specific setting
- Simulation
  Simulation mode is indicated by an '!S' icon next to the icon.
- Note
  A note change to this block instance can be indicated next to the icon. The corresponding selection box has to be selected in the control dialog / tab 'About' / Note.
- The options "Simulation" and "Note" are only valid for direct object addressing, they have no function for the other source types.

Color settings:

- The representation type of the value to be displayed can be set.
- The adjustments are active if the option "Display value" is selected.

- The colors of the value, the background and the frames as well as the number of digits, unit, decimal places and width of the frame to be displayed can be changed here.
- Is also possible to switch over between global (`"Controls.ini"`) and the instance-specific setting

Note

The border width `ColorBorderWith=` can only be set on an instance-specific basis, i.e. the change is always enabled.

Control dialog:

- Modal:
  The control dialog is shown as a modal window, i.e. the user cannot work with the same application in other windows as long as the dialog window is displayed.
- Always on top
- Dialog position X: ... Y:
  The position of the control dialog in the process diagram can be prescribed here.

Monitoring tab

It is also possible to signal the status of a measured value as a colored frame at the analog control. The source of the status value (max. 32Bit) is configured in the ‘Monitoring’ tab. The available source types for the status value are ‘Direct object address’ and ‘Direct address type’.

The max. 32 different border colors can be defined centrally in the 'controls.ini' file or individually in the dialog of the instance.

File: `...<proj-path>\sys\controls.ini`	

Key:
`[Analog]`
`ColorStatusBorder<Bit>=<R>,<G>,<B>`	

<Bit>:	Bit 1 to 32
<R>,<G>,<B>:	RGB values for red, green, blue
Default value	0,0,0 (black)
Example:	
`[Analog]`	
`ColorStatusBorder1=128,0,0`	(Bit 1, dark red)
`ColorStatusBorder2=0,255,0`	(Bit 2, light green)
…	
`ColorStatusBorder32=0,0,0`	(Bit 32, black)

The border color is set according to the following rule:

- Status value=0:
  Standard border color configured for the control
- Status value>0:
  Color for the highest set bit, as defined in controls.ini.

Trend Tab

In the 'Trend' tab, the scaling of the time- and y-axis is defined.

- Duration:
  Time range, with which the changes to the connected variable are recorded.
- Line distance/Y:
  Scaling of the Y-axis
- Line distance/time:
  Scaling of the time-axis.
- Line color:
  Selection of the color of the curve sequence.

Fonts tab

The font type, style and size may be selected with this tab

Note

This faceplate does not support symbolic characters such as "Ω".

6.1.3.3 Controller Control

Basic principles

The control is used for monitoring and simulating the block instances controllers. In draft mode, the following tabs are available for connecting an analog variable and for setting the representation in the process diagram (=control dialog):

General tab

- Source type:
  In the ‘source type’ area the PID or 3-Step controller type is defined.
- Source:
  The individual variable may be selected with the following combo boxes:
  - Area → Area of the variables to be interconnected
  - PCU → PCU of the selected AREA
  - Instance → Object instance in the selected PCU

- Display mode:
  - Unit:
    here the physical unit can be entered.
  - Dec.Point:
    Display format with decimal point.
- Symbol file:
  The symbol for the control be selected here

Extended tab

The 'Extended' tab can be used to define the display type of the Digital Control Faceplate in the runtime system (process images).

View:

- Fixed size:
  The size of the control icon cannot be changed.
- Transparent:
  The control icon is displayed transparently.
- Tooltip
  Information related to the connected variable is displayed by touching the control icon with the mouse.
- PID: Y 0..100:
  Scaling of Y values in the range 0..100 (PID only)

Signaling:

- Global settings
  Switchover between global (`("Controls.ini"`) and instance-specific setting
- Manual mode
  Manual mode is indicated with a 'hand' icon next to the controller icon.
- Internal/external
  Internal mode is signalized with an 'i' icon next to the control icon.
- Y-Tracking
  The state of the block attribute 'YN' (YN flag) is signalized with a '=' icon next to the control icon.
- Note
  A note change to this block instance can be indicated next to the icon. The corresponding selection box has to be selected in the control dialog / tab 'About' / Note.

Control dialog:

The position of the control is specified:

- Modal
  A new control instance cannot be opened without closing the first one.
- Always on top
- Dialog position: X: ... Y: ...
  The position of the control dialog in the process diagram can be prescribed.

Tab Flash

The adjustments for the overflow and underflow limits are defined here for the following values:

- Process valueX
- Manipulated valueY
- SetpointW

The values are displayed flashing if the options are active.

Trend Tab

- Duration:
  a range of time is entered, in which the changes of the connected variable are recorded.
- Line distance/time:
  Scaling of the time-axis.
- Line distance Y:
  Scaling of the Y-axis
- Selection boxes for activating the trend curves and for prescribing the colors:
  Manipulated variable Y, setpoint W, actual value X, fault signal, X – difference value)

6.1.3.4 Date-Time Control

Basics

With this control a clock may be displayed in an analog or digital view. In addition, you have the option of having the date and the current operator displayed.

General tab

With this tab the view type of the clock may be defined

- Show analog clock
- Show digital clock
- Show date field

Annotations: The display format is derived from the actual windows system settings. At least one of the options must be selected.

- 'Show current system user' – shows the user currently logged in

Extended tab

You can set two color properties for the clock:

Background	Color of the analog clock's background or face.
Font color	Color setting for the clock lines and for the font displaying the date and the user.

Fonts tab

The font, font style and font size for date, time and operator are set on the "Fonts" tab.

Note

This faceplate does not support symbolic characters such as "Ω".

6.1.3.5 Digital Control

Basic principles

The control is used for watching and simulating the digital inputs and outputs of the timer. In draft mode, the following tabs are available for connecting a binary variable and for setting the representation in the process diagram (=control dialog):

General tab

- Source type
  The type 'Timer group 1' or 'Timer group 2' is selected here.
- Source
  The individual variable may be selected with the following combo boxes:
  - Area → Area of the variables to be interconnected
  - PCU → PCU of the selected AREA
  - Instance → Object instance in the selected PCU
- Symbol file
  An individual icon can be prescribed here.

Extended tab

View:

- Original size
  The size of the icon cannot be changed.
- Transparent
  The icon is displayed transparently.
- Tooltip:
  Information related to the interconnected variable is displayed by touching the icon with the mouse pointer.

Signaling:

- Global settings
  Switchover between global (`"Controls.ini"`) and instance-specific setting
- Simulation
  Simulation mode is indicated by an '!S' icon next to the icon.
- Note
  A note change to this block instance can be indicated next to the icon. The corresponding selection box has to be selected in the control dialog / tab 'About' / Note.

Icon represents:

The signal input, the block's positive or negative signal output, can be selected for the representation of the icon (in icon, out icon). The attribute "Flashing" can be selected for these signal states.

Control dialog:

- Modal:
  The control dialog is shown as a modal window, i.e. the user cannot work with the same application in other windows as long as the dialog window is displayed.
- Always on top
- Dialog position X: ... Y:
  The position of the control dialog in the process diagram can be prescribed here.

Trend Tab

In the tab, the scaling of the time axis and the trend color is defined.

- Duration:
  a time period is entered in which the changes to the connected variable are recorded.
- Line interlace:
  Scaling of the time-axis.
- Line color:
  Selection of the colors for the curves 'Input', 'Positive output' and 'Negative output'

6.1.3.6 Entity Control

Basic principles

With this control, a set of PCU variables may be displayed in a tabular format.

General Tab

The individual entity object is selected with the following combo boxes:

- AREA → plant area for the entity object to be connected
- Group → entity type of the selected AREA
  Entity types can be created with the 'Entity-Definition' application.
- Entity → Entity instance within the selected type
  Creation of the entity instances and allocation to the entity types occurs using the 'Entity-Definition' application.
- Symbol file:
  An individual icon can be prescribed here.

Extended tab

View:

- Fixed size:
  The size of the icon cannot be changed.
- Transparent:
  The icon is displayed transparently.
- Tooltip:
  Information related to the connected variable is displayed by touching the icon with the mouse in the runtime system.

Color settings:

- The font color for the data actual values can be selected here.

Control dialog:

- Modal:
  The control dialog is shown as a modal window, i.e. the user cannot work with the same application in other windows as long as the dialog window is displayed.
- Always On Top
- Dialog position X: ... Y:
  The position of the control dialog in the process diagram can be prescribed here.

Entity Control configuration

Basics

The data to be displayed in the entity control may be structured. Entity types are defined for this and entity instances are then allocated to these. An entity template which defines a certain set of data points in the PCU is allocated to each instance. The data points of the template allocated are displayed in control dialog of the entity instance.

Entity type

The type summarizes entities. Every type can contain up to 255 entities. A grouping of similar entities in the same type is recommended.

Note

Basic configuration via the 'Configuration / Maintenance groups' application

The types as well as the instances allocated of the data entities can initially be configured with the "Maintenance groups" (Entitydef.exe) application. There you normally configure the assignment of the ICMs to the maintenance groups. If allocation of the ICMs is omitted then an entity type defined this way including the allocated entities can also be used for the data entities. In this case subsequent additions must just be made to the definition files created 'Type_xxx.ini' (templates, address) and the entity templates must be created again using the 'Temp_xxx.ini' files.

Entity Template

- Parameter definition:
  The template contains a list of parameters to be displayed in the data entity control (‘Properties’ tab). For every parameter the following properties may be defined:
  - the access (R/W)
  - the data type (fixed-point number; Bin; Hex; Real; Text list)
  - the name
  - the dimension name
  - the decimal point
  - minimum and maximum input limits
  - the file name for text list
  - address; Offset in the DB or absolute address
- Status definition
  From the parameters list, one value can be defined as a status word to be displayed in the 'Status' view. It can be defined for the status.
  - Number of the parameter
  - Meaning of Bit 0 - Bit 31 via information of a text file. This text file contains in line 0..31 the text strings which correspond to the bits.

Relative addressing

Normally, only the offsets to the data elements are stored in the template within the PCU data block. The base address (PCU, DB, DBW) is stored in the relevant entity instance via key `'Address'`. With this, a structured data set is assigned in the PCU. The entity instances can also be stored in different PCUs.

Fixed addressing

In the template a fixed address can be stored, too. In this case the template is only used for one entity.

Data storage

Item	Definition file	Description
Entity configuration	'<proj-path>\Entities\entities.ini'	Contains the number of types and templates
Entity types	'<proj-path>\texte.x\EntTypes.txt' '<proj-path>\Entities\Type_xxx.ini'	Description (language depend-ent) Definition / xxx = number of the entity type
Entity templates	'<proj-path>\Entities\Temp_xxx.ini'	Definition / xxx = number of the entity template
Entity instances	'<proj-path>\texte.x\Ent_xxx.txt'	Description (language depend-ent) xxx = number of the entity type

Description of the Configuration files

entities.ini

```
[Global]

;Number of types to be supported
Types=3
Templates=4
```

This means that the following files exist in the '<proj-path>\Entities\...' folder:

Type_001.INI ...... Type_003.INI
Temp_001.INI ...... Temp_004.INI

Type_xxx.ini

```
[GLOBAL]
Entities=3

[Entity_001]
Template=1
;Address=pcu,db,dw
Address=1,4100,34
```

```
[Entity_002]
Template=1
Address=1,1000,5

[Entity_003]
Template=3
Address=1,1000,6
```

The individual entities of the type are entered here. One template and one absolute address is assigned to each entity. This address is used as the base for the template parameters (if the template parameters are not absolute).

Temp_xxx.ini

```
[Global]
Parameters=4

Parameter1=SW/TEXT/BIN/HEX/
REAL,Name,dim,dep,min,max,file,Offset:DBBx/DBXx.y/DBWx/DBDx

Parameter2=SW/TEXT/BIN/HEX/
REAL,Name,dim,dep,min,max,file,Abs:pcu,db,DBBx/DBXx.y/DBWx/DBDx

Parameter3=R,SW/TEXT/BIN/HEX/
REAL,Name,dim,dep,min,max,file,Offset:DBBx/DBXx.y/DBWx/DBDx

Parameter4=W,SW/TEXT/BIN/HEX/
REAL,Name,dim,dep,min,max,file,Offset:DBBx/DBXx.y/DBWx/DBDx

[Status]
Parameter=2
;Textfile containing status texts
File=enttempl.txt
```

First of all the number of parameters and then the definitions are provided in the `[GLOBAL]` section. These parameters are displayed in the 'Properties' tab.

- "R","W" as a leading sign for read only or writeable are optional. Default is "W".

Now the type of parameter is indicated:

- SW=decimal possibly with decimal point
- TEXT=Text list
- BIN=binary input
- HEX=hexadecimal input
- REAL=real input

Then the character strings for name and unit follow, the values for decimal point, minimum and maximum (e.g. -100 / 100.4 / 100.45), the file name for the texts, (for "TEXT" format), and the address at the end.

This address can be identified with "Abs:" (= absolute address) instead of with "Offset:". The base address of the entity is without any importance in this case. Instead, you also need to specify the PCU and DB with the address parameters "pcu" and "db".

Note:

- All attributes should always be indicated.
- Empty attributes must contain a blank at least!
- Changes in parameters or attributes are only accepted in 'LzSys.exe' after a restart.
- The parameter text files are also stored under '<proj-path>\Texte.x' in a language-dependent way.

The `[Status]` section contains the following specifications: These parameters are displayed in the 'Status' tab.

- Parameter=2
  Defines the parameter from the parameter list that is to be interpreted in the status view. The text and the binary state are displayed for each bit.
- File=
  Determines the text file (e.g. '<proj-path>\texte.x\EntTempl.txt') with the status texts of the individual bit

6.1.3.7 ICM Control

Basics

The control is used to monitor individual control elements and to simulate their states.

Version as seat lifting block:

The SLB class (Seat Lifting Block) is available for controlling seat lifting blocks (see section SLB - Seat lifting for double seat valves (Page 357)). The complete functionality is achieved through the connection of an SLB instance to an ICM instance (parameterized reference, one-to-one relationship). Visualization and operation of the two connected block instances takes place over the same ICM OCX; however, in this case must be interconnected to the SLB instance.

In draft mode, the following tabs are available for interconnecting an ICM control and for setting the representation in the process picture (=control dialog):

General tab

- Source type
  - ICM → Standard single control element (motor, valve)
  - Double seat valve → Combination of SLB and ICM instance. The additional "Seat lifting" tab becomes visible during runtime for operator control and monitoring of the SLB. **Notice: For this application, the OCX must be interconnected to the SLB instance instead of the ICM.**
- Source:
  The individual variable may be selected with the following combo boxes:
  - Area → Area of the object instance to be interconnected
  - PCU → PCU of the selected AREA
  - Instance → Object instance in the selected PCU
- Symbol file:
  The symbol for the control be selected here
- flashing:
  The state, in which the ICM will flash in the runtime system, can be determined here.

Extended tab

The 'Extended' tab can be used to define the display type of the Digital Control Faceplate in the runtime system (process images).

Display:

- Fixed size:
  The size of the icon cannot be changed.
- Transparent:
  The icon is displayed transparently.
- Tooltip:
  Information related to the interconnected variable is displayed by touching the icon with the mouse pointer.

Signaling

- Global settings
  Switchover between global (`"Controls.ini"`) and instance-specific setting
- Manual mode
  Manual mode is indicated with a 'hand' icon next to the controller icon.
- Simulation
  Simulation mode is indicated by an '!S' icon next to the icon.
- Interlock
  The interlocking status is signalized next to the control icon.

- Maintenance
  Maintenance mode is signalized next to the control icon.
- Note
  A note change to this block instance can be indicated next to the icon. The corresponding selection box has to be selected in the control dialog / tab 'About' / Note.

Border color settings

- The border around the icon can be shown in the colors defined here for the ICM statuses 'Automatic', 'Manual', 'Maintenance' and 'ABM' (affected by maintenance).

Note

The border width `ColorBorderWith=` can only be set on an instance-specific basis, i.e. the change is always enabled.

Control dialog:

- Modal:
  The control dialog is shown as a modal window, i.e. the user cannot work with the same application in other windows as long as the dialog window is displayed.
- Always on top
- Dialog position X: ... Y:
  The position of the control dialog in the process diagram can be prescribed here.

Additional settings

You can configure the signal indicating a missing operation interlock (OI) centrally in the 'controls.ini' file.

File:	<proj-path>\sys\controls.ini
Key:	`[ICM3]` `SignalInterlock=1`
Value 0:	missing OI not indicated
Value 1:	missing OI indicated (default)

6.1.3.8 Image Navigator Control

Basic principles

The image change control encapsulates the functionality of the standard variables “static text” and “image change” in a control. Therefore the settings follow closely these two standard variables, which are distributed on three register tabs:

Tab General:

Convert process diagram to:

- Area:
  The image will be loaded from this area.
- Image:
  Path of the picture for switching over
- Shortcut:
  Shortcut which leads to automatic switch-over.
- New instance:
  This property indicates whether a new instance of the runtime system has to be launched.

Display:

- Text:
  Static text, which is displayed in the control.
- Tooltip:
  A tooltip with the information for the properties “Image” and “Shortcut” is displayed, if this property is set.
- Fixed size:
  If the setting “fixed size” is activated, the width of the control is given by the property “Width” and the height is adjusted according to the font. Otherwise, the size of the OCX can be modified as desired in the image construction.

Format tab

- Colors:
  Both text color and background color can be adjusted for the control. A transparent control can be achieved by activating the checkbox “Transparent”.
- Border:
  Color and width of the border can be modified by two combo boxes.
- Alignment:
  Horizontal and vertical alignment of the text.

Font tab

The font, style and font size for the display text are set here.

6.1.3.9 LineCtrl Control

Basics

The following functions can be checked and controlled for a pipeline using the control:

- Display and operation (manual operation) and monitoring of the runtimes for a freely definable line status
- Display of the associated control group and control type
- Display and control (manual mode) of the line status
- Manual/Automatic mode change

The different modes and operating functions are specified during image configuration using the following properties dialogs:

General tab

- Source:
  The required object instance can be selected with the following combo boxes:
  - Area → Area of the variables to be interconnected
  - PCU → PCU of the selected AREA
  - Instance → Object instance in the selected PCU
- Symbol file:
  An individual icon for the control in the runtime system can be set here.

Extended tab

Display:

- Original size:
  The size of the icon corresponds to the bitmap dimensions ("Icon" layout) or the number of lines and 32 character width cannot be changed.
- Transparent:
  The icon is displayed transparently.
- Tooltip:
  Information related to the interconnected variable is displayed by touching the icon with the mouse pointer.
- Grid lines:
  Show separating lines in the icon's value table

Signaling:

- global signaling
  enable all available signals
- Manual mode
  Manual mode is indicated by a "hand" icon next to the block icon.

- Manual mode material change
  A material change is indicated by an "M" icon next to the block icon.
- Note
  A note change to this block instance can be indicated next to the icon. To be able to do this, the relevant list box "Signal note in process picture" must be selected in the control dialog / "About" tab / "Note" button.

Color settings

- The color attributes for the remaining time display field in the control dialog can be set here.

Note

The border width `ColorBorderWith=` can only be set on an instance-specific basis, i.e. the change is always enabled.

Control dialog:

- Modal:
  The runtime is shown as a modal window, i.e. the user cannot work with the same application in other windows as long as the dialog window is displayed.
- Always on top
- Dialog position X: ... Y:
  The position of the control dialog in the process diagram can be prescribed here.

Layout tab

The display properties of the block icon are defined here:

- Shape:
  - Icon
  - Value view
  - Value view with icon
- Value table - Selection, font/size and sequence for the parameters:
  - Line name
  - Material
  - Line status
  - Time monitoring
  - Batch information
  - Recipe category
  - Recipe
  - Batch number
  - Order number

Tips for designing the faceplate icon:

- The line, column and text width for the value view can be specified for each parameter.
- The text width only has an effect on the column width when the attribute "Original size" is disabled in the "Advanced" tab
  (otherwise, the column width is always limited to 32 characters with the selected font and font size)
- You must pay attention to the validity of line and column especially when using multiple columns because fields may be overwritten or not displayed at all.
- Gaps and/or lines or columns that are not used are not displayed.
- The result of all layout changes can be tracked simultaneously in the icon window.
- The icon border and thus the size during runtime must be adapted manually to the configured value table with "Original size=Off".

6.1.3.10 Message View Control

Basic principles:

The control selects and displays the messages of one plant area. For better clarity, the number of messages can be limited by using filter settings.

Filter settings:

- AREA: → Area (= server) selection from which the message archives are read in.
- The messages are selected according to the SQL filter string in the 'Filter expression' box. Prerequisite is a formally correct expression, which can be examined with the ‘Check’ button. If the filter string is empty, all messages of the active area are then displayed.
- The attributes listed in the ‘Field identifier’ list box are available as filter criteria. The selected item is inserted into the filter string at the cursor position by double-clicking. Compare and Logical operators as well as brackets have to be added manually.
- With the ‘Check’ button, the syntactic correctness of the current filter string can be checked at any time.
- With the button ‘Simple unit filter’, a new dialog with selection boxes can be called. With this a filter string is created without SQL knowledge by selecting the corresponding fields and boxes - however only for one PCU/Unit combination.
- The so created filter string can be modified with the normal edit functions (Mark, Delete, Copy/Ctrl-C, Paste/Ctrl-V).

Note

- An already present filter string (manually edited or generated) is completely replaced by clicking the 'OK' button of the dialog ‘Simple unit filter’. If needed later, this string should be copied for example into a ‘Notepad’ window before calling the dialog.
- For numerical data, text strings can be indicated (e.g. ‘001') or direct numbers (e.g. 1)
- The numerical data is preferable in this case since leading zeros or blanks remain unconsidered and thus a secure comparison is possible.
- As wildcard character only the ‘*' character is admitted and indeed only as the right-most string character (example: PCU = ’00*’)
- The following compare operators are available:
- =, <> for text strings
  =, <>, <, >, <=. >= for numbers
- The logical operators AND/OR are available
- Additionally for the ‘Type’ (message type) identifier, the IN operator is possible (e.g. Type IN ‘FBM’) which describes a shortened OR operation of the given message types
- All the filter string elements (identifiers, operators, numbers/text strings) should be separated by blank characters, e.g.:
- PCU=1 leads to a syntax error, PCU = 1 is OK.

6.1.3.11 Panel Control

Basics

With the control, applications can be started and/or already running applications can be brought into the display foreground. In addition, fast picture changes via the navigation bar can be implemented. A time display with a configurable display format rounds up the control.

General tab

With this property page the appearance of the buttons is configured.

Tab navigation panel

The navigation panel can be used to configure the image navigation bar.

- Display navigation field
- Align right
  This affects the display of the navigation panel in terms of whether it is located left or right of the above-mentioned application panel

- Show Tooltips
- Control file:
  Several different files can be selected for each faceplate using the file selection dialog (default path and file name '...<proj-path>\texte.x\paneltree.ini').
  For editing, the standard editor of the operating system can be opened with the button on the right.

Example navigation bar:

Content paneltree.ini:

```
Allgemein\\over.bik Drinktec 20xx
begin;MALZ
 Sudhaus\MALT-6.bik; 2; Malztransport; 1
 Sudhaus\MALT-1.bik; 1; Malztransport; 1
 Sudhaus\MALT-2.bik; 2; Malzsilos 1-8; 1
 Sudhaus\MST.bik; 4; Millstar 1; 1
end
begin;SUDHAUS
 Allgemein\overv_1.bik; 1; Sudhaus Übersicht;1
 Sudhaus\MTK.bik; 6; Maischepfanne 1; 1
 Sudhaus\LT.bik; 9; Läuterbottich; 1
 Sudhaus\PRT.bik; 11; Würzevorlauftank; 1
 Sudhaus\WK.bik; 12; Würzepfanne; 1
 Sudhaus\WHP.bik; 14; Whirlpool; 1
 Sudhaus\WATER.bik; 14; Schwachwürze; 1
 Sudhaus\CIP.bik; 16; CIP Sudhaus; 1
End
```

The structure of the configuration file considers the following aspects:

- The first line defines the overall and main picture
- By default the image files are stored in the folder tree "...<proj-path>\bilder". In addition, further subdirectories may follow which must be specified in front of the relevant picture name. A supplementary description maybe defined for the runtime system, which is displayed as a context menu (right mouse click on the navigation button) ("Drinktec 20xx").
- Related picture groups are defined after that.
- The area is opened with the word "begin" followed by a descriptive String for the area (‘MALT’ in our example). This name is indicated in the context menu.
- The respective area is closed with the Word "end". Further areas can follow.

- Pictures that appertain to an area are described between "begin" and "end". The setup provides the following syntax:
  - `folder\picture file name; Unit; Description for context menu; Area ID`
- The specification of a unit will not have an effect on the representation and you can replace it with a space. Both semicolons are necessary, but they may not be listed one right after the other.

IMPORTANT!!! The respective parameters are separated with a semicolon.

Tab application panel

The specific settings for the application panel are:

- Show application field
- Show Tooltips
  Selection box must also be clicked to display the configurable tooltip for the application buttons.
- Number of buttons
  If this value is less than the count of configured applications, forward/backward buttons are displayed automatically
- Button width:
  The button may be varied additionally for the display of longer text strings
- Control file:
  Several different files can be selected for each faceplate using the file selection dialog (default path and file name '...<proj-path>\texte.x\panel.ini').
  For editing, the standard editor of the operating system can be opened with the button on the right.

Example application bar:

Content panel.ini:

```
;

;

; This file is used by the following modules:

; - PANEL.OCX

;*******************************************************************

// [App1] // Section

// BtnName=Menu // Button Description

// App=Newmenu.exe // Application name

// BtnToolTip=Sistar Menu // Button tooltip description

// SistarApp=1 // Sistar or foreign application (1/0)

;*******************************************************************

[App1]
```

```
BtnName=Basic menu

App=Newmenu.exe

BtnToolTip=Sistar Menu

SistarApp=1
```

Here you can select up to 100 applications of your choice. Beginning with the section [AppX], this is followed by the description of the button and application, the tooltip and the means of distinguishing whether or not the application is a system application.

AREA overlapping picture navigation

- The panel OCX does not contain any AREA reference in the object properties but it reads the AREA ID per process image line from the configuration file ‘PANELTREE.INI’.
- With this, an AREA-overlapping picture navigation is possible, regardless of where (MC client or MC server) and/or in which AREA the runtime session (LZSYS) is started, where the process diagrams are loaded individually by the relevant server.
- The files "PANELTREE.INI" and "PANEL.INI" can be created individually for the relevant IOS and saved on the IOS server (see section General (Page 409)).
- The search order for the path is as follows:
  - Local search (e.g. for client-specific navigation sequence)
  - AREA Server of loaded picture, client-specific IOSxxx subdirectory
  - AREA Server of loaded picture
- The same search order is now also valid for the configuration file ‘PANEL.INI’ for the application buttons while no AREA reference is found here. The applications are started with the most recently valid AREA settings.
- If the same navigation hierarchies are required in the different AREAs, the ‘PANELTREE.INI’ files must be copied manually between the IOS stations after each change.
- For existing PANELTREE.INI files without AREA entries, the AREA of the loaded picture is always valid.

Tab time panel

For rounding up the shortcut tab you may fade in a time and/or date field. They can be configured together in the time panel tab.

- Activate/deactivate time display:
  In the corresponding edit field you can affect the display format.
  The following parameters are available:
  - %H: Hour indication
  - %M: Minute indication
  - %S: Second indication
- Activate/deactivate date display
  In the corresponding edit field you can affect the display format:
  The following parameters are available:
  - %Y: Year (4 digits)
  - %M: month (2 digits)
  - %D: day (2 digits)

  Order and separator characters must be assigned according the country specific standards.

  Note

  If the edit fields are left empty with the time/date display still activated, the system format is used at runtime.

Fonts tab

The font, style and font size for the date and time are set here.

6.1.3.12 Smart Unit Control

Basic principles

The control monitors a unit. Error messages, warnings, process messages and operation messages are displayed as icons and can be reset and acknowledged. The associated 'full' unit faceplate may be activated by an Info Button

General Tab

- Parameter:
  The individual variable may be selected with the following combo boxes:
  - Area → plant area for the variable to be connected
  - PCU → PCU of the selected AREA
  - Unit → object instance of selected PCU
- Image:
  A process diagram can be prescribed here, which is opened by clicking on the "Unit" screen change button. The process diagram, in which the main objects of this unit are located, is recommended here.

Extended tab

View:

- Small icons:
  The smaller version of the SmartUnit Control is shown.
- Large icons:
  The larger version of the SmartUnit Control is shown.
- Width for screen change button:
  Width of the unit button for the screen change, in pixels.
- Transparent:
  The faceplate icon is displayed transparently.
- Tooltip:
  Information related to the faceplate icon is displayed by touching the icon with the mouse.

Control dialog:

- Modal:
  The runtime is shown as a modal window, i.e. the user cannot work with the same application in other windows as long as the dialog window is displayed.
- Always On Top
- Dialog position X: ... Y:
  The position of the control dialog in the process diagram can be prescribed here.

Note

Highlighting the operator request button

A pending operator request can be displayed as flashing for all instances. This is done in the "...<proj-path>\sys\controls.ini" file with the following key:

```
[SmartUnit]
...
;Flash operator request button
FlashOpReqBtn=1
```

Fonts tab

The font, style and font size for the unit name are set here.

Note

This faceplate does not support symbolic characters such as "Ω".

6.1.3.13 Statusbyte Control

Basics

The control serves the symbolic representation of particular values of any desired variable. There are 2 methods for implementing the value on the icon bitmap to be displayed. The configuration of both methods is described below.

Address Tab

In the address dialog the requested variable is specified. The input of the variable can occur in two ways: via the object name or via direct address, The input type is selected in the source type area.

If ‘object name’ is selected, the following dialog appears:

- Area → Area of the variables to be interconnected
- PCU → PCU of the selected AREA
- Class → Data block class in the PCU (only classes with byte attributes can be selected!)
- Instance → Object instance in the selected PCU
- Attribute → requested attribute of the object

If the option 'direct address' is selected, the following dialog appears:

- Area → Area of the variables to be interconnected
- PCU → PCU of the selected AREA
- Data type: → Input of the data type (data block, flag byte, input byte, output byte).
- Block no.: → Input of the block number, if the data type "Data block" was selected.
- Datebyte no.:→ Input of the requested byte number

Tab View

The display type of the control in the runtime system is defined in the display dialog.

- Original size → The size of the faceplate cannot be changed.
- Transparent: → the faceplate is displayed transparently.

- Tooltip: → information related to the connected variable is displayed by touching the faceplate with the mouse in the runtime system.
- Symbol file:
  The following two methods are available to select and display a status symbol:

1. Method - Definition of a bitmap file with 8 subsymbols:
   The selected bitmap must consist of eight subsymbols of the same size. Each subsymbol represents a value of the status byte. Only the first three bits of the status byte are evaluated, which means the values n mod 8 + 1 correspond to the subsymbols 1...8.
   Flashing - here you can specify in which status the symbol is to flash in the runtime system.
2. Method - Definition of a text list with up to 256 symbol files.
   It is possible to display 256 states of the status byte. All eight bits of the status byte are evaluated. Every state of the status byte is displayed by a separate bitmap file. A text file is required containing the path information of the bitmaps. Here the values 0...255 of the status byte correspond to the bitmap files in the lines 1...256 of the text file.

- The corresponding display method is automatically specified by selecting a bitmap file or text file.
  The "Flashing" attribute is not available for this display method.

6.1.3.14 TankCtrl Control

Basics

Via the Control the following functions can be checked and controlled for a tank or container:

- Display of the tank fill level, pressure and temperature (up to 8 zones and average temperature)
- Display and control (manual operation) and monitoring of the runtimes for cleaning, sterility and quality status
- Display of the associated tank group and tank type
- Display and control (manual operation) of tank status and quality status
- Manual/Automatic mode change

The different modes and operating functions are specified during image configuration using the following properties dialogs:

General tab

- Source:
  The required object instance can be selected with the following combo boxes:
  - Area → Area of the tank instance
  - PCU → PCU of the selected AREA
  - IID → Object instance within the selected PCU
- Symbol file:
  An individual icon for the control in the runtime system can be set here.

Extended tab

Display:

- Original size:
  The size of the icon corresponds to the bitmap dimensions ("Icon" layout) or the number of lines and 32 character width cannot be changed.
- Transparent:
  The icon is displayed transparently.
- Tooltip:
  Information related to the interconnected variable is displayed by touching the icon with the mouse pointer.
- Grid lines:
  Show separating lines in the icon's value table
- In addition to this, the decimal places for the display boxes "Temperature", "Pressure" and "Volume" can be set here

Signaling:

- Global settings
  Switchover between global (`"Controls.ini"`) and instance-specific setting
- Manual mode
  Manual mode is indicated with a 'hand' icon next to the block icon.
- Manual mode material change
  Manual mode for material selection is signalized with a 'hand + M' icon next to the block icon.
- Note
  A note change to this block instance can be indicated next to the icon. The corresponding selection box has to be selected in the control dialog / tab 'About' / Note.

Color settings

- Switchover between global (`"Controls.ini"`) and instance-specific setting
- Setting of the color for remaining time in the control dialog
- Setting of border color and background color of the block icon.

Note

The border width `ColorBorderWith=` can only be set on an instance-specific basis, i.e. the change is always enabled.

Control dialog:

- Modal:
  The runtime is shown as a modal window, i.e. the user cannot work with the same application in other windows as long as the dialog window is displayed.
- Always on top
- Dialog position X: ... Y:
  The position of the control dialog in the process diagram can be prescribed here.

Layout tab

The display properties of the block icon are defined here:

- Shape:
  - Icon
  - Value view
  - Value view with icon
- Value table - selection, font/font size, alignment, line, column and text width for the following parameters:
  - Tank name
  - average temperature (average temperature of the interconnected/activated sensors)
  - Pressure
  - Quantity
  - Material
  - Tank status
  - Remaining time monitoring (depending on active counter: cleaning, sterility, quality)
  - Batch information in full or, optionally, batch number/order number separately
  - Recipe category
  - Recipe
  - Quality status
  - Temperature sensor 1 .. 8 (the sensors selected here are displayed, however only if they are interconnected/activated)
  - Batch number
  - Order number

Tips for designing the faceplate icon:

- The line, column and text width for the value view can be specified for each parameter.
- The text width only has an effect on the column width when the attribute "Original size" is disabled in the "Advanced" tab
  (otherwise, the column width is always limited to 32 characters with the selected font and font size)
- You must pay attention to the validity of line and column especially when using multiple columns because fields may be overwritten or not displayed at all.
- Gaps and/or lines or columns that are not used are not displayed.
- The result of all layout changes can be tracked simultaneously in the icon window.
- The icon border and thus the size during runtime must be adapted manually to the configured value table with "Original size=Off".

Trend Tab

In the 'Trend' tab, the scaling of the time- and y-axis is defined.

- Duration:
  Time range, with which the changes to the connected variable are recorded.
- Line distance/Y:
  Number of division marks of the Y-axis
- Line distance/time:
  Number of division marks of the time axis
- Trend curves - Selection and color selection
  - Average temperature
  - Pressure
  - Quantity
- Upper limit / lower limit for temperature, pressure and quantity
  - decides the axis scaling in the trend window for temperature, pressure and quantity

6.1.3.15 Unit Control

Unit Control

Basics

The Unit Control can be used to check and control the following functions for a unit:

- Recipe selection, start, stop, step selection, hold, restart, manual/automatic, auxiliary equipment On/Off, operator prompt.
- Separate windows for the Unit DFMs and the EOP DFMs can also be viewed and hidden.
- The different modes and operating functions are specified during image configuration using the following properties dialogs:

General Tab

- Area → Area of the variables to be interconnected
- PCU No. → PCU of the selected AREA
- Sequence: → Here the required sequence within the PCU is selected.
- Unit selection operable: → When selected, the selection list is activated for the unit, which means the operator can select the allocated unit dynamically.
- Step selection operable: → When selected, the button for the step selection is enabled, which means the operator can select a new recipe step.

- Assigned process cell view:
  - This corresponds to the number of the configurable plant in the "Plant overview" application (internal reference to "...<proj-path>/etc/kposxxx.ini"),
  - This is loaded when the plant overview is called up via the command button
    A "3" starts the plant overview, for example with "...<proj-path>/etc/kpos003.ini".
- Symbol file:
  the symbol for the 'Icon' mode can be selected here (see 'Extended' tab). With IOS clients if a network path is selected with a drive letter then this is automatically replaced by the UNC network path ('\\Server\Releasename\...')

Extended Tab

View:

- Shape:
  - Icons → Representation as symbol. The window representation opens when the icon is clicked.
  - Line with title → Representation in 2 lines (title line and info line)
  - Line → Display in 1 line (only info line)
  - Window → representation as operator control dialog
  - Line can be configured → The information and operating elements displayed can be configured
  - Window can be configured → The information and operating elements displayed can be configured
- Layout:
  For the "Line configurable" and "Window configurable" display types, a variable configuration of content, size, position and font of the included individual objects is supported. All settings are saved in the configuration file "unitctrl.ini" (standard path "...<proj-path>\bilder"). This configuration file is set up automatically by clicking the button next to the "Layout" field if it is not already available. Otherwise it is opened in the Notepad Editor and can be changed and/or supplemented by the user (description see section UnitCtrl.ini (Page 473)).

- Index:
  Sets the display variant. The index corresponds to the relevant section in the "unitctrl.ini" configuration file as follows:
  Index = 0 → section `[Dialog] / [LINE]`
  Index = 1 → section `[Dialog1] / [LINE1]`
  ... etc.
  With a new setup the configuration file contains the following sections and therefore display variants (see section Unit Control (Page 500)):
  – `[LINE] → line display with header bar`
  – `[LINE1] → line display without header bar`
  – `[LINE2] → line display with header bar and all fields`
  – `[Dialog] → 'Dialog window small' window display`
  – `[Dialog1] → 'Dialog window large' window display`
- Original size (only active for 'Icon' view):
  The size of the symbol is oriented to the bitmap and cannot be changed.
- Transparent:
  The symbol is displayed transparently against the background. With this, all the parts of the bitmap which are white (RGB value 255, 255, 255) become transparent.
- Tooltip
  Information related to the connected variable is displayed by hovering the mouse in the runtime system.

Control dialog:

- Modal:
  Display during runtime is as a modal window, i.e. the user cannot work with the same application in other windows as long as the dialog window is displayed.
- Always on top:
  The dialog is always displayed on top of the image. It remains visible when the application is changed.
- Dialog position:
  Through this the input fields are enabled for the screen position. If this option is not selected, then the control dialog is placed in such a way that the upper left corner of the icon and of the dialog are on top of each other.
- X / Y The screen position of the control dialog is entered in these input fields. With this the coordinates are calculated relative to the display window. The upper left corner corresponds to the coordinate 0/0.

Note

Highlighting the operator request button

A pending operator request can be displayed as flashing for all instances. This is done in the "...<proj-path>\sys\controls.ini" file with the following key:

```
[UNIT_CTRL]
...
;Flash operator request button
FlashOpReqBtn=1
```

Setpoint tab

Up to 3 setpoint value variables (DFMs) can be defined here. The display in the runtime view of the UNIT control faceplate must also be configured in the relevant mode (see "Layout Setting" in the previous section). The instance data is selected via the following fields:

- PCU: Selecting the PCU
- Class: Selecting the DFM group
- Instance: Selecting the DFM instance
- Attribute - selecting the object attribute
  - SOLL_DINT = The current DFM setpoint value is displayed
  - IST_DINT = The current DFM actual value is displayed
  - SOLL_DINT && IST_DINT = Both values are displayed
- active=Data request and online display are activated
- operable=The DFM setpoint can be operated.

Bit variable tab

Up to 3 bit variables can be defined here. The display in the runtime view of the UNIT control faceplate must also be configured in the relevant mode (see "Layout Setting" in the previous section). The instance data for both variables is selected via the following fields:

- Selecting the addressing type
  - Direct object addressing (PCU, class, instance, attribute)
  - Direct addressing (PCU, module/flag/I/O, index, data bit)
  - For further details see Specify data sources (Dialog) (Page 429) section **"PCU - direct address source information"**
  - With direct addressing an object instance name can be defined directly in the 'Name' field.
- Active = data request and online display are activated
- operable=The bit variable can be operated.

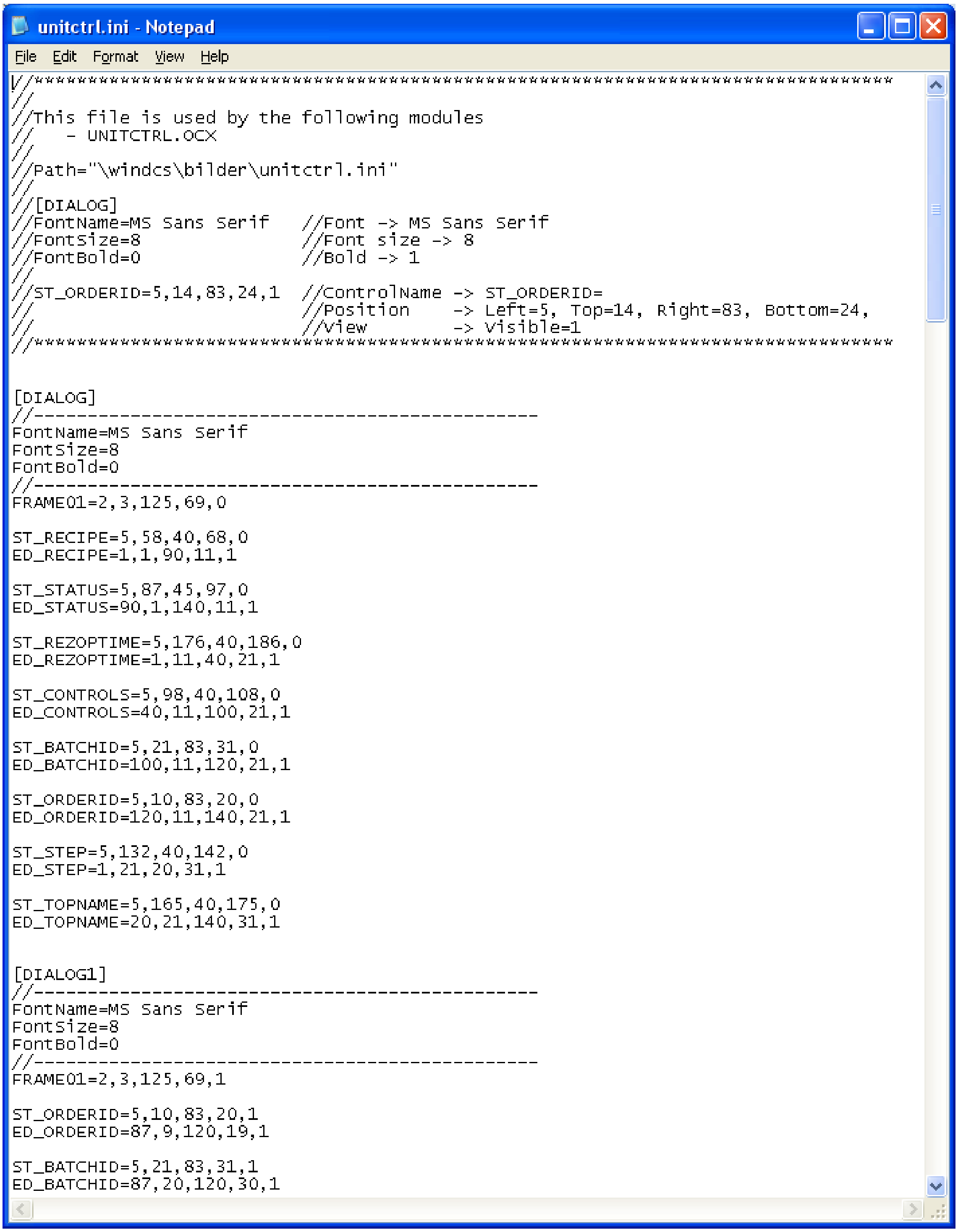
Additional attributes for the "Direct object address" address type":

- PCU: Selecting the PCU
- Class: Selecting the object class
- Instance: Selecting the object instance
- Attribute: Selecting the object attribute from the bit variable type
- File: here the symbol for the bit variable can be selected.

UnitCtrl.ini

Configuration file 'unitctrl.ini'

The figure below shows a newly created configuration file:




```
//*****************************************************************************************
//
//This file is used by the following modules
//    - UNITCTRL.OCX
//
//Path="\windcs\bilder\unitctrl.ini"
//
//[DIALOG]
//FontName=MS Sans Serif    //Font -> MS Sans Serif
//FontSize=8                //Font size -> 8
//FontBold=0                //Bold -> 1
//
//ST_ORDERID=5,14,83,24,1   //ControlName -> ST_ORDERID=
//                          //Position    -> Left=5, Top=14, Right=83, Bottom=24,
//                          //View        -> Visible=1
//*****************************************************************************************


[DIALOG]
//-------------------------------------------------
FontName=MS Sans Serif
FontSize=8
FontBold=0
//-------------------------------------------------
FRAME01=2,3,125,69,0

ST_RECIPE=5,58,40,68,0
ED_RECIPE=1,1,90,11,1

ST_STATUS=5,87,45,97,0
ED_STATUS=90,1,140,11,1

ST_REZOPTIME=5,176,40,186,0
ED_REZOPTIME=1,11,40,21,1

ST_CONTROLS=5,98,40,108,0
ED_CONTROLS=40,11,100,21,1

ST_BATCHID=5,21,83,31,0
ED_BATCHID=100,11,120,21,1

ST_ORDERID=5,10,83,20,0
ED_ORDERID=120,11,140,21,1

ST_STEP=5,132,40,142,0
ED_STEP=1,21,20,31,1

ST_TOPNAME=5,165,40,175,0
ED_TOPNAME=20,21,140,31,1


[DIALOG1]
//-------------------------------------------------
FontName=MS Sans Serif
FontSize=8
FontBold=0
//-------------------------------------------------
FRAME01=2,3,125,69,1

ST_ORDERID=5,10,83,20,1
ED_ORDERID=87,9,120,19,1

ST_BATCHID=5,21,83,31,1
ED_BATCHID=87,20,120,30,1
```

- The entries have the following meaning:
  - `[Dialog] / [LINE]`
    Start of layout variant section. Multiple variants of the relevant view can be defined via an additional index (1, 2, 3, ...). This index corresponds with the "Index" display attribute described subsequently
  - `Fontname, FontSize, FontBold =`
    The font style, font sizes and font variation for this view are specified here
  - `FRAME01, 02, 03=`
    Defines a frame as a collective criterion for the objects defined subsequently (until the next "`FRAMExx`" key.). An object grouping e.g. for recipe, sequence and step objects can be created in the faceplate using this.
  - Object definitions (e.g. `ST_ORDERID/ED_ORDERID, ST_SEQUENCE/ ED_SEQUENCE`, ...)
    The objects are defined in each case with information for ST_xxxx ("Static Text") and ED_xxxx ("Edit Object"). The 5 position values are described in the header section of the configuration file using the example "`ST_ORDERID= ...`". The first 4 numeric values define the upper left corner as well as the lower right corner of the object. The last value determines whether the object is set as visible/invisible.

Note

- The layout sections set up from the dialog should be retained and the user should create separate layouts from these (copy & paste) and provide them with their own index. As such, the standard layouts serve as a basis and can be used at any time for reference purposes.

List of available image objects

All of the objects supported are listed in the table below. They can be used as needed in the individual sections. After creating the new configuration file 'unitctrl.ini', the entire list of object definitions is available in the sections `[Dialog1]` and `[LINE2]`.

Table 6-4 Text and information objects

Object name Static text	Object name Content object	Meaning
ST_ORDERID	ED_ORDERID	Order number
ST_BATCHID	ED_BATCHID	Batch number
ST_BATCHNAME	ED_BATCHNAME	Batch
ST_RECTYP	ED_RECTYP	Recipe category
ST_RECIPE	ED_RECIPE	Recipe
ST_SEQNR	ED_SEQNR	No. sequence/unit
ST_SEQUENCE	ED_SEQUENCE	Sequence/unit
ST_STATUS	ED_STATUS	Status (unit)
ST_CONTROLS	ED_CONTROLS	Display
	ED_MESSAGE	Operator prompt
ST_STEP	ED_STEP	Step number
ST_TOPID	ED_TOPID	TOP number

Object name Static text	Object name Content object	Meaning
	ED_SEQNR	Sequence number
ST_TOPSTATUS	ED_TOPSTATUS	Status (Techn. Operation)
ST_TOPNAME	ED_TOPNAME	Name (Techn. Operation)
ST_ROPTIME	ED_ROPTIME	Runtime (recipe operation)
ST_DFM_NAME01	ED_DFM_NAME01	DFM 1 Name
ST_DFM_SW01	ED_DFM_SW01	DFM 1 setpoint
ST_DFM_IW01	ED_DFM_IW01	DFM 1 actual value
ST_DFM_NAME02	ED_DFM_NAME02	DFM 2 Name
ST_DFM_SW02	ED_DFM_SW02	DFM 2 setpoint
ST_DFM_IW02	ED_DFM_IW02	DFM 2 actual value
ST_DFM_NAME03	ED_DFM_NAME03	DFM 3 Name
ST_DFM_SW03	ED_DFM_SW03	DFM 3 setpoint
ST_DFM_IW03	ED_DFM_IW03	DFM 3 actual value
ST_BIT_NAME01	ED_BIT_NAME01	Bit variable 1 name
	IMG_BITMAP01	Bit variable 1 symbol bitmap
ST_BIT_NAME02	ED_BIT_NAME02	Bit variable 2 name
	IMG_BITMAP02	Bit variable 2 symbol bitmap
ST_BIT_NAME03	ED_BIT_NAME03	Bit variable 3 name
	IMG_BITMAP03	Bit variable 3 symbol bitmap

Table 6-5 Buttons

Object name	Call of ...
BTN_RECCONTROL	Control recipe application
BTN_PROCPARA	Job parameters dialog
BTN_PROCINPUT	List of input materials dialog
BTN_SEQCONTROL	Plant overview application
BTN_SEQPERMCOND	Permanent condition diagnostic sequence (S7Status or BLR runtime)
BTN_ADDUNIT	Toggle additional unit on/off
BTN_SEQAUTO	Sequence automatic/manual
BTN_SEQSTART	Start sequence
BTN_SEQABORT	Abort sequence
BTN_SEQRELEASE	Release sequence
BTN_SEQSTOP	Hold sequence
BTN_SETSTEP	Step selection dialog (if it was enabled in the "Properties" dialog)
BTN_LEAVESYNC	Skip synchronization sequence
BTN_EOPSTART	EOP S88 command "Start"
BTN_EOPPAUSE	EOP S88 command "Pause"
BTN_EOPHOLD	EOP S88 command "Hold"
BTN_EOPSTOP	EOP S88 command "Stop"

Object name	Call of ...
BTN_EOPABORT	EOP S88 command "Abort"
BTN_OPREQUEST	Operator request dialog
BTN_S7STATUS	"S7Status" application
BTN_SEQSETPOINT	Sequence setpoint window
BTN_EOPSETPOINT	EOP setpoint window

6.1.3.16 ‘Specify data sources’ (dialog)

Basics

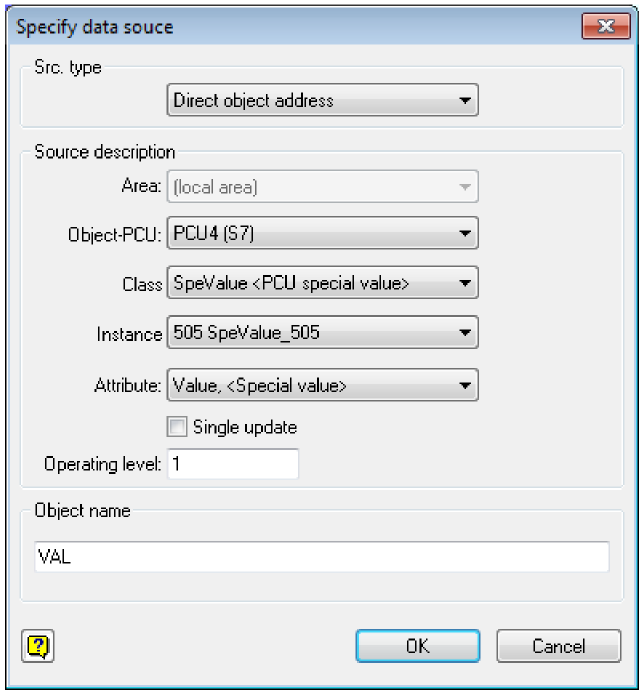
The picture objects are normally connected in the respective properties dialog in the General tab to the PCU data element. In the case of "Indirect addressing", however, the dialog "Specify data source" has to be opened in the control icon's context menu. This calculates all data required by the system in order to save a unique allocation of process diagram objects to the PCU data elements or object classes.

Indirect addressing

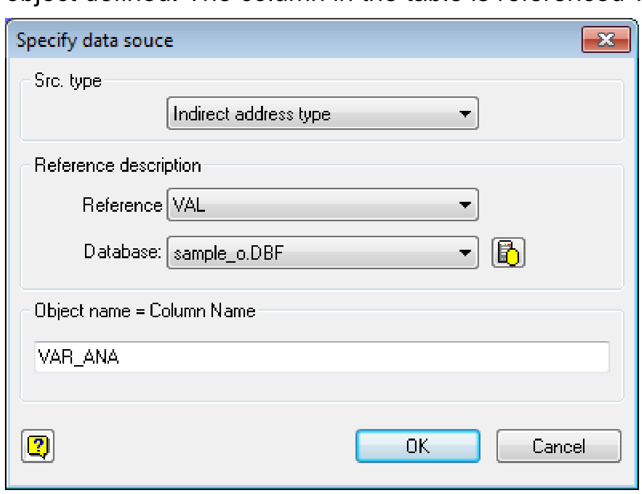
In order to be able to observe pumping processes in a process diagram, particular picture objects should be dynamically connected, depending on the tank or product path currently being used, to the physical PCU data elements. The representation of the control is fixed here, the data sources have to be connected dynamically, however, depending on the reference variables.

To configure the indirect addressing of a group of picture objects, the following steps are necessary, whereby step 1 (reference variable) and step 3 (address table) apply together to the entire group:

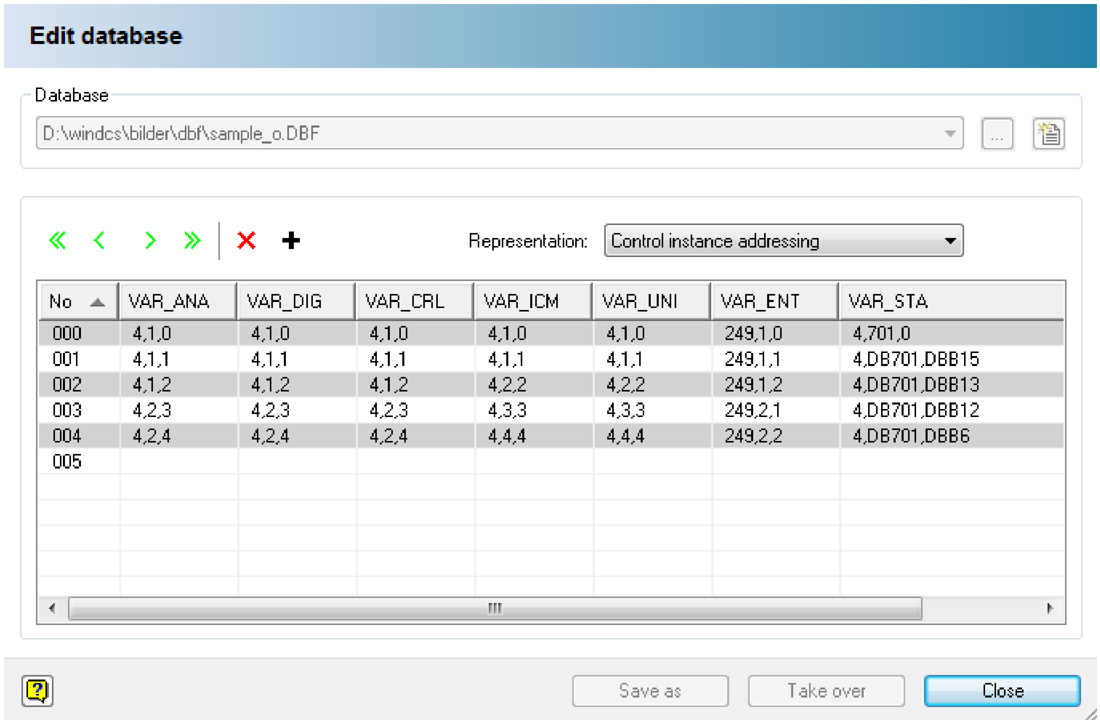
1. The reference variable is defined with the connection type 'Direct object addressing' and the attribute 'one-off updating'. A unique object name has to be issued for this.



2. The connection type "indirect addressing" has to be selected in the "Specify data source" dialog for the referencing controls. In the field 'Reference variable', the object name of the previously defined reference variables has to be selected. In addition, the database file (*.DBF) with the address information has to be selected, and the object name of the picture object defined. The column in the table is referenced via this object name.



3. The button next to the selected database table takes you to the editor widow for the address information of the picture objects.



Structure of the table and editor functions

- The dbf-file to be edited is selected in the "Database" field. The list box contains the dbf files already disclosed. An existing dbf file not included in the list box can be opened via the '...' button.
- A new dbf file can be set up and added to the list.
- Columns or lines can be added or deleted in the table window. To do this, select the column/ line and then click '+' or 'x'.
- Lines can also be added or deleted.

Column "No." - cannot be edited, contains the line index. The corresponding lines are selected for the duration via the reference variables. The address information for the picture objects referenced in the columns is calculated from this.

The following applies to all other columns:

- The column name must correspond to the object name of the referenced picture objects (in the example "VAR_ANA", "VAR_DIG", ...). The column name can be edited via the context menu "Rename column".
- The field content per line contains the address information for the picture objects referenced. The field content can be edited via the button "..." in the selected field.

Note

The structure of the field content varies depending on the type of picture object referenced, i.e. the field content is interested by the picture object itself and is not known in the editor dialog. Consequently, the project manager has to select one of the following settings via the "Display" selection box before editing field content:

- For standard variables (VAL, HEX, Bit, ICM) → "Direct object addressing" or "Direct addressing"
- For controls (analog, digital, ICM, UNIT, ... → "control instance addressing" (see below)

Address information for picture objects of type 'Controls'

The address information in the table columns, to which picture objects of the type Controls refer, have a different syntax from the standard variables, as the following table shows:

Table 6-6 VARx = <Value 1>,<Value 2>,<Value 3>

Assigned Control	Value1	Value2	Value3
Analog.ocx	<PCU-No>	<Group> 1 = ANAU 2 = MESS	<Instance>
Ctrl3.ocx	<PCU-No>	<Group> 1 = Dreip 2 = PID	<Instance>
Digital.ocx	<PCU-No>	<Group> 1 = TIMER01 2 = TIMER02	<Instance>
ICM3.ocx	<PCU-No>	<ICM group> 1, 2, 3, 4	<Instance>
StatByte.ocx	<PCU-No>	DBxxx	DBBxxx
Unit_Ctrl.ocx	<PCU-No>	1 = Sequences/Unit	<Instance>
TankCtrl.ocx	<PCU-No>	1 = TANK (PCU V7 or higher)	<Instance
LineCtrl.ocx	<PCU-No>	1 = LINE (PCU V7 or higher)	<Instance
Entity.ocx	<PCU-No> = 249	<Entity-Type>	<Instance>

NOTICE
Correspondence between control type and referenced object class • An object class belonging to the respective control type always has to be selected for the attribute "Class" according to column "Value 2" in the dialog "Data source". If this is not the case, the control cannot currently interpret the value set for the duration, under some circumstances, and there is no error message and no data display. • If the group = "0" is set (e.g. for manual input), there is no error message and no data display. • Indirect addressing is not supported for the other controls not listed here and therefore a context menu dialog "Data source" is not offered.

See also

Tools - Menu (Page 413)

General (Page 154)

6.2 Runtime system

6.2.1 General

Functionality

The application process diagrams offer a plant overview based on the configured full graphic images by viewing process data from the PCU. The display of the transmitted process data occurs on the objects parameterized in the image construction. Via control elements, the user can manipulate the running process and access the different sequences.

Client Area

The Client Area includes a process diagram. The size of the process diagram is aligned in general with the entire size of the Client Area. The size of the process diagram is determined from the scales of the static image made in the image construction. The connected variables are displayed by the symbols selected by the user during the image construction, and "lie" above the process diagram.

Working with the application

Working with the application involves the following activities:

- Print process diagrams
  By clicking the printer symbol on the toolbar or the **Print** menu item in the **File** menu, the current loaded process diagram can be printed with the values of variables.
- Printer setup
  With the item **Printer setup** in the **File** menu, the requested default printer can be selected. Fine adjustment of the respective printer model is possible according to the installed printer driver.
- Viewing a process diagram
  When starting the application the root screen (default = BILD000.BIK) is loaded. In addition, the root screen can also be selected at any time via the menu item **General overview** in the **Process diagrams** menu.
  The process diagram to be displayed when the application starts can be defined by the following entry in the "BM_AppCenter.ini" file.
  Example:
  `App1=2;Prozessbilder;lzsys.exe _Overview_Systest.bik`
  Usage is described in Chapter Register of the Application (Page 151).
- Change to full-screen mode
  With the <Ctrl + Backspace> keyboard entry the view can be changed between "Without menu" and "With menu".

- With the buttons inserted by the user during the image construction, and/or via objects, menus etc. a branching out to any further process diagrams of the desired unit is possible. In addition, process diagrams can be called via the menu command **Open process diagram** in the menu **Process diagrams.**
  Within a specific dialog window, the desired area or 'local' is to be selected first. The process diagrams are listed below. In the case of a selected area, this list is built up from the client-specific picture subfolder of the area server (if present/ 1st priority) or server picture folder (2nd priority).
  The optionally selectable preview picture facilitates the image selection.

  Note

  With this dialog, the call of executable files is prevented which is relevant to security.

- Viewing a memo
  For each image a memo can be stored. Clicking on the pen icon on the toolbar or selecting the Process diagrams\memo menu item brings up a memo window. In this window the memo which was stored last is displayed. For editing the memo the window must be switched into editing mode by clicking on the 'edit' button.
- Variable information and variable search
  Interconnection information can be displayed for images. In addition, it is possible to search for variables in the current picture or across all pictures. For additional information on this topic, please refer to chapter Variables information (Page 509).
- **Replay mode**
  If the "Replay mode" option has been selected for the current IOS station in Administration / Configuration, the menu command "Options / Replay mode" is enabled. In addition, a projector symbol is displayed in the toolbar, which can be used to enable/disable replay mode (see section Replay mode operator panel (Page 523)).

Context-sensitive opening of the changes and message archives

For fast diagnostics of visualization and operator control objects in process diagrams, it is possible to branch to the changes and message archives with a mouse click. The object is transferred as a filter string in the process.

Procedure:

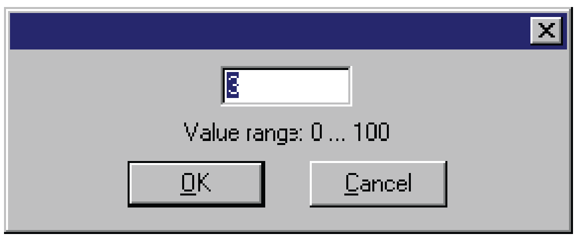
- Move cursor onto the desired object
- Open shortcut menu (right-click) and select the "Operator inputs and messages" command (left-click). The Changes and Messages application opens.
- Only messages/operator inputs for this object and the current tag are then displayed in the archive viewer.

6.2.2 Standard variables

Operation of variables

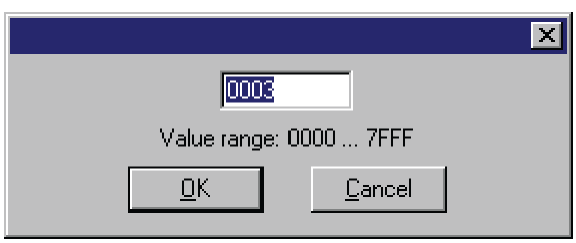
A hand appears on operable variables when they are touched by the mouse. Clicking on the operable variable will open a dialog box, which is specific to each type of variable.

6.2.2.1 Operating variable type ‘Value’



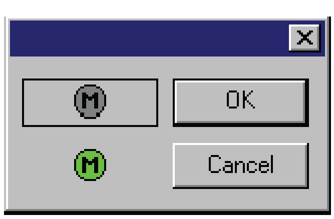
Double-clicking on an operable integer number or real number will make a dialog box for the input of the new value of variables appear. The permissible value range in accordance with configuration is displayed.

6.2.2.2 Operating ‘Hexadecimal’ variable type



With a double-click on an operable hexadecimal number a dialog box appears for the input of the new value of variables. The permissible value range in accordance with configuration is displayed.

6.2.2.3 Operating Variable type ‘Binary state' (Bit)

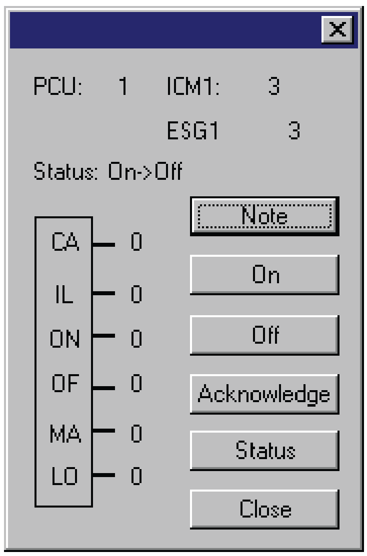


With a double-click on the operable binary variables a dialog box appears. The requested switch state clicked on by the mouse receives the framed box. With the **OK** button the switch is tuned correspondingly.

Example:

- above M gray => Motor off (Bit=0)
- below M blue => Motor on (Bit=1)

6.2.2.4 Operating ‘individual control module’ variable type (ICM)



By double-clicking on an operable ICM a dialog box with the following command buttons appears:

• Memo	Call the memo editor
• On	Switch on the ICMs in manual operation
• Off	Switch off the ICMs in manual operation
• Acknowledge	Acknowledgement of the ICM fault
• Status	Jump to the application 'status' with the relevant operating lock
• Close	Close the **ICM** dialog box

The dialog box contains the following information:

- **PCU number**
- **ICM group** (for example ICM1 corresponds to group1)
- **ICM number** and symbolic name of the current ICM

Table 6-7 Representation of the interface signals:

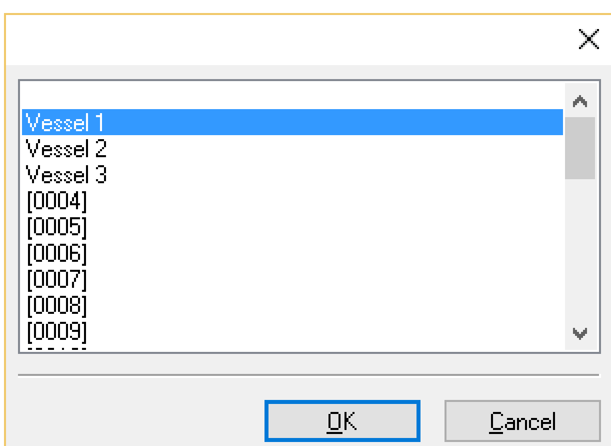
1	=	logical "1" is signaled to the element
0	=	logical "0" is signaled to the element
BA	=	Command Automatic
		Control from procedure or user program
		0 = automatic command OFF
		1 = automatic command ON

BV	=	Operating lock	
		1 = enabling:	ICM can be enabled
		0 = block:	ICM is blocked automatically
		re-enabling is not possible	
RE	=	Acknowledgment ON	
		1 = response from ICM "ON"	
RA	=	Response OFF	
		1 = response from ICM "OFF"	
HD	=	Manual operation	
		1 = manual operation is active	
		0 = automatic operation is active	
QL	=	Load output	
		1 = Load output is signaled	
Status	=	display of the current operating state of the ICM	
		• On	ICM is switched ON
		• Off	ICM is switched OFF
		• On->Off	ICM is switched off at the moment
		• Off->On	ICM is switched on at the moment

6.2.2.5 Operating Variable type 'text'

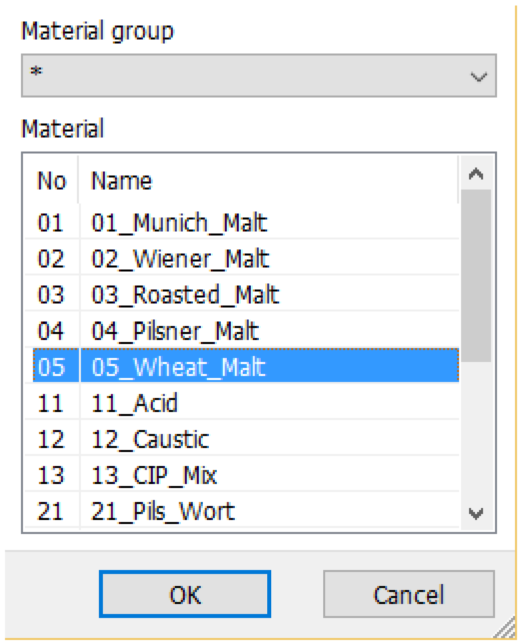
'Text' object type

With a mouse click on an operable text, a dialog box for selecting a text line appears.



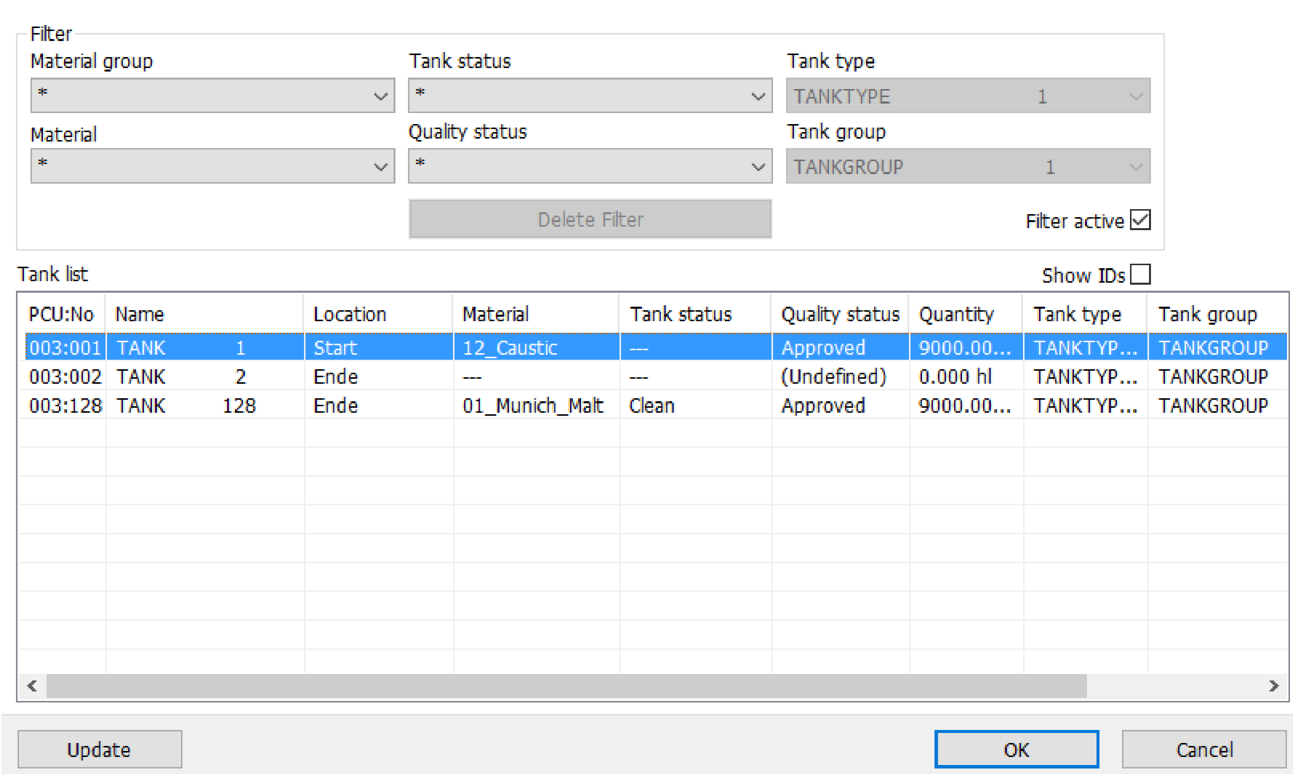
'Material' object type

With a mouse click on an operable text, a dialog box for selecting a material appears.



'Tank' or 'Tank location' object type

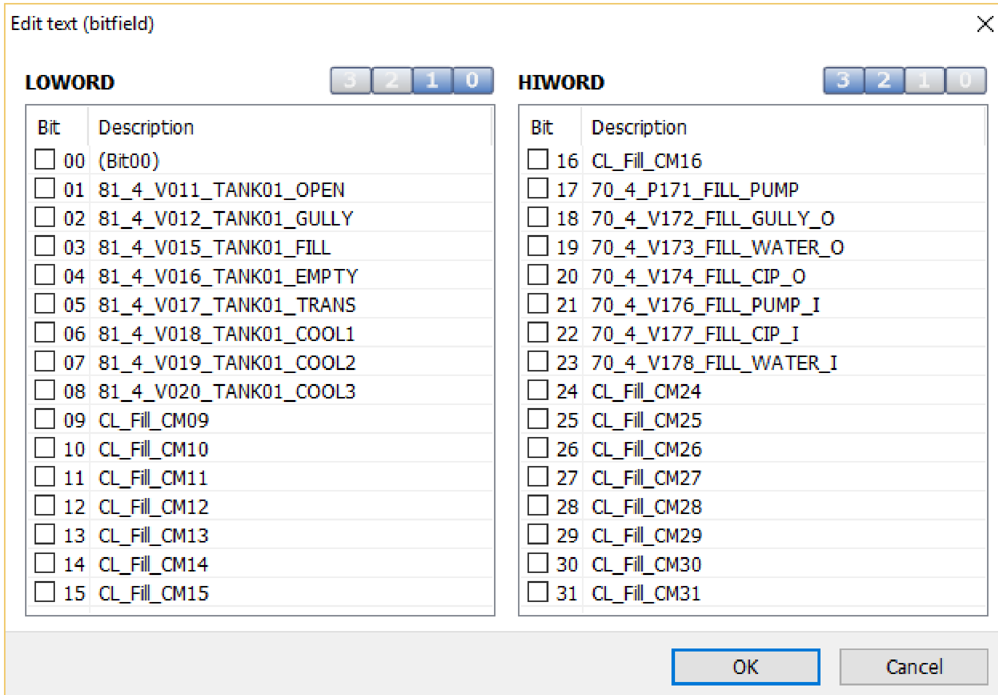
With a mouse click on an operable text, the tank selection dialog appears. For more details on selection and filter options, see section Dialog 'order parameter' (Page 741).



Text(Bitfield) object type

With a mouse click on this object type, the bit field dialog opens:

- The setpoint is defined by a bit pattern (32 bits).
- Check mark is set → setpoint bit = true
- Check mark is not set → setpoint bit = false



6.2.2.6 Operation Sound

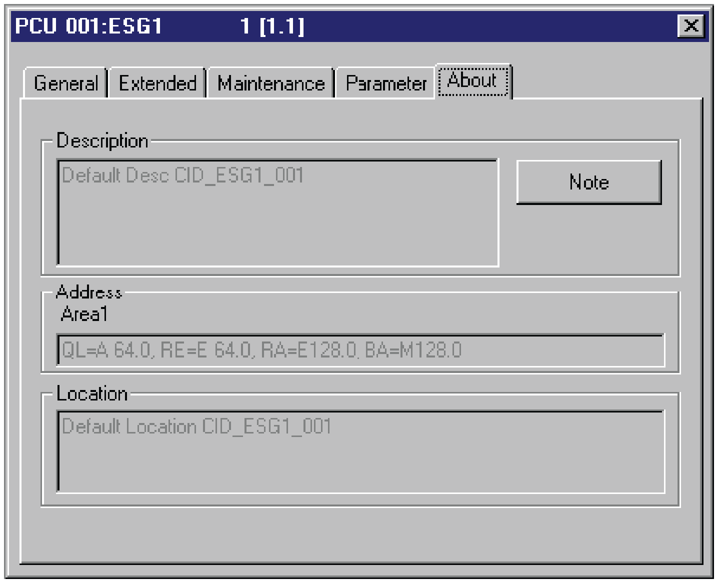
With a double-click on the operable variable 'Sound' a dialog box appears. The requested switch position clicked on with the mouse can be adjusted accordingly with the OK button.

6.2.3 Controls

Clicking on an operable control faceplate will open a dialog, in which the configured data can be viewed, changed and simulated. For all operable control faceplates there is the 'About' tab.

Tab About:

In the tab 'About' configured information relating to the connected variable is displayed (Description, Address, Location).



Memo	Call the memo editor

6.2.3.1 Analog control

Basics

The control is used for viewing and simulating analog input and output values.

Note

Information on the data source

Because different data sources can be visualized with these controls, the currently interconnected block class or the absolute address is displayed in addition to the instance in the caption bar.

Examples:

- Analog1: AIN, AOUT
- Analog2: all classes that can be interconnected or the absolute address

---

General Tab

- The current value is displayed in the form of a bar.
- With the control box 'Simulation' the analog values can be simulated.
- When the "Change" control box is activated, the block parameters (lower limit, upper limit, ramp gradient, output inhibit and limit) can be changed.

Trend Tab

The trend curve of the analog value is scaled graphically in a coordinate system

6.2.3.2 Controller Control

Basic principles

The control is used for viewing and simulating PID or Three-point controllers.

General Tab

- In the values area, the setpoint "W", the actual value "X" and the controller difference "XD" are displayed in bar form.
- In the control area the user can transfer between automatic and manual operation via the corresponding buttons.
- With the buttons 'External' and 'Internal' the source of the setpoint can be transferred.
- In the position 'External' the configured setpoint is used for controlling.
- In the position 'Internal' the setpoint is entered manually. For this the W field becomes active.
- In the parameter area the parameters of the controller (KP, TV, TN) can be changed.
- In the lower area the set value, the actual value and the output value are displayed in numerical format.

Parameter Tabs

- The parameters of the controller, such as lower limit, upper limit, sampling time, dead band and unit assignment can be changed by activating the "Change" control box.
- Reversing operation as well as X-Tracking can be adjusted.
- The YN flag can be simulated here.

Trend Tab

The following values of the controller can be observed graphically in a configured time interval:

- Output value Y
- Setpoint W
- Actual value X

- Fault signal Z
- Control difference XD

The parameters KP, TV, TN and the sampling time can be changed here.

6.2.3.3 Digital Control

Basic principles

The control is used for watching and simulating the digital inputs and outputs of the timer.

General Tab

- The states of the digital outputs and the input of the connected timer are viewed graphically in a coordinate system (with scaling).
- By activating the control box **Simulation**, the input of the timer can be simulated.

Parameter Tabs

- By activating the control box "Change" , the times of the positive and negative outputs of the timer can be changed.
- The outputs can also be inverted.
- By activating the control box Simulation , the input of the timer can be simulated.
- The timer type is shown in the area "Type".

6.2.3.4 Entity Control

Basic principles

The control is used to display groups of connected data in a defined format. The data grouping is established via the assigned template.

Properties Tab

- In the Properties tab, the configured data records and their values and units are displayed in tables.
- If one data record has writing access, the limits of the indicated value are displayed in the part below.

Status Tab

The status of a configured data record is displayed.

- One value can be defined as a status word from the list of parameters.
- An info text and the binary state is displayed per status bit (0 … 31).

See also

Entity Control (Page 447)

6.2.3.5 LineCtrl Control

Basics

The following functions can be checked and controlled for a pipeline using the line status control:

- Display and operation (manual mode) of the durations for any desired process state
- Display of the associated control group and control type
- Manual/Automatic mode change
- Display and operation (manual mode) of process state and material movements

'General' tab

Grouping of elements	
Displaying and switching the operating mode	
The block's operating modes can be shown and executed here: • Automatic • Manual	
Displaying and switching the line status parameters	
• Name	Name of the block instance
• Group	Control group from block parameter `LineGroup`
• Type	Control type from block parameter `LineType`
The following control state can be displayed here:	
• Process status	Current process status from block parameter `LineStatusAct`
In Manual operating mode the new value can be selected from a list through the additional dialog	
• Process status	List from enumeration `LineStatusOp`
Displaying and switching the material status parameters	
• Operating mode	Display or specification of the material tracking operating mode
• Material	Display or specification (Manual mode) of the last material transfer in the line. The list of the available materials is determined from the material configuration dialog.
Note: The material is selected via a list box in the faceplate. After confirmation with the Save button, the material is imported.	

Grouping of elements
Monitoring time
During activated monitoring this field shown the remaining time for a predefined process status. The field below it contains the start time of this process status. After the monitoring time has expired, the field for the remaining runtime is colored red.

Times tab

The monitoring times for the process status defined with the `StartLineStatus` attribute (line status for start of monitoring time) are displayed and activated in this view. The display fields are interlinked with the corresponding block parameters:

Status monitoring time

- Remaining time → `LineRmTimeAct`
- Duration → `LineTimeSp`
- Monitoring active → `LineTimeMon`

'Batch' tab

The display contains the currently applicable batch and recipe allocations for the controls.

- Order number
- Batch number
- Recipe category
- Recipe
- Sequence
- Status

6.2.3.6 ICM Control

Basics

The control is to watch and control the individual control elements and to simulate their states.

General tab

- In the general tab the state of the connected ICM is represented and the control of the ICM is executed.
- In the control area the user can transfer between automatic and manual operation via the corresponding buttons.
- With the buttons ON and OFF the ICM is forced to switch on or off.

- In the program area the configured unit is displayed. The unit which has requested the route is displayed in the second field when the ICM is occupied by RCS. When no unit assignment has been configured, the RCS unit is displayed in the left field.
- The number of monitoring time violations ("ErrorCounter" attribute) is displayed in the error counter. The counter can be reset with the "Reset" button.

Seat Lifting tab

This tab is displayed when the ICM control is interconnected to an instance of the class "SLB" (seat lifting). The SLB instance must be connected to an ICM instance by parameter assignment in this case. If this is not the case, a grayed out symbol or no symbol is displayed in runtime.

Here the additional activations of the respective seat lifting top and bottom for double seat valves can be visualized and operated:

- Valve symbol with signaling of the seat liftings and display of the operating mode (manual, simulation, error)
- Display field seat lifting on, off, locked
- Command buttons On, Off, Q (acknowledge error)
- Monitoring time in seconds
- Pulse time current and setpoint in seconds
- Pause time current and setpoint in seconds
- "External" selection box for pulse/pause mode

Extended tab

- In the "Maintenance groups" area, the interconnected ICM and all ICMs belonging to the same maintenance group can be set to maintenance.
- The maintenance group configuration is described in section Maintenance groups (Page 1007).
- The mode is set with the **Activate** and **Deactivate** buttons. The successful changeover is reported with a box ‘Miniport entity’.
- In the "Single maintenance" area, this ICM instance can be set to maintenance in manual mode. It depends on whether the ICM is a maintenance group or not.
- In the "Emergency operation" area, the ICM feedback simulation or the forced operation can be activated independent of each other.

Sequence of function Active.Maint – switch ICM to maintenance mode

- The faceplate sets Bit SetM in the ICM data record.
- The ICM block transmits a message if the Bit fits (SetM message)
- This message will be received by the Entity PCU Server
- Analysis of which entities are concerned.

- Settings of all ReqABM Bits for all concerned ICMs
- ICMs send messages based on the ReqABM Bits AckABM message (yes or no). If possible, all ICMs occupy the state ABM (AffectedByMaintenance).
- Waiting for the acknowledgments of the ICMs. If all ICMs entered the state ABM, set the Bit ReqM on the original ICM.
  If not all ABM transfers were accepted, the function would be aborted, the states would be updated and an error message would be displayed.
- The ICM responds with the AckM message and occupies the state M (Maintenance)

Sequence of function Deactiv.Maint. – switch back ICM from maintenance mode

- The faceplate sets the Bit ResetM in the ICM data record.
- ICM sends as a result a ResetM message.
- The message will be received by the Entity PCU server.
- Analysis of which entities are concerned.
- Determination of whether ICM goes back to Hand/Automatic or only to ABM
- IOS resets M or ABM Bits respectively.
- Determination of whether one or more entities are reset completely as a result.

In the **Affected entities** area the states of 2 entities are displayed (ICM can be a member of a maximum of 2 entities):

- ABM entity in the ABM status
- IDLE - ABM deactivated
- SWO-ABM, SWO-IDLE - transition status

In the '**Error**' area the cause of the error is displayed in the case of activation failure.

With the command buttons **Details** and **Close** you are activating/deactivating 2 list boxes for diagnostic purposes with the information relating to the entity and display of all ICMs assigned to this entity.

The display is a snapshot which can be refreshed by clicking on the command button ‘Details’.

The displayed ICMs can have the following states:

- Au/Hd → Maint deactivated, ICM in the auto/manual mode
- M → For this ICM maintenance was activated. If the ICM is a member of an entity, this entity goes into ABM state.
- ABM → These ICMs are members of an entity in the ABM state.

Push Function for ICM and entity:

By activating the **‘!’** key the entity miniport is initiated for:

- Deleting dynamic errors in queue.
- Checking the state of the element and adapting it to the states of the concerned objects (e.g ICM state from ABM to IDLE, if no ICM in M).

In the area ‘Emergency operation’, the acknowledgments can be simulated and the release of the ICMs can be forced.

Maintenance Tab

- In the section type, the user can switch between power-on time counter and switch sequence counter.
- Up to 5 trigger values are displayed in the Messages area. A note window can be opened using the buttons to the right, and acknowledgment can be issued.
- The current meter state is shown in the 'Actual value' area.
- The configuration for this, and the central display application, are described in chapter Maintenance data (Page 998).

Parameter Tabs

- The parameters of the connected ICMs can be changed here, such as ‘Switch on delay’, ‘Switch off delay’ and ‘Monitoring time’.
- In the Expanded area, the unit assignment, manual group assignment and ICM type can be changed with the Change control box.

Detailed status display on bitmap

The following additional icons are displayed to the right of the bitmap for a detailed representation of the current status:

Priority	Icon	Meaning
1		Affected by maintenance (ABM)
2		In maintenance
3		Manual operation is active
4		'Force release’ active
5		'Simulation feedback’ active
6		Missing 'operation interlock’

Up to two icons can be displayed simultaneously.

6.2.3.7 Message View Control

The control shows the PCU messages selected corresponding to the filter settings.

Note

'Loop in Alarm' function

- If the message has the following attributes and there are picture information files for this block instance under '<proj-path>\PCU.xxx\IMAGE\<class>.txt', a process image with this distance can be opened directly from the message.
  - PCU number
  - Class (block type)
  - Instance number
- The corresponding picture is opened by double-clicking on the message or with a right-click on the message and subsequent left-click on the information box.
- The picture information files are used together with the "Variable search".
- The generation of these text files is described in chapter Global variable search (Page 511).

6.2.3.8 Statusbyte Control

The control serves to observe the Statusbyte of the configured variables.

- The control cannot be manipulated.
- The control serves the symbolic representation of particular values of any desired variable. There are 2 methods for implementing the value on the symbol bitmap to be displayed.
  - Method 1: Representation of a symbol from a bitmap file with 8 subsymbols corresponding to the bits 0, 1, 2 of the interconnected variables. The values n mod 8 + 1 correspond to the subsymbols 1...8.
    Specific subsymbols can also be displayed as flashing.
  - Method 2: Representation of a bitmap from a text list with a maximum of 256 bitmap files corresponding to the values 0...255 of the interconnected variables.

6.2.3.9 TankCtrl Control

Basics

Via the Tank states Control the following functions can be checked and controlled for a tank or container:

- Display of the tank fill level, pressure and temperature (up to 8 zones and average temperature)
- Display and control (manual operation) and monitoring of the runtimes for cleaning, sterility and quality status
- Display of the associated tank group and tank type
- Display and control (manual operation) of tank status and quality status
- Manual/Automatic mode change

'General' tab

<table>
<tr><th colspan="2">Grouping of elements</th></tr>
<tr><td colspan="2">Operating mode</td></tr>
<tr><td colspan="2">The modes of the block can be shown and changed here:
• Automatic
• Manual</td></tr>
<tr><td colspan="2">Tank data</td></tr>
<tr><td colspan="2">The following tank parameters can be shown here:</td></tr>
<tr><td>• Name</td><td>Name of the block instance</td></tr>
<tr><td>• Group</td><td>Tank group from block parameter</td></tr>
<tr><td>• Type</td><td>Tank type from block parameter TankType</td></tr>
<tr><td>• Tank status</td><td>Current tank process status from block parameter</td></tr>
<tr><td>• Quality status</td><td>Current tank quality status from block parameter</td></tr>
<tr><td colspan="2">In Manual operating mode the new value can be selected from a list through the additional dialog</td></tr>
<tr><td>• Tank status</td><td>List from enumeration</td></tr>
<tr><td>• Quality status</td><td>List from enumeration</td></tr>
<tr><td colspan="2">Material movement</td></tr>
<tr><td>• Operating mode</td><td>Automatic mode: Material control is handled by the user program
Manual: Material control is handled by the operator</td></tr>
<tr><td>• Material</td><td>Display or specification (Manual mode) of the last material transfer in the tank. The list of the available materials is derived from the material configuration dialog of the recipe editor.</td></tr>
<tr><td>• Correction quantity</td><td>Correction value that is added to the "QuantityTotal"</td></tr>
<tr><td>• "Save" button</td><td>The manual entry is applied.</td></tr>
<tr><td>• "Back" button</td><td>As long as the data has not yet been saved for manual entry, the last material input value can be restored again with the "Back" button.</td></tr>
<tr><td>• Total quantity</td><td>Contains the "QuantityTotal" of the material movements/correction entries since the last reset.</td></tr>
<tr><td>• Number</td><td>Contains the "QuantityMovements" of the material movements/correction entries since the last reset.</td></tr>
<tr><td colspan="2">Display of other parameters beside or under the tank icon</td></tr>
<tr><td colspan="2">• S1...S8
max. 8 temperatures of the interconnected ((Tx_ValueRef)) or patched ((Tx_ValueAct) temperature sensors
• Symbol for outlet valve
if the block parameter TankValueRef is interconnected, the corresponding IMC icon is shown here
• Icons for full and empty signaling probes - display of the interconnected block parameters
– FullDetector<Ref/Signal> = Tank HighLevel probe
– EmptyDetector<Ref/Signal> = Tank LowLevel probe
• Average temperature TemperatureAvg and unit TemperatureUom
Display of the average temperature of the valid/wet temperature sensors (calculated by the block)
• Pressure Pressure<Ref/Act> and unit PessureUom
• Quantity Quantity<Ref/Act> and unit QuantityUom</td></tr>
</table>

Grouping of elements
Cleaning time - only with tank status = "Clean"
If the monitoring is activated, the two boxes below show the runtime remaining and the start time for the "Clean" tank status. After the monitoring time has expired, the field for the remaining runtime is colored red.
Sterile time - only with tank status = "Sterile"
If the monitoring is activated, the two boxes below show the runtime remaining and the start time for the "Sterile" status. After the monitoring time has expired, the field for the remaining runtime is colored red.
Quality time - only with quality status="Allowed" and tank status= "Filled"
If the monitoring is activated, the two boxes below show the runtime remaining and the start time for the "Allowed" quality status. After the monitoring time has expired, the field for the remaining runtime is colored red.

Times tab

The monitoring times for the "Quality status" and for the tank status "Clean" and "Sterile" are displayed and activated in this view. The display fields are interlinked with the corresponding block parameters:

Explanation of the screen shot	
	Quality time monitoring
	• Remaining time→ • Duration→ • Monitoring active → `QualityTimeMon`
	Cleaning monitoring time
	• Remaining time→ • Duration→ • Monitoring active → `CleaningTimeMon`
	Sterile monitoring time
	• Remaining time→ • Duration → • Monitoring active → `SterileTimeMon`

'Batch' tab

The display contains the currently applicable batch and recipe allocations for the tank.

- Order number
- Batch number
- Recipe category
- Recipe
- Number of batches
  from the ""`BatchCounter`"" block parameter; the parameter is set by the user program

- Sequence - assigned unit
- Status - unit status

Trend Tab

Up to 3 curve properties from the following tank parameters are displayed here:

- Average temperature `TemperatureAvg` and unit `TemperatureUom`
- Pressure `Pressure<Ref/Act>` and unit `PessureUom`
- Quantity `Quantity<Ref/Act>` and unit `QuantityUom`

6.2.3.10 Unit Control

Views

Basics

The Unit Control can be used to check and control the following functions for a unit:

- Recipe selection, start, stop, step selection, hold, restart, manual/automatic, auxiliary equipment On/Off, operation prompt.

> **Note**
>
> **Temporary change to the allocated unit**
>
> The unit allocated via the configuration dialog via connection can be temporarily overwritten
>
> - Requirement: The 'Operable' attribute has been activated in the configuration dialog
> - The new unit is selected in the 'Sequence' listbox

> **Note**
>
> - The step selection dialog reads the recipe structure from the offline data management. If a recipe procedure is enabled for editing and changed while batches based on it are still running, if the step selection dialog is opened later, there may be a deviation compared with the currently running batch.
> - **Changes to recipes should therefore only be made when no more batches are running with this recipe procedure.**

- Separate windows for the Unit DFMs and the EOP DFMs can also be viewed and hidden.

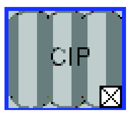
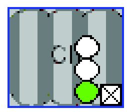
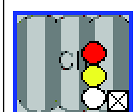
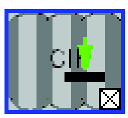
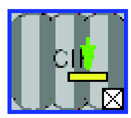
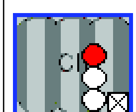
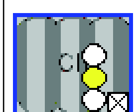
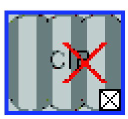
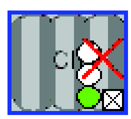
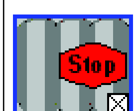
The 'Unit Control' mode can be specified in the 'Properties' tab upon configuration (see Section Unit Control (Page 468)) as follows:

1. Icon
2. Row with title
3. Row
4. Window

5. Row configurable
6. Window configurable

"Icon " representation

In the process diagram, the selected bitmap is represented. Using the bitmap, the status of the sequence is displayed in the right lower corner.

Idle 	Running 	Starting is not shown	Restarting 
Paused 	Pausing 	Held 	Holding 
Canceled 	Canceling 	Stopped 	Stopping 
Finished 			

Note

- The full display of all statuses is only possible for recipe controls from Version V5 because ISA-88 is only supported from this version onwards.
- Clicking on the bitmap opens the operator dialog (refer to the section Control dialog (Page 506)).

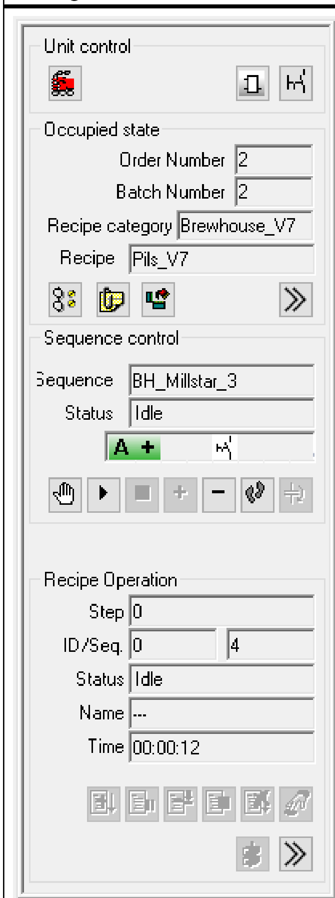
"Line with title" mode

"Line without title" mode

The unit control line display consists of:

- The title line – can be displayed or hidden statically upon configuration via the mode. A table can be set up inside the process image with both variants.
- Different fields with attributes of the selected sequence
- The scalable windows UNIT DFMs, EOP DFMs – can be hidden/shown dynamically using the button row.
- The setpoint value windows are only opened for one sequence in each case. If the setpoint value window is opened for an additional sequence then the window that is currently open is closed beforehand.
- The sequence command buttons always remain displayed in this view.

"Window" mode

Dialog	Content
	The displays and control objects are broken down and displayed grouped into the following areas: • Unit control • Occupied state • Unit control • Recipe operation • The command buttons always remain displayed in this view.

Unit control

- Call of the application process cell overview
- Call of the "Units permanent condition" A distinction is made automatically between bit memory allocation or a BLR interconnection and either "S7-Status" or "BLR-Runtime-View" is opened.
- Toggle additional unit on/off

Occupied state

Here, the current allocation of the unit is displayed.

- Order number
- Batch number
- Recipe category
- Recipe

The following buttons are also available for branching (from left to right):

- Call of the control recipe view for the current batch of the unit
- Shows the order parameters for the batch
- Shows the process input of the batch
- The ">> /<<" button opens/closes the setpoint window of the sequence (refer to the section Setpoint window (Page 507)).

Sequence control

Sequence control section

- The following commands can be executed using the buttons (from left to right):
  Auto/manual, Start, Cancel, Release , Hold, Step selection (if this has been enabled in the "Properties" dialog)

Recipe operation

The current step status is displayed here.

- Step number
- Identifier and unit
- Status (S88 status)
- Name
- Runtime

The following commands can be executed using the buttons (from left to right):

- Continue, Hold, Pause, Stop, Cancel, Operator request
- With the "Details" button, you can branch to the status program for the block
- The ">> /<<" button opens/closes the setpoint window of the recipe operation

"Line can be configured" display mode

One of the following variants can be selected in this mode during configuration in the "Properties" tab via the "Index" attribute:

- Display variant "Index=0" (with title)

- Display variant "Index=1" (without title)

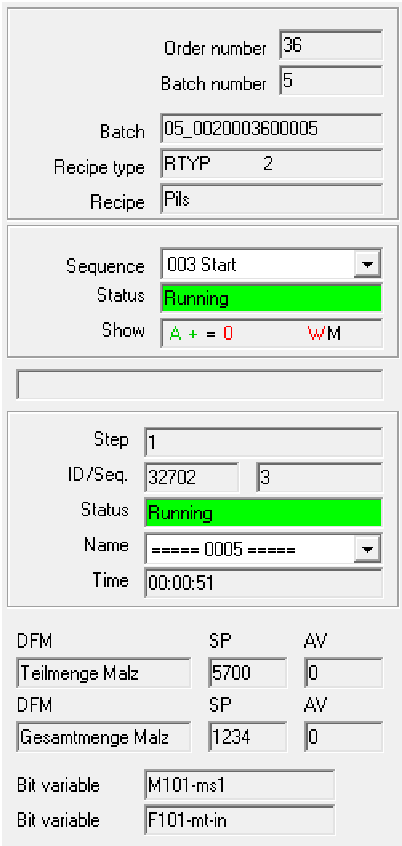
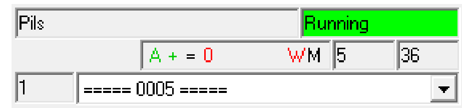
- Display variant "Index=2" (with title and all fields)

The information displayed here corresponds in principle with the normal line display but with the following differences:

- The configured info objects and control buttons are displayed
- A separate control dialog is opened by clicking the mouse on the line (see section "Control dialog" (Page 506))
- The size and the displayed objects can be configured (see section)

"Window can be configured" mode

One of the following variants can be selected in this mode during configuration in the "Properties" tab via the "Index" attribute:

<table>
<tr><th>Variant Index=1 (with all objects)</th><th>Variant Index=0 - (standard selection)</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

The information displayed here corresponds in principle with the normal window display but with the following differences:

- The configured info objects and control buttons are displayed
- A separate control dialog is opened by clicking the mouse on the window (see Control dialog (Page 506) section "Control dialog for the "Line/window without control dialog" mode)
- The following additional values are displayed in the left dialog window:
  - Up to 3 DFMs with name strings for each along with current setpoint and actual value
  - Up to 3 bit variables with name string in each case and current actual value
  - The connection is made in the Properties dialog
  - The display and the size of the object fields can be configured (see section Unit Control Faceplate (Page 468))

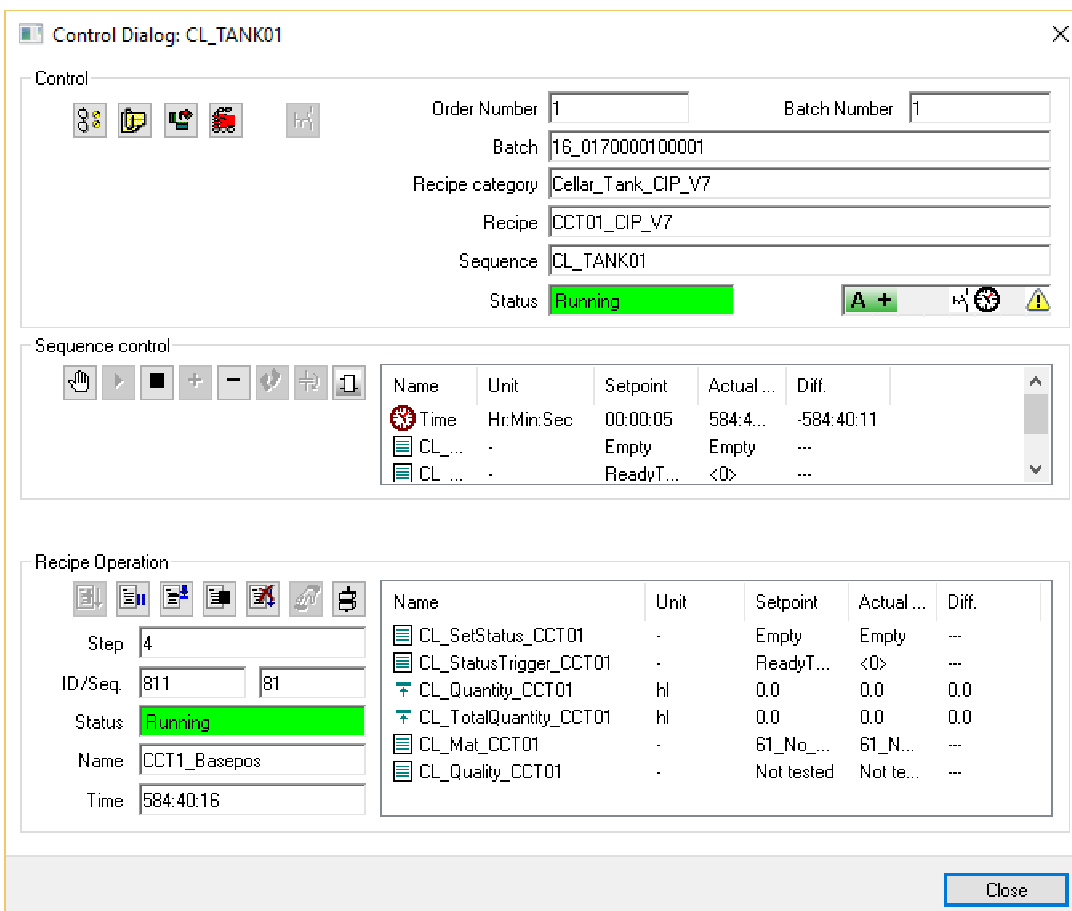
Acknowledgments in the sequence status display

The display of errors, warnings and process messages are removed from the display by actuating the corresponding button in the toolbar.

	Removal of messages configured as "Error" message class
	Removal of messages configured as "Warning" message class
	Removal of messages configured as "Process message" message class

Control dialog

Control dialog for the "Row/window configurable" representation



The dialog is split into the following sections:

The control and display elements correspond in principle with those of the "Window" faceplate view (see previous section). It is just the special features that are described here:

- Control section
  - The following additional windows can be opened using the buttons (from left to right): Current recipe, Order parameters, Process input, Process cell overview, Additional unit on/off
  - The current batch data are displayed for this unit in the right part of the image
  - Another unit can be selected via the selection list in the "Sequence" field if this has been enabled in the "Properties" dialog
- Sequence control section
  - The following commands can be executed using the buttons (from left to right): Auto/manual, Start, Cancel, Release , Hold, Step selection (if this has been enabled in the "Properties" dialog)
  - Call of the "Units permanent condition" A distinction is made automatically between bit memory marshalling or a BLR interconnection and either "S7-Status" or "BLR-Runtime-View" is opened.
  - The setpoint window of the unit is displayed in the right part of the image
- Recipe operation section
  - The following commands can be executed using the buttons (from left to right): Continue, Hold, Pause, Stop, Termination, Control request, Details with "Status.exe" selection
  - The setpoint window of the recipe operation is displayed in the right part of the image

See also

Views (Page 500)

Setpoint window

Basics

Name	Dim.	SP	AV	Dev.
Time	Hr:Min:Sec	00:00:00	00:00:00	00:00:00
Destination silo	-	Silo 1		
Consecutive silo	-	2	2	0

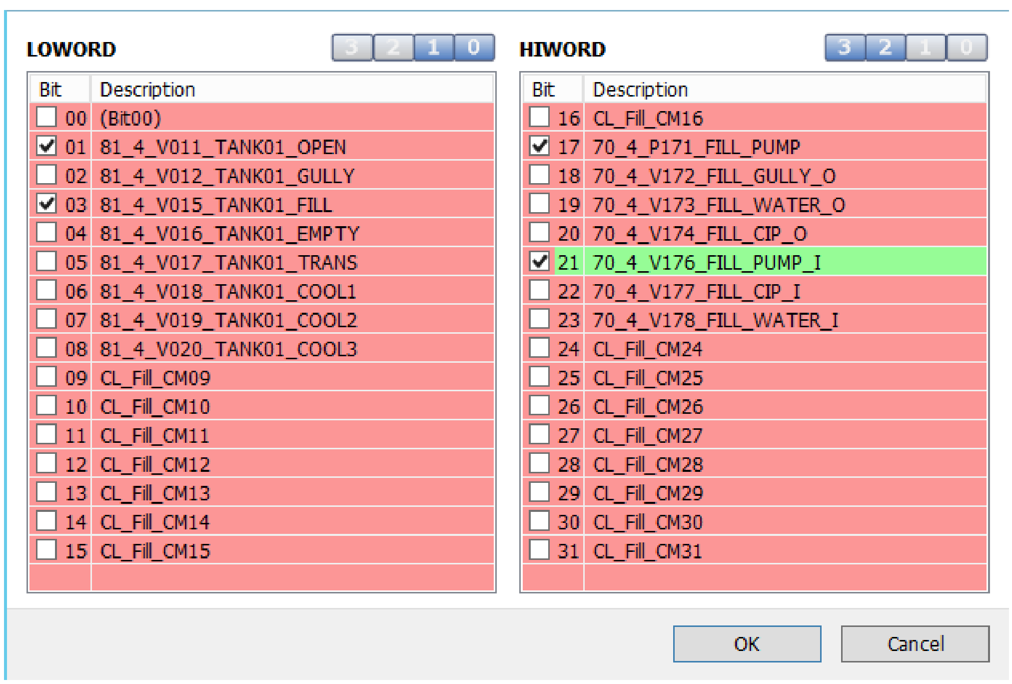
In the window, the sequencer- or step-related setpoints are displayed.

- Depending on the parameter type, the setpoints can be edited or selected in special selection dialogs.
- The special features of the individual parameter types are described in the section 'Edit DFM definitions' dialog (Page 581).

- The size and column width of the setpoint windows can be changed.
- The affected adjustments are filed.
- The window for the setpoints of the step and the sequencer have different adjustments. The settings are saved to the file '<local-proj-path>\sys\controls.ini'.
- [unitctrlUSP]
  For the setpoint window of the sequencer-related setpoint.
- [unitctrlSP]
  For the setpoint window of the step-related setpoint.

"Text (Bitfield)" parameter type (PCU from V7)

- The setpoint is defined by a bit pattern (32 bits).
- The display uses a separate dialog in which the 32 bit positions can be set/reset using select boxes.



Note

Update and representation of the setpoints as well as the associated actual values

The displayed bit information from the PCU DFM object is formed when the dialog opens.

- Check mark is set → setpoint bit = true
- Check mark is not set → setpoint bit = false
- Line green → actual actual value = setpoint
- Line red → actual value ≠ setpoint

Caution:

The statuses are not updated when the dialog is open. The statuses visible here are written to the DFM object in case of changes to the setpoint bits and confirmation with the OK button, regardless of whether the statuses have changed in the PCU in the meantime (e.g. through step forwarding).

Suppressing adjustments filing

Filing the adjustments can be suppressed.

WriteSettings=0

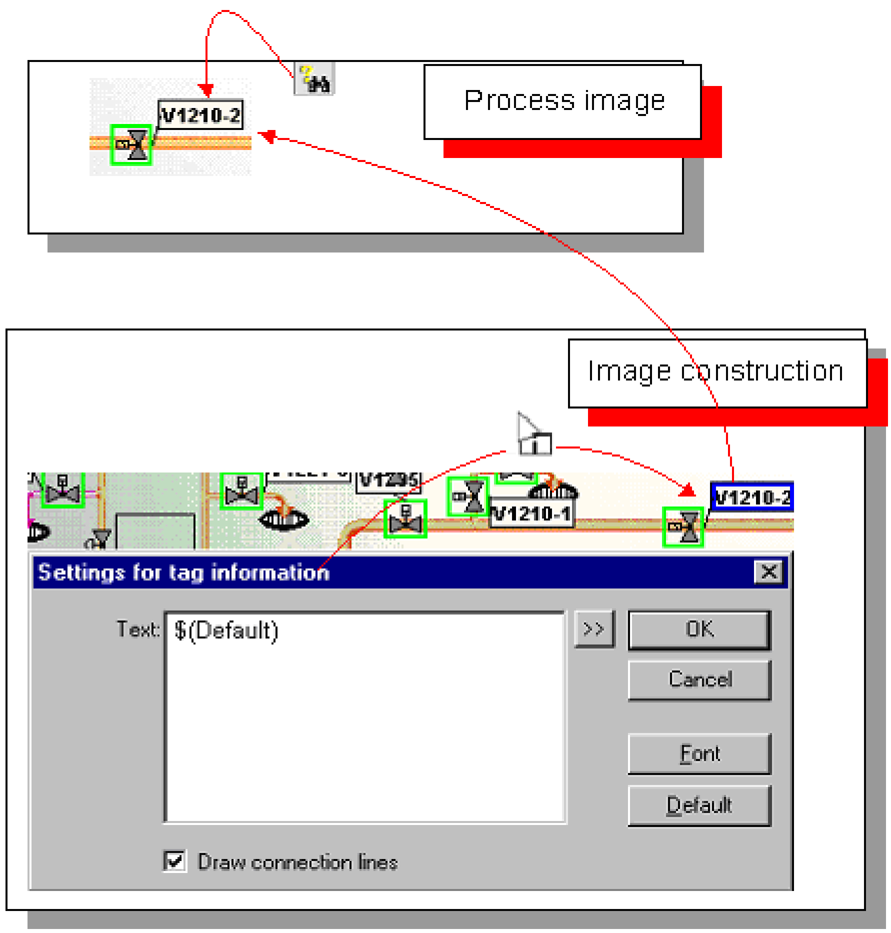
By opening the window the adjustment that has been set one time is always set consequently. Manual adjustments aren't filed.

6.2.4 Variable search

6.2.4.1 Variables information

Make the variable information visible in the runtime system

By activating the symbol in the button bar or the menu item ‘Process diagrams’->’Variable information’, the configured information from the image construction becomes visible.



Switch On/Off variable information of an object

If individual info fields have been made visible/invisible then proceed as follows:

1. Open the image with the image construction.
2. Activate the variable info function via the binocular symbol in the symbol bar or menu command Variables / Info. The mouse cursor is enhanced with an I symbol as long as this function is active.
3. Using the mouse click directly on the element (not on the info text) in order to remove the configured info text.
4. The info text can be made visible or invisible with each further mouse click on the element itself.
5. Save the image with the desired setting.

6.2.4.2 Variable search in the image

- When clicking on the symbol in the icon bar or with the selection of the menu item Process diagrams →Variable search within the image, a dialog appears, in which all existing objects of the image are listed.
- The variables can be displayed filtered.
- The requested object group is selected in the "Filter" list box.
- After selection of the object group, all available objects of this group appear in the overview window in the image.
- After clicking the found object, it flashes on the screen and is shown with green marking.
- In the Information are, the name from the text parameterization is shown with the selected object.

Note

- If a newly selected object is in a covered or invisible area of the screen, the green marking remains on the previously selected object, in some cases, but no Io flashes.
- The newly selected object has to be made visible by moving the window.

6.2.4.3 Global variable search

Basics

By clicking on the symbol or selecting the menu item Process diagrams → Variable search → global a dialog window with two tabs, "Search" and "Scroll" appears.

Seek Tab

The search is performed here by object name.

- Searches can be performed by object name (also partly qualified, but without '?' or '*') with selectable search options.
- The "start" button is used to start the search across all process diagrams.
- The objects found are then listed in the "Search results" section.
- Detailed information about a selected object (PCU, module, number) is shown in the fields underneath.
- After clicking on the button "Goto process diagram", a jump into the corresponding image is made.

Browser Tab

The object is selected as follows:

- Selecting the PCU
- Selecting the object class (module)

- Selecting the object instance
- After clicking on the button "Goto process diagram", a jump into the corresponding image is made.

Additional information

- The pictures that are to be branched to are stored in a text file for each PCU / for each object class under '<proj-path>\PCU.xxx\IMAGE\<ObjectClass>.txt'.
- The syntax is the same as for the instance name file, but with a Len parameter of 128 instead of 32.
- The class-related picture directory files are created or updated automatically in the picture configuration 'BiKo.exe' in the system delivery state.
- Alternatively, they can also be created or updated by the user via text parameterization.
- For generation via BIKO, the following button has to be set in the configuration file 'biko.ini'.
  ```
  [Options]
  SaveImageInfo=1
  ```
- If instances exist more than once, the last saved image receives the cross-reference.
- These picture directory files are also used by the 'Loop in Alarm' function of the message window (see Message View Control (Page 496)).

6.3 Replay mode

6.3.1 Basics

Core statement

The **"Replay-Mode"** feature is not to be understood as the visual recording and subsequent replaying of selected process diagrams with the specifically performed operator inputs. Instead it involves the monitoring and replay of any process diagrams and visualization objects also those not previously selected which are then supplied with recorded historical process data instead of online data. The start time and the time interval can be preselected in much the same way as with a video recorder or media player. As long as a process diagram instance is set to Replay mode, only operator input such as change process diagram and opening faceplates and their dialogs/tabs is possible. All other operator input in particular entries, changing values or switching operations are however blocked.

Application

The main application is that with the Replay mode error statuses that occurred in the past can be totally and conveniently observed and localized in terms of their origins and their potential effects.

6.3.2 System architecture

Basics

In replay mode the tag source requirements of a control application (currently only "Process diagrams" application) are supplied with values from a "DataLog" archive instead of online values from the S7 controller. The components of the recording and replay infrastructure required for this have been completely integrated in the existing system architecture. This results in a number of benefits:

- The Replay mode is available parallel to online mode on all client and server IOS stations.
- No dedicated replay servers are needed for the data logging.
- No particular configuration of the process diagrams is necessary and even process diagrams created before the upgrade are supported without change.
- No interventions are required in the AS software, neither in the runtime system nor in the user blocks.

What is recorded and how?

In principle, the data recording includes all the block parameters that are needed for visualization and operator control in the process diagrams for HMI objects (tags, controls). To do this, in the parameter assignment these attributes were given the new "LOG identifier" by the system. When generating the requirement list for the data logging (TAG list) then only these are taken into account. During the data logging a distinction is made between Runtime- and Engineering-attributes. The latter are subjected to a slower acquisition cycle. In addition to this all indirectly connected objects and all addresses referenced in block parameters (P, I, Q, M, DB) and objects (BLR / Interlock, Perm-Cond, Next-Step-Cond) are included in the data logging. Finally the DFMs referenced in the sequences and EOPs are analyzed and their setpoint and actual values are recorded as Engineering- or Runtime-attributes.

Process data logging

In addition to the previous online communication via the "S7 Port" a further communications port "S7 Logging Port" is now introduced per IOS server. This ensures the cyclic requesting of the data points stored in a TAG list and passes them on to the subsequent processing (compressing) and archiving. The TAG list has the range of validity of the AREA associated with this IOS server, it therefore includes the process data of all PCUs from this AREA released for the logging.

Functions:

- Establishment of 1...4 additional S7-LOG connections to every PCU
- Processing of the TAG lists per PCU and cyclic reading of the data points in continuous operation
- The cycle time results from the data volume and from the communication performance of the CPU type used.
- This communication performance can be influenced within limits in the hardware configuration of the CPU module.
- The current cycle time per PCU is displayed in the PCU server window.
- The process values are buffered in a cache and stored with a time stamp only if they change (with analog values taking a hysteresis into account).

Data storage

The compressed process values are stored in the archive path of the project. To optimize performance and the data volume, the binary format was selected. The data points of a PCU are acquired per address area (P, I, Q, M, DB) and per data block and saved day-related in individual files. The data structure is organized as follows:

- "`<arch-path>\LOGGING\DataLog\...`" Basic directory
- "`...\DL_yy_mm\...`" one folder per month
- "`...\DL_yy_mm_dd\..`" one folder per day
- "`...\PCU_nnn\...`" one folder per PCU
- "`...\DB_nnnnn.dat`" one file per address area or per data block

This results in the following path system:

- Archive path: "`<arch-path>\LOGGING\DataLog\DL_yy_mm\DL_yy_mm_dd\PCU_nnn\DB_nnnnn.dat`"

Replay mode of process diagrams

The tag requests of process diagrams in replay mode are supplied from the LOGGING archive via the new, additional PCU server miniport "Replay Port".

It has the following properties:

- Dedicated state machine per logged-on client IOS → support of individual operator actions (Start, Pause, Time, Interval)
- Synchronization of the replay parameters for process diagram instances of the same IOS station
- Setting of replay time from other applications (Message Control, Alarm Viewer, Batch Archive)

6.3.3 Configuration limits and diagnostics

Basics

Because the data volume and expected performance precludes recording of all relevant process data points of a PCU, a selection of the object classes/instances and address references actually used in the project must be made. This selection can be implemented most effectively by searching through the process diagrams. In so doing, a tool in the "Diagram creation" application is used to analyze the tag information of the interconnected objects and absolute addresses and this information is saved as a TAG list for each IOS server. This is saved as a dBase file in the project on the following path:

- "`<proj-path>\LOGGING\DataLog\DataLogDef.dbf`"

Depending on the number of process diagrams selected for the Replay mode (ideally all the pictures of a project) and the connected visualization objects, the number of TAGs to be recorded per AS can vary greatly. The extra communications load resulting from this must not, however, disturb the normal process operation significantly, in other words reaction times with online operator input, recipe control, route control and the standard logging functions (trending, step logs, change logs, ...) must be available without restrictions.

Based on the technical conditions of the utilized CPU types and performance measurements during system testing, limits have been defined for the maximum number of tags per AS as a function of the CPU type. These limits can be found in section Quantity schedule of functions and classes (Page 65). Note that the numbers named there apply only to the CPU types of the upper performance segment and the latest generation.

Note

Exceeding the maximum possible number of TAGs

- In projects with PCUs under a higher load, it can occur that the maximum number of TAGs is exceeded. This is reported already when the TAGs are generated and results in the generation being aborted with an error message.
- The user then has the opportunity by deselecting process diagrams less important for the Replay mode of coming within the limits again.
- The critical number of TAGs per type and PCU can be followed in the generation log.

Recording cycle - RT and ENG attributes

When reading in the TAG list into the "S7 Logging Port" at the time of the PCU server startup, the TAGs are divided into two different internal request lists. Depending on whether runtime or engineering attributes are involved, a fast or a slow recording cycle is used.

1. PI tags - runtime attributes (parameter assignment / A.type = RT)
   The recording cycle is as fast as possible so that brief switching operations can also be captured. A cycle time of 1 ... 1.5 sec. is adequate here.
   The cycle actually reached sets itself individually per S7 logging connection and depends on the number of TAGs and the communication performance of the CPU used (CPU type and generation).
   The current number of RT TAGs and the cycle achieved is displayed on the PCU server / Coupling status in the "PA" column.
2. PI TAGs - engineering attributes (parameter assignment / A.type = "ENG")
   The system limits the cycle time to a minimum value of 8 sec. so that more communication performance is available for the runtime attributes.
   A recording cycle of several seconds can be tolerated because in particular following commissioning these parameters are seldom changed.
   The current number of ENG TAGs and the actual cycle is displayed on the PCU server / Coupling status in the "PI" column.

Recording cycle - limits and recommendation

Due to the stringent requirements for communication performance, the use of the "Replay Mode" feature can only be recommended for CPU types of the top performance segment and the latest generation.

For older S7-400 CPU generations (especially before the "05" generation) or for CPU types of the lower performance segment, cycle times are already long even with a low tag count and the expectation regarding extensive coverage of all process data cannot be fulfilled. Replay mode must then be limited to a few process diagrams.

Note

Recommended CPU types for replay mode

Specifications for performance of CPU types with process data recording can be found in the readme document / section "HW requirements".

Effect on the cross-coupling and the Route Control System (S7-400 only)

In particular, when cross-coupling that is also used by the Route Control System is used, attention must be paid to the effect of the additional S7 connections and communication load of the CPU caused by the process data archiving. As a result, the cross coupling jobs can also be executed more slowly.

Increasing the communications performance of the CPU

The prioritization of the communication jobs in the SIMATIC S7 CPUs can be influenced within certain limits in the CPU settings. This also affects achievable cycle time in TAG logging.

- **S7-400: Step 7 / CPU HW Configuration** / CPU Properties / Cycle+Clock Memory tab / Cycle load due to communication(%) → Default setting = 20%
- **S7-1500: Step 7 TIA Portal / Device Configuration** / CPU Properties / Communication Load / Cycle load due to communication(%) → Default setting = 20%
  The default setting can be increased to 30% or 40%.
- **Influence on the OB1 cycle**
  If user blocks are called in the OB1 cycle in the project a slower OB1 cycle must be taken into account. The cycle time monitoring may need to be adapted.

Note

Less communications load by reducing the number of TAGs

When generating (see section "Configuring data recording for Replay mode (Page 519)") by deselecting process diagrams less important for the Replay mode the number of TAGs and the CPU communication load due to process data logging can be reduced. This is recommended in the following cases:

- If the cycle time of the PA TAGs of S7 logging connections (Runtime attributes) reaches or exceeds values of 5 sec.
  The warning limit is 5 sec. and is signaled by a blue font color.
- If the detection time of signal changes in Replay mode is no longer satisfactory and a faster recording cycle is required.
- If the cross-coupling jobs (including RCS) are no longer executed fast enough.
- if the OB1 cycle exceeds values that can no longer be tolerated by certain user blocks.

6.3.4 Configuration

6.3.4.1 Configuration settings

Basics

After system installation or after an update using the product setup, the new available configuration settings for the "Replay-Mode" are initially deactivated. They are activated in the "Management/Configuration" application using release buttons in the settings dialogs of IOS and PCU stations involved.

The following constraints should be kept in mind:

- The visualization objects interconnected on a particular PCU can only be supplied with historical data when PCU data logging is activated.
- With redundant servers, to avoid communication load, data logging may only be activated on one of the servers. If this is ignored, when the configuration is saved a dialog pointing this out is displayed.
  Recommendation: The server that runs predominantly as the standby server should be used for the data recording.
- The logged data ("DataLog" archive) is automatically distributed to both redundant IOS servers. This means that client access in Replay mode is handled in an identical way as in online mode.

Step 1: Enable data logging

Replaying process diagrams with historical data is only possible if data logging of the variables contained was performed previously. To do this the following configuration steps are necessary:

- Data logging must be enabled in the settings of the relevant IOS and PCU stations in the "Administration / Configuration" application:
  - for each desired AREA on an IOS server, see Modifying IOS settings (Page 114),
  - for each desired PCU, see Modifying PCU settings (Page 121)
- The PCU server must be restarted. This starts the communications ports for data logging and Replay mode

Step 2: Configure data logging for replay mode

In the "Engineering / Process diagram design" application, a TAG list must be generated for all process diagrams that are relevant for replay mode (see Configuring data recording for Replay mode (Page 519)).

Step 3: Enable replay mode

In the "Administration / Configuration" application, replay mode must be enabled for the desired IOS stations (see Modifying IOS settings (Page 114)).

Limiting archiving period

Depending on the number of TAGs per area server and depending on the size and frequency of the batch throughput, in time the DataLog archive grows more slowly or more quickly. To limit the memory required on the archive path (**"<arch-path>\LOGGING\DataLog\..."**) the archiving period can be configured in days (see section Area settings (Page 111)).

Note

Limiting the storage space

- The default setting is 90 days. This period can be varied between1 and 365 days.
- This means that the system automatically deletes older data so limiting the storage space. This also applies to the redundant IOS server.
- If the value is reduced, the user is informed that at the next system start older process data will automatically be deleted.
- This also affects older archives perhaps played back by the user that are not write protected.
- Value = 0 days means: The deleted function is turned off and the archiving period and occupied storage space can grow freely.
  - Disadvantage of this setting:
    The archive drive (possibly identical to the project drive) can be filled up the limit of the disk capacity.
  - Advantage of this setting:
    Older process data that is no longer required can be swapped out by the system administrator to an external data medium and later played back.
- **In any case, rotational monitoring of the occupied storage space and possibly a manual intervention by the system administrator is necessary and is strongly recommended.**

6.3.4.2 Configuring data recording for Replay mode

Generating the TAG list for the process data logging

In the application "Process diagram design" there is a dialog available for generating an AREA-related TAG list for the process data logging. Here the process diagrams are selected that will later be used in Replay mode. The TAG list can be generated on every client or server IOS. This always a complete generation that replaces an already existing TAG list. It is stored on both redundant IOS servers.

Note

Updating the TAG list after changes to existing diagrams

If existing process diagrams are extended or new process diagrams created for which the Replay mode is required, the TAG list must be generated again.

Note

Ending creation of diagrams on other IOS stations

During generation none of the process diagrams selected here may be open with the "Process diagram design" application on another IOS station otherwise correct saving of the selection markers is not possible.

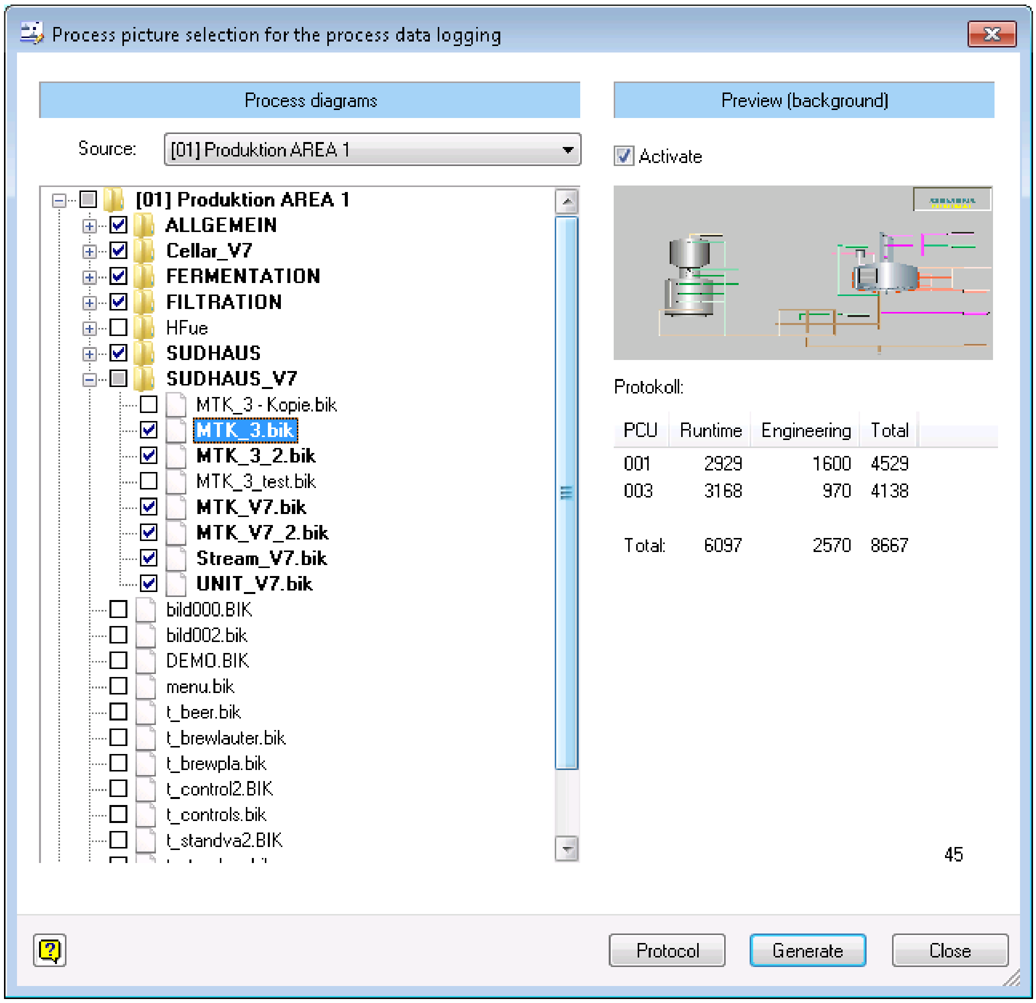
Note

Updating the offline DBs before generating the TAG list.

If configuration changes are made in online mode, prior to generating make sure that the offline DBs are up to date. This applies especially for the following parameter assignments:

- Sequences: Reference permanent condition `"PermCondRef"` to BLR logic element
- ICM: Reference interlock `"RefInterlock"` to BLR logic element
- SLB: Reference of the type OREF `"PartnerICM"`
- General reference attributes to absolute addresses or object instances in various blocks (type GREF, PREF)

Control dialog



Operator actions when generating TAGs

Action	Result
Call menu item "Tools/Create tags for process data logging"	• The dialog shown is opened. • The "Bilder" directory is read in regardless of the currently open process diagram. • The process diagrams selected for the last generation are marked again.
Select the AREA in the "Origin" drop-down list	The picture directory of the selected AREA is read in and displayed in a tree view.
Select the Process diagrams in the tree view Single diagrams, folders/levels or mixed selections are possible	• The selection for the next generation is made. • The number of selected diagrams is shown in the total box at the bottom right • The diagram selection is **not saved yet!**

Action	Result
Enable preview (background)	The background picture of the currently selected process diagram in the tree view is displayed as a miniature.
"Generate" button	• The TAG list for the selected process diagrams is generated • It is stored in the file `"<proj-path>\Logging\DataLog\DataLogDef.dbf"` on both redundant servers. • **The previous TAG list is always completely overwritten.** • The current diagram selection **is stored (on both redundant servers).** This is achieved by setting a flag in the relevant process diagram. There is no separate list for diagram selection. • An empty TAG list (no process diagram selected) is supported. There is then no data recording for this AREA.
Analyze log brief information	Contains the summary of the last generation per PCU with the following values: • PCU → PCU number • Runtime → Number of runtime TAGs • Configuration → Number of configuration TAGs • Total → Total TAGs (of the PCU) For further information, see the section below....
"Log" button	The log file of the last generation is displayed in a new window.
"Close" button	The "Selection ..." dialog is closed..

Supported types of connection

The visualization objects configured in the process diagram (variables, controls) are analyzed in terms of their connection. The following source types may occur:

- Direct addressing
  The configured absolute addresses are adopted in the TAG list.
- Indirect addressing
  The fields referenced in the corresponding database file (*.DBF) via the reference variables are analyzed and the configured object instance or absolute addresses are resolved. From this the recording tags are obtained and added to the TAG list.
- Direct object addressing
  The LOGGING attributes for the connected object classes/instances are identified.
  For this, the "LOG" attributes are specified in the definition files of the object classes ("PARAMS7.xxx"). During TAG generation these decide which data points per object are logged.

Displaying the LOG attributes in the "Parametrization" application

In the application "Engineering / Block parameterization" the LOG attributes can be filtered by object class. For this the Options menu contains the menu item "Display LOG attributes" and a corresponding button in the toolbar. When using this all other attribute switches should also be set otherwise you may not obtain the full LOG attribute list.

Referenced BLR elements

The BLR instances to be logged are identified depending on their category of use based on the objects connected in the process diagrams:

- Sequences: Reference permanent condition `"PermCondRef"` to BLR logic element
- ICM: Reference interlock `"RefInterlock"` to BLR logic element
- BLR for EOP step enabling condition are identified from "epe.ini" ("NSCBlrObj")

Identifying the DFM instances to be logged

To display historical data in the setpoint windows of the unit control, the used DFM instances are identified as follows:

- Unit-related DFMs are identified from "sequence.ini"
- EOP-related DFMs are identified from "epe.ini"
- At the unit control itself up to 3 DFMs and 3 bit variables can be set

6.3.5 Replay mode

6.3.5.1 Replay mode operator panel

Basic principle

Besides online mode, the "Production / Process diagrams" application has a new operating mode – "Replay mode". The Replay mode can be activated or deactivated with a password query in the open application. If an instance of the application is in Replay mode, it interrupts its online connection to the SIMATIC S7 controller and connects itself instead via the PCU server/Replay port to the "Datalog" archive of the IOS server.

Operator panel: Input boxes and buttons

In Replay mode at the bottom edge of the process diagram an additional operator panel is displayed that allows the following operator input:

- Preselection of the Replay time (date, time)
- Slider for fast changing of the Replay time (min:sec) to the preselected hour
- Entry of the Replay interval (in sec where n = 1 ... 60)

The Replay interval defines the time interval between Replay mode points in Start mode and therefore the ratio of the Replay mode speed to real time (n=1 : Real time replay; n>1 : time lapse replay with n * real time)

- Start button - Start/continue replay with interval
- Pause button- Pause replay (retain replay time)
- Fast forward/backward button - Set replay time 1h forward/back
- Forward/backward button - Set replay time 1min forward/back

Operator panel: Fast operation with keyboard

Fast changing of the replay time with keyboard:

Shortcut key	Function
Arrow left	1 sec back
Arrow right	1 sec forward
Arrow up	1 min back
Arrow down	1 min forward
Pos 1	Start of the current h (xx:00:00)
End	End of the current h (xx:59:59)

6.3.5.2 Behavior of the process diagrams

Basic properties of the Replay mode

- Existing process diagrams are supported without any changes. With a few exceptions (RCS control, Exec_FB, Status_S7...) all images can be visualized in Replay mode.
- The TAG lists for data logging can be generated directly in the Process diagram design application.
- Several instances of the application can be open at the same time in each case in online or in Replay mode.
- On the same IOS station Replay mode is possible with several instances with the same, in other words synchronized replay parameters (time, interval). All operator input (Start/Pause/Forward/Back) and the replay data are automatically synchronized.
- On different IOS stations individual reply parameters are supported.

(See also section Properties of the replay port (Page 529))

Special features in the Replay mode

- Only the images that are connected to data sources or process values in the AS can be supplied with historical data.
- All the information (instance names, info texts, notes, ...) read in from data storage (files on the project path) reflect the current status also in Replay mode. Configuration changes to these files are not detected by the data logging.
- Status identification of the display fields when there is no process value at the replay time or when there is no connection to the Replay-Port of the IOS server.
  - "#" = numeric fields
  - "---" = text fields

Note

Operator buttons and input fields

In replay mode, only operator inputs for navigation are possible, for example, changing the diagram, opening faceplates and their dialogs/tabs. All other operator inputs, especially entries, value changes and switching operations, are blocked however.

6.3.5.3 Behavior of the visualization objects

Basics

Only the images that are connected to data sources or process values in the AS can be supplied with historical data. All the information (instance names, info texts, notes, ...) read in from data storage (files on the project path) reflect the current status also in Replay mode. and are not acquired by the data polling.

In the following sections you will find details on the behavior of the visualization objects in Replay mode:

Standard variables

These standard variables are supplied with historical data if they are connected to data sources in the AS.

- Value
- Hexadecimal
- Bit
- ICM
- Text
- Line
- Bar
- Sound

In Replay mode the following standard variables have the same behavior as in online mode since they are not based on data sources or process values from the controller.

- Image change
- Static text
- Static picture

Special situations:

- Exec IOS is released in Replay mode For example when "Kurven.exe" is called the transfer parameters derived from standard variables are also supplied with historical data (year, order no., batch no.).
- Exec-FB is blocked in Replay mode because an AS block call only makes sense in online mode, in other words in a real-time context.

Controls

The controls listed below are supplied with full historical data if they are connected to data sources in the AS.

- Analog Control
- Digital Control
- Controller Control
- Entity Control
- Line Control
- Tank Control
- Status Byte Control
- Smart Unit Control

The following controls have the same behavior in replay mode as in online mode since they are not based on data sources or process values from the controller:

- Panel Control
- Image Navigator Control

Note

Info tab

The Info tab (description, address, location, note) does not contain any historical data but the current status of the files in the project path (<proj-path>\PCU.nnn\ADDRESS.nn, ...\DESC.nn, ...\LOCATION.nn, ...\NOTETXT.nn).)

Behavior of Date-Time Control

- The date and time information shows the replay time.
- The symbol for replay mode is displayed instead of the logged-in operator.
  The reason for this is that user logons are always IOS-related and generally several users are logged on. The later examination od process diagrams in Replay mode in terms of IOS station and logged on users bears no relation to the conditions at the time of logging.

Behavior of ICM Control

- The status button is not active with bit memory interlocks
  Only if there is BLR interlock logic configured is a branch made to the BLR window and the logic is supplied with historical data.
- The dynamic RCS unit data is not displayed.
- The maintenance data actual values (hours / switching cycles) are not displayed.
  Maintenance message texts (setpoint 1 .. 5) always show the current configuration status.

Behavior of Message View Control

- If the Replay mode is active the messages shown by the control are displayed time coordinated at the relevant replay time.
- Compared with online mode, the messages are then read from the message archive and displayed only up to the currently valid replay time.
- The replay time is supplied cyclically by Replay-Port according to the specification in the replay operator input box.
- An earlier message line can be specifically selected in the list, and the replay time can be set to this exact message time.

Behavior of Unit Control

The types of display of Unit Control are also supported in Replay mode:

1. Icon / operator input dialog
2. Line with title
3. Line
4. Window
5. Line configurable / operator input dialog
6. Window configurable / operator input dialog

The display types and the operator input dialog show the following special features:

- All display boxes are supplied with historical data.
- The setpoint windows (sequencer and step-related setpoints) can be opened normally and all boxes show the historical data of the replay time.
- Here the following also applies:
  operator input boxes, selection boxes and buttons are switched inactive in Replay mode.
- The operator buttons for the sequencer permanent condition and for the EOP step enabling condition behave as follows:
  - With memory bit assignment the button is switched inactive because the display of the S7 status cannot be supplied with historical data.
  - With a BLR connection, the "BLR runtime" window is opened. This is supplied with the historical data of the replay time.

Behavior of the BLR Runtime window

If a BLR object is referenced as a sequencer permanent condition, as an EOP step enabling condition or as an ICM interlock, in Replay mode when the status view in the relevant control is opened, the BLR Runtime window is opened and supplied with the replay data.

Note the following here:

- The binary signals (inputs, output) and logic statuses (TRUE/FALSE coloring) are filled with historical data.
- **The logic structure itself (gate type / size and connections) is read from the current online/ offline DBs and therefore always corresponds to the current configuration.**
- **As a result, the signal status display is only valid if the logic configuration has not been changed after the current replay time!**
- A change to the logic configuration can be recognized in the change log.

Note

Opening the archive viewer via the shortcut menu

It is possible to branch to the message archive via the "Operator inputs and messages ..." shortcut menu. In so doing, the currently selected BLR instance and the replay time are transferred as the filter criterion. You then receive the filtered list of all operator inputs and messages recorded for this BLR instance and this tag. A line can be specifically selected in the archive viewer, and the replay time can be set to this exact message time.

Behavior of RCS Control

Since no trending is available in "Route Control", the **RCS Control is switched inactive in Replay mode** (icon not visible).

6.3.5.4 Properties of the replay port

Basics

In the "replay port" of the PCU server the replay instances (e.g. process diagrams in Replay mode) of an IOS station are controlled via their own state machine (technical programming "Thread") in terms of replay parameters and status and supplied with the requested process tags from the LOGGING archive. A separate Thread is kept for each connected IOS station. This allows different IOS stations to act in Replay mode independently of each other.

The PCU server window "Coupling State" provides a detailed status of the replay port over all logged on IOS stations. There for each connected IOS station a status row with the following parameters is displayed.

Parameters	Description
IOS xxx:	IOS number The following values include all replay instances (e.g. process diagrams in Replay mode) of an IOS station.
InternalTags=	Number of internal TAGs
PaTags=	Number of registered variable process TAGs [... sec] = Update cycle
Cycle=	Interval replay time (Cycle in sec) Is set by the replay operator input box.
Time=	Replay time (DateTime) - standing or running value Can be preset with the replay operator input box or by the command "Set Replay Time" in "Message-ocx" / "Archive viewer".

Parameters	Description
RUNNING, HELD, IDLE ...- Time-To-Held (9 sec countdown) ...- Time-To-Idle (59 sec countdown)	Status of the replay "Thread" • RUNNING: Replay running, variables "PaTags" are returned from the archive at the interval "Cycle". Supplement "Time-To-Held": Waiting time for the automatic switchover RUNNING → HELD • HELD: Replay held, variables "PaTags" are returned with a frozen replay time "Time". Supplement "Time-To-Idle": Waiting time for the automatic switchover HELD → IDLE • IDLE: no Replay mode is requested in the IOS The waiting times are automatically inserted by the system if there is a diagram change or if process diagrams are closed in Replay mode.
Text color (green) Text color (blue) Text color (black)	Status = RUNNING Status = HELD Status = IDLE

At the end of the section the Total row follows with the number of: Clients, Internal Tags, PA-Tags

Setting the replay time by other applications

Apart from setting the parameter "Replay Time" IOS related by operator input in the process diagram, it can also be derived using the shortcut menu "Set Replay Time" from messages and change logs in other applications. In this case it does not matter which status the replay Thread currently has, in other words whether or not a replay instance is open. The replay time remains stored in the replay port and is used automatically with the next request.

This shortcut menu function "Set replay time" is supported in the following situations:

- Message window in process diagram or in system overview → Time of selected message
- Archive viewer application (messages and changes) → Time of selected message or operator input
- Batch list / Batch archive application → Start time of the selected batch

6.3.6 Typical applications

Basic principle

To support the operator with the efficient identification and diagnostics of exceptional statuses based on the Replay mode it is necessary to establish a simple and intuitive connection between the error event (signal/status in the process diagram or message in the message window or message archive) and the replay time. Since the replay time is a property of the Replay-Ports of the IOS, it can also be set outside the operator panel in the process diagram, for example in the message archive or batch archive.

Below a couple of practical applications for daily system operation will be shown.

1. Starting point is a visualization object (tag, ICM, sequence, etc.) in the process diagram or the BLR online view.
2. The starting point is any message or change in the 'Archive viewer' application
3. The starting point is a message in Message Control (system overview or process diagram)
4. The starting point is a finished batch in the batch archive

Operating sequence for 1: Open the message archive with preset filter definition from the process diagram

1. Selection of the image with additional signal: manual, simulation, fault, forced release, ... or logic view
2. Branch to the message archive via shortcut menu "Operator inputs and messages..." (with preset filter definition)
3. Change display period (default setting=current day) as desired via File/Open
4. Selection of message or change in the list
5. Use the shortcut menu (right-click) to set the replay time to the message time and optionally change to the process diagram

Operating sequence for 2: Derive the replay time from the selected message in the Archive viewer

1. Set object filter
2. Decide on the display period with File/Open
3. Select the message or change to the object in the list
4. Use the shortcut menu to set the replay time to the message time and optionally change to the process diagram

Operating sequence for 3: Derive the replay time from the selected message in the message window

1. Select a message in Message Control (system overview or process diagram)
2. Use the shortcut menu to set the replay time to the message time and optionally change to the process diagram

Operating sequence for 4: Derive the replay time from the selected batch in the batch archive

1. Selection of batch in Batch list application / Batch archive tab
2. Use the shortcut menu (right-click) to set the replay time to the start time of the selected batch

Result:

If Replay mode is then activated on this IOS, the process diagram shows the precise system status at the time of the selected message or change. Following this, the play function and the replay time can be varied very simply in the operator input box.

Recipe System 7

7.1 Basic principles

7.1.1 Preliminary note

The description of the recipe system is structured into three parts:

1. The first part describes the recipe system itself. It explains keywords and functions of the recipe and the recipe control. Furthermore the handling of plants with parallel lines is described.
2. The second part describes the recipe editor of the system. Master recipes, process parameter lists and process input lists as well as recipe procedures can be created with the recipe editor. It enables a central processing and configuration of equipment recipe elements like equipment operations, functions, DFMs and units. Materials and material sequences can also be configured with the recipe editor.
3. The third part strengthens the knowledge of the recipe system by discussing several use cases.

7.1.2 Brief Description

The recipe system creates control recipes for batch jobs in cooperation with the order system. This is done at runtime. The recipe system transfers the generated unit control recipes into the PLCs and controls their execution. The application ‘Sequence Control’ which is part of the recipe system too, allows manual interaction with a running control recipe.

Furthermore, the processing of a batch can be monitored at ROP resolution.

The recipe system divides recipes into recipe procedures (how something is done), into process parameters (with which parameter it is done) and into process input (with which material something is done).

This division has the following advantages:

- Lists of materials for process input may be imported from a superpositioned PPS.
- The amount of recipe procedures can be minimized. The creation of recipe procedures is the most extensive part of the plant configuration. Fewer recipe procedures means less commissioning expenditure and also less expenditure for the maintenance.
- The operator can change process parameters without access to the procedure.

7.1.3 Elements of the recipe system

Recipe system terminology

Terminology	Abbreviations
Recipe category	RCAT
Master recipe	GR/MK
Recipe procedure	RP
Recipe unit procedure	TRP/RUP
Unit	TA/UNIT
Recipe operation	ROP
Technical (Equipment) Operation	EOP
Technical parameter	TPAR/SW
Digital function module	DFM
Process parameter list	PPL
Process input / list *)	PIL

The master recipe consists of

1. Master recipe header,
2. Process parameter list,
3. Process input list *)
4. Recipe procedure

A master recipe is assigned to a recipe category (RCAT) and it is assigned one recipe procedure. The same recipe procedure can be assigned to several master recipes within an RCAT.

Note

***) The 'process input list" functionality and component dosing is deactivated in the factory settings.**

As of BRAUMAT version V7, the functionality of the process input list in conjunction with the component and storage location management is deactivated as default. The options available in earlier system versions "Component management" with the sub view "Components" in the master recipes and the corresponding recipe editor dialogs "Engineering/components" previously "Material stock" and "Engineering/Depot locations" (refer to the Appendix section Weighing and weighers (S7-400) (Page 1232)) can only be used, when necessary after setting the following INI switch:

File `'<proj-path>\recipe\project\plant.ini'`

```
[Componentlist]

Enable=1
```

A recipe procedure consists of

1. Recipe procedure header,
2. Recipe unit procedure and
3. Global properties.

For a definition of the configuration limits, refer to the section Quantity schedule of functions and classes (Page 65)

7.1.3.1 Recipe categories

The master recipes of a plant area are assigned to recipe categories. This enables a division into different product groups or a division into production and cleaning according to the plant structure.

- A maximum of 255 recipe categories can be defined.
- The recipe categories are unique within an area.
- The naming and creation of recipe categories is described in chapter "The recipe overview (Page 588)"

7.1.3.2 Relation between recipe categories and process parameters: Fixed assignment

The process parameter list always has the same ID number and the same name as the master recipe.

One process parameter list can be used for one master recipe exclusively.

The free assignment of the process parameter list to one or more master recipes introduced in previous versions is no longer possible. When migrating an existing system with free assigned process parameter lists the data files for both process parameter definition and actual process parameter values have to be copied manually.

7.1.3.3 Master recipe header

The header of a master recipe includes parameters like name and long name of the master recipe, the assigned recipe procedure. The nominal batch size for both the scaling of process parameters and process input materials is stored in the master recipe header as well as it gives information about the date of creation and the last change.

The master recipe header is part of the master recipe view of the recipe editor and is explained there (see Section "The master recipe view. (Page 595)")

7.1.3.4 Process parameter

For each recipe category a table of parameter types can be defined. These parameters are called process parameters and can be inserted in the control recipe as setpoints (actual parameters). For this purpose, it is necessary to define formal parameters (‘placeholder’) in the recipe procedure. These placeholders are substituted by the actual parameters from the process parameter list when creating the control recipe.

Every process parameter can be scaled when it is put into the control recipe. For instance, in batch processing it is often useful to adapt quantities or times to the batch size. Whether a value should be scaled or not is defined in the process parameter list.

The values of the process parameter list are added up. If the sum of the parameter values does not match the nominal batch size in the recipe header, a message box is displayed.

The display of this message can be deactivated by entering [mainrecipe] ChkSum=0 in the file `'<proj-path>\recipe\project\plant.ini'`.

Parameter assignment

The number and definition of the process parameters can be configured for each recipe category (see chapter 2.5 "The master recipe view")

The values of the DFMs in the recipe procedure can be substituted by the actual values of a process parameter. Therefore the definition of a process parameter is based on the definition of a DFM. The definitions of the selected DFM templates are copied into the process parameters' definition table. During configuration the definition of DFMs (name, dimension, decimal places...) is copied as the process parameters' definition. After copying the parameters there is no link between the process parameter and the DFM that was the template for its definition. Later changes to the DFM do not have any influence on the process parameter. Therefore a change of the process parameter definition has no effect on the DFM definition and vice versa.

The DFMs of 16BIT or 32BIT type cannot be used as template for the type of a process parameter.

Note: In this documentation, the terms 'parameter element' and DFM (digital function module) are synonymous and refer to containers for setpoints implemented by PCUs.

System delivery state

The system as delivered includes a configuration example for recipe category 1. This file is set up for 20 process parameters. If more or fewer parameters are required, this list can be lengthened or shortened accordingly.

The process parameter rows may be displayed in different colors: black, red, green and blue. The color information is coded into the comment of each parameter. The process parameter definition dialog offers buttons to select the text color
(see also: Dialog 'Configuration process parameters').

7.1.3.5 Process input list

Basics

The process input list is a list of components in the form of a table that are required for dosing operations in a recipe. One row in the table is used for dosing a component.

The recipe system transforms the recipe procedure into control recipes. When the recipe system encounters a Dosing ROP, a Dosing EOP is inserted for each entry in the component list. The entries for the relevant unit are filtered out of the component list. The order of entries in the component list determines the order of weighing. This order can be further affected by the dose group.

The 'Process input list' functionality can only be used in conjunction with silo management and the 'Weigher' basic function after it has been specifically activated (refer to the note in "Elements of the recipe system (Page 534)"). You will find a description of the configuration and use of this function in the Appendix in the section "Weighing and weighers (S7-400) (Page 1232)".

7.1.3.6 Recipe procedure

A recipe procedure consists of

- Recipe procedure header,
- Recipe unit procedures (sequentially ordered),
- Steps (recipe operations, elements of the recipe operation library),
- Setpoints,
- Synchronizations and alternatives,
- Dosing operations and
- Recipe unit procedures (for each unit).

Recipe procedure header

In the recipe procedure header, the name, batch sizes (nominal, minimum, maximum), production time, start sequence, recipe category, status and creation date are stored.

Recipe unit procedure sequence

For each recipe procedure, it is necessary to define which recipe unit procedures are required. Furthermore, units (physical or equipment objects) from the technological hierarchy must be assigned to these recipe unit procedures (recipe objects).

Units from several PCUs can be assigned. Exactly one unit may be assigned to one recipe unit procedure. This unit contains one sequence object. A sequence is represented by one column in the graphical recipe procedure editor.

The sequential order of the units is insignificant for processing, but defines the graphic presentation in the recipe procedure editor. This sequence is also used in the order system ('batch monitoring' view).

Recipe operations and setpoint values

Each PCU has equipment operations which are permanently assigned to one, several or all units. An equipment operation in BRAUMAT/SISTAR Classic V6.0 is the smallest self-contained operation of a unit.

Each equipment operation has at least one special and up to 13 regular setpoints. Setpoints are always assigned to DFMs (digital function modules). How many and which DFMs the equipment operation uses must be configured on the IOS. Up to 24 setpoints can be defined for each equipment operation.

An equipment operation is one to one related to a recipe operation. Values can be assigned to the setpoints of a recipe operation in the step. Each step has an additional setpoint, which is the step monitoring time.

Values can be assigned to the setpoints in various ways:

- The value is defined in the recipe procedure (direct input);
- The value is defined in the recipe procedure and should be adapted to the true batch size using a scaling function;

Two scaling functions are included in the standard version;

1. A linear scaling without offset
   (displayed in the recipe procedure as >-Lin< setpoint in recipe)
2. A linear scaling with 10% offset
   (displayed in the recipe procedure as > -Off< setpoint);

The scaling functions are implemented as DLLs, and can be extended for individual plants.

- A reference to a process parameter can be defined;
  - when the control recipe is generated, the value of the current master's process parameter is entered as the setpoint;
  - This is a recipe-dependent substitution (displayed in the recipe procedure as > %Sx<parameter name, where x is the parameter number in the process parameter list).
- A reference to an order parameter can be defined;
  - when the control recipe is generated, the value of the order parameter for the current batch is entered as the setpoint;
  - This is a batch-dependent substitution (displayed in the recipe procedure as > $Bx<parameter name, where x is the parameter number in the batch).
- A setpoint value directly set in the recipe procedure view can multiplied by an order parameter; (displayed in the recipe procedure as >*$Bx<setpoint, where x is the parameter number in the batch).

Recipe unit procedures

A recipe procedure consists of one or more recipe unit procedures (short RUP). Each recipe unit procedure is assigned to a unit from the technological hierarchy in the dialog 'RP properties'.

One recipe unit procedure is an ordered sequence of recipe operations. Each recipe operation references an equipment operation.

One recipe unit procedure must have the Start ROP as the first step. This recipe unit procedure is referred to as the start recipe unit procedure or start sequence. It can be started by the order system.

One or more recipe unit procedures must have the End ROP as the last step. The recipe procedure should be configured in such a way that there is only one End ROP. Furthermore 'free ends' (last step in a recipe unit procedure is an ROP and not a synchronization or End ROP) should be avoided.

More RUPs can be started by synchronizations and alternatives.

Synchronization ROPs may also synchronize several concurrent RUPs.

Start recipe operation

This operation is a special recipe operation since it does not reference an equipment operation from the PCUs' 'stock of equipment operations' (i.e. an FC). The Start ROP does not need configuration. Only one Start ROP is allowed per recipe procedure and it is possible only as the first step of a recipe unit procedure.

If this equipment operation 'Start' is processed by the PCU's sequence control, a telegram of type 15 is generated and sent to the IOS (Batch processing/request).

The Start ROP has the number 32700.

A synchronization or alternative must not follow directly below the Start ROP. The Start ROP may be followed by a normal ROP or NOP (no operation).

End recipe operation

This operation is a special recipe operation since it does not reference an equipment operation from the PCU's 'stock of equipment operations' (i.e. an FC). The Stop ROP does not need configuration. It can be used several times per recipe procedure, but only as the last step of a recipe unit procedure.

If this equipment operation is processed, the PCU sends a telegram of type 15 to its IOS.

The End ROP has the number 32701.

A synchronization or alternative must not precede the end ROP directly. The End ROP may be preceded by a normal ROP or NOP (no operation).

If the recipe procedure is terminated with an OR synchronization, each participating recipe unit procedure has to be terminated with an End ROP. Only in that way is the secure processing of the batch guaranteed.

Note

The End ROP must be the last ROP in the recipe procedure to be processed. At this time, no more recipe unit procedures must run within the scope of this batch.

This can easily be achieved by synchronizations. If the End ROP is used several times in the recipe procedure, or recipe unit procedures do not end with a synchronization (free ends), the user must ensure that all RUPs of the recipe procedure are finished before the End ROP is reached.

If the End ROP is processed although one or more RUPs are still running in the scope of the batch, all synchronizations which have already been generated for this batch are set to 'error'. The recipe procedure is no longer processed automatically, or only with restrictions.

Synchronization

Synchronizations - as the name indicates - are used when synchronizing the concurrent processing of several recipe unit procedures.

Note

Synchronizations may occur only within the own area of the given recipe procedure as the recipe server may reference PCUs only from that area via the PCU server miniports.

The recipe unit procedures which are involved in the synchronization wait until the synchronization is complete, and then continue processing.

This operation is a special recipe operation since it does not reference an equipment operation from the PCU's ‘stock of equipment operations’ (i.e. an FC).

Synchronizations are identified within the recipe procedure by their synchronization number.

Recipe unit procedures can take part in synchronization in four ways.

1. This RUP is waited for (logical AND operation);
2. This or another RUP included in the synchronization is waited for (logical OR operation); this type of synchronization must be used for RUPs which are started using an alternative;

3. The RUP is started ('non-blocking'). This kind of synchronization can only be added as the first step of a unit. The 'non blocking sync' (short NBS) is used for non-blocking the units which are waiting and ready for starting. The starting is buffered. It is started as soon as it is free.

Note

To enable this function, you need to set the INI key "EnableNonBlockingSync=1" in "`<proj-path>\recipe\project\plant.ini`" of the section [Version4.60]

---

4. The RUP is started (with blocking). This kind of synchronization can only be added as the first step of a unit. The units which are taking part in the synchronization are waiting until this unit is free and the continuing operation is performed.

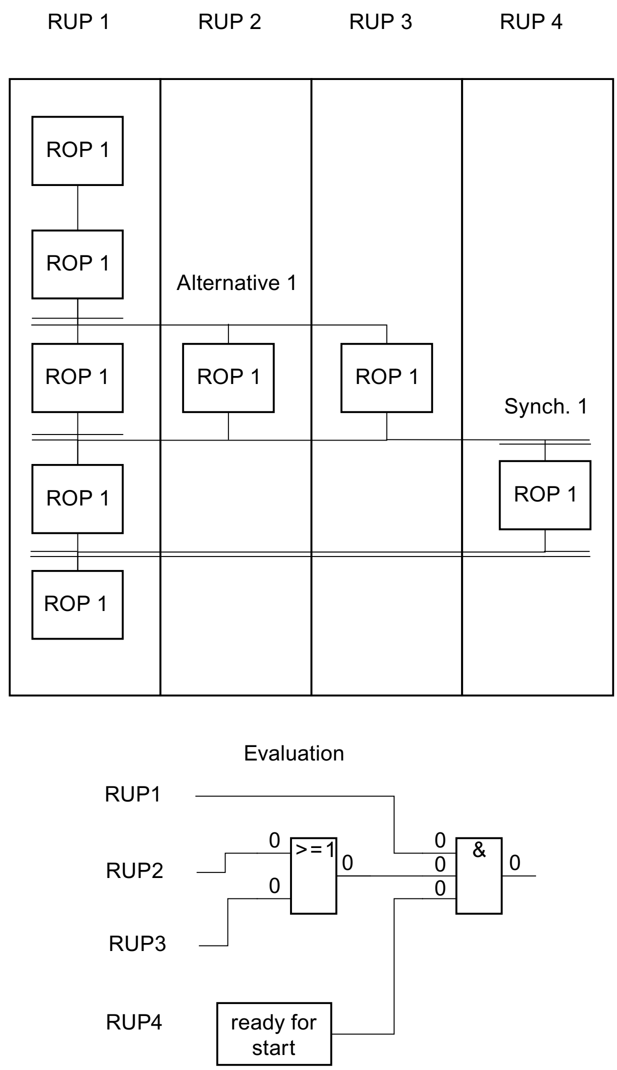


Figure 7-1 Example of a synchronization with 'OR' and 'AND' linkage

Processing the synchronization

During processing of the recipe procedure, every change to an RUP is reported to the IOS. The production can be monitored with the 'control recipe' tool from the 'process monitoring' tab in the basic menu.

Completing a synchronization

A synchronization has completed if:

1. All units which are waited for with ‘AND’ have reached the synchronization;
   each of these units must be in automatic mode, and the permanent condition must be met;
2. At least one of the units which are waited for with ‘OR’ has reached the synchronization; this unit must be in automatic mode, and the permanent condition must be met;
3. And if all units which should be started are free and not in manual operation, and the permanent condition of these units is set.

Before the synchronization progresses, all units which are to be started are started. These meet a synchronization in the first step, processing stops and they report the unit status to the IOS (acknowledgment on unit start).

Alternative

Alternatives (as the name indicates) are used for the alternative start of recipe unit procedures.

The RUP which ‘produces’ the alternative decision can continue. RUPs which are started by an alternative can be brought together again by a synchronization of ‘OR’ type.

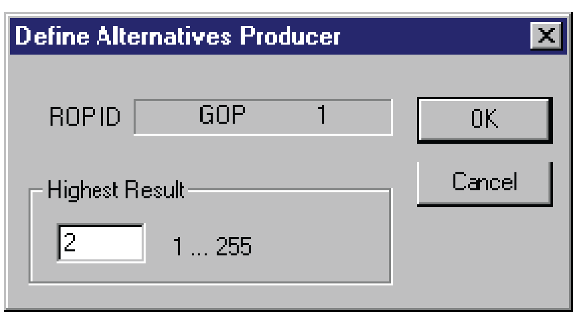


Figure 7-2 Dialog for the configuration of an alternative-producer-operation.

Before every alternative, there must be an equipment operation, which produces an alternative result. This equipment operation has to be configured as an alternative producer. The parameters of the equipment operation specify which alternative results are generated by this ROP. The configuration as alternative producer may be established in the equipment data dialog or in the EOP definitions dialog.

The alternative result value has to be calculated in the associated equipment operation in the PCU, which is called 'alternative result producer' EPE therefore.

For additional information on this topic, please refer to chapter Create engineering operations (EOP) as function (Page 652).

During recipe creation, the results are uniquely assigned to the RUPs which are to be started by the alternative (so-called alternative destinations). If the equipment operation generates a result that is not assigned to a destination RUP, the recipe control stops until a valid result is supplied.

This operation is a special recipe operation since it does not reference an equipment operation from the PCU's ‘stock of equipment operations’ (i.e. an FC). It has the number 32703.

The alternative destination has the number 32704.

Processing the alternative selection

An alternative selection is processed in the following way:

- If the destination RUP assigned to the generated result is not busy, not in manual mode and the permanent condition is met, the destination RUP is started.
- The sequence control finds the alternative destination step on the first position, stops processing and notifies the sequence state to the IOS (acknowledgement of unit start).

Note

At each of the recipe unit procedure inputs in the list recipe view, a setpoint 1 of type "text" is defined. The setpoint is predefined by the system ("virtual“ DFM 3.0) and cannot be configured via the "Edit SP Def." dialog. The indicated textstring is extracted from the configuration file "_altres.txt" in the PCU-specific folder '..\Texte’ or '..\Texte.x’. As an index the alternative result 1..255 is used (first line of textfile is blank). These texts can be adjusted individually by copying and editing of the supplied standard file "`<proj-path>\PCU.xxx \texte\_altres.txt`".

Transition

Basics

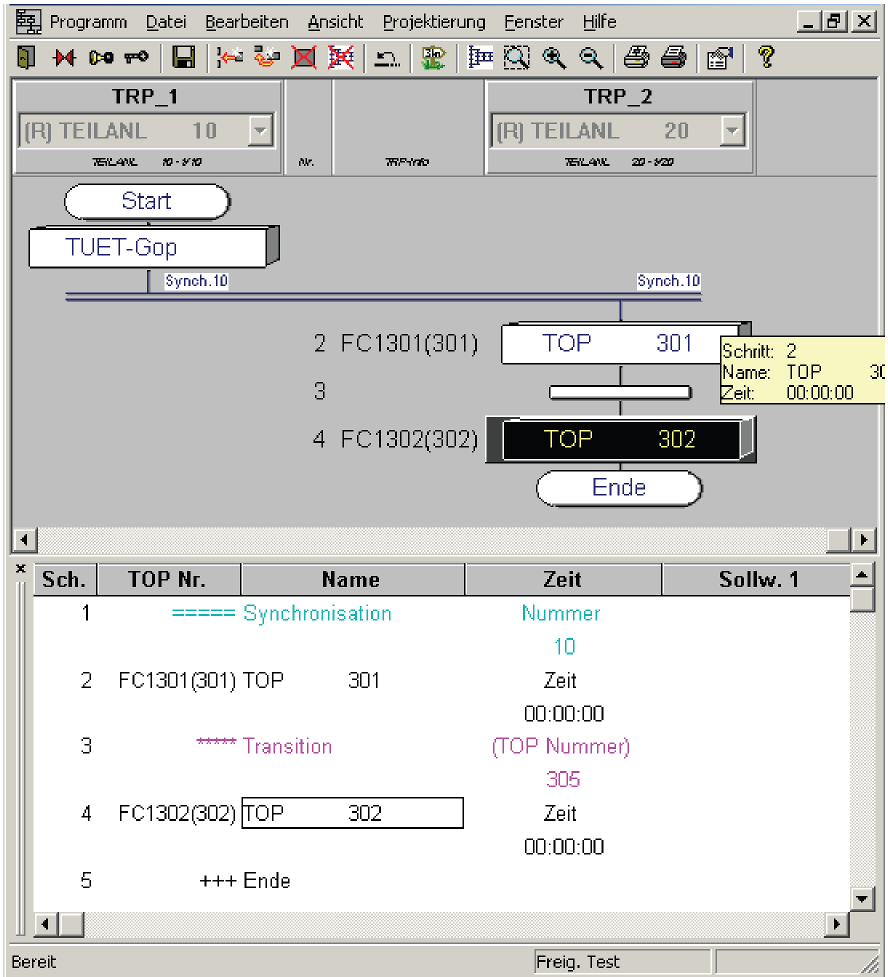
A transition is a special equipment operation.

It has exactly one setpoint which is an equipment operation number (indirect call of a basic operation).

The transition is fulfilled if the unit operation which is specified by the setpoint supplies RLO = 1.

The basic operation before the transition is processed until the transition is met which means that two ROPs are processed simultaneously.

Example:



Sequence:

If EOP 301 is exited with RLO=1, the EOP 305 is processed via the transition. As long as EOP 301 returns RLO=1, EOP 305 is processed, i.e. EOP 301 und EOP 305 are processed in parallel.

When EOP 305 returns RLO=1, EOP 305 and EOP 301 are ended and the processing of EOP 302 starts.

Note

The EOP referenced by the transition (EOP 305 in the example) is not run through with EopStop at a step change. Thus all assignments remain active and opened lids, for example, are not closed any more. It can only be used with EOPs that ultimately only carry out a step enabling condition.

Jump destination / Jump

Basic principle

This operation allows jumps within the unit recipe which must be based on a unit. The jump target must be configured in two places: in the PCU and in the IOS.

PCU

The jump result must be created by an equipment operation and stored in the byAlterResult –variable (PCU < V7: DB725.DBB177, PCU V7: DB725.DBB193)

IOS

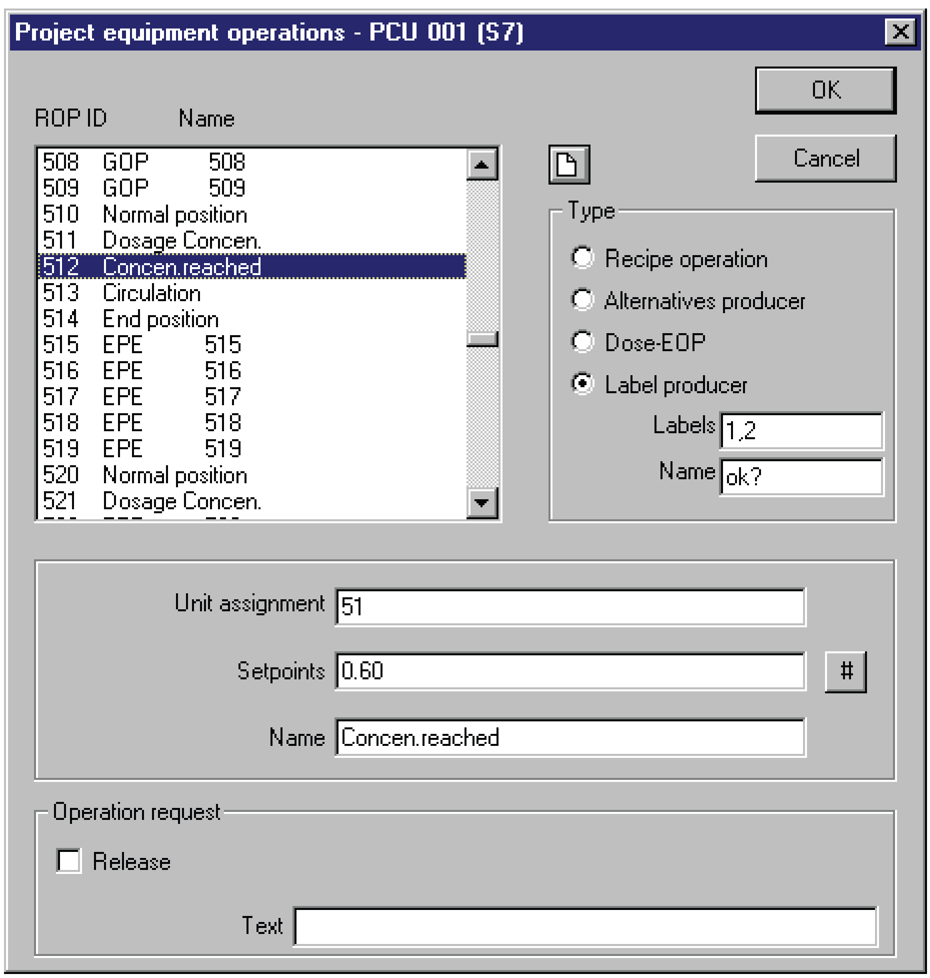


Figure 7-3 Definition of an equipment operation as jump destination producer by selecting the radio button ‘Label-Producer’ and the input fields for the labels (comma separated) and the name of the jump

The configuration is performed in the EOP definition dialog. In this dialog window the corresponding EOP will be defined as jump destination producer, the jump destinations are entered and the name for the jump is allocated. The jump name may be up to 8 characters long. The jump destinations have values between 1 and 255. In the editbox ‘Labels’ several destinations are filled in separated by commas.

In the recipe procedure jumps can only be added after normal ROPs which are configured as jump destination producers (EOP Def. Dialog or equipment data dialog). The actual jump operation is configured by entering a name and an ID code number.

The jump destinations are added in any place in the recipe unit procedure. If a matching jump number is found for an inserted jump destination operation, the jump's name will be shown next to the destination operation symbol.

In the file, "`<proj-path>\recipe\project\plant.ini`" the entry 'JumpMode' can be adapted, if jumps are branching execution across synchronizations. The user has to make sure that all RUPs taking part in the synchronization are processed in the way expected.

JumpMode may be set to the following values:

1. Jump across a synchronization produces an error
2. Jump across a synchronization produces a warning
3. Jumps across synchronizations are ignored, no warning or error is produced

If no entry is made in file 'plant.ini‘ or if the entry is invalid, the default value ‘1’ (Jump across a synchronization produces an error) will be set.

Uniqueness of jump destination IDs

The user must ensure all jump destination IDs are unique.

If a jump is executed the system branches execution to the first jump destination with the corresponding ID that is found.

Jump execution

At runtime, the value 'byAlterResult' (PCU < V7: DB725.DBB177, PCU V7: DB725.DBB193) of the unit data set is evaluated.

If the value is 0 no jump is conducted.

If the value is not equal to zero, the recipe data block is searched for the jump destination by ID. If a jump destination is found, execution will continue at this step.

If no jump destination with a corresponding ID is found, execution will – as with a result of 0 – not be branched. Processing continues as if no jump was executed.

Dosing Operation

This recipe operation is used for dosing materials from the process input list.

You will find a description of this function in the special documentation "Dosing and Weighers" (see Appendix).

The process input list data is taken over to the control recipe when this ROP is processed by the recipe control.

At runtime, a control recipe is generated using the order and batch numbers. This 'dosing' equipment operation is inserted as often as there are entries in the component list.

If the ‘intermediate drain’ quantity is exceeded, an emptying operation is inserted.

The last emptying operation must be inserted explicitly.

See also

Weighing and weighers (S7-400) (Page 1232)

Releasing a unit

Basic principle

An option is available which lets you temporarily release a unit during a recipe procedure run and then re-use it.

This function is useful if, at runtime certain batches require a unit more than once, but cannot permanently occupy that unit because it is required by other batches.

Example:

Two batches require a unit 3 several times at different times during their execution (e.g. a heater). First, batch 1 (started via unit 1) uses unit 3. When batch 1 no longer requires unit 3 (e.g. temperature reached), unit 3 is released.
This means that although unit 3 will be required again later for the batch 1 recipe, in the meantime it may be used by batch 2.

Advantage:

Unit 2 does not have to wait until the recipe being processed by unit 1 and 3 is finished. Unit 1 and 2 can run at the same time, whereby it must be ensured that they do not attempt to access unit 3 simultaneously. It is therefore necessary to coordinate use of unit 3 by the use of a ‘Release unit’ ROP.

On delivery, this function is disabled. It can be enabled in the file "`<proj-path\recipe\project\plant.ini`":

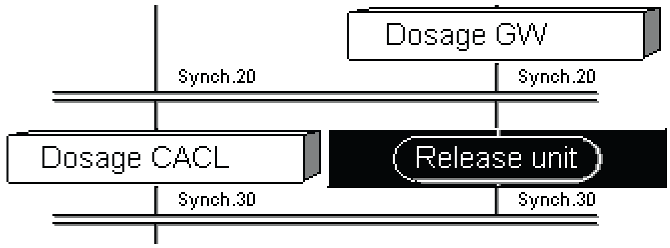
```
[ReleasePlantSection]

Enable=1
```

If Enable is set to 1, the function to release a unit during the recipe procedure run is enabled.

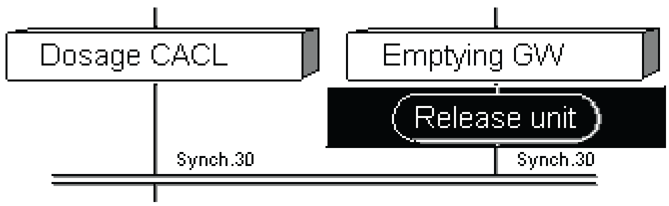
In order to enable the ‘Release unit’ functionality, it is necessary to add a step in the recipe procedure editor. In the dialog for defining the step, the ‘Release Unit’ button must be selected.

It is only possible to release a unit in two instances:

1. There is a synchronization before and after the release



2. There is a recipe operation before the release and a synchronization after the release.



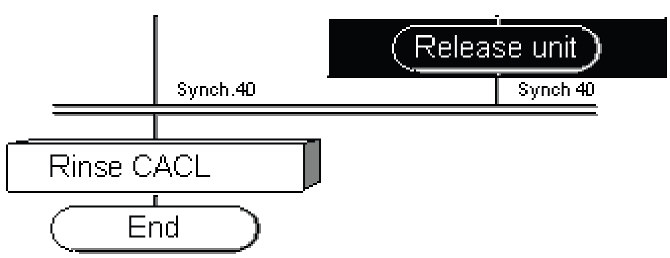
There must be a synchronization after each release, which restarts the plant section.

The synchronizations must be of ‘AND’ type.

‘OR’ synchronizations are not permitted before or after a release.

Safe Open End

If there are no further steps following a synchronization after a release, the release is interpreted as ‘safe open end’.



Example:

RUP 2 is released immediately after completion of step ‘Emptied’. If RUP 1 reaches synchronization 40, a check is carried out to see if the batch is still running in RUP 2. If this is the case, RUP 1 stops at Sync. 40 and waits for the step ‘Emptied’ to finish.
Thus the advantages of an open end can be utilized in the recipe procedure without making any sacrifices to safety. As soon as the step ‘Emptied’ is finished, RUP 2 can be used by other recipe procedures (open end), whereby RUP 1 only progresses to the end step when a check has ensured that the batch in unit 1 is no longer running.

Deadlocks

With the possibility to release one unit in the recipe procedure temporarily there is also the risk of deadlocks during processing.

A deadlock can't be detected in the recipe editor.

This type of 'batch-interlocks' in the plant depends on the sequence of the batches (and thus the sequence of the recipe procedures).

It is the task of the plant operator to prevent this type of jamming via the batch sequence input.

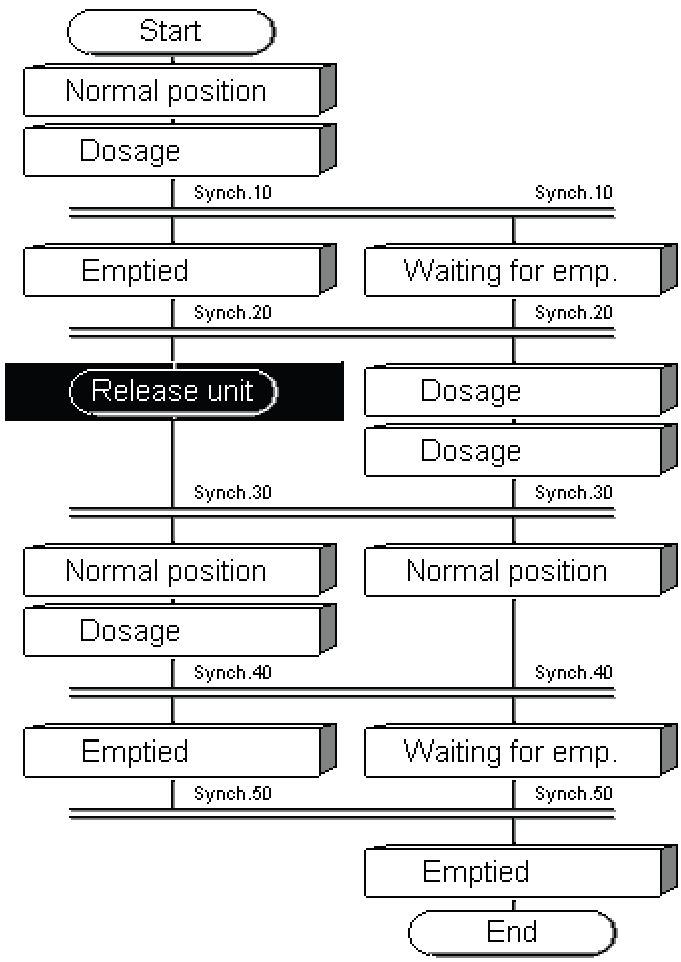


Figure 7-4 Example of a deadlock due to unit releases: Recipe procedure 1

Example of a deadlock (see recipe procedures 1 and 2):

- Batch C1 is using recipe procedure 1 (weigher 1 is released after synchronization 20). Batch C1 is running in weigher 2.
- If the next batch C2 is now started with this recipe procedure 1, it reserves weigher 1. Batch C2 runs until synchronization 10 is reached and there it waits for weigher 2 to continue.
- Batch C1 runs until synchronization 30 is reached and there it waits for weigher 1 to continue.

These two batches are now blocking each other.

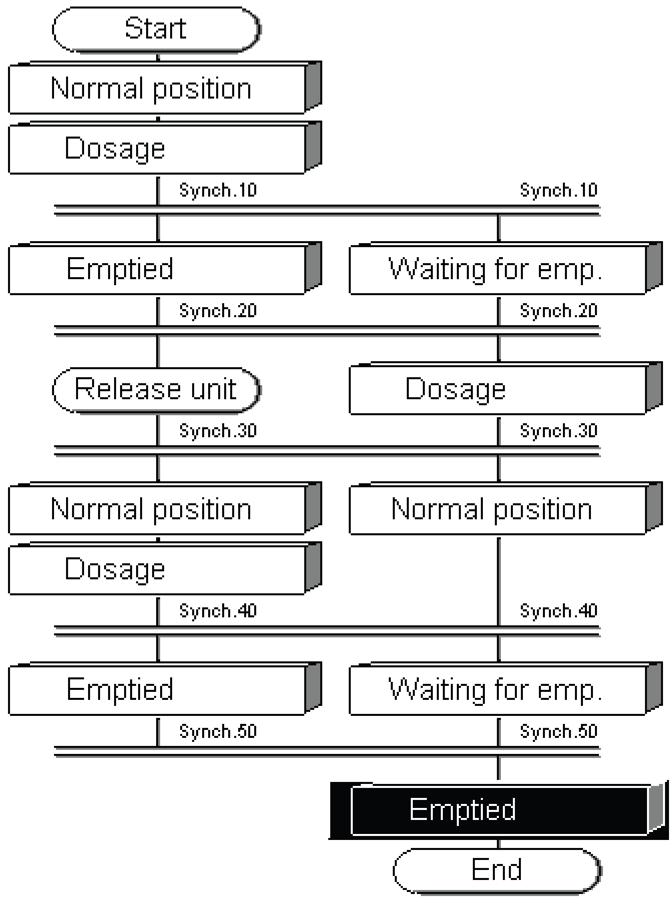


Figure 7-5 Example of a deadlock due to unit releases: Recipe procedure 2

Solution: If a batch is started after the batch C1 which uses a slightly different recipe procedure 2, there will be no deadlock.

Batch C2 reserves weigher 1. Batch C2 runs until synchronization 10 is reached and waits there for weigher 3 in order to continue. If weigher 3 is free, weigher 1 will be emptied and released by its Release-Unit-ROP after synchronization 20. Weigher W2 is now available to be used by batch C1 again (synchronization 30).

Therefore it is important to ensure that recipe procedures 1 and 2 are started by turns. Starting the same recipe procedure twice consecutively will result in a deadlock.

7.1.3.7 Lines

There are similar lines in many plants e.g. brewing lines, etc. which are able to process the same recipe procedures. For a definition of the term 'line' see also: System description Classification of plants). The recipe procedures of the lines differ in the use of different units with their own equipment operations and DFMs. Speaking of parallel lines one can assume that the units of the same lines show the same installation. Therefore it is possible to create the recipe procedures of the parallel lines from the recipe procedure of the projected lines (reference line).

For each recipe procedure, it is possible to define the lines on which the recipe procedure can be processed. By selecting the recipe procedure type in the RP properties dialog (see also: Recipe procedure properties dialog), the user has the choice to decide if a recipe procedure can only be processed on one reference line (a so-called simple procedure) or if it will be possible to deduce the parallel lines recipe procedures from this reference procedure (lines procedure).

If there are no lines with structural similarities in a plant this functionality is not needed and line 1 is always used as the reference line.

Lines are unique over the whole plant. The configuration is made with the 'line configuration' and 'line assignment' dialogs. The dialogs are activated from the ‘RP properties’ dialog and can be displayed only if the procedure is of type 'line recipe'.

Reference Line

The reference line can be a real line of the plant. A recipe procedure must be created for the reference line. The recipe procedures of the parallel lines are automatically generated using assignment tables.

If the lines are not exactly equal, it may be necessary to define a virtual reference line. This line represents a superset of units, recipe unit procedures and recipe operations.

Using the assignment tables, the recipe procedure of this virtual line is converted to recipe procedures of real lines, and thus adapted to a different function. For the various parallel lines, the recipe unit procedures or equipment operations which are not required are removed from the virtual recipe procedure using the ‘ZERO conversion’.

Parallel Lines

These are plant lines which resemble a reference line to the extent that recipe procedures can be generated automatically from the recipe procedure of the reference line.

Recipe procedures are always converted when the recipe procedure of the reference line is saved to hard disk.

It is not possible to edit recipe procedures of the parallel lines.

Line assignment

A reference line can be assigned for each recipe procedure. If the plant consists of only one line, no definition is necessary and line 1 is always used as the reference line.

For the reference line it can be defined if it is a virtual line or not.

With the line assignment dialog it is defined, on which lines the RP can be processed. With it, a line may be configured as a reference line. All other lines must be configured as parallel lines to the reference line.

Assignment tables

Assignment tables are required for the automatic conversion of the parallel lines’ recipe procedures. They must be configured. These tables are in dBase format.

The assignment tables are defined in the following dBase files inside folder `'<proj-path>\recipe\project'`:

dBase table	Description	Max line count
• 'sectass.dbf'	for plant section assignment table	65535
• 'bopass.dbf'	for basic operation assignment table.	65535

The tables have to be created using MS Access, MS Excel or another editor (which supports the format).

Note

When creating the tables, care must be taken to ensure that the tables are exactly in the format described in the definition file. You may find it extremely difficult to meet this requirement in MS Excel. Hence, we strongly suggest that you use MS Access and integrate the *.dbf file. The Access layout view gives you more control over the table/column format.

An example of the unit assignment table and recipe operation assignment table is supplied with the system.

After changing the tables the system must be restarted in order to implement these changes.

Conversions for unit assignments

Basic principle

Three conversion types are distinguished:

1. 'ZERO' conversion:
   With this conversion, a unit is not converted into the parallel line at all. This function is useful if the reference line contains units which do not exist in the parallel line.
2. '1:1' conversion:
   With this conversion, the same unit with the same basic operations can be used in several lines. This is useful in plants in which not all sections exist more than once.
3. 'Normal' conversion:
   With this conversion, one unit is converted to another unit. This conversion is used if the unit of the reference line corresponds to another unit of a parallel line.

Table 7-1 Structure of the 'SectAss.dbf' file

Column	Meaning
No_ref_lin	number of reference line
Pcu_ref_l	PCU number of reference line
Sect_ref_l	Unit No. of Reference line
No_par_lin	Number of parallel line
Pcu_par_l	PCU No. of Parallel line
Sect_par_l	Unit No. of Parallel line

Example

Table 7-2 Section from 'SectAss.dbf' file. Line '3 'is the reference line, the lines '1' and '2' were configured as parallel lines.

No_ref_lin	Pcu_ref_l	Sect_ref_l	No_par_lin	Pcu_par_l	Sect_par_l
3	2	12	1	2	22
3	2	13	1	2	23
3	2	12	2	4	12
3	2	13	2	4	13

Here the Line ‘3’ is the Reference line. The unit 12 of PCU 2 is converted to unit 22 of PCU 2 on conversion for Parallel line ‘1’ and for Parallel line ‘2’ to unit 12 of PCU 4. In this case only so-called ‘normal’ conversions take place. One unit of the reference line is displaced to another unit of the Parallel line.

The implementation provision of this example is highlighted further using the following representation. On saving the recipe procedure, one recipe procedure each for both parallel lines are generated based on the SectAss table

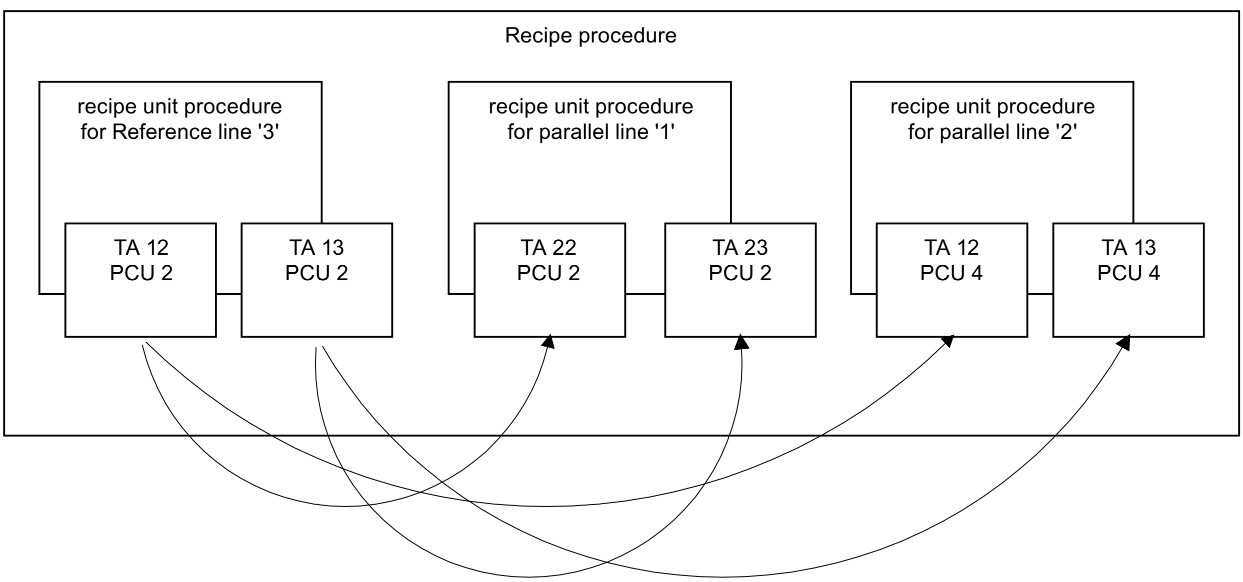


Figure 7-6 Conversion of units from reference lines to parallel lines, based on the table in 'SectAss.dbf'.

Conversions for Operation assignments

The operation allocation table is saved to the file 'bopass.dbf'. This table specifies how the Technical Operations of the reference line on the parallel lines are to be represented. There has to be one entry per parallel line for each Technical Operation of the reference line.

PCU numbers are not stated in the table for the parallel lines because they have already been specified in the unit allocation.

With the conversion of the operations the following four conversion types may be distinguished:

- ‘ZERO’ conversion:
  With this conversion, a basic operation is not converted into the parallel line at all. This function is useful if the reference line contains EOPs which do not exist in the parallel line.
  Limit: the basic operation of the reference line must not be between two synchronizations.
- ‘1:1’ conversion:
  With this conversion, the same basic operation can be used in several lines. The same setpoints and setpoint numbers are used.
  In association with another setpoint definition (line-dependent setpoint definition Line-dependent setpoint definition (Page 554)) for the basic operation, the same basic operation can be used in different plant sections. However, for this it is necessary to arrange the setpoints before and after the unit processing.
  When using the same technical operation row dependent BLR definitions for the step enabling condition can be configured using the BLR object (see Line-dependent step enabling condition via BLR object (Page 554))
- ‘Normal’ conversion:
  With this conversion, one EOP is converted to another. This conversion is used if the EOP of the reference line corresponds to another EOP of a parallel line.

Column	Meaning
No_ref_lin	number of reference line
Pcu_ref_l	PCU number of reference line
Bop_ref_l	EOP number of reference line
No_par_lin	Number of parallel line
Bop_par_l	EOP number of parallel line

Line-dependent setpoint definition

As well as the unit assignment table and operation assignment table, one DFM reference definition per parallel line can be defined for each EOP.

This definition of assigned DFMs only becomes effective when the EOP is converted for a parallel line.

The definition must be entered in the `'<proj-path>\pcu.xxx\recipe\epe.ini'` file using a text editor.

Example	Description
[EPE012]	Equipment Operation 12
SetPoint=1.1,1.2	ICMs for Recipe of the Reference line
SetPoint_lin2=2.1,2.2	ICMs for use in a unit of Line 2
SetPoint_lin3=2.11,2.12	ICMs for use in a unit of Line 3

Line-dependent step enabling condition via BLR object

As well as the unit assignment table and operation assignment table, a BLR reference for the step enabling condition per parallel line can also be defined for every equipment operation (EOP) (only V7 PCU!).

This link to a BLR is only effective with the parallel line conversion.

The definition must be entered in the `'\pcu.xxx\recipe\epe.ini'` " file using a text editor.

Example	Description
`[EPE012]`	Equipment operation 12
`NscBlrObj=CID_BLR1,1`	BLR for the recipe of the reference line (specified in the dialog Equipment Operation)
`NscBlrObj_Lin2=CID_BLR1,21`	BLR for use in a unit of line 2
`NscBlrObj_Lin3=CID_BLR1,31`	BLR for use in a unit of line 3

Possible values:

Line number 1-999

BLR class: CID_BLR1-CID_BLR9

BLR instance: 1-256

7.1.3.8 Cross-Streaming

General

The 'cross-streaming' feature allows the definition of possible streams that the produced batch may take through a plant (standard streams, auxiliary streams, etc.).

When defining these streams, the physical connections of the plant’s units have to be considered, for not every combination of units may be possible, e.g. connected via pipes. In the brewhouses of breweries there are often several equivalent vessels for one process operation (several mash tuns, lauter tuns and wort coppers etc.). The functional equivalence does not imply the obligatory equivalence of recipes, e.g. in a brewhouse there may be more than one lauter tun that are all used for sweet wort separation, but that differ in their technical equipment.

The different vessels are often used in preferred combinations that are called streams. In certain cases, e.g. failures of vessels or for special functions that are only needed in the course of a batch production, the combination used can be changed. This change of the combination can be made while the batch is running.

The difference between a stream and a line is the missing physical connection between the units of parallel lines (using ISA-88.1 terminology: ‘multiple path structure‘). The line as an ordered sequence of units cannot be defined explicitly. It is defined by the recipe procedure’s unit sort and its line assignment tables. When using stream recipe procedures, the batches‘ paths may cross (using ISA-88.1 terminology: ‘network structure‘). The paths of a stream recipe procedure are defined in a special editor dialog and may be changed-over in the batch-list-editor at a batch’s runtime. Unlike in the case of line recipe procedures of parallel lines, which can be generated by the recipe editor using the conversion tables, the individual recipe procedures of all units involved in the streaming process must be configured explicitly and after changing-over streams.

Stream selection

General

The system supports the configuring of streams. A stream is a sequential combination of a plant's units (path of a batch). The sequence of the units in a stream is not necessarily identical to the physical sequence of units occupied by the batch. Processing is defined by the recipe procedure. Paths are generated for a recipe procedure and are unique in this RP.

A stream is defined by the following data:

- Name of the stream
- Number (as assigned by the system)
- Bitmap file
- Assignment of the unit candidates to the recipe unit procedures
- Primary or secondary stream

The bitmap file shall show a graphical representation of the stream and is displayed in the batch-list-editor for improved user orientation.

Streams that do not contain a unit with a starting operation are called ‘secondary streams‘. They may only be selected at runtime after a regular stream has been used to start the batch.

When changing-over a stream at a recipe’s runtime, only streams that contain the units allocated at present may be chosen.

Another definition of interest at this point is the so-called stream fragment. If a stream is a 'fragment', one of the recipe unit procedures has no unit assigned to it. The empty recipe unit procedure does not have to be a start unit.

Example:

RUP 3	Lautering apparatus
Candidate 1:	Lauter tun 1
Candidate 2:	Mash filter 1
RUP	Auxiliary lautering procedure
Candidate 1 (used with lauter tun 1):	Raking control
Candidate 2 (used with mash filter 1):	Empty
Therefore streams containing the mash filter unit are fragment streams.	

Configuration

Streams are defined in the recipe editor. A recipe procedure may be defined as a stream procedure.

Note

The "stream procedure" option can be activated in the specified dialog only if this feature has been enabled via the following button in the configuration file "Plant.ini":

```
[Version4.60]

;Enable stream function for recipe procedures

EnableStreams=1
```

If a procedure is configured this way the button to open the stream editor dialog (‘Allocation’) becomes active.

After selecting the recipe procedure type ‘streams‘, an RUP can have more than one unit assigned to it as candidates. The first assigned candidate becomes the so-called ‘reference candidate’ and is labeled with the letter ‘R‘. The following candidates, whose recipe procedures have to be defined manually and are not derived automatically are labeled with an ‘M’.

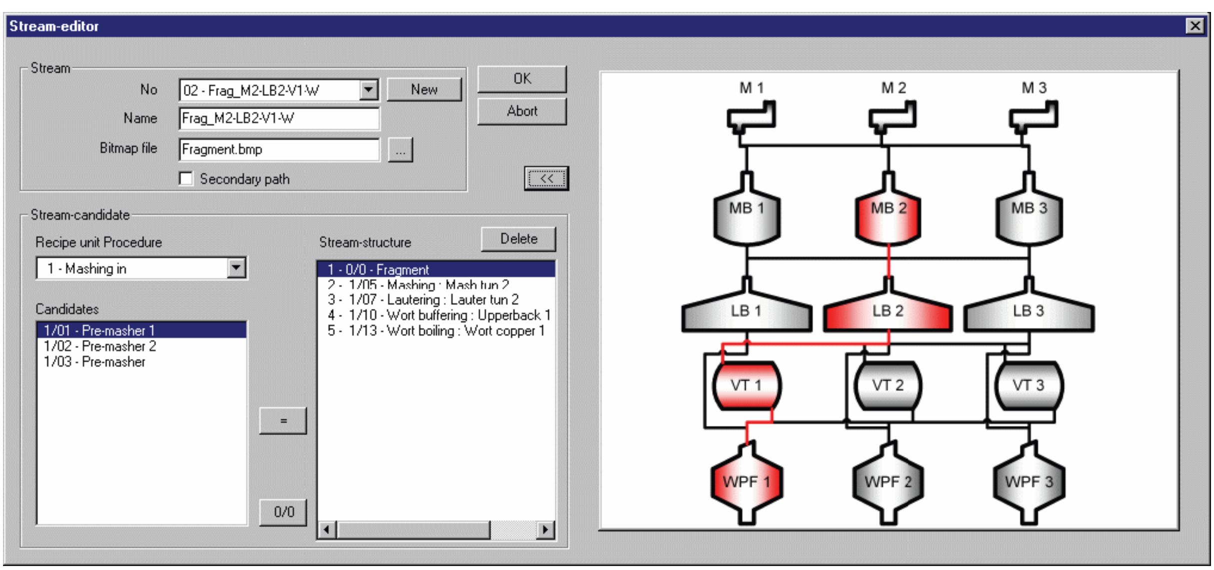


Figure 7-7 The stream-editor-dialog with a fragment stream selected

Combining the candidates into a stream

The candidates that were assigned to a recipe unit procedure have to be combined into a stream in the stream-editor-dialog. In this dialog additional parameters of a stream (name, bitmap file, primary or secondary) are set as well.

Communication between candidates

Stream recipe procedures are transformed into control recipes when they are started. They do not differ from control recipes that were created from simple or line recipe procedures. The streaming functionality is completely controlled on the IOS. To enable communication between the dynamically combined recipe unit procedure, a new kind of parameter has been introduced: the so-called recipe-procedure-parameter or short procedure parameter.

RP parameters are used similarly to process or batch parameters. They are substituted for setpoints in the recipe procedure. RP parameters may be of the data type ‘fixed point number‘ (SW) or ‘text list index‘ (TEXT). They have a minimum and a maximum value.

RP parameters are either read or written by a recipe operation. At runtime both the DFM substituted with the RP parameter for writing and the DFM substituted with the parameter for reading contain the RP parameter value as setpoint value. The RP parameter value is entered as a constant in the recipe procedure editor. When creating the control recipe, the recipe control searches all recipe unit procedures of the stream for writing access on RP parameters and stores the first value found. In a second search all reading RP parameters are substituted by the values found in the first pass. After this the control recipe is transferred into the PLC.

By this it is possible e.g. to determine the following unit by reading an RP parameter simply by writing the following candidate’s ID into the parameter in every candidate RUP.

7.1.4 Status of recipe objects

The two recipe objects master recipe and recipe procedure have a status parameter. This status describes the state of the objects‘ generation and indicates their read/write status as well.

The current system version distinguishes between the following states:

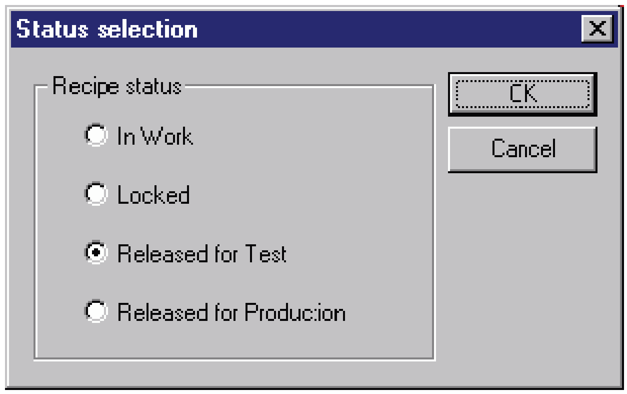
- In Work
- Enabled for Test
- Enabled for Production
- Locked

The status ‘In Work‘ indicates incomplete recipe objects that are not suitable for batch production.

Objects that are *‘Enabled for Test‘* may be used for the production of batches. Master recipes and recipe procedures are read- and writable in this state.

If *‘Enabled for Production‘*, the objects are read-only and may no longer be changed. They may of course be used for the production of batches.

The status *‘Locked‘* identifies an object that potentially could be used to produce with, but that was locked by the operator for some reason.



The change-over from one status into another is done with the ‘Status selection‘ dialog. The user has to check the conditions necessary for the change. The status changeover is password-protected and logged by the system.

Note

A status change from 'Released for Production' to other states (e.g. 'In Work' for editing the recipe procedure) should be done only after all batches with that recipe procedure are completed. Otherwise batches still running could 'hang' at synchronizations or alternatives if changes are made to those objects in the recipe procedure.

The master recipe’s status is shown in the master recipe view of the recipe editor and can be changed in this view as well.

The status of the master recipe is shown in the recipe editor's statebar, too. If in status ‘Enabled for Production‘ the text ‘read-only‘ is added.

If the master recipe has a recipe procedure assigned to it the status of this recipe procedure is displayed in the master recipe view as well (see figure).

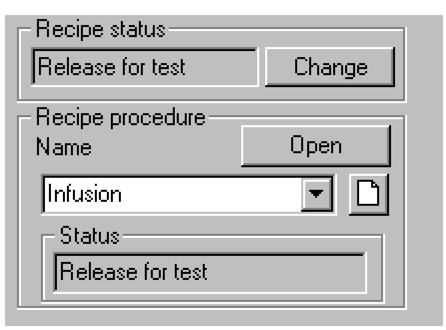


Figure 7-8 Status display of a master recipe and the assigned recipe procedure in the master recipe view.

The status of a recipe procedure can be changed via the menu entry ‘Edit > Change RP status‘.

In the recipe procedure view the status is shown in the statusbar.

7.1.5 Determining the Start Sequence

When a new order is entered, the recipe with which and the line on which it is to be processed are specified.

The order control system requires a start sequence for the CAS module.

The order system determines the recipe procedure's number from the master recipe header and the start sequence from the recipe procedure's header.

If the line has the ‘reference line’ code, the determined unit is the start unit. If the line has the ‘parallel line’ code, the start unit must be determined from the unit assignment table.

7.1.6 Relationship between process input list and recipe

The materials of a recipe are defined in this list.

A process input list is always needed if something has to be weighed or dosed, that means if a dosing operation is included in the recipe procedure.

This list may be downloaded by a CIS system. There is one batch component list for each batch. The list (from CIS) consists of a CIS component ID and a quantity.

The following must be added to this list in the IOS:

- IOS component ID,
- Unit number (weigher),
- Dosing group (dosing sequence).

.

For the additions to the list, a component database and silo management data is accessed. Each component must be configured in silo management

If CIS is inactive, the process input list is transferred to the IOS component list during batch generation. However, all components are first recalculated to the size of the batch on behalf of the 'nominal batch size of process input list' parameter. The list must then be extended in the same manner as the CIS list.

7.2 The recipe editor

7.2.1 Start of the recipe editor

The recipe editor is started in the basic menu under the tab 'Production'.

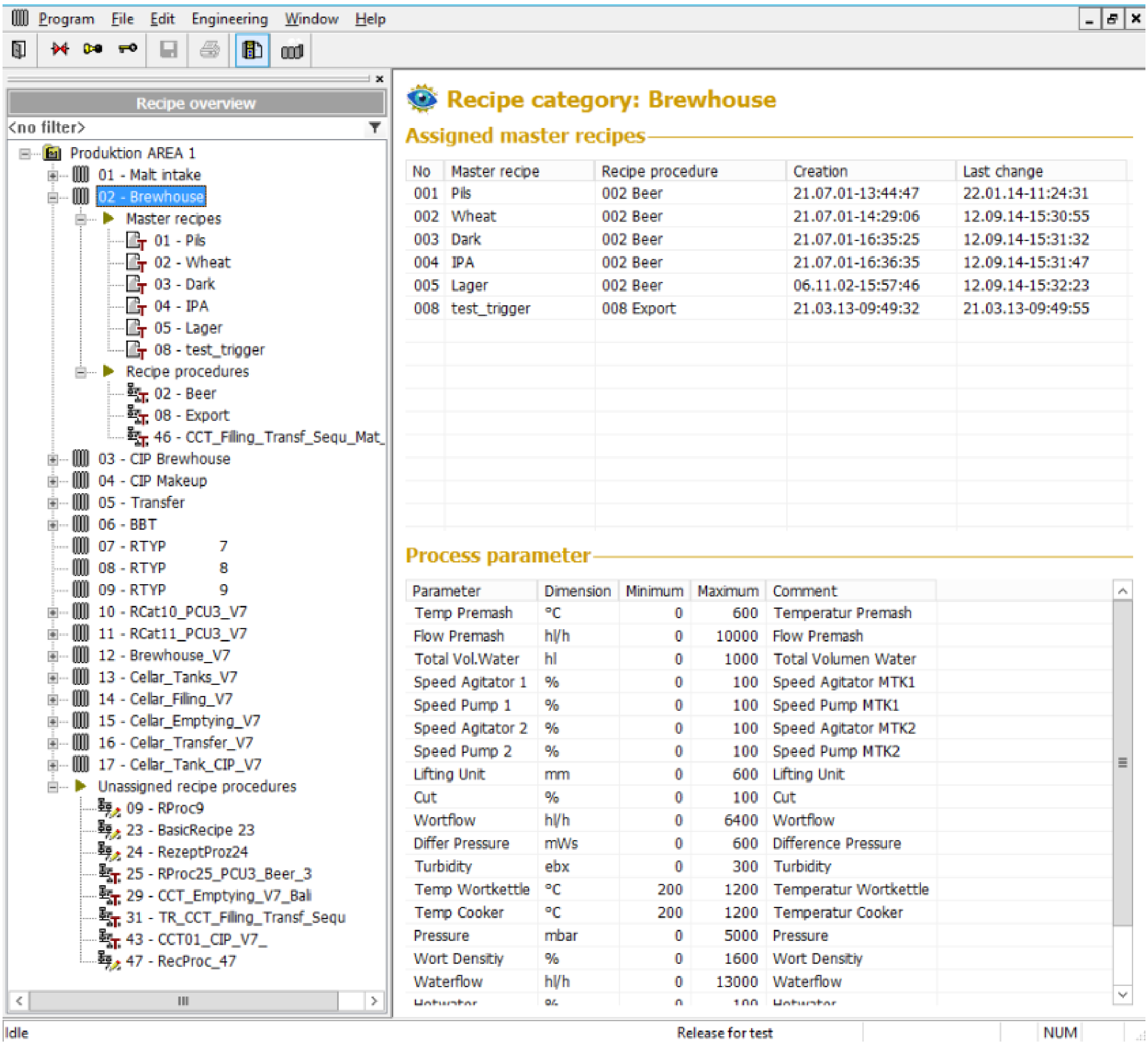
Note

The recipe editor does not support representation in other areas in one instance. You can enable the start of multiple instances of the recipe editor by setting the INI key [App]EnableMultiInst=1 in the editrec.ini file.

7.2.2 Overview - The views of the recipe editor

The recipe editor has three basic views used for generation and administration of the recipe systems elements:

- Recipe overview
  Presents the recipe categories in a tree view along with the master recipes and recipe procedures that have been assigned in each case. It also lists the "free" recipe procedures, i.e. those that have not yet been assigned. A mouse click on a recipe category opens the following display:



(To open a preview first, click a master recipe or a recipe procedure.)

- Master recipe view
  This view opens when you double-click a master recipe. It contains the master recipe header, the process parameter list, and the process input list.
- Recipe procedure view
  This view opens when you double-click a recipe procedure. In the initial state, it contains the graphical recipe procedure, the recipe unit procedure list, and the setpoints for the selected recipe unit procedure.

In addition to these main views any further elements of the recipe system can be processed e.g equipment operations, process parameter definitions, etc. by dialogs and menu inputs of the corresponding view. The corresponding dialogs are described in the context of the corresponding view.

7.2.3 Common control elements of all views

7.2.3.1 Menu commands

Menu 'File'

New > Master recipe

Creates a new master recipe and opens the master recipe view of the editor in order to process the new 'master recipe' immediately.

New > Recipe procedure

Creates a new recipe procedure and opens the following dialogs and the recipe procedure view in this order:

- Recipe procedure header dialog,
- RP properties dialog,
- Recipe procedure view.

The recipe procedure view is described in more detail in the chapter 'The recipe procedure view'.

Open > Recipe overview

Opens the recipe overview if it was closed or brings it to the front again.

Open > Recipe procedure

Opens the recipe procedure selection dialog and opens the selected recipe procedure in the recipe procedure view.

Save as

Selecting a master recipe causes a dialog for duplicating the master recipe to appear. You must specify the name of the new master recipe.

The following options are available for the ID:

- Append (new ID=<x>) → Generates a new (consecutive) master recipe ID
- Overwrite deleted master recipe → The list of deleted master recipes is active. When you select a line and confirm with the "OK" button, this is overwritten by the new master recipe and the "deleted master recipe" status is removed.

Preview > ...

Opens the print preview of different sections of the current view.

Print

Opens the Windows Print dialog.

Prints a partial section of the current view e.g. the process parameter list, the input list or the recipe category overview, or a total report of the recipe which has currently been loaded.

Printer setup

Opens the Windows dialog for setting up printers.

Menu 'Project planning'

Material management

- 'Materials and groups' command (see chapter Material and material groups (Page 585))
- 'Material sequences' command (see chapter Material assignment (Page 586))

Components

Opens the dialog for configuration of materials, material groups and material-weigher-assignments. The dialog is described in the document 'Dosing and Weighers'.

Storage stock

Opens the dialog for the storage location management. The dialog is described in the document 'Dosing and Weighers'.

Equipment data

Opens the equipment data dialog (see chapter Dialog ‘Engineering equipment data' (Page 574)) that is used to manage and configure equipment elements (units, sequences, operation, setpoints etc.).

Menu 'Window'

Cascade, tile vertically, tile horizontally

Specifies how windows should be arranged when you have several open at the same time.

Selection of opened windows

In this list all windows which are opened in the recipe editor are displayed. The window which is active at the moment is marked. The selected window is brought to the front.

7.2.3.2 Common dialogs of all views

Recipe procedure header dialog

Basics

This dialog is displayed

- if a new recipe procedure is created
- via the Edit / RP Properties / Header
- by pressing the button 'header data' in the RP-property-dialog.

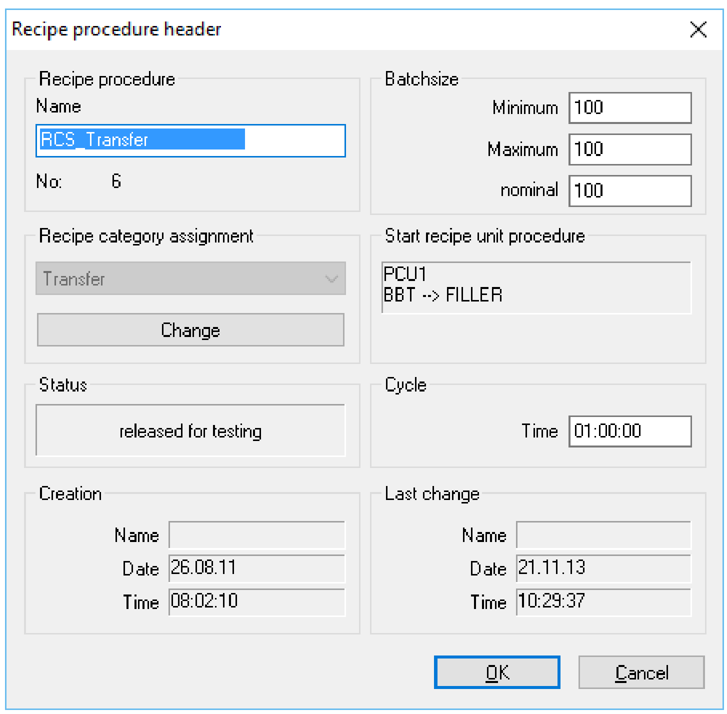


Figure 7-9 The recipe procedure dialog is opened on creation of a new RP. To make changes, the dialog can be reopened later via menu as well as via the RP property dialog.

The dialog is used to set the basic parameters of a recipe procedure. The parameters can be changed if the status of the RP is not 'released for production'.

The dialog has the following control elements and displays:

Recipe procedure name

Name of the recipe procedure

Recipe category assignment

'Change' button
Activates the combination field for the recipe category assignment.

Combination field 'Assignment'
Selection of a recipe category and assignment of the recipe procedure to this recipe category.

Note

For the substitution of order and process parameters an assignment to a recipe category is necessary. This assignment is registered automatically as soon as the recipe procedure is assigned to a master recipe. If no assignment is selected for new recipe procedures, the parameter substitution is lost. **If parameter substitution should be used in RPs, the recipe category, to which the parameters should be substituted later, has to be defined prior to RP creation.**

If a new recipe procedure is to be created with the menu command "File > Save as …", the dialog "Recipe procedure header" appears automatically. Here also, attention should be paid to the RKAT assignment for the above-mentioned reasons.

A recipe procedure can be assigned to one recipe category only.

Status

Display of the status of the recipe procedure. The status can be changed in the recipe procedure view via the menu item 'Edit' > Change RP status' after previous validation.

Creation

- Display
  When and by whom the recipe procedure was created.

Batch size

- Input 'Minimum batch size'
  Minimum batch size which can be produced with the recipe procedure.
- Input 'Maximum batch size'
  Maximum batch size which can be produced with the recipe procedure.
- Input 'Nominal batch size'
  Nominal batch size which is produced with the recipe procedure.

The batch sizes are important when scaling setpoints of the recipe procedure. Furthermore, these parameters are used for the calculation of the amount of material input when producing a batch size different from the nominal size.

Start sequence

- Display
  Identification of the sequence with the start recipe procedure.
- The start sequence is determined automatically after creating the recipe procedure.
- The start sequence is identified by the Start ROP.

Cycle

- Input time
  Scheduled time for processing the recipe procedure
- The cycle is used as the time between two batches in the order system in 'by time' and 'by time/automatic' start mode.

Last Change

- Display
  Date recipe procedure was last changed and the name of the user who made this change.

Button 'Ok'

Closes the dialog and saves the changes.

'Abort' button

Closes the dialog without saving the changes.

Dialog 'RP properties'

General

This dialog is used for the structure definition of a new recipe procedure.

This dialog can be opened in any view via the 'File / New / Recipe procedure' menu item or in the recipe procedure view via the 'Edit / RP Properties / Properties' menu item.

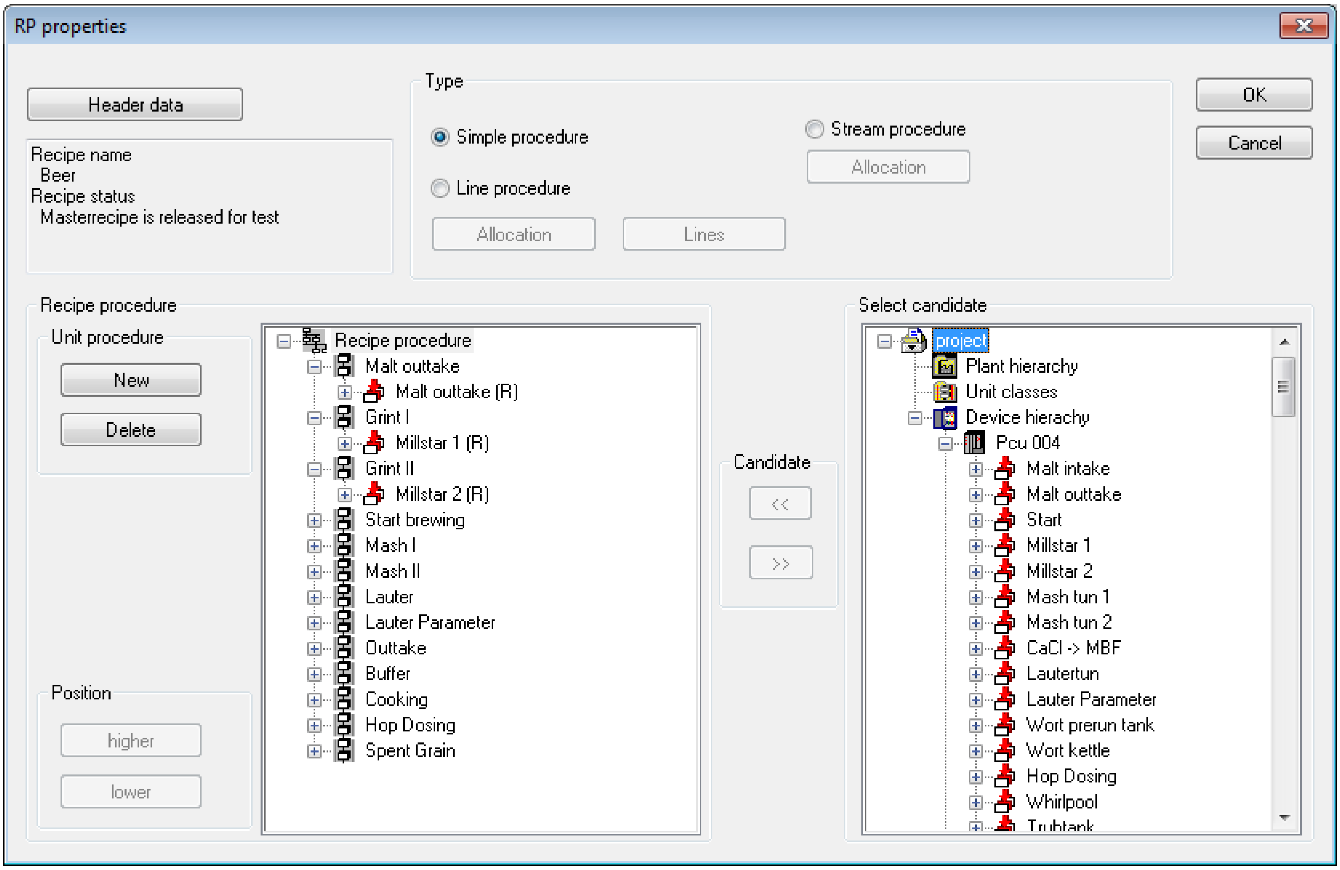


Figure 7-10 On creation of a new RP, the RP property dialog is used to define the recipe unit procedure sequence and for defining the recipe type. In the tree view on the right the units of the equipment hierarchy are displayed which can be assigned to the recipe unit procedure in the left tree view.

Button 'header data'

Opens the recipe procedure header dialog.

Text field 'Properties'

In this text field, the name and status of the recipe procedure are displayed. The name of the recipe procedure can be changed in the recipe procedure header dialog, the status via the menu item 'Edit' > 'Change RP status' in the recipe procedure view.

Type

Determines the type of the recipe procedure. The recipe types 'Simple procedure', 'Path procedure' and 'Line procedure' are supported in the current system version.

The 'simple procedure' is characterized by 1:1 assignment of recipe unit procedures and units. It doesn't have any parallel lines. If a simple procedure is assigned to a master recipe, exactly one recipe procedure can be started via this master recipe.

The 'Path procedure' allows one or more units and associated technical equipment groupings suited to producing one material batch to be grouped together. The 'Path editor' dialog is opened via the 'Assignment' button.

The 'lines procedure' is used for plants with repetitive equipment and permits – starting from a reference line - the same parallel lines to be generated for the same or approximately the same parallel lines.

Button 'assignment'

Opens the dialog for the assignment of reference and parallel lines to this recipe procedure.

Button 'lines'

Opens the dialog for the configuration of lines.

Recipe procedure

The left tree view shows the configured sequential order of defined recipe unit procedures and their assigned units and sequences. With the buttons of the 'recipe unit procedure' group, new recipe unit procedures can be created or deleted.

The button 'New' opens the dialog 'create new RUP'.

The current cursor position in the tree view determines the position of the recipe unit procedure that will be added:

- If the root element of the tree ('recipe procedure') is selected, the new recipe unit procedure will be added directly underneath the root element. If the structure of recipe unit procedures is not empty, the new RUP is inserted in front of the existing RUPs.
- If an RUP, an RUP assigned unit or its sequence is selected when inserting a new recipe unit procedure, the new RUP is appended behind the selected object and in front of already existing RUPs. If the structure of recipe unit procedures is not empty, the new RUP is inserted in front of the existing RUPs.

The unit which is assigned to a recipe unit procedure is also called 'candidate' for this recipe unit procedure.

The recipe unit procedures can be moved within the tree view with the two 'Position' buttons up ('higher') or down ('lower'). In the graphical recipe procedure view the uppermost recipe unit procedure is indicated to the left, the one below to the right.

The indicator 'R' behind the candidate's name identifies it as the reference candidate. It has no meaning in the path procedures.

The names of RUPs that are already selected by a second click (no double-click!) become editable.

Candidate selection

In the tree view to the right the hierarchy views of the project are displayed. The PCU-dependent units can be selected from the device hierarchy. By pressing the ‘<<’ button, units are selected as candidates for an RUP selected in the left tree view. Candidates can be deleted from the left tree view by pressing the ‘>>’ button.

Candidate

The two buttons of this group can be used to assign a unit as a candidate to a recipe unit procedure or to delete candidates from RUPs.

The assignment button ('<<') is only activated if the selected recipe unit procedure has not yet been assigned a candidate.

The delete-button ('>>') is only activated if the selected object in the left-hand tree control is a unit/candidate.

A unit must be assigned to exactly one recipe unit procedure. An individual unit must not be the candidate unit for more than one RUP in one recipe procedure, Only a single candidate may be allocated to a recipe unit procedure (except in path procedures).

Display of the assignment in the recipe procedure view of the recipe editor

In the recipe procedure view, the assignment of units to recipe unit procedures is displayed in the function header of the recipe columns as shown in the illustration:

1. Line	Name of the recipe unit procedure (here: 'Weighing 3')
2. Line	Name of the unit ('Weigher 3')
3. Line	Name of the sequence which is assigned to the unit followed by a dash, the PCU number and the sequence number (e.g. 'WeighingSeq 3 - 2 / 32' means sequence name is 'Weighing Seq', it is part of PCU 2 equipment objects and has the ID 32).

Figure 7-11 Function header of a recipe column in the RP view of the recipe editor

Button 'Ok'

Closes the dialog and saves the changes. The recipe unit procedure structure created is checked for syntactic correctness before the dialog closes. If errors are found (e.g. recipe unit procedures without candidates), the dialog does not close.

'Abort' button

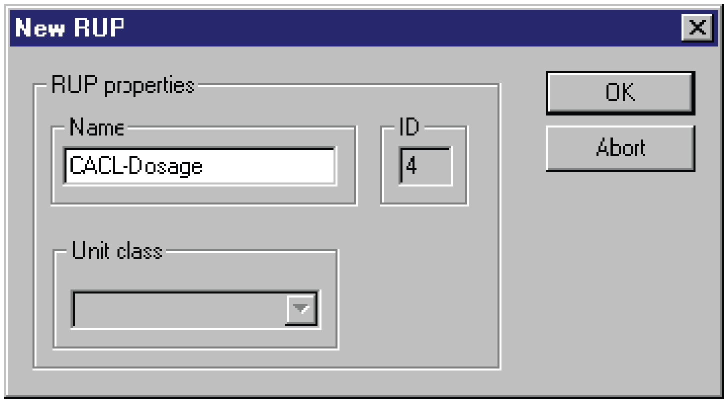
Closes the dialog without saving the changes.

Change unit or sequence configuration

If changes are made to the configuration of units or sequences in files '<proj-path>\pcu.xxx \recipe\unit' (xxx represents the PCU number) or '<proj-path>\pcu.xxx\recipe\sequence.ini' the assigned units and sequences are not automatically updated in the recipe procedure structure. If the combination of units and sequences is changed, i.e. a unit is assigned a new sequence, the unit must be deleted from the left tree view and reinserted before the change in assignment takes effect.

Dialog 'New RUP'

This dialog is opened by pressing the 'New' button in the recipe unit procedure group of the 'RP properties' dialog.



Name

Name of the new recipe unit procedure, preset with 'RUP_abc' where 'abc' is the consecutively numbered ID of the recipe unit procedure.

ID

Consecutively numbered ID of the recipe unit procedure. The ID is created by the system and cannot be changed.

Button 'Ok'

Closes the dialog and saves the changes.

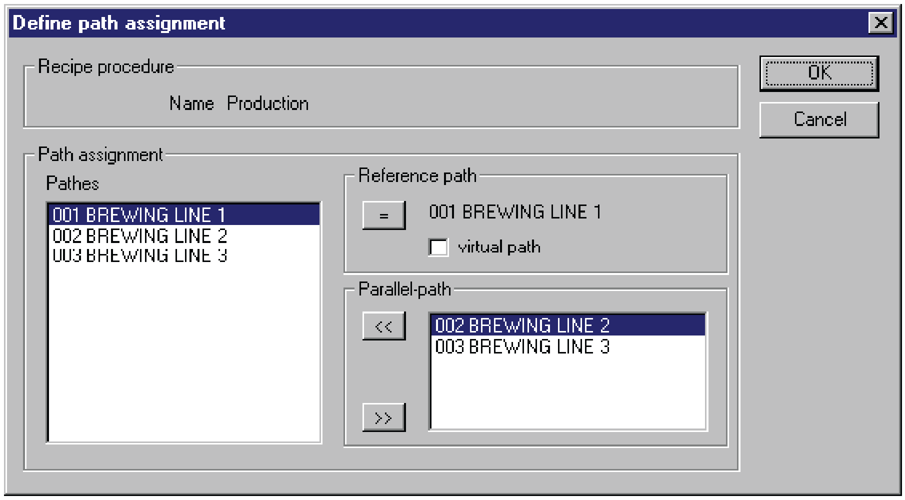
Button 'Abort'

Closes the dialog without saving the changes.

Dialog 'Define line assignment'

The dialog 'Define line assignment' is opened via the button 'Assignment' in the dialog 'RP property' in the 'Type' groupbox.

The 'Assignment' button is activated only if the recipe procedures type was selected as 'line procedure'.



Recipe procedure

Display of the recipe procedure name.

Line assignment

The 'Line' listbox on the left shows all lines configured in the project. The configured lines are valid globally. Lines are not reference or parallel lines implicitly.

When a line from the list is assigned to a line procedure, this line becomes the reference line (Button '=' in the 'reference line' group) or a parallel line (Button '>>' in the 'parallel lines' group) for this line procedure. If there is mark set in the virtual path checkbox, the reference line becomes a so-called virtual line that cannot be used for the production of a batch. Instead it is only a template line for the definition of parallel lines. Its elements are a superset of all parallel lines' elements. Elements of the reference line can be omitted during the conversion ('zero conversion').

See also: Lines

The '<<' button removes the line from the parallel line listbox.

Each line can only be assigned to a recipe procedure once. Each recipe procedure is assigned at least one reference line. Not all lines have to be assigned.

Button 'Ok'

Closes the dialog and saves the changes.

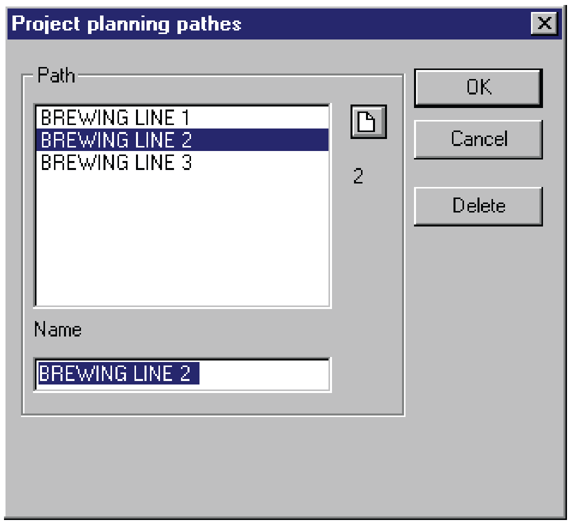
Button 'Abort'

Closes the dialog without saving the changes.

Dialog 'Configuration lines'

The dialog 'Configuration lines' is opened via the 'Lines' button in the operating element group 'type' in the dialog 'RP properties'.

The line configuration is activated for recipe procedures of the type 'Line procedure' only.



As mentioned above, line assignment tables must be created manually. Only the name is defined in the input field and the number of all lines via this dialog (button 'Delete' and button  for creating new lines).

Button 'Ok'

Closes the dialog and saves the changes.

Button 'Abort'

Closes the dialog without saving the changes.

Dialog 'Open recipe procedure'

This dialog is displayed via the menu item 'File' > 'Open' > 'Recipe procedure'.

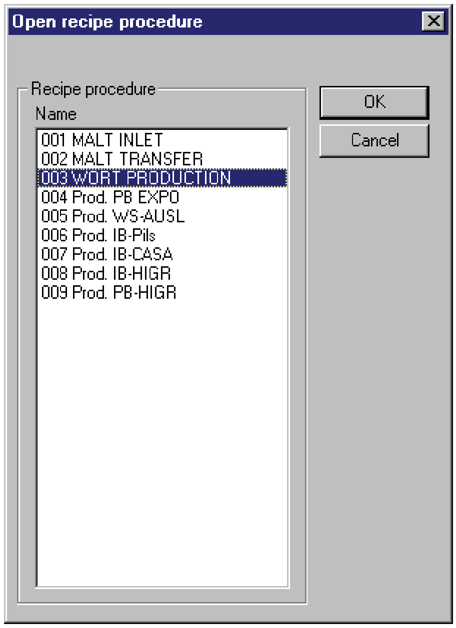


Figure 7-12 Dialog for opening an RP.

The required recipe procedure can be selected from the list. The selected recipe procedure is opened by pressing the ‘OK’ button or by double-clicking the procedure name. The recipe procedure which was loaded last is preselected when the dialog opens. The dialog is closed without opening a recipe procedure by pressing the 'Cancel' button.

Dialog ‘Engineering equipment data'

System hierarchy tree view

The dialog is opened via the 'Configuration / Equipment data...' menu.

The tree view on the left side of the dialog shows the device hierarchy of the system configuration for each range. The configured PCUs of the system can be found under device hierarchy. The equipment elements of the recipe system are assigned to the PCUs:

- Units and sequences 'unit'
- Equipment operations and functions
- Parameter elements (=DFMs, setpoint container)

The detailed view of the individual hierarchy levels is only available for the active area.

Property list

If one object is selected in the tree view its properties will be listed to the right. The lines of the list are dynamically adapted to the selected object. The properties are shown write-protected.

Context menu

Right-clicking on the tree view object (operation, parameter element, unit) or clicking on the "Properties" button opens a dialog to configure the object definitions.

Close

Closes the dialog. Changes to individual equipment objects are implemented by pressing the ‘OK’ button from their properties dialogs.

Dialog 'Units Properties'

This dialog is opened if the menu item 'properties' is selected in the context menu of one unit in the equipment data dialog.

The name of a unit can be configured in this dialog.

Names of units and sequences can also be configured using the text parameterization application.

Button 'Ok'

Closes the dialog and saves the changes.

'Abort' button

Closes the dialog without saving the changes.

Dialog 'Configure sequences'

Basics

The sequence configuration dialog is opened via the menu item 'properties’ of the context menu of the sequences in the equipment data dialog.

Description

A sequence is uniquely identified using the sequence ID and the PCU of the sequence. Both entry fields are therefore write-protected. The name of the sequence can be edited here.

Unit assignment

The PCU No. and the unit are permanently allocated to the sequence and therefore write-protected in this dialog.

Sequence-related setpoints

A list of setpoints of this sequence, separated by commas. The setpoints are entered in the format DFM-group.DFM-number.

Example: Setpoint from DFM group 2, DFM number 123 and setpoint from DFM group 3, DFM number 12:

Input: 2.123,3.12

Note

Setpoints defined here serve only to permanently display selected setpoints of those setpoints used in the EOPs (see next chapter). They must not be considered as additional setpoints to those related to the operation.

'...' button

Opens the dialog for setpoint selection.

The assignment of the setpoints by input field or dialog is equivalent to assigning them during configuration. However, the dialog is much more convenient for it displays possible setpoints with setpoint names and always results in a valid formatted list.

Recipe categories

Recipe categories and master recipes can be assigned to the unit here, allowing the unit to be started manually in the process cell overview. If the text box is left blank, the unit cannot be started manually in the process cell overview. The following entries are possible:

- One recipe category → All the master recipes that fall into this category can be selected
- Multiple recipe categories (comma-separated list)
- A selection of specific master recipes can be defined for a recipe category by adding them in brackets
  (comma-separated list or <from> - <to>)

Example:

Recipe categories = "1(22,23,24),2(1-9,11-22,24,25)"
→ The following recipe categories/master recipes can be selected when starting the unit manually:

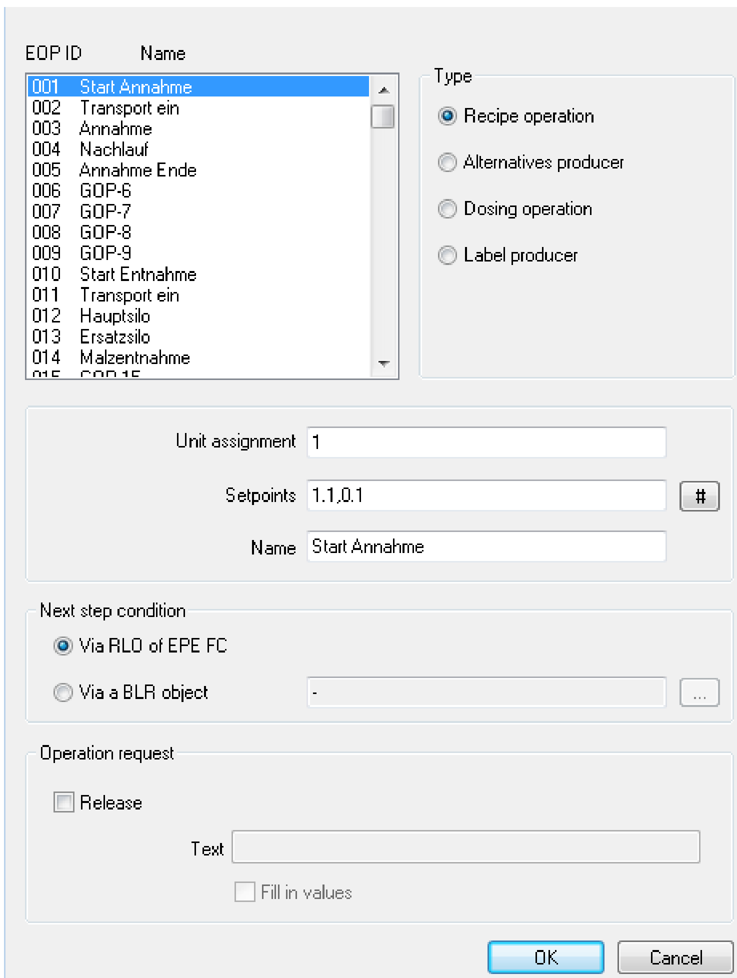
Recipe category	Master recipes
1	22, 23, 24
2	1 to 9, 11 to 22, 24, 25

Note: A "!" highlighted in yellow is displayed as long as the input is syntactically incomplete.

Dialog 'Project equipment operation'

Basic principles:

This dialog can be opened in every view via the context menu in the equipment data dialog. In the recipe procedure view it is directly accessible from the configuration menu (EOP definition editing) and via the context menu of the recipe operations.



List of equipment operations

In this list all equipment operations of one PCU are shown. All other parameters which are displayed in this dialog window refer to the operation which is selected in the list.

Type

With the radio button of this operating element group the type of the equipment operation is determined:

- Equipment operation
  No special parameters or properties
- Alternative producer
  The equipment operation produces alternative decisions which use an alternative in the recipe control to start one of several possible units.
  See Alternatives (Page 707)
- Weighing EOP
  The equipment operation executes a metering operation.
  See supplemental documentation "Weighing and weighers" chapter "Configure weighing operation".
- Label producer (selectable only for S7 PCUs)
  The equipment operation generates jump destinations which the recipe control uses to determine where to branch recipe executing through a jump operation.
  See Jump Operation (Page 708)

Sequence assignment

A comma-separated list of the sequences indicating in which sequence a selected EOP may be used. If this field is empty, the EOP cannot be executed on any sequence, while a ‘0’ indicates an EOP that may be used on any sequence.

Note

The properties dialog box opens when you insert or append data, or call the shortcut menu in an ROP. Its combo box at the top shows all EOPs which are assigned to the currently active unit/sequence. In default factory state, all EOPs are assigned to all sequences ("0"). This is why the combo box always shows all of the EOPs. If you prefer a less complex view, you can easily hide the EOPs which are not explicitly assigned to this unit or sequence. To do so, open the EOP definition configuration files in a suitable ASCII-based editor (e.g. Microsoft's Notepad.exe) and replace the "UnitAss=0" entry in all sections with "UnitAss=65" (PCU < V7) or "UnitAss=129" (PCU V7). This modification assigns the corresponding EOPs to the non-existing unit/sequence 65 or 129, which hides the EOPs in the properties dialog. **Always take particular care when you edit the configuration files!**

Note

Changes to EOP definitions do not become active until each linked recipe procedure or procedures (RP) is/are opened and resaved in "Edit" or "Release test" status. This is a system feature and attention should be paid in order to maintain consistent EOP processing.

Setpoints

A comma-separated list of the setpoints of this equipment operation. The setpoints are entered in the format EPAR-group.EPAR-number.

A setpoint can be write-protected by concatenating an ‘-e‘ to the Group Number. **This only has an effect on the setpoint input in the recipe editor/list recipe and not on the setpoint dialogs during runtime (plant overview, control recipe, order system).**

Example: Setpoint from DFM group 2, DFM number 123, write-protected and setpoint from DFM group 3, DFM number 12, not write-protected

Input: 2.123-e,3.12

Button '#'

This command button opens the dialog for the assignment of setpoints.

The assignment of the setpoints by input field or dialog is equivalent to assigning them during configuration. However, the dialog is much more convenient for it displays possible setpoints with setpoint names and always results in a valid formatted list.

Name

Name of the equipment operation.

Step enabling condition

- Via RLO of the EOP FC (mandatory for PCU versions prior to V7)
  The step enabling condition must be formulated in the AWL program of the EOP function.
- Via BLR object (possible for PCU version from V7)
  The step enabling condition is supplied by a 'Binary Logic Result' object. Interconnection with the BLR instance takes place via the '...' button. The inputs and the connection logic of this object instance is configured with a separate editor (see BLR - Binary connection logic (Page 225)).

Operator prompt

The controls in this group are used to configure an EOP to display an 'operator prompt' message in the sequence control application. To do so the box ‘release’ must be checked. The text then pops up in the plant overview as an operator prompt dialog (see User request (Page 825)).

You can enter the process value of a DFM in the acknowledgement dialog window of the plant overview when you set the "Value input" check box.

Button 'Ok'

Closes the dialog and saves the changes.

'Abort' button

Closes the dialog without saving the changes.

'Configure parameters for equipment operation', setpoint dialog

This dialog is opened by pressing the '#' in the 'Configure equipment operation' dialog.

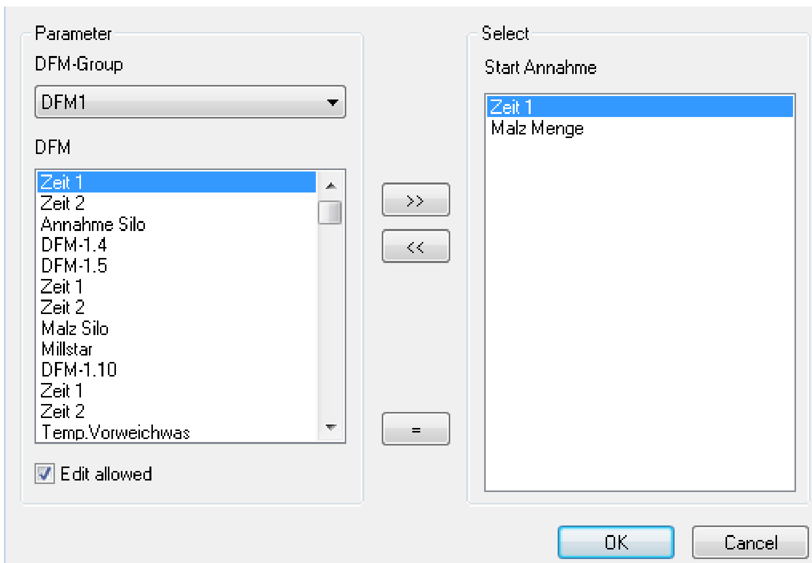


Figure 7-13 Parameter assignment dialog

Parameter

In this control element group the DFMs which shall be assigned to an EOP can be selected. With the combination field 'DFM group' the group is preselected. In the list 'DFM' below an individual parameter is selected. The parameters are editable by default. By removing the check in the check box 'Editable', the parameter becomes write-protected in the recipe editor.

Assignment button '>>'

This button adds the parameter selected to the list in the setpoint interface of the equipment operation.

Delete button '<<'

This button removes the parameter selected from the list 'selection' of the equipment operations setpoint interface.

Replace button '='

This button replaces the DFM selected in the list 'selection' by the parameter selected in the list 'DFM'.

Selection

List of DFMs which are inserted in the setpoint interface of an equipment operation.

Button 'Ok'

Closes the dialog and saves the changes.

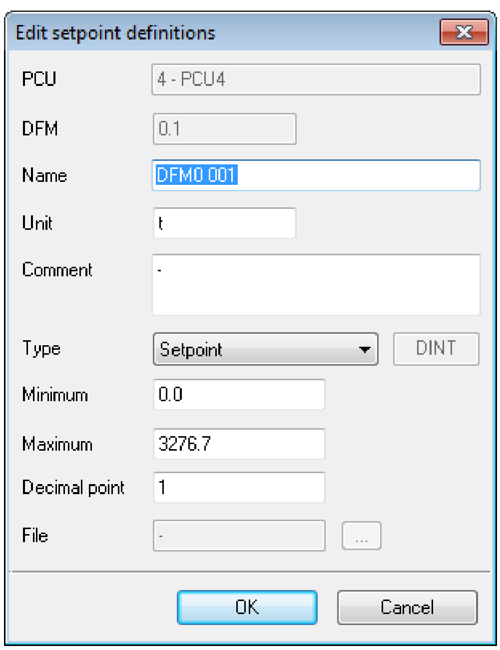
Button 'Abort'

Closes the dialog without saving the changes.

'Edit DFM definitions' dialog

Basics

This dialog can be opened in every view via the context menu of a DFM in the equipment data dialog. In the recipe procedure view it is directly accessible from the configuration menu (DFM-Def. edit) and is in the context menu of the setpoints.



Parameter information

- PCU - PCU number and name
- DFM - DFM group and number.
- Name - DFM name
- Unit - measured value unit of the parameter
- Comment - A comment can be edited for this parameter element definition.

Parameter types and value limits

The following input and list boxes specify the attributes, parameter type, data type, minimum and maximum value and if applicable associated text lists. Depending on the selected type the input fields of the group are activated or deactivated.

The following parameter types are defined:

Setpoint

Numeric data type

The value of this parameter element is a number with the minimum value 'Min.' and the maximum value 'Max.'.

- **Data type PCU < V7**
  only DINT variable is supported
- **Data type PCU as of V7**
  Format for DFM 0, 1, 2, 3, 4: DINT
  Format for DFM 5, 6, 7, 8: REAL
- Minimum and maximum values
  according to the range of values for the formats DINT and REAL (see section Parameter types (Page 204))
  Important: Changes here are also directly reflected in the display for parametrization (online and offline).
- Dec. pt.
  The position of the fixed point is determined in the field 'Dec.Pt'. This therefore decides the number of decimal places for the two number formats (DINT, REAL).
  Example:
  DINT, Dec.Pl. '0': no decimal places, abs. value range -2147483648 to +2147483647
  DINT, 'Dec.Pl. '2': one decimal place, abs. value range -21474836.48 to +21474836.47
  The input of the values for 'Min.‘ and 'Max.‘ limits the range of value further.

Text

- The setpoint is represented by text from a PCU-specific text list.
- You use the "..." button next to the 'File' input box to select the text file. The selection dialog shows the text files in the project directory "`<proj-path>\Pcu.nnn\Texte.x`" (nnn= PCU no.; x= language number).
- The setpoint is entered in the parameter dialogs using a list box that shows the lines of the text file defined by the values 'Min' and 'Max'.
- With this type, the decimal point is meaningless and is therefore grayed out.

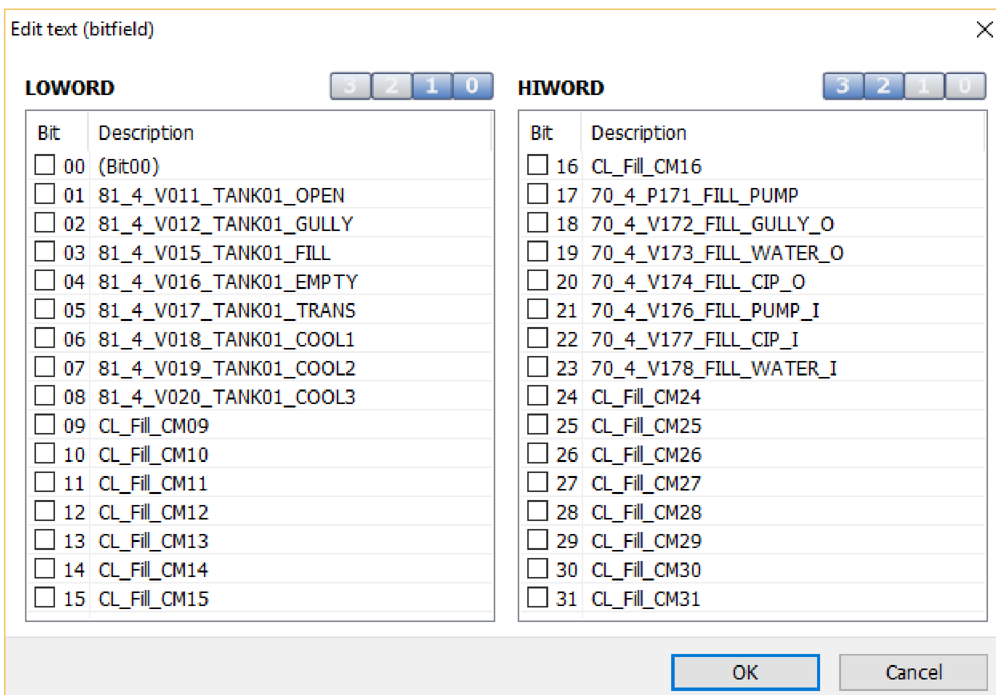
Text (global)

- The setpoint is represented by text from a global text list.
- You use the "..." button next to the 'File' input box to select the text file. The selection dialog shows the text files in the project directory "`<proj-path>\Texte.x`" (x= language number).

- The setpoint is entered in the parameter dialogs using a list box that shows the lines of the text file defined by the values 'Min' and 'Max'.
- With this type, the decimal point is meaningless and is therefore grayed out.

Text (Bitfield) (PCU from V7)

- The setpoint is defined by a bit pattern (32 bits).
- The description of the bits can be saved in the file prescribed in the entry field 'File' in the SW definition. The file name is entered without the '.txt' suffix. The test file must be in the project directory "`<proj-path>\Pcu.nnn\Texte.x`" (nnn= PCU No.; x= language number).
- The file can contain a maximum of 32 lines, each with 32 characters.
- The setpoint is entered in the parameter dialogs using a separate dialog in which the 32 bit positions can be set/reset using select boxes.



Text (indirect)

- The setpoint is represented by text in various text lists.
- The function is comparable with TEXT type but is implemented in 2 stages. It makes it possible to select elements from groups again.
- See chapter Setpoint definition type ‘COND’ (Page 1229) for further information about the configuration and function of the setpoint type.

Material

- The setpoint is represented by a material from the material engineering. The material engineering is created in the recipe editor with the menu command "Engineering / Materials and groups".
- Entering the set point in the parameter dialogs allows the selection of a material from the full list (material group "*") or from a defined material group.
- With this type, the decimal point is meaningless and is therefore grayed out.

Tank (PCU as of V7)

- The setpoint is represented by a tank from the PCU-specific tank list ('<proj-path>\pcu.nnn\texte.x\Tank.txt' file). The text file is created during the parameter assignment of the tank block instances by the text parameter assignment.
- The setpoint, when entered directly, is entered using a list box that shows the lines of the text file that are defined by the values 'Min' and 'Max'.
- In runtime, the setpoint can be set using a dynamic filter dialog generated based on the current tank attributes (refer to the section 'Tank selection' dialog (Page 796)).
  This can be called with a mouse click on a setpoint of the "Tank" type in the following operator control applications:
  – Order system: "Batch parameters" pane
  – Process cell overview: Step-related or sequence-related setpoint list
  – UnitCtrl: Setpoint list of the units or basic operation
- With this type, the decimal point is meaningless and is therefore grayed out.

Tank location (PCU as of V7)

The setpoint is represented by a global location ('<proj-path>\texte.x\rcs_loc.txt' file). The locations are configured in the following two ways, depending on the use of the "Route Control System" option:

1. Separate from RCS using the text parameter assignment of global texts
2. Using the RCS configuration/locations and then loading the RCS server. The configured locations are transferred to the text file.

- The setpoint, when entered directly, is entered using a list box that shows the lines of the text file that are defined by the values 'Min' and 'Max'.
- In runtime, the setpoint can be set using a dynamic filter dialog generated based on the current tank attributes (refer to the section 'Tank selection' dialog (Page 796)).
  This can be called with a mouse click on a setpoint of the "Tank location" type in the following operator control applications:
  – Order system: "Batch parameters" pane
  – Process cell overview: Step-related or sequence-related setpoint list
  – UnitCtrl: Setpoint list of the units or basic operation
- With this type, the decimal point is meaningless and is therefore grayed out.

Bit pattern (16 bits)

- The 16 lower bits of the setpoint are shown in binary.
- The input occurs by keyboard in the binary format ('0‘ and '1‘).
- With this type, the decimal point, minimum and maximum is meaningless and is therefore grayed out.

Bit pattern (32 bits)

- The setpoint is represented as an 8-digit hexadecimal number.
- The input occurs by keyboard in the 'hex' format.
- With this type, the decimal point, minimum and maximum is meaningless and is therefore grayed out.

Button 'OK'

Closes the dialog and saves the changes.

'Abort' button

Closes the dialog without saving the changes.

Material and material groups

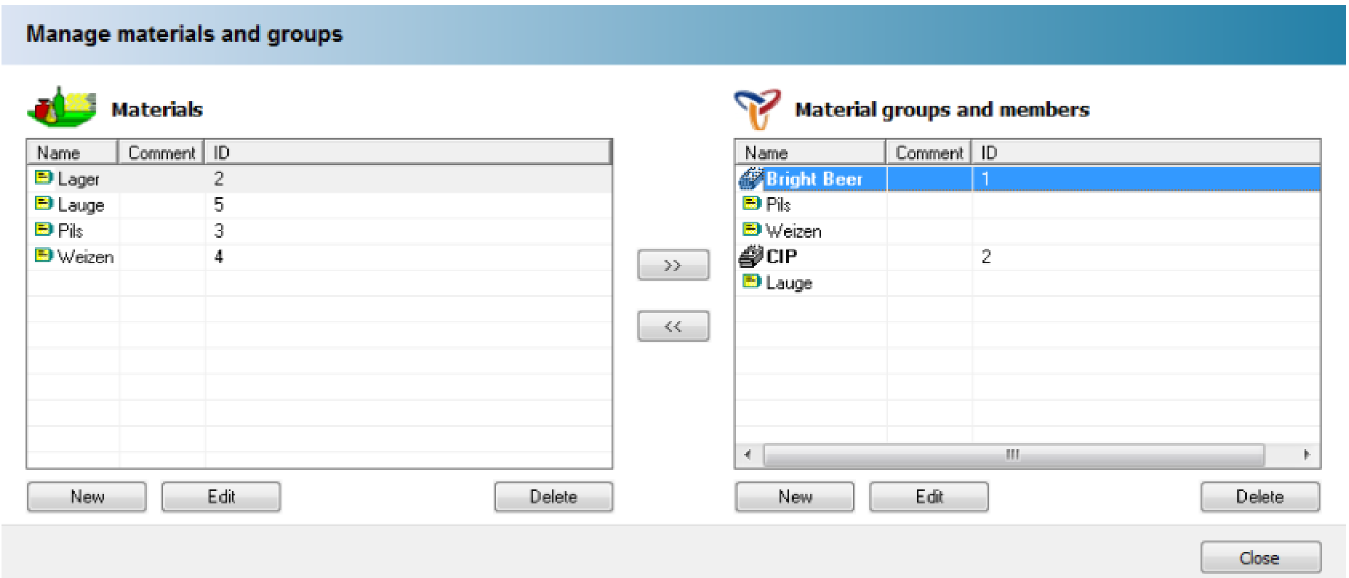
Basic principles:

In many fields of the industry specific materials may follow after each other or may use the same transport route. For example the beer product should never follow in the brewery after the acid cleaning directly, but only after the flushing out with water. Therefore, it is possible in the system to parameterize a material compatibility and/or material order.

- Case 1, Route control:
  The information as to which material is used in which Partial Route is buffered in a connected Link Element. With this information the Route Control checks whether the next material may follow or not.
- Case 2, Batch processing:
  - In the job planning
    When selecting a tank for a batch parameter, tanks with unsuitable content (material) can be filtered out.
  - When carrying out a step chain in the PCU:
    It is possible for each user EOP to check 2 material IDs for suitability.

Material configuration

Open the dialog via the recipe editor menu ‘Configuration → Material management → Materials and Groups' to configure different materials. Here you define the materials that should be transported and their group assignment.



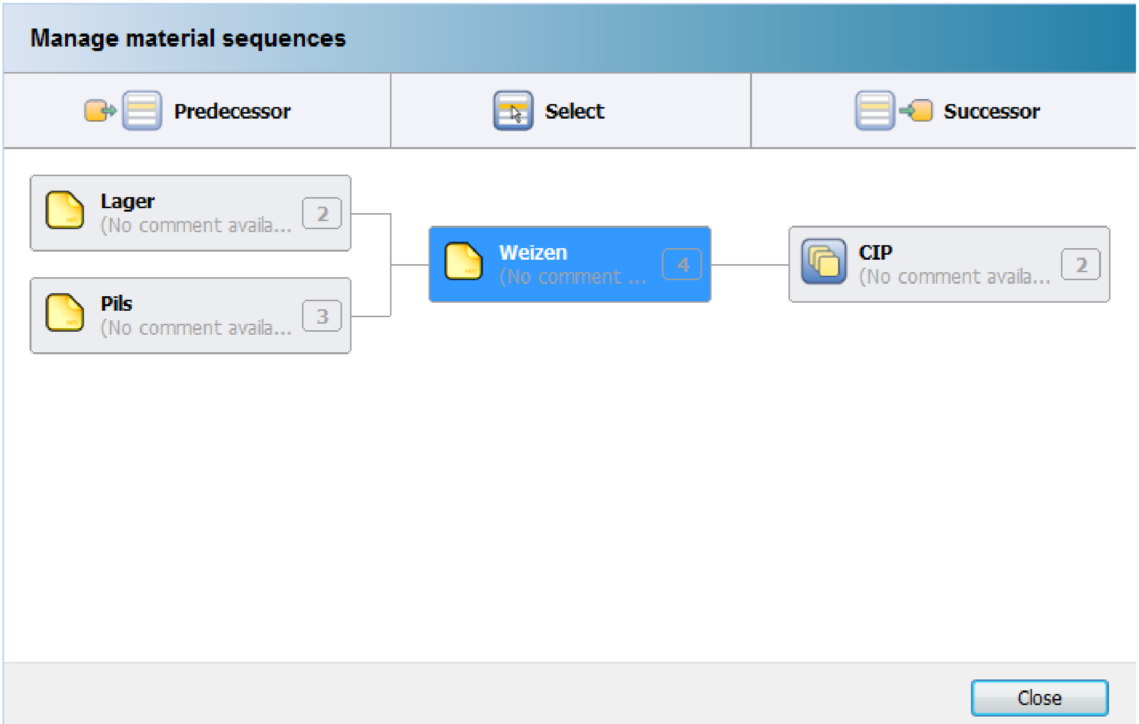
Creating a new material:

- After pressing the button "New material“ the dialog will appear for the input of a new material. The ID input field is preset with the highest number. However, it is possible to change this number. The necessary input fields are the ID number and the name field of the new material. The comment field can be used but isn't mandatory. If all fields are entered, the material will be created in the database and displayed in the list window for materials after pressing the "Add" button.
- To change, select the material in the list window and open the change dialog by pressing the button "Edit".
- After creating the materials it is possible to define material groups. You take the same steps as were used for creating materials. Here the same input conditions are valid, i.e. the ID number and the name of the material group should be entered for the definition.
- The desired material and the desired group can be selected and allocated or removed with the buttons '>>' and '<<'.

Material assignment

To define the material sequence designations, open the dialog 'Material sequences' in the menu 'Configuration →Material management →Material sequences’.

1. First of all, select the reference element via a list box, i.e. the material or material group for which a sequence relationship is to be configured.
2. Then the following view opens (example with pre-existing relationships):



The predecessor and successor elements of the selected element are displayed in the middle column (material or material group).

Operating concept:

Successors can be added to or removed from the current selection element.

The following functions are available by moving the mouse over the various elements:

- Double clicking the predecessor or successor element moves them to the selection position.
- Double clicking the selection element open the list box so that a new element can be selected

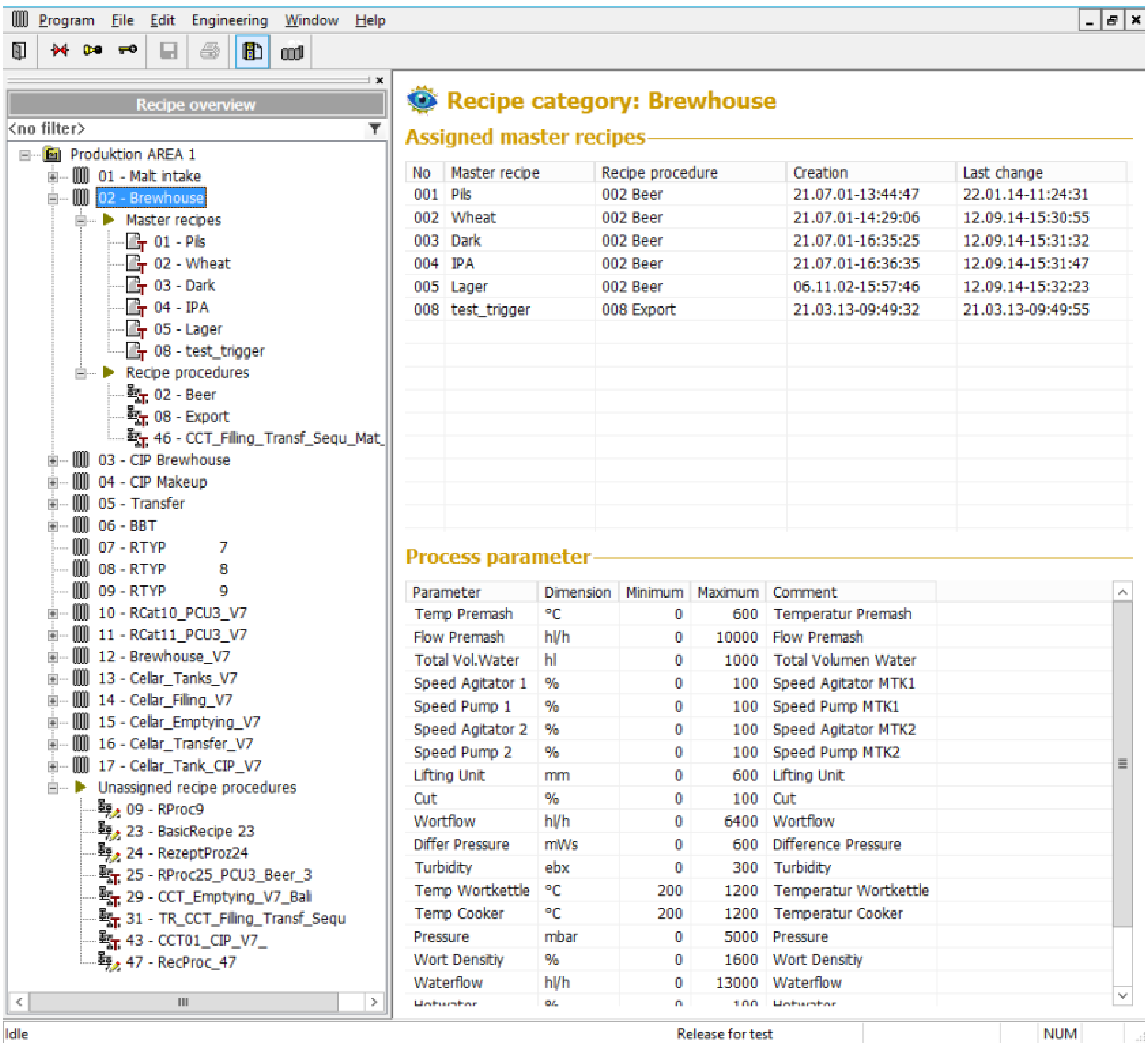
The following commands are available in the shortcut menu (by right-clicking):

- Predecessor / Selection → Moves the element to the selection position
- Select / Select other object → Opens the list box so that a new element can be selected
- Select / Add successor → Defines a new /additional successor element
- Successor / Selection → Moves the element to the selection position
- Successor / Remove → Removes the successor relationship from the selection element

7.2.4 The recipe overview

Basic principle

The recipe overview is used to navigate in the recipe system. Therefore it is laid out similarly to a file management tool. On the left side there is a tree view with the recipe system objects. The right side subviews provide further information about the object selected in the tree control and lets the user open these objects in their special editor views.



Recipe overview

In this tree view, the recipe objects for the active area are displayed hierarchically.

- The first hierarchy level below the area contains the recipe categories.
- The master recipes and recipe procedures are located in the second level. For each element, the **current release status is identified by a symbol in front of it**.
  - Pen = In preparation
  - "T" = Test release
  - Lock = locked
  - Green check mark: Production release
- If you select the process cell or a recipe category in the tree view, all the elements associated with this category are listed in the overview in the top right-hand corner. In the case of the recipe category, the process parameters are displayed in the area underneath this.
- When you select a master recipe or a recipe procedure, the right-hand pane shows the associated preview together with the key header data and relevant state.
- Whenever you double-click a master recipe or recipe procedure (in any view), the relevant element opens in a new window. Open elements are indicated in the tree view by means of "bold" font.
- In the delivery state, the tree view is configured so that no object IDs (recipe category, master recipe, and recipe procedure numbers) are displayed in front of the object name. You can change this behavior by setting the following INI switch in file `'<proj-path>\sys\editrec.ini'`.
  `[RecipeBrowser]ShowIDs=1`

Note

Delivery state:

In the delivery state, the system contains exactly 1 recipe category with 1 master recipe and 1 recipe procedure.

Shortcut menus

The recipe system tree view offers possible operations via shortcut menus. The possible operations that are available in the shortcut menus depend on what has been selected in the tree view as per the table below.

Selection	Available shortcut menus
Process cell	• New recipe category Inserts a new recipe category at the end of the structure tree. Displays the "Define recipe category" dialog for the new recipe category.
Recipe category	• Process parameter definition Opens the "Define process parameter list" dialog. • New master recipe Generates the data required for a new master recipe and opens this new master recipe in the master recipe view. When you create the master recipe via the shortcut menu for a recipe category, it is assigned to this category. Once a master recipe has been assigned to a recipe category, you cannot change this subsequently. When the new master recipe is created, it is immediately saved by the system automatically. • Undelete master recipe If a master recipe within this recipe category has been deleted, you can restore it via a list dialog. • Properties Displays the "Define recipe category" dialog.
Master recipe	• Open master recipe Opens the master recipe that has been selected in the master recipe view. • Delete master recipe Deletes the master recipe that has been selected in the tree view. The recipe is not actually deleted completely (see "Undelete master recipe")
Recipe procedure	• Open recipe procedure Opens the recipe procedure that has been selected in the recipe procedure view.

Automatic acceptance of tree view changes

Changes made in the tree view (such as the addition of a new recipe category or the renaming of a recipe category) are saved automatically.

Behavior associated with language dependency

The "engineering language" system setting is responsible for determining how the recipe names are saved (see section Plant configuration with "SiteCfg" – Application / Factory settings). The following table provides information about the applicable data storage method for descriptions of

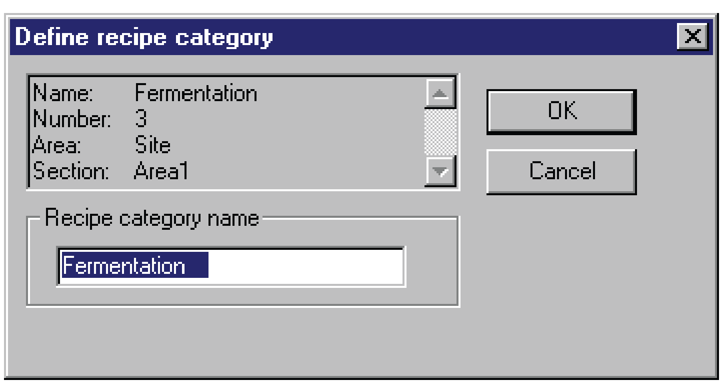
- Recipe categories
- Master recipes
- Recipe procedures

	Engineering language disabled	Engineering language active
Recipe category	'<proj-path>\texte.#\RTYP.txt' # follows the system language	'<proj-path>\texte.#\RTYP.txt' # follows the engineering language
Master recipe	'<proj-path>\texte.#\SORTn.txt' # follows the system language **Note:** SORTn.txt is only *written* by 'Recedit' (direct changes to text are lost)	'<proj-path>\texte.#\SORTn.txt' # follows the engineering language
Recipe procedure	'<proj-path>\texte.#\BASICREC.txt' # follows the system language **Note:** BASICREC.txt is only *written* by 'Recedit' (direct changes to text are lost)	'<proj-path>\texte.#\BASICREC.txt' # follows the engineering language

7.2.4.1 Dialogs of the recipe overview

Dialog 'Define recipe category'

This dialog is opened in the tree view via the context menu of the active area (Menu item 'New recipe category') and via the context menu of the recipe category (Menu item 'Properties').



It shows the write-protected parameters of the recipe category. The only parameter that is not filled in automatically is the recipe category's name, which can be changed in the text input field.

Button 'Ok'

Closes the dialog and saves the changes.

Button 'Abort'

Closes the dialog without saving the changes.

‘Configuration process parameter definition’ dialog

Basic principles

This dialog is opened via the tree view context menu of a selected recipe category (Menu item 'process parameter definition').

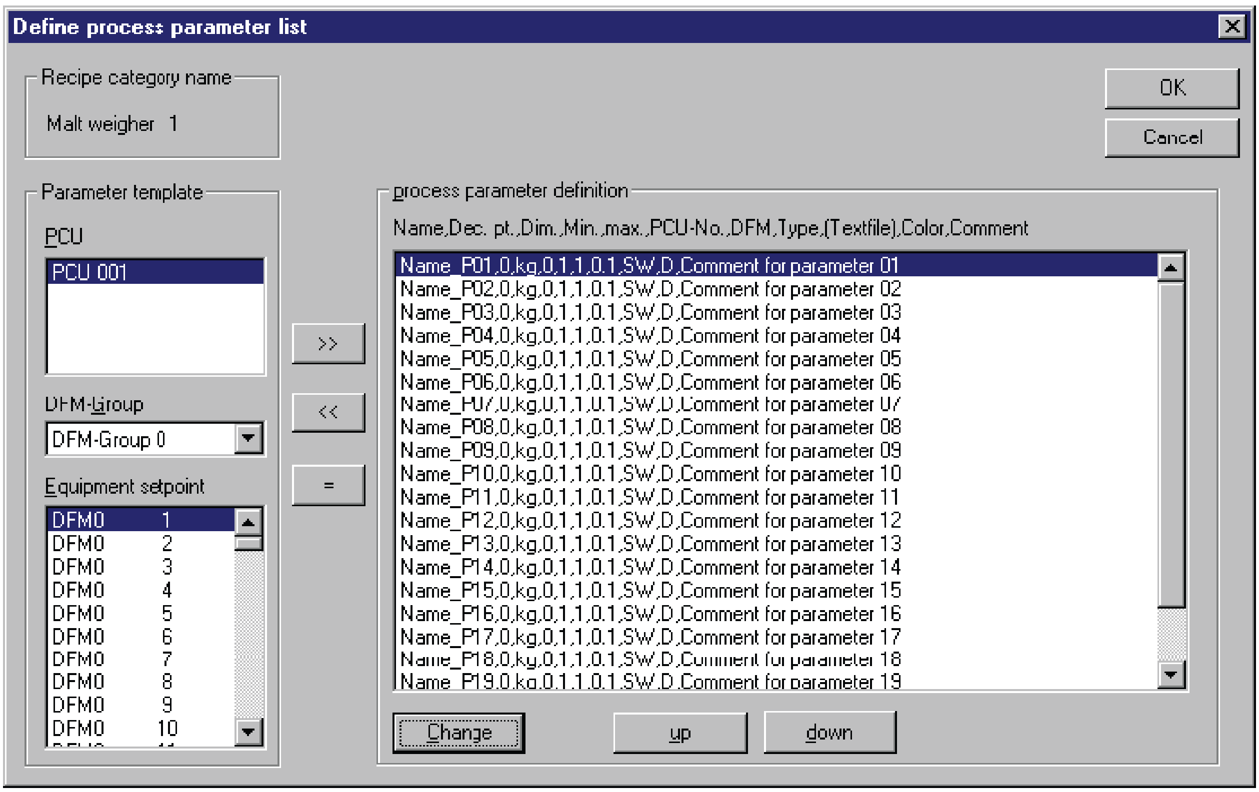


Figure 7-14 The dialog ‘Define process parameter definition'. The definition of an assigned DFM is used as a template for the process parameters type. All process parameters form the process parameter list which is displayed in the master recipe view.

Recipe category name

A process parameter list is always dependent on a recipe category. Its name is shown in the text field to the upper left.

Parameter template

As described above, the definitions of DFMs are used as type definition templates for the process parameter type. With the list and the combination field of the control element group 'DFM assignment' a DFM is determined from the different DFM groups.

Assignment button '>>'

By pressing this button the selected DFM’s definition is used as type definition for a process parameter in the process parameter list.

Delete button '<<'

Removes the selected process parameter from the list.

Replace button '='

Replaces the selected process parameter in the list with the definition of the selected DFM.

Process parameter lists definition

In this list field the defined process parameters which have already been defined are listed. With the buttons 'up' and 'down' the position of a process parameter can be changed in the list.

Change

Opens the dialog 'Edit process parameter'

Button 'Ok'

Closes the dialog and saves the changes.

Button 'Abort'

Closes the dialog without saving the changes.

Changes to the definition of the process parameter do not have any influence on the definition of the underlying DFM!

Note: If there is no process parameter list for a given recipe category or if an existing list is deleted, the system warns when opening a master recipe of this category by showing a dialog window. If the configuration does not make use of process parameters or if the proper functionality is always guaranteed, the dialog may be suppressed by putting the entry 'SuppressPPLMsg=1' into the section ‘[processparameterlist]’ in the file `'<proj-path>\sys\editrec.ini'`.

Dialog 'Edit process parameter'

This dialog is opened via the ‘Change’ button in the dialog ‘Define process parameter definition'. The dialog displays the process parameter definition, which is based on a DFM definition. Although this definition is derived from a DFM definition, as soon as it is accepted (one-off operation) there is no longer any direct connection between the process parameter and DFM. This means in particular that changes in this dialog do not affect the DFM definition.

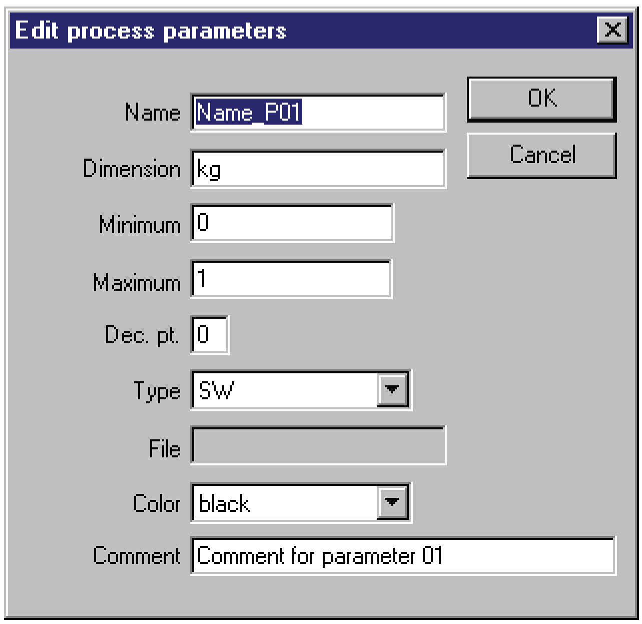


Figure 7-15 The dialog for the configuration of the process parameter definition.

The text boxes (with the exception of the "Color" text box) are responsible for determining the column values for the process parameter definition in the previous dialog.

Color

With the combination field 'color', the text color of the entire process parameter line in the process parameter list of the master recipe view can be set. The color information is coded into the comment text and is displayed in the dialog 'Configuration process parameter definition'.

Button 'Ok'

Closes the dialog and saves the changes.

Button 'Abort'

Closes the dialog without saving the changes.

7.2.5 The master recipe view.

The master recipe view is used for the display and configuration of a master recipe. It is opened e.g. via the 'Open' button or a double-click on the selected line in the recipe category overview.

The master recipe view is divided into 3 subviews:

- Master recipe header (left)
- Process parameter list (to the right above)
- Input list (to the right below)
  → The 'process input list' functionality and component dosing is deactivated in factory settings (see section Elements of the recipe system (Page 534)).

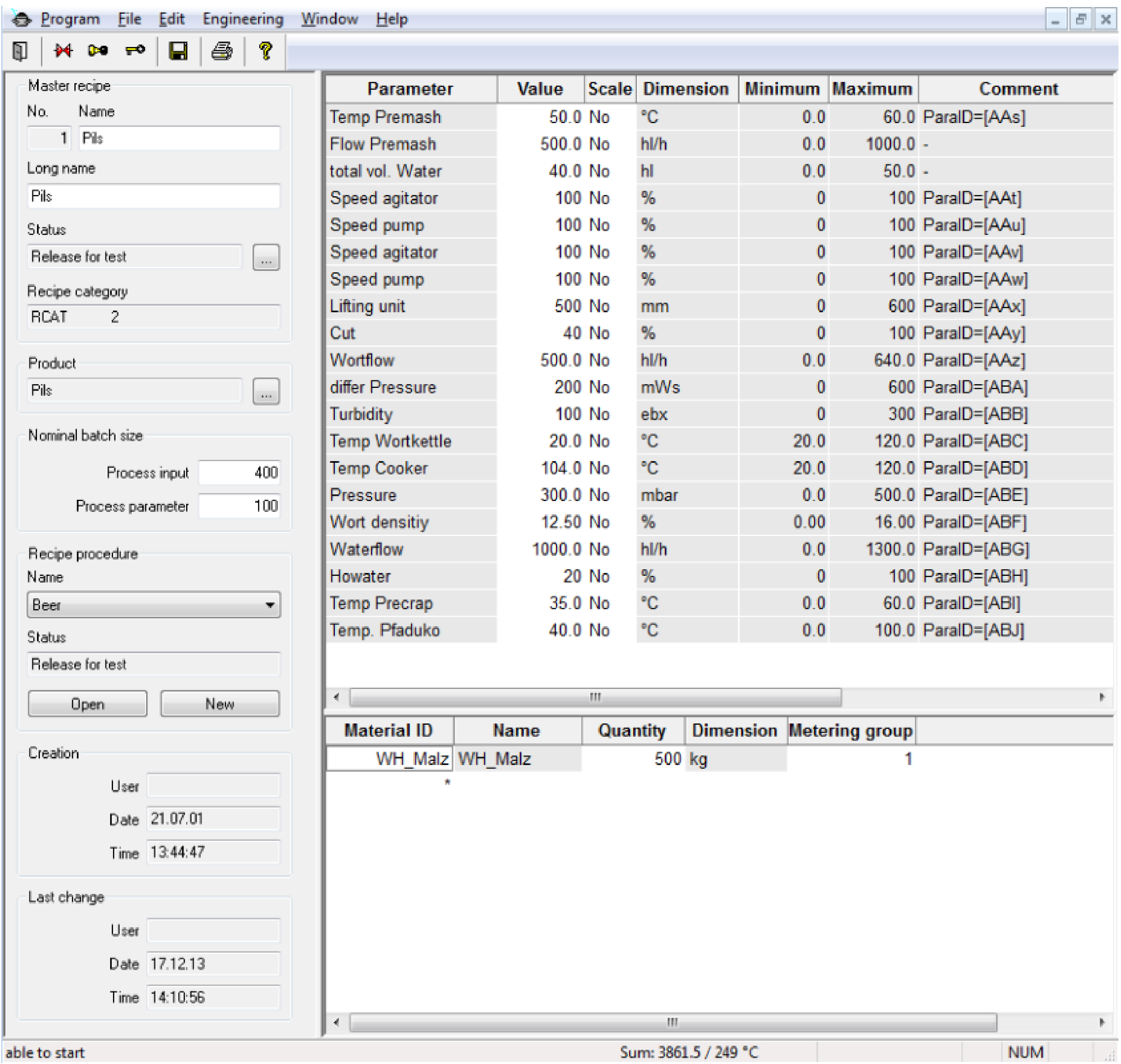


Figure 7-16 The master recipe view of the recipe editor. On the left, the information relating to the master recipe header is presented and configured. On the right above the process parameter list is located, and the process input list on the right below.

7.2.5.1 Information in the title bar

In the title bar of the recipe editor is displayed what kind of recipe object is edited ('master recipe'), to which recipe category the master recipe belongs and the name of the master recipe.

Example: [Master recipe: Cleaning Cold wort line CIP]

7.2.5.2 Information in the status bar

If the master recipe view is in the front, the status bar of the recipe editor displays the master recipe’s status and whether the master recipe is write-protected or not. Furthermore, the standard window keystate indicators are shown.

The following menu items are activated in the recipe procedure view:

File

- Preview > Process parameters
- Preview > Process input
- Print > Process parameters
- Print > Process input

Use these menu commands to print or display the print preview for the relevant subview (process parameter list or process input).

Process

If you place the cursor in the process input list, the following menu commands are available in the "Edit" menu:

- Edit > Insert line
- Edit > Delete line
- Edit > Select material

The menu commands perform exactly the same functions as the corresponding commands in the shortcut menu.

7.2.5.3 Commands of the toolbar

In addition to the standard commands for context-dependent printing and the online help system the save button is shown when in master recipe view mode:

 This button saves the entire master recipe including the process parameter list and input list.

7.2.5.4 Master recipe header

Basic principle

In the left column of the master recipe header view the so-called header data of the master recipe is displayed. These parameters can be configured in this subview. The displayed parameters are:

Master recipe name

Input fields for name and long name of the master recipe.

Recipe category

The recipe category to which the master recipe is assigned. The recipe category cannot be changed.

Product

A product can be assigned to the master recipe via the material selection list. This product is assigned to all batches created using this master recipe.

This makes it possible to implement the following functionality and behavior:

- The type "Tank" or "Tank location" can be used when defining the order parameters.
- The tank selection dialog then opens for order parameters of this type in the "Order parameters" dialog when creating an order or in the batch list.
- The list of available tanks may be pre-filtered in this dialog and then only contains tanks that contain the product assigned to the batch.
- The filter can then be modified as required.

Batch size for process input

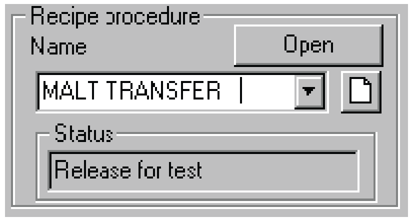
The nominal size of a batch which is produced with summed amounts of the materials in the process input list. This size can differ from the sum of the materials. When creating the order, the material quantity is scaled with the actual batch size and the nominal batch size.

When setting up a new recipe, the quantity stated in the file `'<proj-path>\recipe\project\plant.ini' [Componentlist] DefaultBatchSize=` is entered.

Recipe status

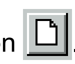
Display of the current master recipe status and button for opening the dialog ‘Release recipe’.

Recipe procedure



The master recipe stores only a reference to the assigned recipe procedure, since one RP may be assigned to several master recipes of a recipe category. The recipe procedure should be assigned either to no recipe category or to the same one as the master recipe, for the combination field only lists RPs which may be assigned to this master recipe. If the RP has not yet been assigned to a recipe category, it is assigned automatically on selection in this combination field.

The selection occurs via the combination field 'name'. The recipe procedure can be opened via the 'Open' button to the upper right.

A new recipe procedure is created with the button .

In the lower display field the current status of the recipe procedure is shown. However, the status of the recipe procedure cannot be changed in the master recipe view.

Note

If a recipe procedure is assigned, which was not assigned to any recipe category before, it is automatically selected into the recipe category of the master recipe. If the assignment is deleted from the master recipe later, the recipe procedure keeps the category assignment. One recipe procedure can only be removed from a recipe category via the recipe procedure header data dialog.

Process parameters

The process parameter list is assigned to the master recipe using the recipe category. This assignment cannot be changed. The nominal batch size of the process parameters will be compared to the sum of process parameters, if configured to do so. In order to activate the comparison, the entry 'ChkSum' in section [mainrecipe] of the configuration file '`<proj-path>\recipe\project\plant.ini`' has to be set to 1. For the deactivation of the test the value '0' is entered.

Generation and last change

In the display fields of these two groups the user identification, the date and the time of the creation or the last change is shown.

Care should be taken when editing the recipe header files to ensure that no changes are made to the structure (field names, order, field type and field size). The file cannot be read if the structure is changed.

7.2.5.5 Process parameter list

The process parameter list shows the defined process parameters in the form of a list. Only the columns 'value' and 'scaling' are editable.

For process parameters of the 'SW' file type the actual value of the process parameter will be entered in the 'value' column using the keyboard. If the process parameter is of text type, a dialog for the selection and the selected text are displayed in the 'value' field. In that case a dialog with a text list can be opened by double-clicking the field or pressing the ENTER key with the value cell selected.

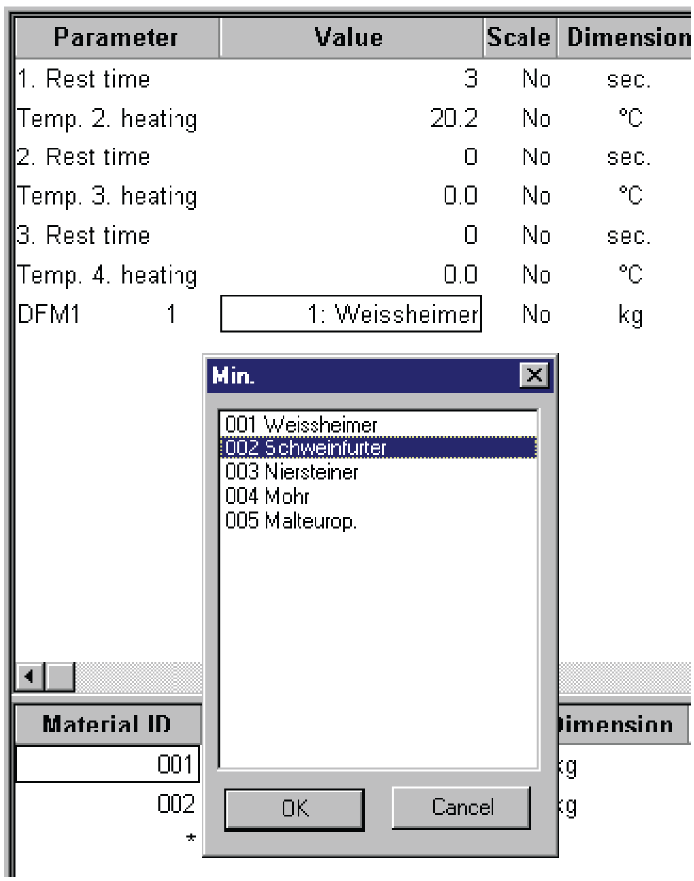


Figure 7-17 If the data type of a parameter is 'text', the value can be selected in a list view.

The limits which are indicated in the process parameter list are checked when a value is entered.

If the value of the process parameter is to be scaled with the actual and nominal batch size, the value in the column ‘Scale’ must be set to ‘Yes’. Otherwise ‘No’ deactivates scaling.

Note

The texts for the activation/deactivation of the scaling function are dependent on the selected language!

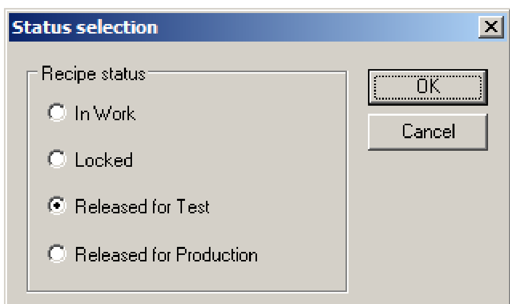
7.2.5.6 Process input list

In the input list the materials which have to be dosed in the recipe procedure are entered. Materials can be inserted via context menu entry 'Add material' or via the same menu entry in the main menu item 'Edit'. The new material is inserted into the selected row of the PIL.

If a new material is to be inserted between two existing ones the user has to insert a new line first via the context menu entry ‘Insert line’. Accordingly, a line is removed, regardless of whether it is empty or not via the menu item 'Delete line'.

See also: Elements of the recipe system (Page 534)

7.2.5.7 'Status selection' dialog box



This dialog box opens when you press the 'Modify' button of the operator control group 'Recipe status' in the submenu of the master recipe header.

Recipe status

The recipe is assigned the current status in the status group. Batch jobs can only be generated when the 'Enable Test' and 'Enable production' status flags are set. The recipe is write-protected when the 'Enable production' status flag is set.

Button 'Ok'

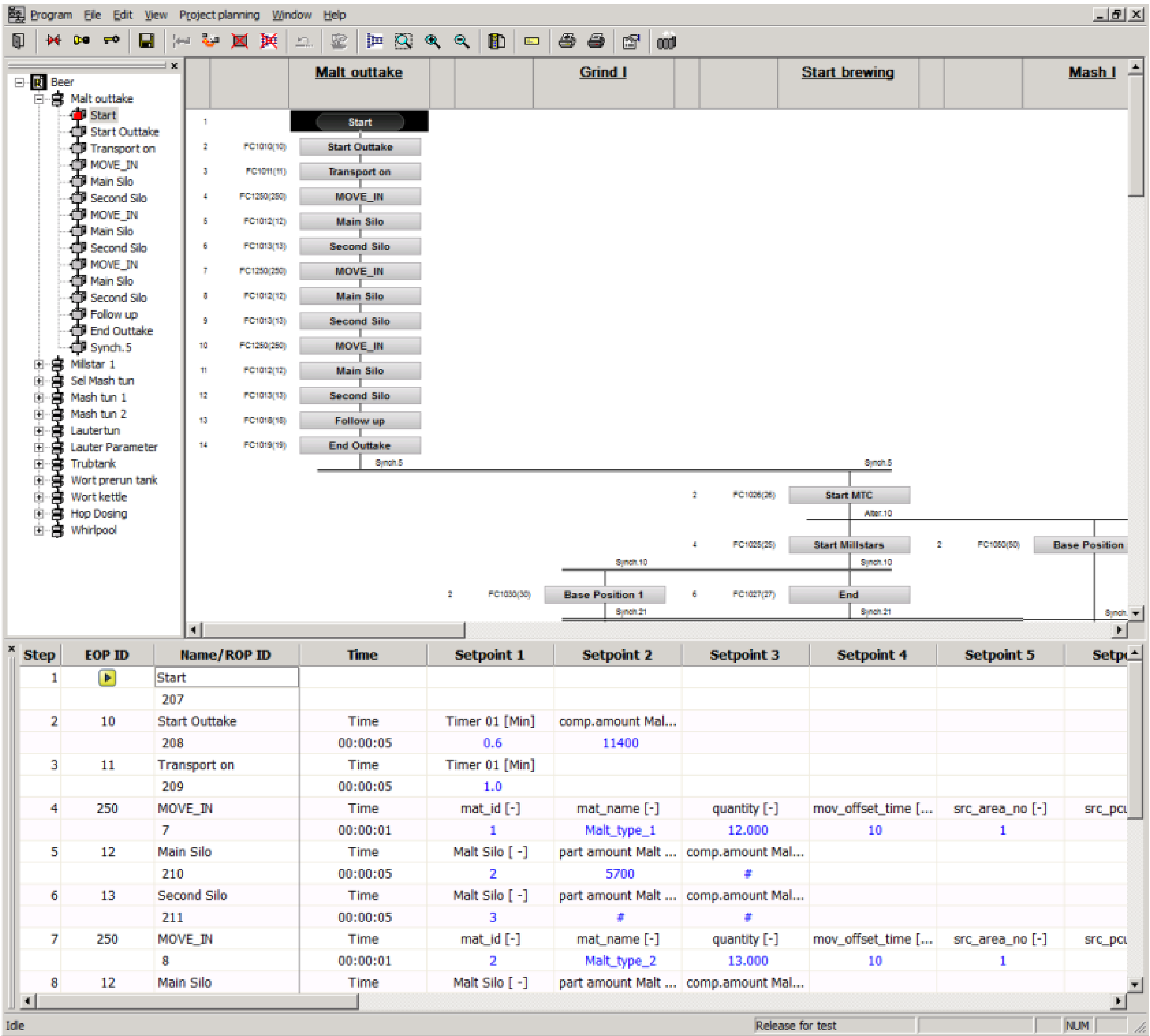
Closes the dialog and saves the changes.

Button 'Abort'

Closes the dialog without saving the changes.

7.2.6 The recipe procedure view

The recipe procedure view is opened via the menu item 'File' > 'New' > 'Recipe procedure', the menu item 'File' > 'Open' > 'Recipe procedure' or the context menu item 'Open recipe procedure‘ in the recipe category subview. It is used for displaying and configuring the recipe procedures.



Recipe procedure view.

- The subviews are freely positionable by the use of dock windows and can be hidden.
- In the figure, the recipe hierarchy can be seen at the top left and across the bottom of the RUP list. The main data area of the view is used to show and edit the graphical recipe procedure.

7.2.6.1 Information in the title bar

The title bar of the recipe editor shows which recipe object is edited ('recipe procedure'), to which recipe category this recipe procedure is assigned and the name of the recipe procedure.

Example: [Recipe procedure: Production India Pale Ale]

7.2.6.2 Information in the status bar

In the status bar of the recipe editor the recipe procedure's status is displayed and whether the recipe procedure is write-protected or not. Furthermore, the standard window keystate indicators are shown.

7.2.6.3 Menus

The following menu commands (which do not feature in the standard menu) are available in the recipe procedure view:

Menu 'File'

Print > Recipe procedure

Prints the graphical recipe procedure either

- in the current size,
- on a single page
- as posters with adjustable page layout.

After selecting print mode, the page partition can be defined in a subsequent dialog when 'Poster printing' is selected.

Printing > Recipe unit procedure list(s)

Opens the dialog to print the RUP list. Either the current recipe unit procedure or all recipe unit procedures as a list may be selected. The format of the printout is text only.

See also

Dialogs of the recipe procedure (Page 611)

Menu 'Edit'

Cut

In the recipe unit procedure, the recipe operation at the current cursor position is copied to the buffer and deleted from its original location.

- This function is only available for "normal" recipe operations and NOPs (no synchronizations, alternatives, jumps, jump destinations)

Copy

In the recipe unit procedure, the recipe operation at the current cursor position is copied to the buffer.

- This function is only available for "normal" recipe operations and NOPs (no synchronizations, alternatives, jumps, jump destinations)

Insert

The last recipe operation to be copied to the buffer is inserted just after the current cursor position.

- This function is only available for "normal" recipe operations and NOPs (no synchronizations, alternatives, jumps, jump destinations)

Note

Insertion of copied/cut NOPs or GOPs

The software does not support the insertion of NOPs or GOPs directly before a recipe jump.

Insert new ROP

In the recipe unit procedure a new recipe operation is inserted above the current cursor position. The recipe operation property dialog is displayed to select the type of ROP. This button is deactivated if the cursor is in the recipe procedure overview.

The inserting operation can be cancelled.

Append new ROP

The new ROP is appended behind the last operation present, even if this is the End ROP. The recipe operation property dialog is displayed to select the type of ROP. This button is deactivated if the cursor is in the recipe procedure overview.

The appending operation can be cancelled.

Delete ROP

Deletes the currently selected ROP. The user must confirm the delete operation. This button is deactivated if the cursor is in the recipe procedure overview.

The delete operation can be cancelled.

Scale setpoint

Opens the dialog 'Scales'. This menu item is only active if the cursor is in the recipe unit procedure list in a setpoint column.

Substitute setpoint

Opens the dialog 'Substitute setpoint'. This menu item is only active if the cursor is in the recipe unit procedure list in a setpoint column.

Update

Synchronizes the equipment data parameters and references to this data from inside the recipe procedure. This function is executed automatically after many operations on the equipment data.

Change RP state

Opens the dialog 'Release recipe procedure' to change the state of a recipe procedure after several syntactical checks.

RP properties > Header

Opens the 'Recipe procedure header' dialog (Page 565).

Properties > RP Properties

Opens the 'RP Properties' dialog (Page 567).

Menu 'Project planning'

Edit EOP definition

Opens the dialog 'Configure equipment operation' (Dialog ‘Configure equipment operation' (Page 577))

Cross Reference List

Creates a cross reference list of the EOPs and DFMs used in this recipe procedure.

Edit setpoint definition

Opens the 'Edit DFM definitions' dialog. ('Edit DFM definitions' dialog (Page 581)).

Material management

- Materials and groups → see Material and material groups (Page 585)
- Material sequences → see Material assignment (Page 586)

Components

Opens the 'Material master' dialog.

Storage stock

Opens the 'Storage location' dialog.

Equipment data

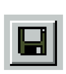
Opens the equipment data dialog ('Equipment data' (Page 574)).

7.2.6.4 Commands of the toolbar

Basic principles

In addition to the standard commands for the context-dependent printing and the online help system the following buttons are available in the toolbar of the recipe procedure view:

Save



Saves the recipe procedure to hard disk or a configured network storage location. This button is always active in the recipe procedure view.

Insert new ROP

In the recipe unit procedure a new recipe operation is inserted above the current cursor position. The recipe operation property dialog is displayed to select the type of ROP. This button is deactivated if the cursor is in the recipe procedure overview.

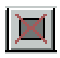
The inserting operation can be cancelled.

Append new ROP

The new ROP is appended behind the last operation present, even if this is the End ROP. The recipe operation property dialog is displayed to select the type of ROP. This button is deactivated if the cursor is in the recipe procedure overview.

The appending operation can be cancelled.

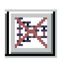
Delete ROP



Deletes the currently selected ROP. The user must confirm the delete operation. This button is deactivated if the cursor is in the recipe procedure overview.

The delete operation can be cancelled.

Delete recipe procedure



Deletes all recipe operation from the recipe procedure. Instead a Start and an End ROP are inserted into the first RUP. The recipe procedure structure as defined in the RP properties dialog stays untouched.

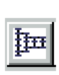
This deletion operation cannot be cancelled. It has to be confirmed in a message box by the user.

Undo

Undoes the following operation in reverse order of its execution (event horizon is the last storage command):

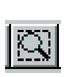
- Insert new ROP,
- Append new ROP,
- Delete ROP.

Show all



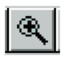
Adjusts the zoom so that the whole graphical recipe procedure is displayed at once.

Free zooming



It enables the user to freely select the zoom adjustment. The zoomed area is selected by left-clicking and dragging the mouse pointer. The selected rectangle is zoomed in on.

Zoom in



Increases the zoom by one step.

Zoom out

Reduces the Zoom by one step.

Tooltip display on/off

Shows or hides the tooltips

Print RP

Prints the graphical recipe procedure either

- in the current size,
- completely on 1 page,
- as posters with adjustable page layout.

If the printing mode ‘poster printing’ is selected, the button for the size selection of the printout is activated.

Print recipe unit procedure list

Opens the dialog ‘Print RUP'. In the dialog the user has the choice between a single RUP or all RUPs of an RP that are printed as text-only lists.

7.2.6.5 Subviews

Navigation with subviews

The partial views of the recipe procedure view show different aspects of an open RP. The movement of the cursor in one subview also moves the cursor in all other views, to ensure that they always display the same, current, part of the RP.

Moving, showing and hiding subviews

The partial views 'Overview', 'Recipe hierarchy' and 'RUP list' are implemented as so-called ‘docking windows’. They may be ‘floating’ over the graphical recipe procedure view or may be docked or snapped to one edge of the main view window. By double-clicking on the handle bar of a docked window or on the title bar of a floating window the state is toggled. The last position in both states is stored.

In the menu 'Window' the partial views of the recipe procedure view can be shown or hidden by clicking on the subview entry. Visible subviews are marked in the menu.

7.2.6.6 Graphical recipe procedure view

The graphical recipe procedure fills the background of the recipe procedure view. It is not a dock window. Therefore it cannot be turned off.

In this subview the recipe unit procedures are indicated in columns next to each other. The graphic representation is used especially for inserting recipe unit procedure crossing elements, e.g. synchronizations or alternatives.

Figure 7-18 Function header of a recipe unit procedure column in the RP view of the recipe editor

The multifunction header of each column shows the following information:

1. Row:	Name of the recipe unit procedure
2. Row:	Name of the unit
3. Row:	Name of the sequence which is assigned to the unit - PCU no./sequence number

If the mouse pointer is moved over the right edge of a column function header the pointer is transformed into a double arrow. By pressing the left mouse button and simultaneously moving the mouse horizontally, the width of the column can be changed. By one click with the right mouse button the width of column is minimized. One further right click extends the column again to the original width.

In the columns the sequence of the operations is configured. With the commands 'Insert', ‘Append’, 'Delete', and 'Properties' the recipe operations are arranged.

The setpoints, monitoring times and the names of the recipe operation are displayed in a tooltip window if the pointer hovers over an ROP for a short time without moving.

7.2.6.7 Recipe procedure overview

The recipe procedure overview represents the entire recipe procedure in the survey. The part of the graphical recipe procedure that is displayed in the main area of the RP view is framed by a dashed rectangle. The top left corner of the window can be moved with the cursor.

7.2.6.8 Recipe procedure hierarchy

This tree view is used for the simple navigation in a complex recipe procedure with many recipe unit procedure columns. A mouse click on the plus symbol of a recipe unit procedure expands its recipe operations as a branch in the tree view. If a recipe operation or a recipe unit procedure is selected in this view, the other subviews move their viewport so the selected object will become visible.

7.2.6.9 Recipe unit procedure list

In this subview the recipe operations of a recipe unit procedure are represented as a table. Apart from the name, the step number and the number of the equipment operation, the monitoring time and the max. 24 (max. 20 with PCU <V7) possible setpoints of a recipe operation are displayed.

- If the setpoints are configured as editable, they can be changed in the list view. The monitoring time is set in this view, too.
- Entering a hash character ("#") prevents the DFM setpoint from being affected during this step (setpoint of a previous line remains valid) and ensures that the actual value is recorded in the step protocol.
- This means that a hash ("#") can be used to record an external value in the step protocol using a DFM as well.
- As in the graphical recipe procedure view the widths of columns can be adjusted with the mouse.

New ROP ID in recipe procedures

For each step, the so-called "ROP ID" is displayed below the EOP name.

- The ROP ID is used as unique ID of an ROP within the recipe procedure in the sense of a "technological step".
- With multiple instances of an EOP in the recipe procedure, the ROP ID distinguishes them from one another.
- The generation of the ROP ID takes place automatically in the recipe editor and cannot be influenced by the user. It is displayed in the setpoint list.
- When batches are running, the ROP ID is also applied to the step logs and saved.

Application case:

- When the "SQL adapter" option is used, the ROP ID is also transferred to the step protocol archives of the SQL database.
- This means the step log data of individual ROPs can be clearly identified as technological steps.
- In this way, it is easier to extract so-called KOPs (key operating parameters) in the reports of the MIS/ERP application.

Note

Behavior after update of existing plants

- Missing or non-unique ROP IDs are detected when recipe procedures are opened.
- If the recipe procedure is not write-protected (status <> "Release for production"), this status is automatically corrected and signaled by a dialog box (RP in change status). The newly assigned ROP IDs become effective with the first "save".
- There is no check and change with released RPs, because they are write-protected and cannot be changed.
- **To use the ROP-ID in connection with the SQL adapter, all RPs should be opened once and set to the status <> "Release for production", if this is not already done. Missing/invalid ROP-IDs are detected in the process and changed with "Save".**

7.2.6.10 Dialogs of the recipe procedure

Basics

The dialogs of the recipe procedure view are explained below. These contain the following common elements:

'Delete' button

The 'Delete' button disables the scaling for the current setpoint.

'OK' button

Closes the dialog and saves the changes.

'Cancel' button

Closes the dialog without saving the changes.

'Scales' dialog

This dialog can be opened using the 'Edit' or the shortcut menu if the cursor is located on a setpoint in the recipe unit procedure list.

Scales

In the Scales list, you can choose between linear scaling and linear scaling with a 10% offset.

'Substitute setpoint' dialog

This dialog can be opened using the 'Edit' or the shortcut menu if the cursor is located on a setpoint in the recipe unit procedure list.

Order system

- With the two radio buttons of this group, an order parameter can be selected as the source for setpoint substitution.
- With this order parameter, the entry of the value in the recipe unit procedure list can be completely replaced ('Replace with parameter') or multiplied by the value ('Multiply with parameter') and transferred to the control recipe.
- Which order parameter is substituted is selected in the list operator control element of the dialog.

Process parameter

The value in the recipe unit procedure list is replaced in the master recipe by the value of a process parameter from the process parameter list.

'Select status' dialog

This dialog is opened after pressing the 'Change" button in the 'Recipe status' group in the sub view of the master recipe header.

Recipe status

- In the status group, the recipe procedure is assigned the current status 'In work', 'Disabled', 'Released for test' or 'Released for production'.
- Batch orders can only be generated in the 'Released for test' and 'Released for production' statuses.
- In the 'Released for production' status, the recipe procedure is write-protected.

Note

A status change from 'Released for production' to other statuses (e.g. 'In Work' for editing the recipe procedure) is only allowed after all batches running with the particular recipe procedure are completed. Otherwise batches still running could 'hang' if , for example, synchronizations or alternatives are changed in the RP.

'Print recipe procedure' dialog

This dialog is activated using the 'File' or the toolbar button for the recipe procedure printout.

Actual size

The picture section is selected as in the graphic recipe procedure partial view.

Whole recipe procedure on one page

Reduces the zoom factor so that the entire recipe procedure can be printed out on one page.

Poster printing

Prints the entire recipe procedure as a poster on several pages. The number of pages can be varied in the 'Page Partition" dialog.

'Page Partition' dialog

This dialog is opened in 'Poster printing' mode with the 'Page partition' button

Page partition

- With the 'greater' and 'less' button, the page partition can be adapted.
- The partitioning is not freely selectable but depends on the height-width ratio of the graphic representation of the recipe procedure.
- By selecting the 'Print preview" check box, and overview window opens in which the position of the page borders is displayed.

'Print' dialog

This dialog is activated with the 'File' > 'Print' > RUP list' or the toolbar button for printing recipe unit procedure lists.

Selection

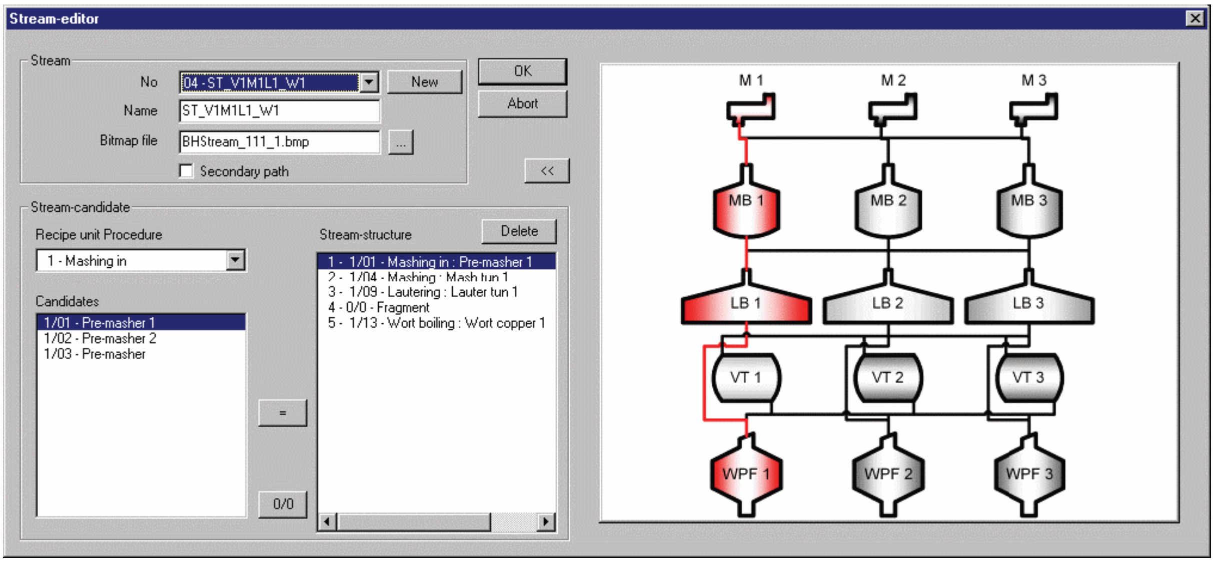
The operator can decide whether only the active recipe unit procedure is printed as a text list or whether all recipe unit procedures of the recipe procedure should be printed out one after the other.

'Setup' button

Opens the standard Windows© dialog for setting up the printer.

'Stream editor' dialog

The stream editor dialog is opened from the RP properties dialog. It is activated only for recipe procedures with stream selection. The candidates of a recipe unit procedure which form the streams need to be assigned in the RP properties dialog of the recipe unit procedure before using the editor. They are then available in the "Candidates" list box.



By selecting a recipe unit procedure from the ‘Recipe unit procedure‘ combo box, the candidates are shown in the list box. Left-click the candidate to select it, and then select it for this stream by clicking the '=' button. It now appears in the "Stream structure" list box, in the line of the corresponding recipe unit procedure.

By left-clicking a recipe unit procedure in the 'Stream-structure' list box the selection in the 'Recipe unit Procedure' list box is updated and the 'Candidates' list box is refilled.

Elements of the 'Stream' control element group

No.	From this combo box you select the stream to be processed. The shown name is automatically taken from the "Name" input box. A new stream and the reference candidate are generated automatically, and a default name is assigned to the stream when you click "New".
Name	Name of the stream.
Secondary path	By selecting this check box, you assign a secondary stream that is only enabled at runtime, and does not contain a start unit or start step.
Bitmap file	Name of the bitmap file that belongs to this stream. The bitmap must be created with a tool such as Windows "Paint", and it should present a schematic view of the stream to the operator. The storage path for bitmap files is '<proj-path>\recipe\project\streams'. For the bitmap, we recommend a size of 384x384 pixels (width x height). Larger formats are scaled prior to display. This may lead to distortion and loss of picture information. With smaller formats the picture is increased in size pixel by pixel.

Elements of the 'Stream candidate' control element group

Recipe unit procedure	Combo box for selecting the recipe unit procedure. The candidate assignment in both list boxes of this group always relate to the selected RUP.
Candidates	List of all candidate units defined in the RP properties dialog.
Stream structure	List of all recipe unit procedures, including the assigned candidates for this stream. The text line specifies the recipe unit procedure, the PCU number and unit number of the selected candidate, the name of the RUP and the name of the unit.
Assignment button '='	Assigns the selected candidate to the recipe unit procedure.
Fragment button '0/0'	Assigns no candidate or the 'empty' unit 0/0 to the recipe unit procedure. This assignment is used for secondary streams or for units not used for all streams, e.g. an optional auxiliary device of a unit that does not exist in other candidates. For this reason, a stream is not automatically identified as secondary stream when the fragment button is used.
Expand button '>>' or '<<'	Hides or shows the selected bitmap.

7.2.6.11 ‘Procedure Parameters‘ dialog box

Basics

This dialog box is opened by means of the shortcut menu of a setpoint in the RUP lists of the RP view. RP parameters can only be used in stream recipe procedures with stream changeover function.

The minimized dialog box shows only a selection dialog. The extended dialog is opened when you click 'Edit Parameters', or automatically if there are no procedure parameters available yet.

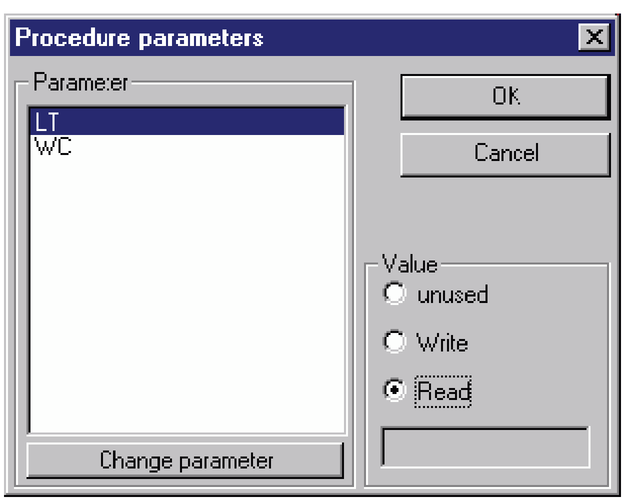


Figure 7-19 View of the minimized RP parameter dialog box for selecting an RP parameter, its use (read, write, unused) and in case of write access.

‘Parameters‘ list

Selection list for predefined parameters.

‘Value‘ control element group

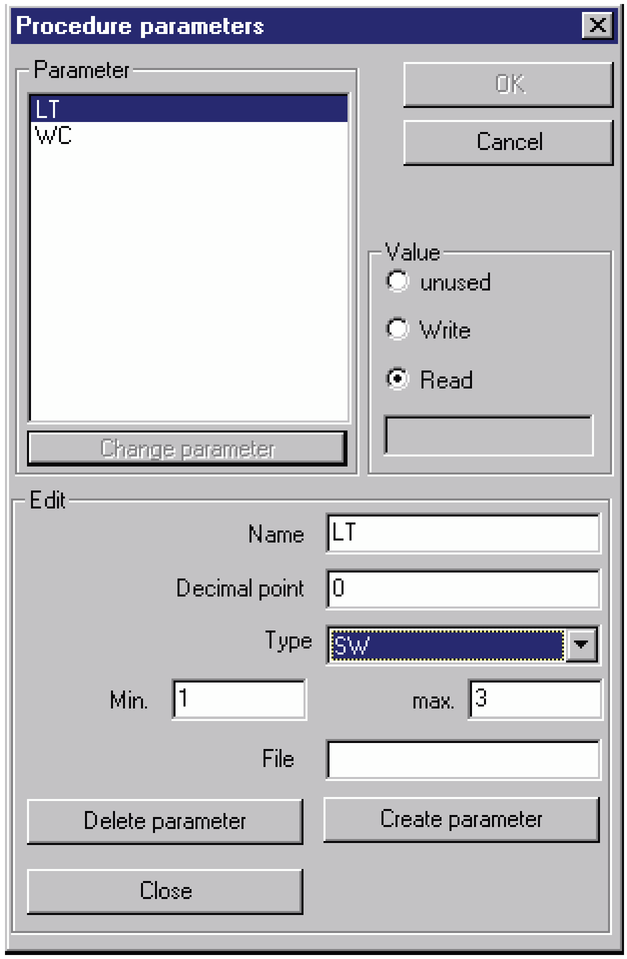
Element for defining the use of the parameter: read/write access to parameter data, or setting the ‘unused‘ attribute for deleting the substitution. When you select 'write' access, a parameter input box opens. If the parameter represents a text list index, the entries of the text list are made available in a combo box.

Application example:

- In the predecessor RUP, the WPF ID in the "WPF" parameter is configured as value = "1" / write
- This WPF ID can then be read in the current RUP.
- The procedure parameters are labeled with the abbreviations "W" for write and "R" for read in the recipe parameter list.

‘Change parameter‘ button

Opens the extended dialog for the definition of new RP parameters or to change properties of an existing parameter definition.



Control group ‘Edit‘

'Name'	Name of the RP parameter
‘Decimal point‘	Number of decimal places
‘Type‘	The parameter type: either ‘SW‘ for fixed-point numbers or ‘TEXT‘ for text list indices.
‘Min.‘ and ‘Max.‘	Minimum or maximum parameter value.
'File'	Text list file name without *.txt extension, if the parameter is of the ‘TEXT‘ type. The file must be stored in the recipe directory, in the folder `'<proj-path>\texte.x'` (x = LANGUAGE NUMBER)
‘Delete Parameter‘	Deletes a procedure parameter from the list.
‘New Parameter‘	Adds a new parameter with currently set declarations to the list.
'Close'	Closes the extended dialog view, and enables the ‘OK‘ button of the dialog.

7.3 Use Cases

7.3.1 Creating a simple recipe

This use case describes the procedure for creating a simple recipe with the recipe editor. The necessary equipment operations and DFMs must already exist in the PCUs.

As a simple example, used cleaning caustic should be drained off.

Table 7-3 Elements required:

Object	No./instance	Description
Unit	50	Caustic tank 1
Sequence	50	Drain caustic solution
Recipe operation	500	Base position
Recipe operation	501	Drain
Recipe operation	502	End position
DFM	1.50	Dribble time
Recipe operation	1	Drain caustic solution
Recipe category	1	

This is what the complete recipe should look like:

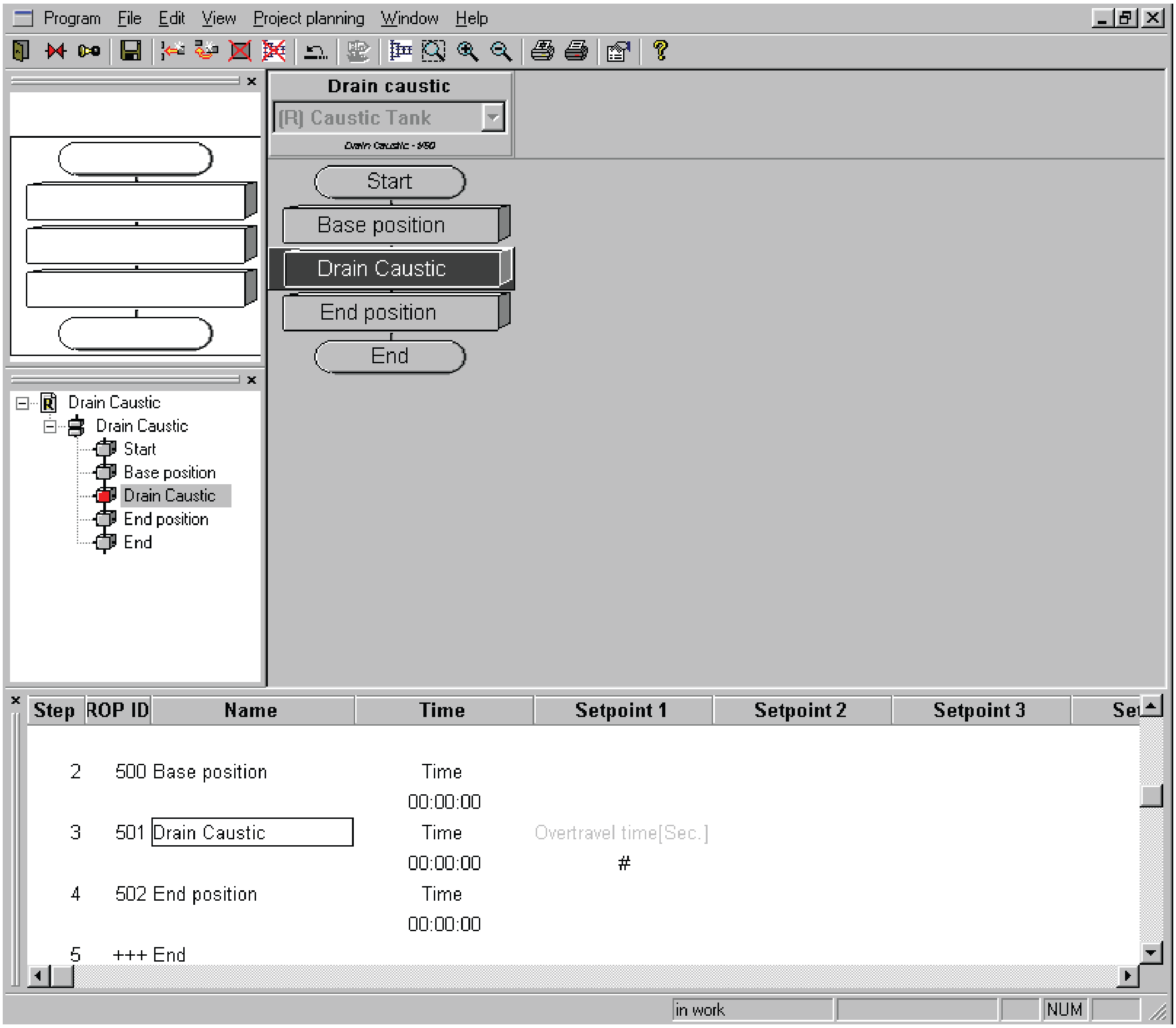


Figure 7-20 Recipe after all steps have been programmed

7.3.1.1 Configuration of the equipment data

Basics

For the example recipe a unit is required with the corresponding sequence and 3 recipe operations, which are defined via the dialog 'Equipment data' (Menu item 'Configuration' > ‘Equipment data' in the recipe overview).

The units are typically arranged into bands (e.g. unit 1 - 20 production units of a system, units 20 - 30 production units of the next system, etc.). In this example the unit 50 of the PCU 1 is used. The units 50 to 60 are supposed to be used for the CIP (cleaning in process) system. The configuration of the unit comprises only the allocation of the names, here ‘Caustic tank 1’, the corresponding sequence is supposed to be called ‘Caustic drain’.

Starting the recipe editor

- Main menu/Process monitoring tab
- Double-click recipes

Editing plant data (unit)

- Configuration/equipment data menu
- Click the symbol next to the device tree
- Click the symbol next to the PCU
- Click the symbol next to the unit

- Right-click 'UNIT 50' and select 'Unit Properties'
  Note: The name is stored in the <proj-path>\pcu.xxx\texte.0\unit.txt file.

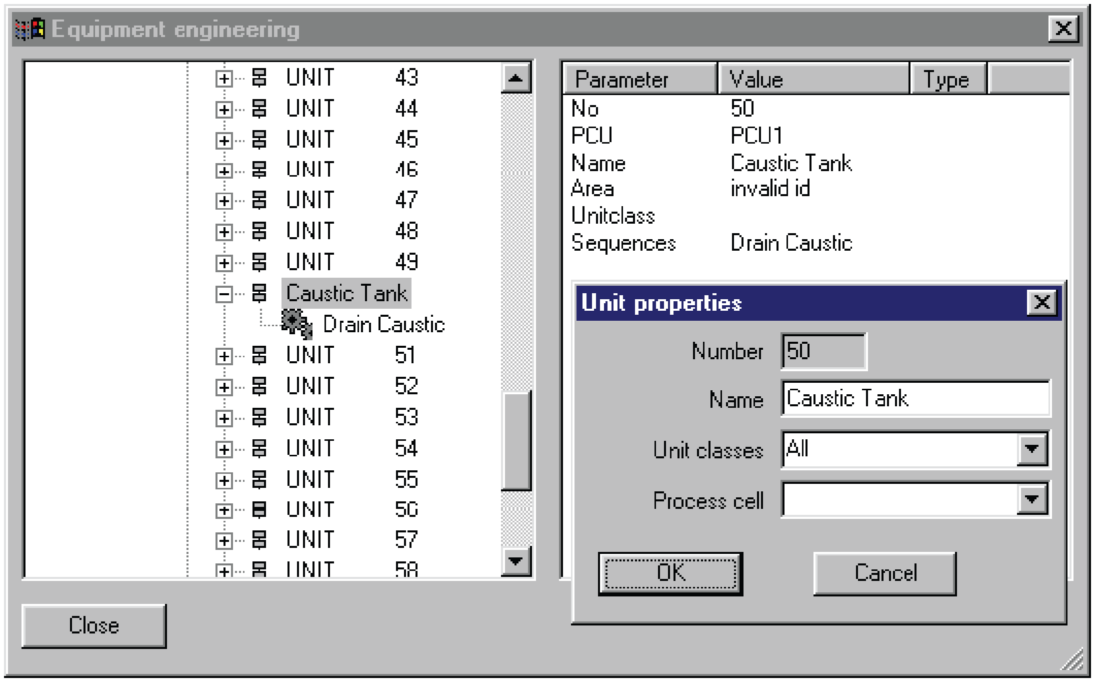


Figure 7-21 Example: Input of a unit name in the equipment engineering dialog box

- Click the sequence name. Right-click to open the properties dialog where you can change the name.
  Note: The name is stored in the <proj-path>\pcu.xxx\texte.0\sequence.txt file.

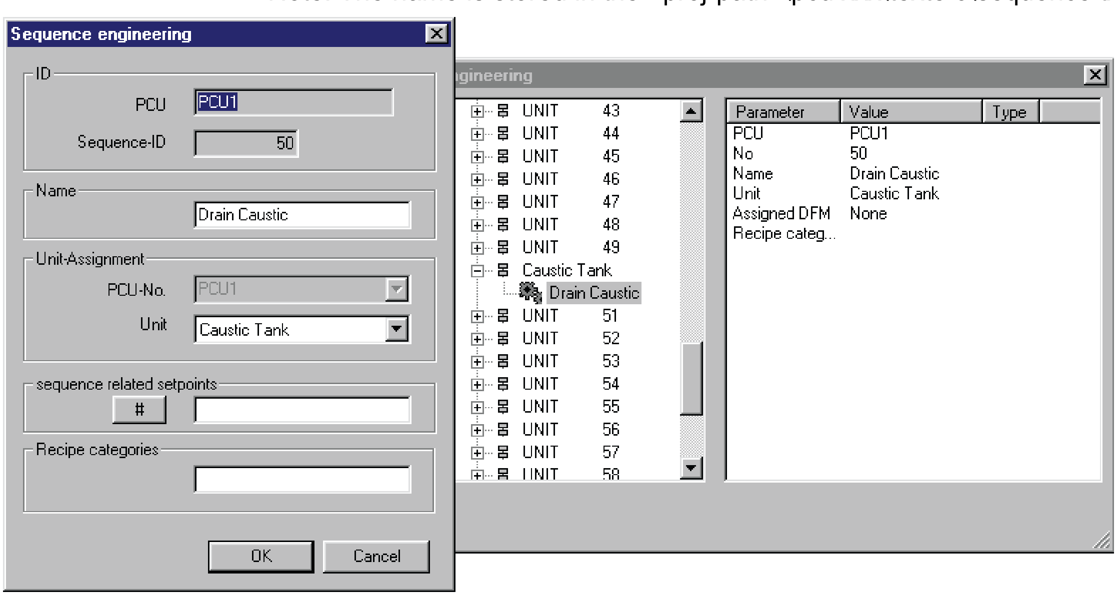


Figure 7-22 Example: Input of a sequence name by means of the equipment engineering dialog box

You can also enter the name as a text-based parameter (main menu/Engineering tools tab/ Text parametrization)

- Please note that changes made to the name via the equipment engineering dialog can be implemented immediately, while name changes carried out via text-based parameters require a system restart.

Entry of the recipe operations

Furthermore, 3 recipe operations are needed:

1. The basic setting for setting the valves
2. the actual draining
3. and the final position of the valves.

As setpoint only in the step ‘drain caustic' an overtravel time is used. For the rest of the steps the monitoring time is sufficient.

Editing system data (recipe operations)

- Menu/Configuration/System data
- Click the symbol next to the device tree
- Click the symbol next to the PCU
- Click the symbol next to Operations/functions
- Click the EPE name PCU.
  Right-click to open the properties dialog where you can change the name.

Note

Note: The name is stored in the <proj-path>\pcu.xxx\texte.0\epe.txt file You can also enter the name as a text-based parameter (main menu/Engineering tools tab/Text parametrization)

DFMs are assigned to a sequence with a corresponding entry in <proj-path>\pcu.xxx\recipe \epe.ini.

In addition to the name, you can assign up to 24 technical parameters (DFMs). The recipe operation can also be assigned to one, to multiple or to all sequences.

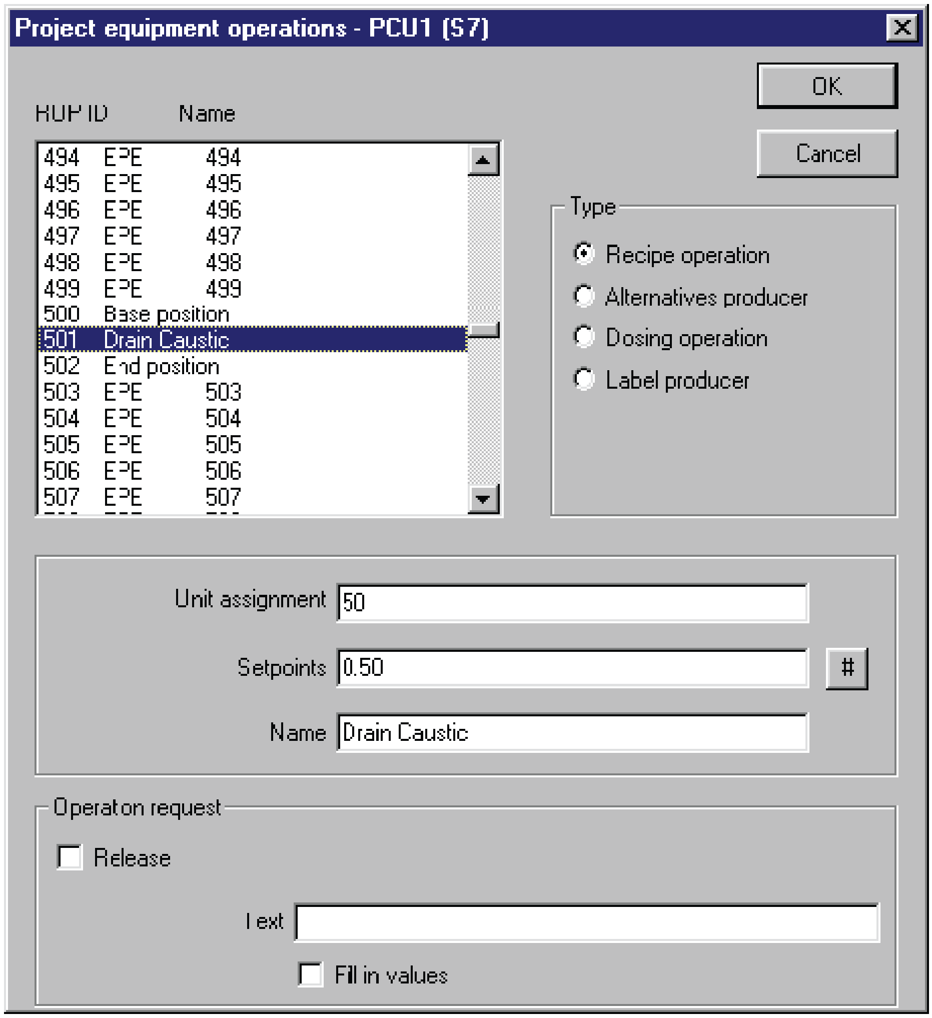


Figure 7-23 Input of data for a recipe operation

The recipe operations 500, 501 and 502 of the PCU 1 are renamed in the EOP definition dialog. They are renamed as 'Base position', 'Drain caustic' and 'End position'. The operation 'Drain caustic' receives the setpoint 0.50.

Editing technical parameters

DFM 50 is to be assigned the name "Lag time" and the unit in seconds.

- Configuration/equipment data menu
- Click the symbol next to the device tree
- Click the symbol next to the PCU
- Click the symbol next to the module
- Click the symbol next to the parameter elements

- Click the symbol next to the DFM group 0
- Click the name of DFM 0.50. Right-click to open the properties dialog.

Note

The name is stored in the <proj-path>\pcu.xxx\texte.0\dfm0.txt file. You can also enter the name as a text-based parameter (main menu/Engineering tools tab/Text parametrization)

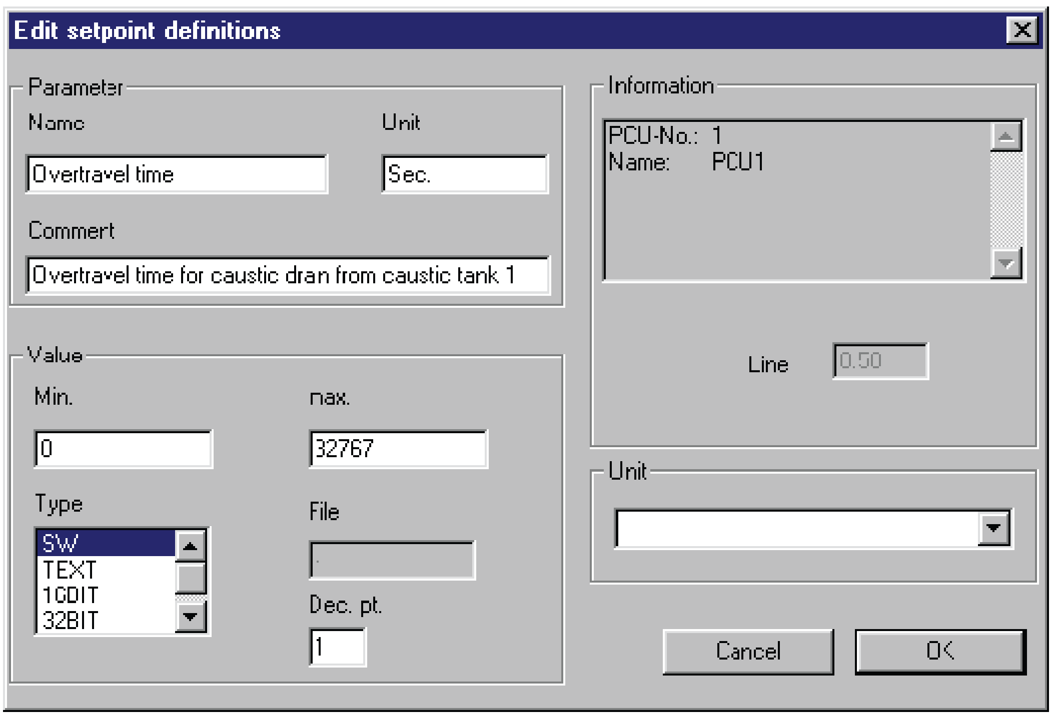


Figure 7-24 Input of the DFM name

7.3.1.2 Create a new recipe procedure

Basics

The caustic drain is configured as a simple recipe procedure without any parallel lines.

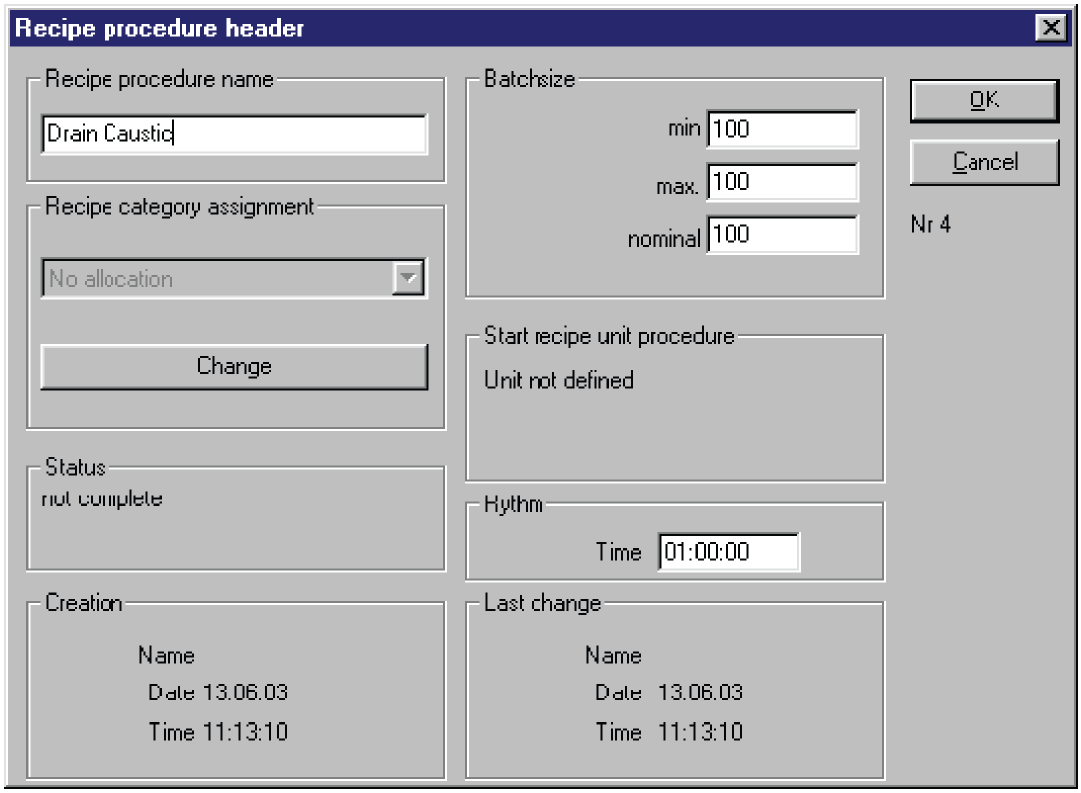
Starting the recipe editor

- Double-click recipes
- File/New/Recipe procedure menu command

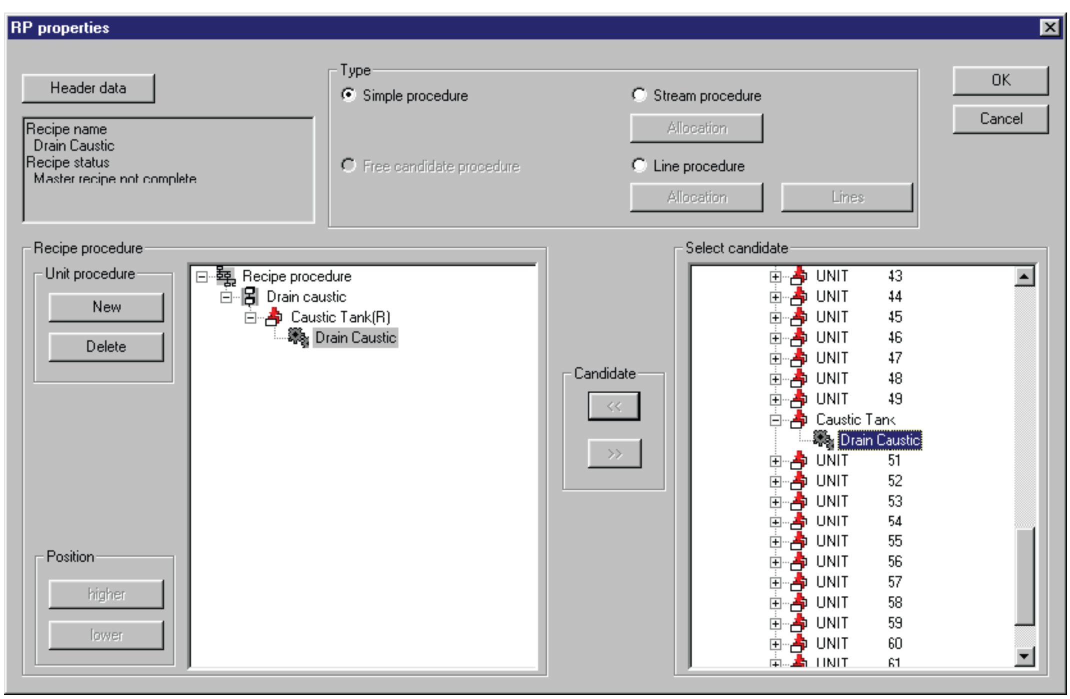
Creating a recipe procedure

First the dialog is opened for the recipe procedure header data.

As the draining of the caustic is monitored via level switch sensor, no entries are needed for batch size and rhythm, only the name is set.



After having closed the dialog with the OK button the necessary equipment data are read in and the RP property dialog opens.



The new recipe procedure has only 1 recipe unit procedure. The recipe unit procedure is created with the 'New' button and named in the 'New RUP' dialog. The unit 50 of the PCU 1 is assigned to the recipe unit procedure.

- Click the "New" button.
  Enter the name of the unit procedure. "Caustic drain"
- + Click the symbol next to the project
- + Click the symbol next to the device tree
- + Click the symbol next to the PCU
- Click the text at UNIT 50 (caustic tank)
  You can assign the unit to the recipe unit procedure by clicking the candidate button.
- End the dialog with OK

The dialog is closed and the recipe procedure view opens after syntactical checks to facilitate the configuration of operations. Because a new recipe procedure contains only an NOP step (No Operation), the following error messages can be acknowledged with 'OK'.

Creating steps of the recipe procedure

- Right-click NOP step
- Click properties.
  The Start type is selected in the dialog box.
- Confirm with OK
- Select Menu/Edit/Append new ROP.
  The "Equipment operation" is preselected in the dialog box. You can now select the second step "Basic position", and then save the selection with OK. Now enter the step monitoring time and the lag time.
- Select Menu/Edit/Append new ROP.
  The "Equipment operation" is preselected in the dialog box. You can now select the second step "Drain", and then save the selection with OK. Now enter the step monitoring time and the lag time.
- Select Menu/Edit/Append new ROP.
  The "Equipment operation" is preselected in the dialog box. You can now select the second step "Final position", and then save the selection with OK. Now enter the step monitoring time and the lag time.
- Select Menu/Edit/Append new ROP.
  The End type is selected in the dialog box.
  Save the selection with OK

- Select Menu/File/Save to save the recipe procedure.

Note

To be able to edit and modify a recipe procedure, the editing mode or 'Enable Test' mode must be set. To enable the recipe procedure, you need to set 'Enable Test' or 'Enable Production'.

- Menu/Edit/Change RP state.
  Select "In work" from the recipe status dialog box.

7.3.1.3 Assignment of the recipe procedure to a master recipe

Creating a recipe category

In order to start an order for draining of the caustic tank 1, the recipe procedure must be assigned to a master recipe. Before the new master recipe is created a new recipe category ‘CIP’ is created.

Starting the recipe editor

- Double-click recipes
- Right-click Area1 (left side in the recipe structure view).
  Select the "New recipe category" command from the shortcut menu. You can now assign the name "Cleaning" to the recipe category.
- Click OK to create the recipe category.

Creating a master recipe in the recipe category

- Right-click "Cleaning" and "New master recipe". Process parameters are not required, which means that you can close the message window with "Yes".
- Editing the master recipe dialog:
  - Master recipe name = "Drain Caustic"
  - Long name = "Drain Caustic"
  - Batch size/nominal = 100
  - Assign the recipe procedure "Drain caustic tank" to the master recipe.
- Select the File/Save command

7.3.1.4 Further information

The following settings are required for you to be able to work with "Recipe 50".

PCU software

In the controller (PCU1 - S7 controller) the corresponding unit blocks must exist.

Unit 50	FB 1050
	A flag is set in FB 1050 to enable/disable runtime of the unit. The flag used for "Unit 50" is 646.1, or the symbolic flag "Teilanl50-DB". A "RLO1" = "Teilanl50-DB"
	"Unit 50" must be edited here, because it uses DFM 1.50.
	The factory version of block FB 1050 can be used for this example.
	Modify the block as follows: Network 2 Action before EOP is running A M 101.4 JC AFTE CALL "DFM1_FC" IDfm:=50 BEU
EOP 500	FC 1500
	Only the step monitoring time needs to be defined for the basic position step. When this time has expired, the block must return RLO=1 to end the step.
	The step monitoring time is polled with the flag 101.0.
	For this example, you can copy the factory version of block FC 1001 to FC 1500, and use it without any modification.
EOP 501	FC 1501
	In addition to the step monitoring time, the "Drain" step also requires a lag time (DFM 1.50). When this time has expired, the block must return RLO=1 to end the step.
	The lag time is monitored by means of the result flag DFM1.50 M 766.1
	For this example, you can copy the factory version of block FC 1001 to FC 1501.
	Modifications required in this block: Network 5 transition A M 101.0 //step monitoring time has expired O M 766.1 //DFM 1.50 completed, i.e. the lag time has expired
EOP 502	FC 1502
	A step monitoring time needs to be defined for the final position step. When this time has expired, the block must return RLO=1 to end the step.
	The step monitoring time is polled with the flag 101.0.
	For this example, you can copy the factory version of block FC 1001 to FC 1502, and use it without any modification.

Enabling unit 50 for the process cell overview by entering unit 50 in the file `...<proj-path>\etc\kpos001.ini`

```
 1 50
 0 0
```

Enabling the start of the unit by means of the plant overview (direct sequence start)

Note: So that the recipe can be started, the master recipe and recipe procedure must be set to "Release for test" or "Release production".

Entry of recipe category 1 in the file `...<proj-path>\pcu.001\recipe\sequence.ini`.

```
[Sequence050]
Comment=Drain caustic solution
RecType=1
SetPoint=
DFMList=
UnitID=50
UnitPcu=1
```

The recipe should now start in the 'Plant overview' program, after the 'Cleaning' recipe for "Unit 50" has been selected, with the following steps:

- Step 2 runtime with step monitoring time of 30 seconds
- Step 3 runtime with the lag time of 180 seconds of DFM 1.50
- Step 4 runtime with step monitoring time of 30 seconds

The unit is then closed.

7.3.2 Create a recipe for parallel lines

Adjusting the concentration of cleaning solutions in the different tanks follows a similar pattern, and the equipment of the tanks is also comparable. With the introduction of so-called parallel lines it is possible to save time when generating the recipes for these kinds of plants and processes.

7.3.2.1 Configuration of the equipment data

As in the previous example, units, sequences, operations and DFMs must be configured. The procedure is the same as the previous example. The following objects are used in this example:

Object type	PCU/Number	Name	Other
Unit	1/51	Caustic tank 1	
Unit	1/52	Caustic tank 2	
Unit	1/53	Caustic tank 3	
Unit	1/54	Caustic tank 4	
Sequence	1/51	Adjust Conc.1	
Sequence	1/52	Adjust Conc.2	
Sequence	1/53	Adjust Conc.3	
Sequence	1/54	Adjust Conc.4	
Equipment operation	1/510	Base position	Line 1
Equipment operation	1/511	Conc. Dosage	Line 1
Equipment operation	1/512	Conc. Reached	Line 1
Equipment operation	1/513	Circulation	Line 1
Equipment operation	1/514	End position	Line 1

Object type	PCU/Number	Name	Other
Equipment operation	1/520	Base position	Line 2
Equipment operation	1/521	Conc. Dosage	Line 2
Equipment operation	1/522	Conc. Reached	Line 2
Equipment operation	1/523	Circulation	Line 2
Equipment operation	1/524	End position	Line 2
Equipment operation	1/530	Base position	Line 3
Equipment operation	1/531	Conc. Dosage	Line 3
Equipment operation	1/532	Conc. Reached	Line 3
Equipment operation	1/532	Circulation	Line 3
Equipment operation	1/534	End position	Line 3
Equipment operation	1/540	Base position	Line 4
Equipment operation	1/541	Conc. Dosage	Line 4
Equipment operation	1/542	Conc. Reached	Line 4
Equipment operation	1/542	Circulation	Line 4
Equipment operation	1/544	End position	Line 4
DFM	0.60	Cond. CT1	
DFM	0.61	Cond. CT2	
DFM	0.62	Cond. CT3	
DFM	0.63	Cond. CT4	
DFM	0.64	Dosing cycl. CT1	
DFM	0.65	Dosing cycl. CT2	
DFM	0.66	Dosing cycl. CT3	
DFM	0.67	Dosing cycl. CT4	

The operations 'Conc. Dosage' (1/511, 1/521, 1/531, 1/541) have a setpoint for the number of the dosing cycles (0.64 to 0.67; Type 'SW', unit 'cycles', minimum 1, maximum 32676).

The operations 'Conc. reached' (1/512,1/522,1/532,1/542) have one setpoint for the conductivity limit (0.60 to 0.63; Type of 'SW', unit 'mS', minimum 1, maximum 1000).

The operations 'Conc. reached' (1/512,1/522,1/532,1/542) are configured as jump destination producers (Jump destinations ‘1,2’, name ‘ok’).

7.3.2.2 Creating line conversion tables

SectAss.dbf

In the SectAss table the conversion of the reference recipe unit procedures is stored for each parallel recipe unit procedure. In this example this diagram is very simple due to the small extent:

NO_REF_LIN	PCU_REF_L	SECT_REF_L	NO_PAR_LIN	PCU_PAR_L	SECT_PAR_L
1	1	51	2	1	52
1	1	51	3	1	53
1	1	51	4	1	54

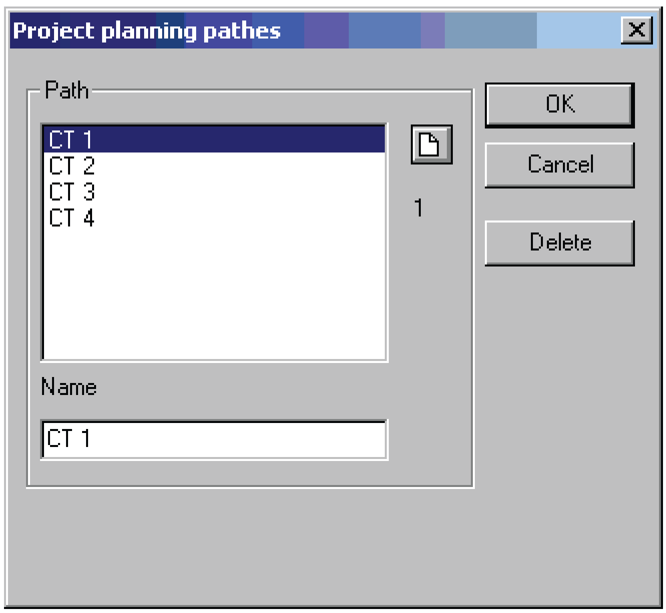
BopAss.dbf

The operations are mapped in a simple way, too:

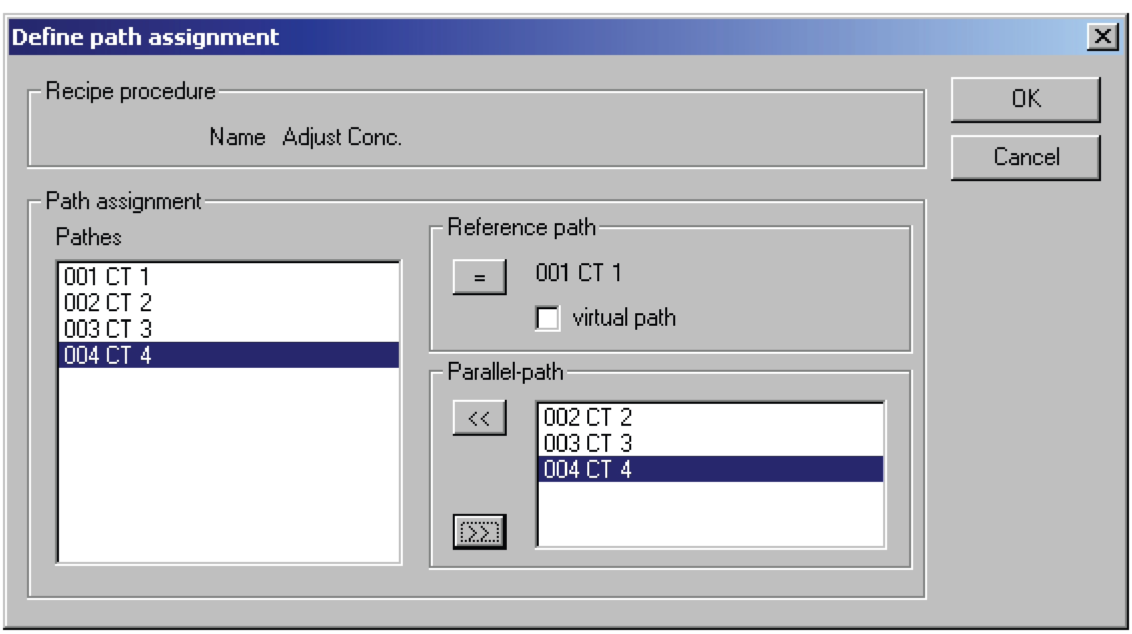
NO_REF_LIN	PCU_REF_L	BOP_REF_L	NO_PAR_LIN	BOP_PAR_L
1	1	510	2	520
1	1	511	2	521
1	1	512	2	522
1	1	513	2	523
1	1	514	2	524
1	1	510	3	530
1	1	511	3	531
1	1	512	3	532
1	1	513	3	533
1	1	514	3	534
1	1	510	4	540
1	1	511	4	541
1	1	512	4	542
1	1	513	4	543
1	1	514	4	544

7.3.2.3 Create recipe procedure

Up to the dialog 'RP properties' the steps for creating the line procedures and the simple procedure are identical. In the property dialog, 'Lines procedure' is selected as recipe procedure type. Then the reference line and the parallel lines should be created. The line definition dialog is opened by the button 'lines'. In this dialog 3 new lines are created and the existing line renamed as 'CT 1'. The rest of the lines are called 'CT 2', 'CT 3' and 'CT 4'.



In the dialog 'Configuration line assignment' that is opened via the button 'assignment' from the ‘RP properties' dialog, the reference and parallel lines are assigned. By selecting line 1 'CT 1' in the left list view and pressing the ‘=' button the line becomes the reference line.



Lines ‘CT 2’, ‘CT3’ and ‘CT 4’ are assigned as parallel lines by selecting them and then by pressing the ‘>>’ button.

In the 'RP properties' dialog a recipe unit procedure 'Adjust conc.' is created, which has unit 'LT 1' as candidate. The three parallel line units must not be assigned.

After having closed the dialog the recipe procedure view is opened. Here the operations 510 to 514 are inserted in the following sequence:

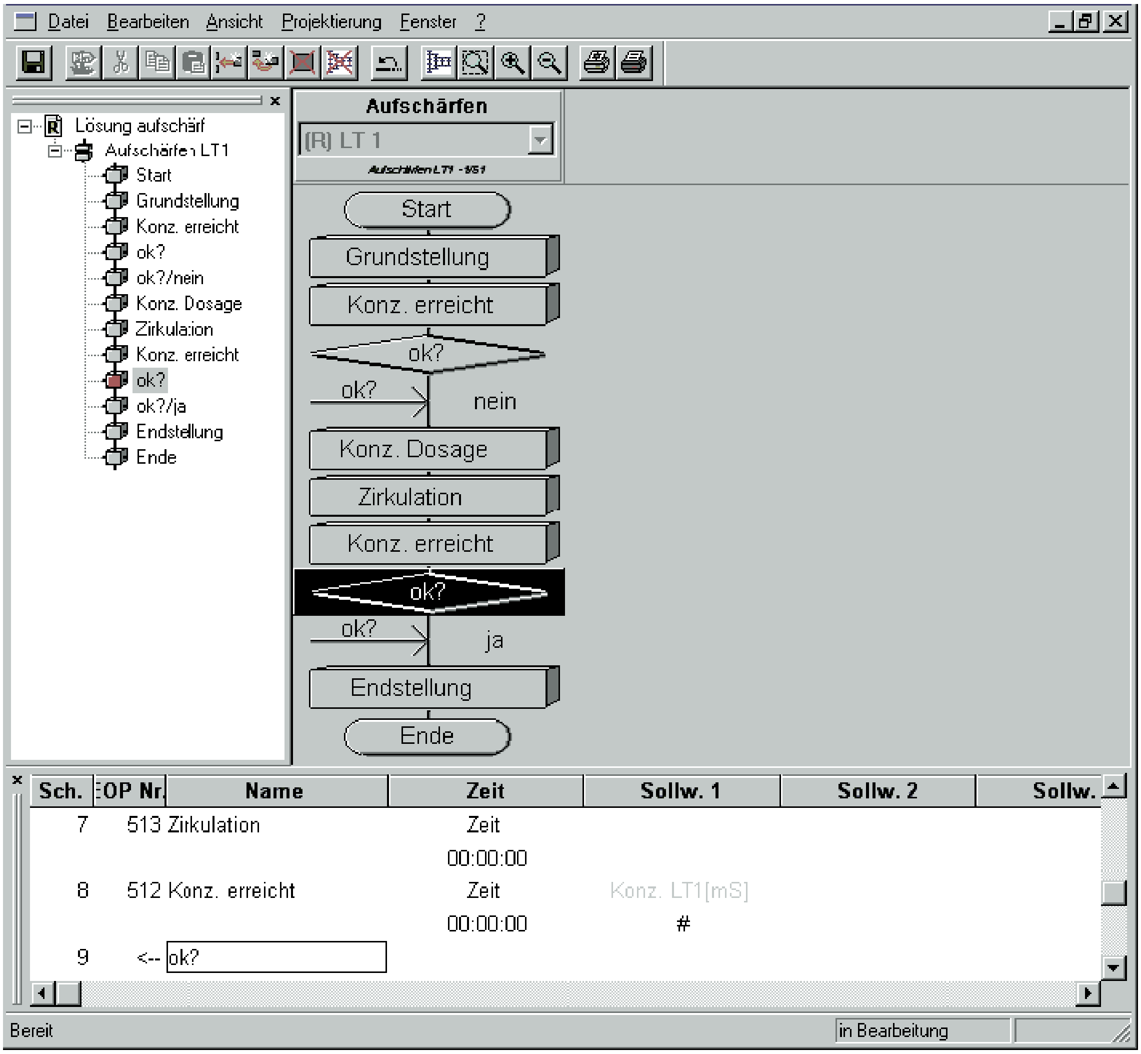
1. Start (- - -)
2. Basic position (510)
3. Conc. reached (512)
4. Conc. Dosage (511)
5. Circulation (513)
6. Conc. reached (512)
7. End position (514)
8. End (- - -)

The concentrate is supposed to be dosed with a piston-type pump. Before the first dosing cycle is started, the conductivity of the caustic is checked. If so no further dosing action is necessary. After the concentrate is dosed and circulated, its conductivity is checked again.

If the conductivity value is reached the procedure is completed, otherwise dosing is repeated.

Jumps and jump destinations must be inserted into the recipe procedure.

One jump destination is inserted above step 4 via the 'Edit object ' dialog (Label 'no', number 1). One jump destination is inserted above step 7 (Label 'yes', number 2). After the two operations 'Conc. reached' one jump is inserted via the 'Edit object' dialog.



During storage, the recipe editor executes the line conversion independently and informs about errors or success in a list box.

7.3.2.4 Create the recipe category and the master recipe

As described in the previous use case, another new recipe category is created and named 'Solutions'. In this recipe category a new master recipe 'Adjust conc.' is created as in the first case of application.

7.3.3 Creating a recipe with process parameters

If recipe procedures of two different processes differ only in the setpoint values of their ROPs an elegant way to save configuration work is the use of process parameters. One example is a bioreactor in which different nutrient solutions are fermented. The different fermentation processes are defined with different master recipes, while the recipe procedure is the same for all products. As process parameter the fermentation temperature is dependent on the master recipe.

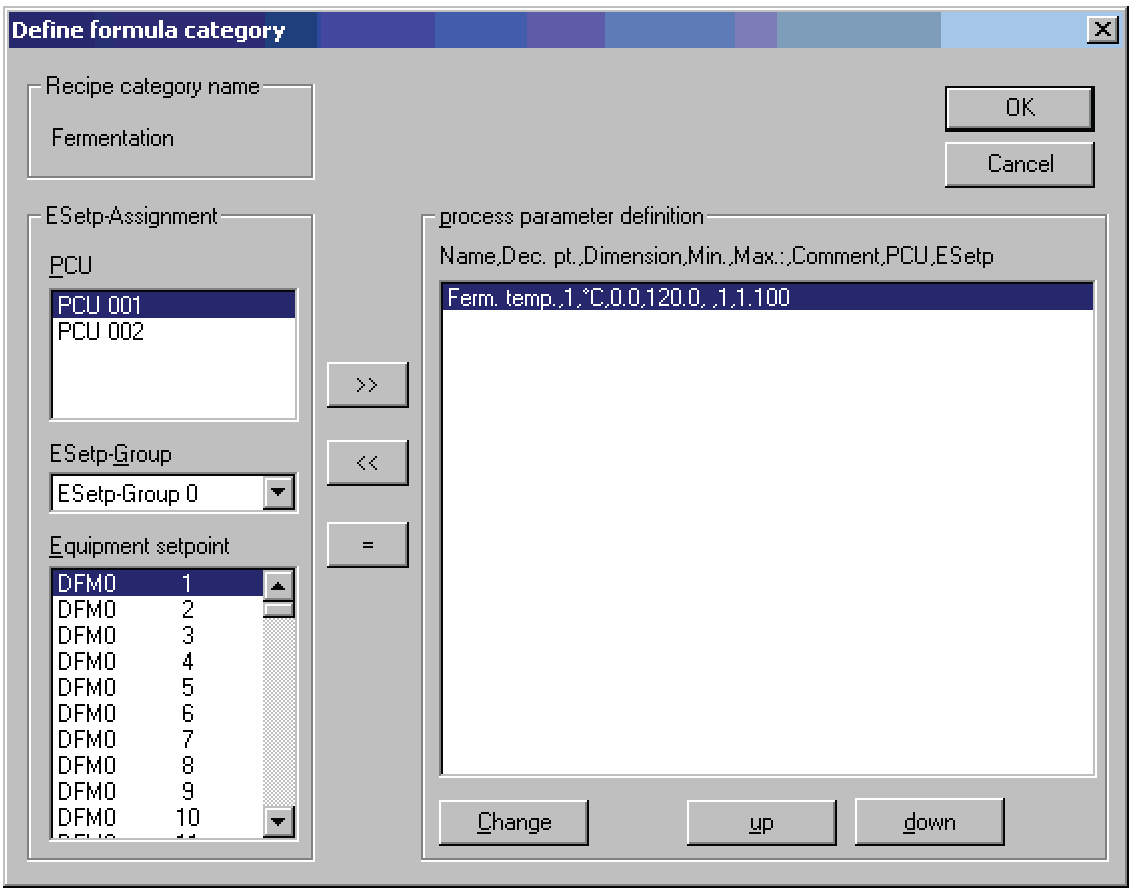
7.3.3.1 Configuration of the equipment data

As in the previous example, units, sequences, operations and DFMs must be configured. The procedure is the same as the previous example. The following objects are used in this example:

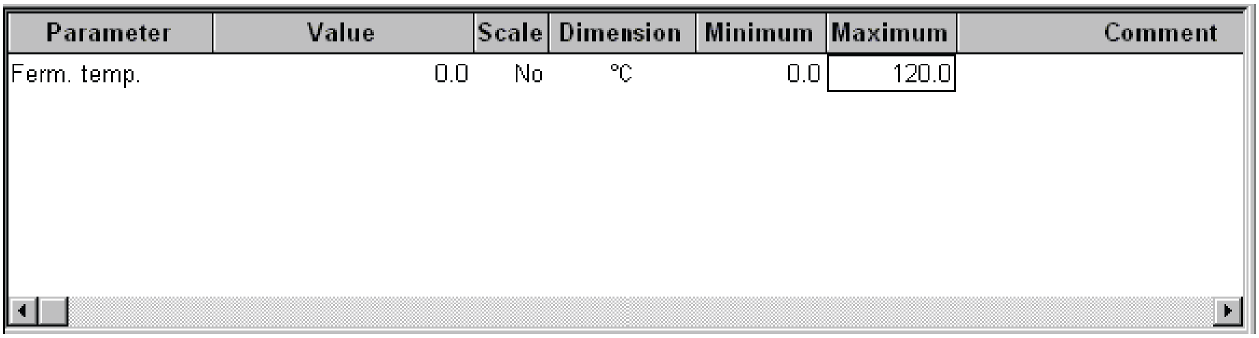
Object type	PCU/Number	Name	Other
Unit	1/60	Fermenter	
Sequence	1/60	Fermentation	
Equipment operation	1/600	Base position	
Equipment operation	1/601	Fill	
Equipment operation	1/602	Fermentation	with setpoint
Equipment operation	1/603	Emptying	
Equipment operation	1/604	End position	
DFM	0.100	Temperature	Setpoint for EOP1/602

7.3.3.2 Create the recipe category and the two master recipes

As already described in the recipe overview one new recipe category ‘Fermentation’ is created. With a right-click on the new recipe category the context menu of this category is opened and the menu item 'Define process parameter' is selected. In the process parameter definition dialog the DFM is chosen as the template for the definition of the process parameter whose value is to be substituted by the process parameter: DFM 0.100. The DFM is selected in the control element group 'DFM assignment' and assigned as the process parameter template by pressing the '>>' button.



Now via the recipe procedures context menu two new master recipes are created and named ‘Product A’ and ‘Product B’. Since the recipe category was assigned a process parameter definition, each of the two master recipes has a process parameter with exactly one parameter ‘Ferm. temp.’.



The temperature setpoint value is entered into the 'value' column for each of the two products independently. The value is not scaled because the fermentation temperature is not dependent on the batch size.

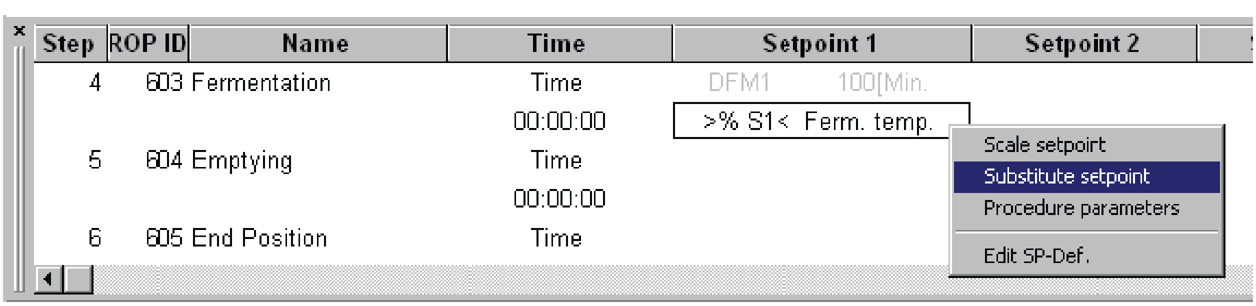
7.3.3.3 Create recipe procedure

As shown in the previous cases of application the recipe procedure is created with a single recipe unit procedure 'Fermenting' and the unit 1/60 is assigned as candidate. The configured operations are inserted in following order:

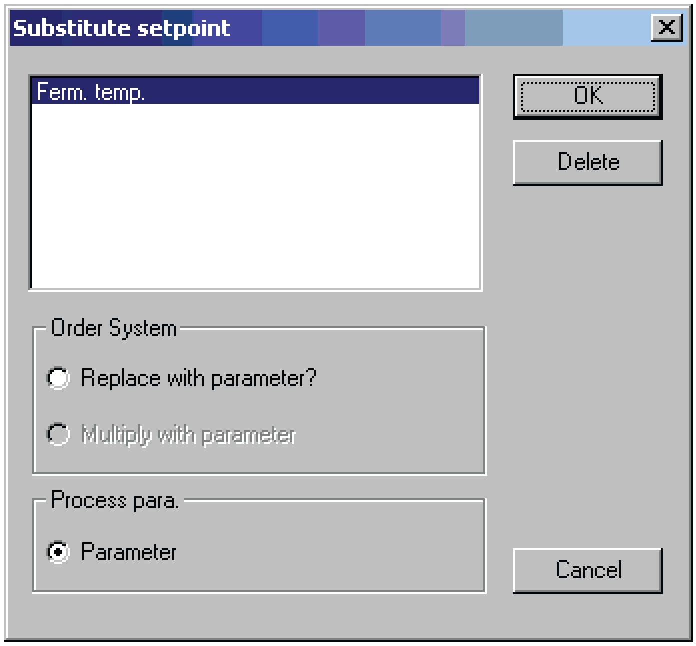
1. Start (- - -)
2. Basic position (601)
3. Fill (602)
4. Fermentation (603)
5. Idle (604)
6. End position (605)
7. End (- - -)

In this case it has to be considered that the recipe procedure must already be assigned to the recipe category 'fermentation' during its generation, because the recipe system needs this information to determine the available process parameters.

The setpoint of the operation ‘Fermentation’ must be substituted with a process parameter. The Substitute setpoint operation is available via the setpoint’s context menu in the RUP list subview.



If the radio button ‘Parameter’ in the ‘Process parameter’ group box is checked the defined process parameter of the recipe category is shown in the list box to the top.



In the RUP list subview the substitution is indicated by the text '>%S<' followed by the name of the process parameter.

7.3.4 Recipe procedure with synchronizations and alternatives

7.3.4.1 Recipe procedure with synchronizations and alternatives

As an example for a recipe with synchronizations and alternatives, malt grinding with one mill and two weighers that may be used alternatively is taken. The weighers are connected to the mill by two conveyors.

7.3.4.2 Configuration of the equipment data

As in the previous example, units, sequences, operations and DFMs must be configured. The procedure is the same as the previous example; the following objects are used in this example:

Object type	PCU/Number	Name	Other
Unit	1/30	Mill 1	
Unit	1/31	Transfer route1	
Unit	1/32	Transfer route2	
Unit	1/33	Weigher 1	
Unit	1/34	Weigher 2	
Sequence	1/30	Milling 1	
Sequence	1/31	Transfer 1	
Sequence	1/32	Transfer 2	
Sequence	1/33	Weighing 1	

Object type	PCU/Number	Name	Other
Sequence	1/34	Weighing 2	
Equipment operation	1/100	Base position	Mill 1
Equipment operation	1/101	Selection	Mill 1, altern. dest. prod.
Equipment operation	1/102	Milling	Mill 1
Equipment operation	1/103	End position	Mill 1
Equipment operation	1/110	Base position	Transfer route 1
Equipment operation	1/111	Transfer	Transfer route 1
Equipment operation	1/112	End position	Transfer route 1
Equipment operation	1/120	Base position	Transfer route 2
Equipment operation	1/121	Transfer	Transfer route 2
Equipment operation	1/122	End position	Transfer route 2
Equipment operation	1/130	Base position	Weigher 1
Equipment operation	1/131	Weighing	Weigher 1
Equipment operation	1/132	End position	Weigher 1
Equipment operation	1/140	Base position	Weigher 2
Equipment operation	1/141	Weighing	Weigher 2
Equipment operation	1/142	End position	Weigher 2

7.3.4.3 Create a new recipe category and a master recipe

As described in the previous use cases a new recipe category ‘Grist handling’ is created. This recipe category gets a single new master recipe called ‘Milling 1’. No special features are to be considered in this case.

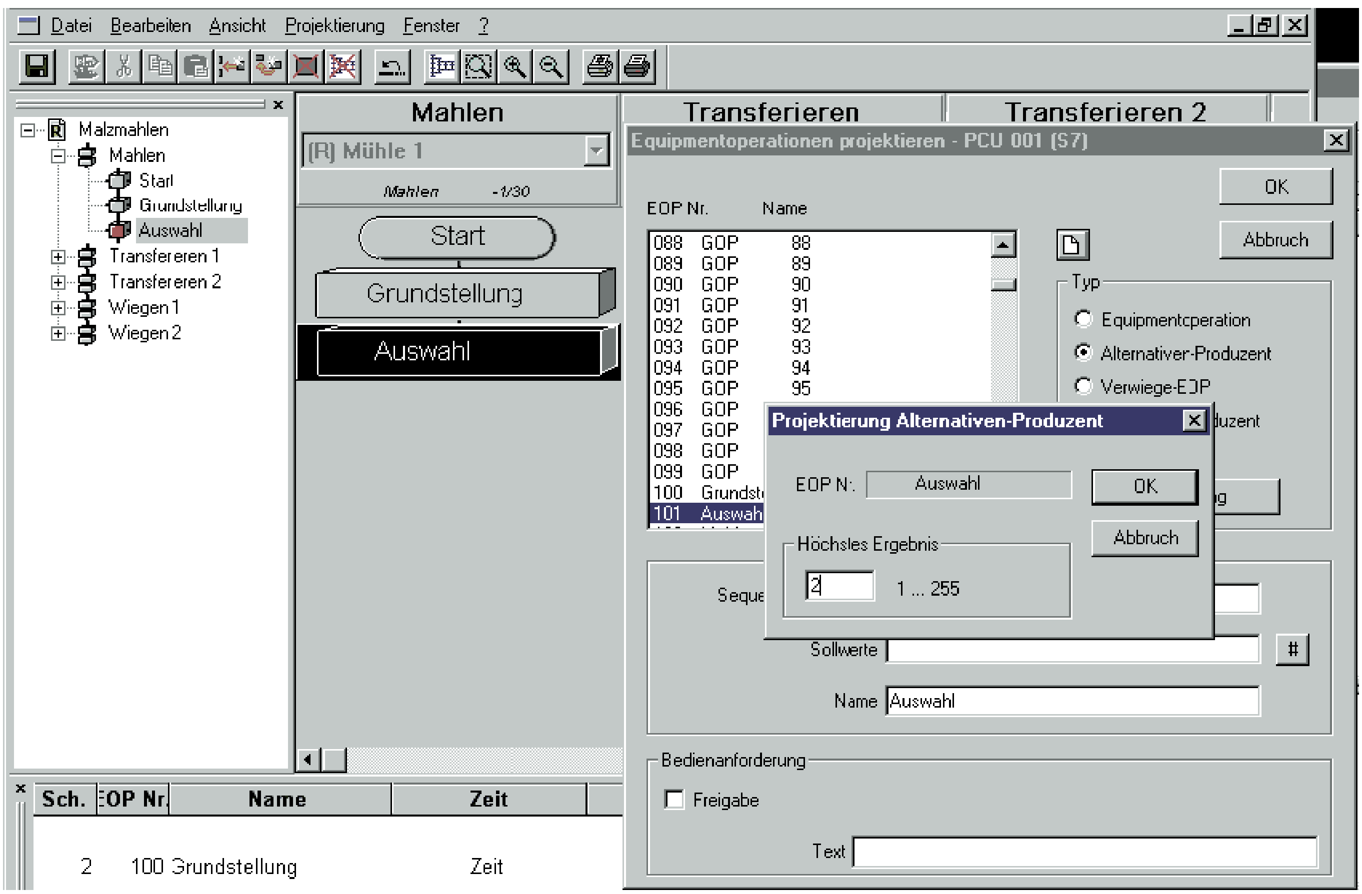
7.3.4.4 Generate the recipe procedure for the grind process

A simple recipe procedure without parallel lines is used for this example. The recipe procedure consists of the following five recipe unit procedures, with the shown units assigned as candidates:

Recipe unit procedure	Unit	Unit sequence
1. Milling	Mill 1	Milling 1
Transfer 1	Transfer route 1	Transfer 1
Weighing 1	Weigher 1	Weighing 1
Transfer 2	Transfer route 2	Transfer 2
Weighing 2	Weigher 2	Weighing 2

7.3.4.5 Insertion and configuration of the operations

The recipe unit procedure 'Milling' is the ‘Start’ RUP. Its operations are ordered as displayed below. The equipment operation 'Selection' is configured as ‘Alternative destination producer’ for both alternative destinations 1 and 2. The logic for the decision between the two alternative RUPs is programmed in the PLC. As a possible criterion for decision-making, there might be the current allocation of both alternative units, one order parameter with which a setpoint is substituted or an algorithm as the 'Round-Robin' arbitration.



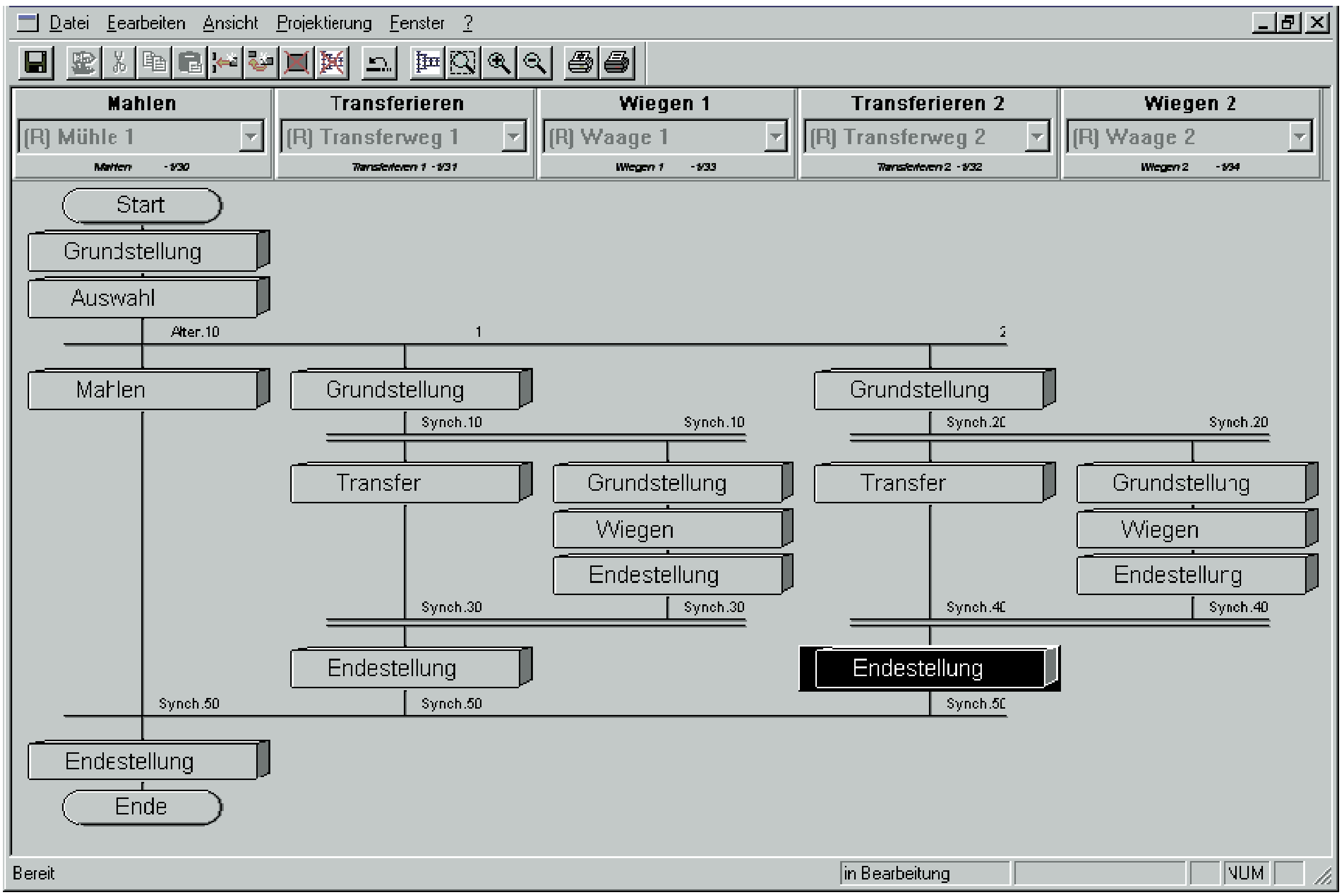
An ROP of the ‘Alternative’ type is appended below the ‘Selection’ operation. The alternative gets the number 10. In the recipe unit procedures ‘Transfer route 1’ and ‘Transfer route 2’ the first steps are changed into 'Alternative destination’ ROPs. The recipe unit procedures are assigned the alternative results 1 and 2 and the alternative number 10.

Below the ‘Alternative destinations’ the 'Base position' ROPs are inserted, directly followed by an ‘AND’ type synchronization. The synchronization is configured via the ‘Edit object’ dialog. The synchronization in the recipe unit procedure 'Transfer route 1' gets number 10, the synchronization in RUP ‘Transfer route 2’ gets number 20.

In order to bring together two recipe unit procedures branched by synchronizations another synchronization has to be inserted in both weigher recipe unit procedures. The connection of the two new synchronization operations is determined by its number.

Now the operations are inserted into the recipe unit procedures as shown. Since the operations were configured in ascending order, this procedure is recommended, as the combination field of the 'Edit object' dialog is preset with the selected EOP when inserting or appending new operations.

Finally, the branches must be reconnected by synchronizations and alternatives. When bringing together two RUPs that were branched by ‘AND’ type synchronizations, in most cases ‘AND’ type synchronizations are used. When collecting branches that were created by alternatives, ‘OR’ type synchronizations have to be inserted, since most of the time only two of several RUPs are executed.



Remark: EOPs executed prior to an AND synchronization may have to be prepared to continue processing, even though the next step condition may have been met.

See also

EOP followed by an AND synchronization (Page 670)

7.3.5 Recipe procedure with cross-stream

The sample plant is a brewhouse with 3 parallel production lines. The batches may swap the lines at several points in the plant. In a brewline the upperbacks may be skipped, e.g. when producing the first brew of the week.

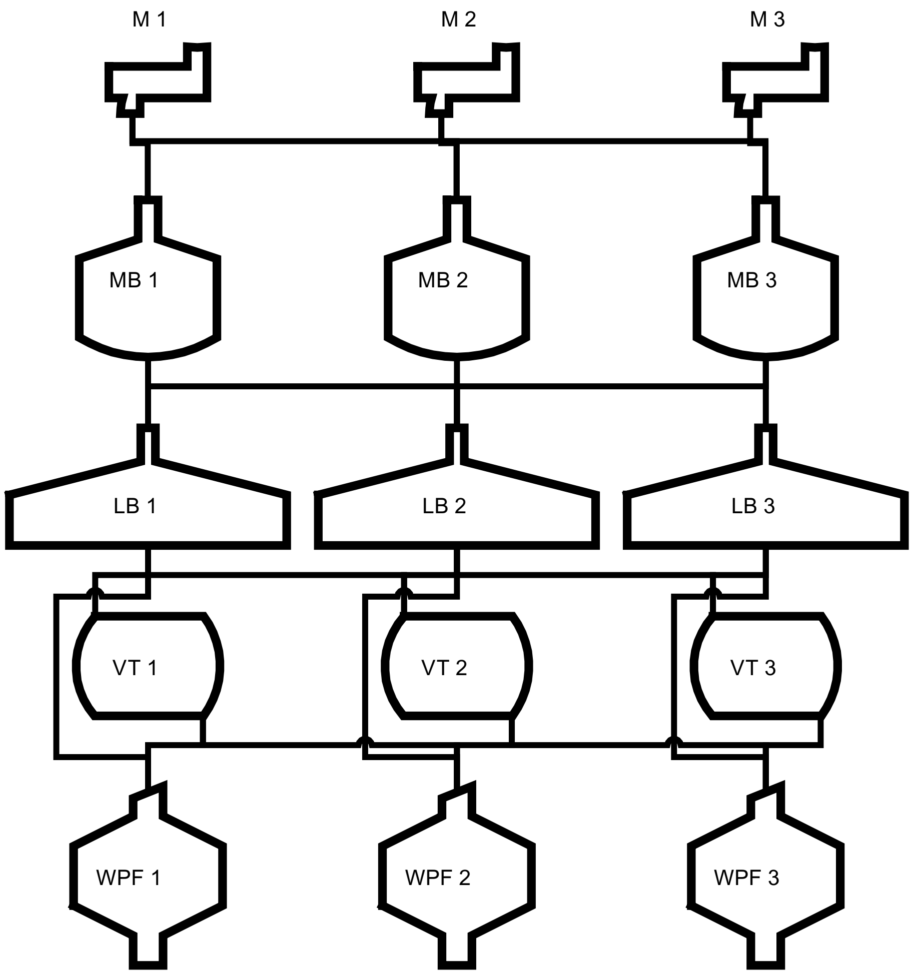


Figure 7-25 Example: 3 mash tuns with 3 pre-mashers, 3 lauter tuns and 3 wort coppers with upperbacks.

As shown in the figure, in the sample plant there are several alternative streams: e.g. the primary streams.

M1 - MT1 – LT1 – WC1,

M2 - MT2 – LT2 – WC2,

M3 - MT3 – LT3 – WC3 or

M1 – MT1 – LT1 – UB1 –WC1.

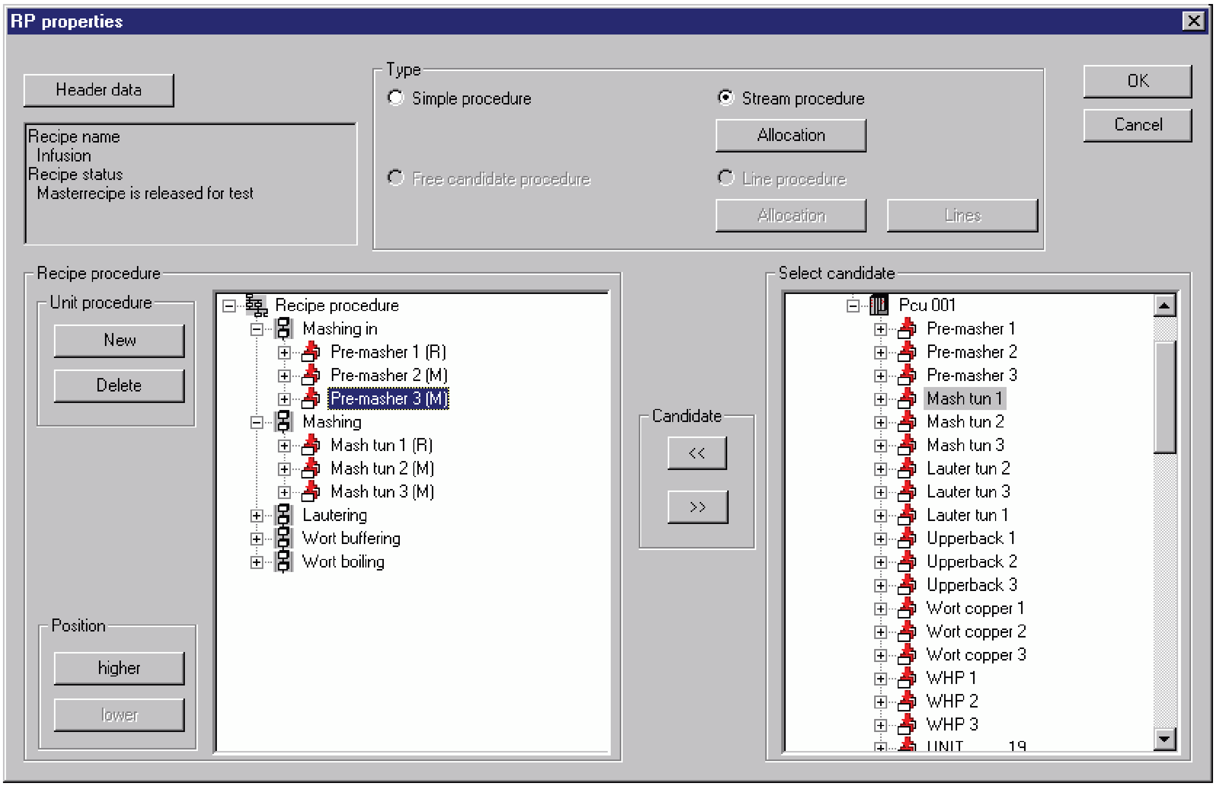
The secondary streams:

MT1 – LT2 – WC3,

LT1 – UB2 - WC2 or

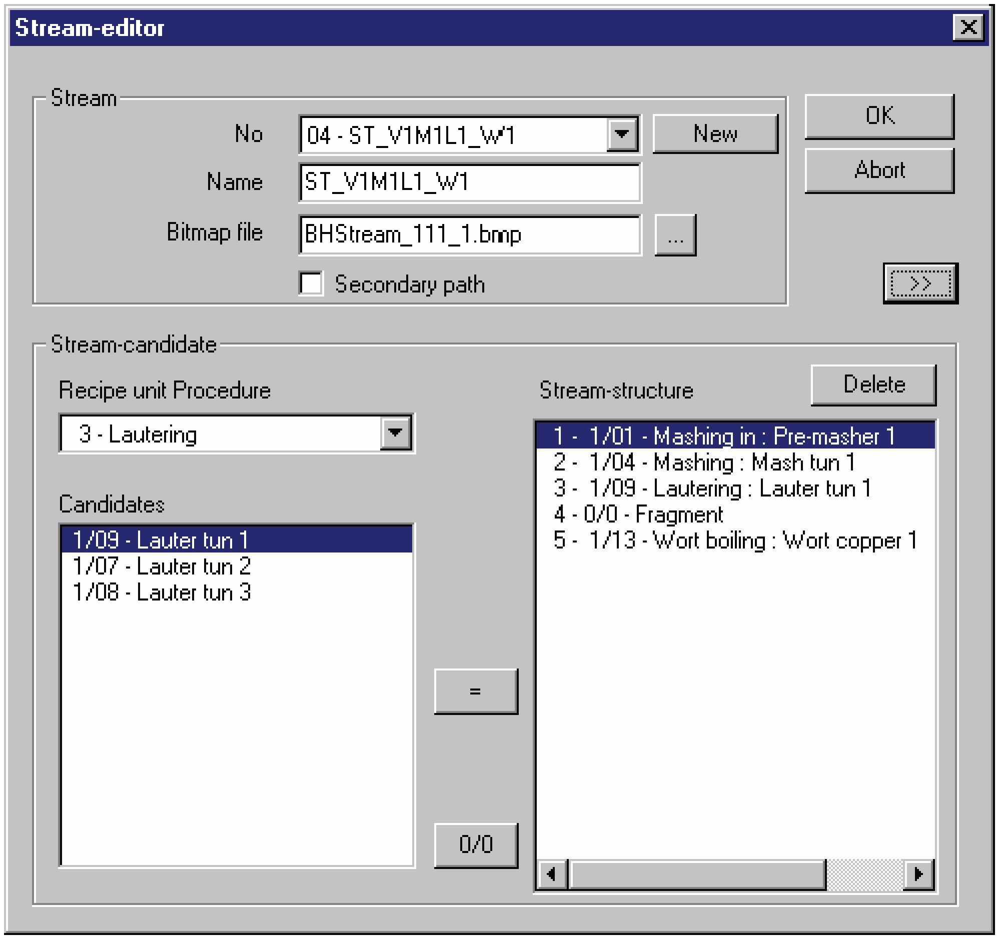
LT2 – WC3.

The English language also distinguishes between standard streams ('streams') and streams which cover or cross units of several standard streams ('cross-streams').



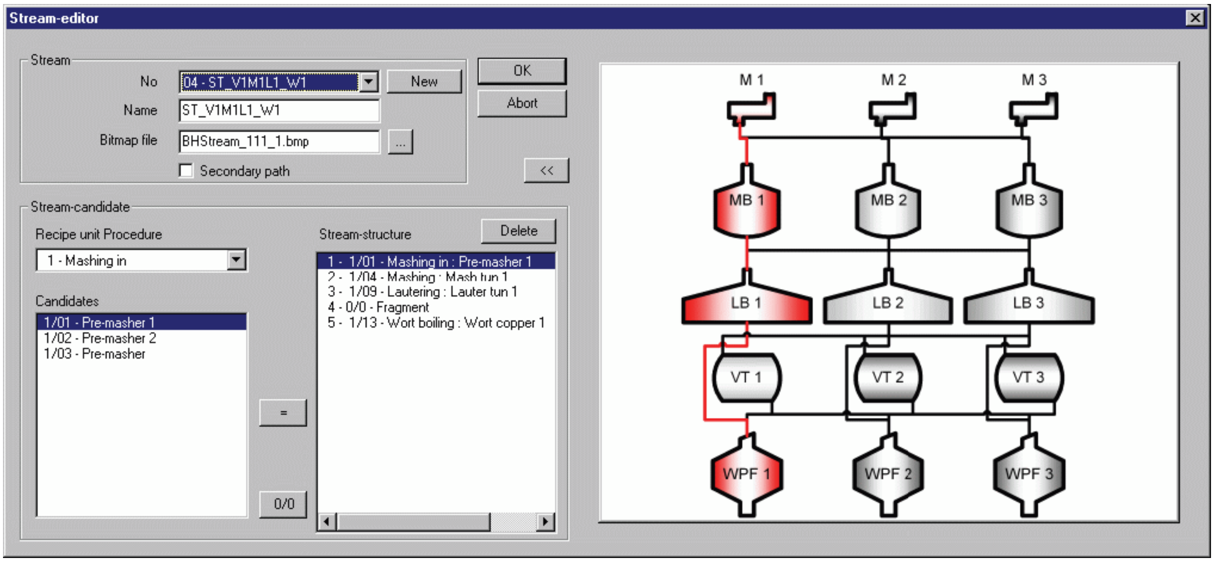
After the new recipe procedure is created the new recipe unit procedures ‘Mashing in‘, ‘Mashing‘, ‘Lautering‘, ‘Wort buffering‘ and ‘Wort boiling‘ are inserted in the RP properties dialog. The units shown in the figure are configured in the equipment data dialog, and then assigned to the new RUPs. Note: assigning more than one candidate to an RUP is only possible if the RP type has been chosen as ‘streams‘. The first assigned candidate is called ‘reference candidate‘ and labeled with the letter ‘R‘. This candidate is taken as the default in most of the following selection operations. The additional candidates are so-called manually converted candidates and therefore are labeled with an ‘M‘.

After the allocation of all candidates, the stream editor is opened by pressing the ‘Allocation‘ button.



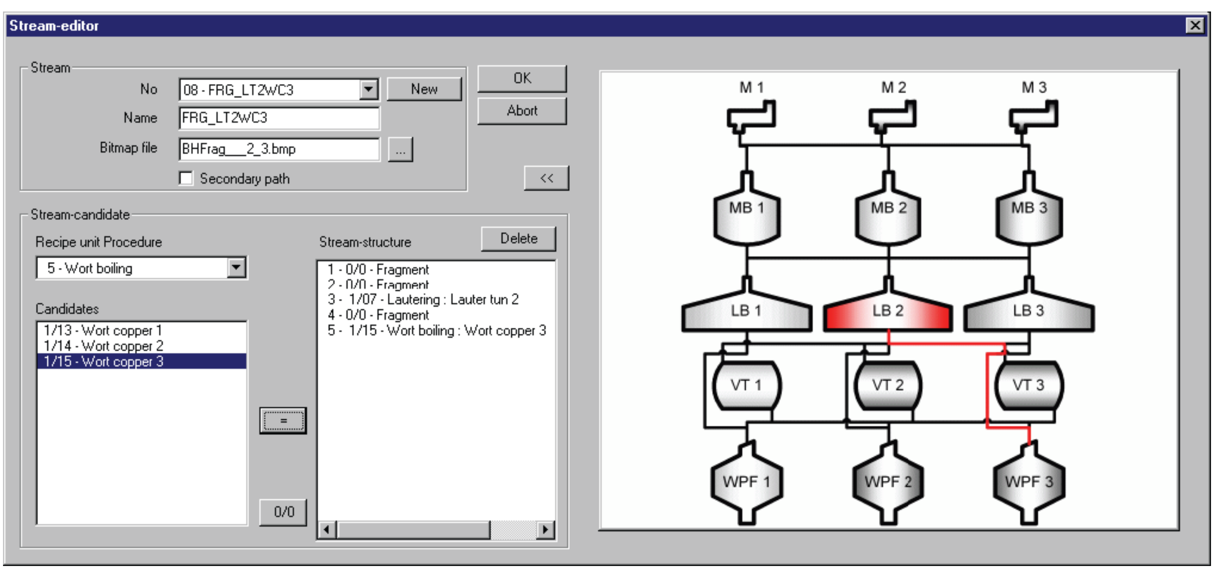
Click ‘New‘ to generate the first stream which is assigned the reference candidates as units. Assign the systematic name (‘ST_P1M1L1U1W1‘ Starting stream containing Pre-masher 1, Mash-tun 1, etc.) and select a bitmap file. You can now click 'New' again to generate the second stream (‘ST_P2M2L2U2W2‘), and then select the corresponding bitmap file. Select the RUPs successively from the "RUP" combo box. From the 'Candidates' list, select 'Unit 2', and then click '=' to assign it to each RUP. The ‘ST_P3M3L3U3W3‘ stream is created in the same way.

When brewing the first batch on a stream at the beginning of the week, the wort copper is not in use. The upperback can be skipped in this case:



The stream can be created like the default stream ‘ST_P1M1L1U1W1‘ with one difference: RUP 3 ‘Wort buffering‘ has the empty unit (‘0/0‘ button) assigned as candidate.

To react to delays flexibly or to use empty vessels, streams are created that swap the path a batch is processed on, e.g. from lauter tun 2 directly into wort copper 3. This stream is a so-called secondary stream, for it has no starting unit. This means the RUPs ‘Mashing in‘, ‘Mashing‘ and ‘Wort buffering‘ have the empty unit assigned to them. The box ‘Secondary stream‘ has to be checked manually.



Creation of recipe procedures with streams

Stream recipe procedures are fully integrated into the recipe system. Therefore they are configured as simple RPs. Contrary to line procedures that are generated by the system automatically, the user has to create all RUPs manually. The recipe editor only shows the RUPs and candidates of the selected path. The candidates may be changed over via the stream selection dialog. The dialog is displayed by pressing the ‘streams‘ button in the toolbar or via the menu entry 'View'->'Stream selection'. The dialog contains a combo box for the stream selection and the configured bitmap for easy orientation.

When inserting synchronizations or alternative objects the user must ensure a consistent number allocation over all streams.

AS configuration with STEP 7 V5.x (S7-400) 8

8.1 Free system resources for users

The system uses a variety of different CPU block and flag areas. Access by user programs to these and to other resources reserved for future or project-specific expansions is locked.

The tables below therefore show the resources available to the user:

Table 8-1 PCU Version < V7

	S7 CPU 414		**S7 CPU 416**		**S7 CPU 417**	
Type	CPU range	User	CPU range	User	CPU range	User
FB	0 - 7999 max. 3000	>= FB1500	0 - 7999 max. 5000	>= FB1500	0 - 7999 max. 8000	>= FB1500
FC	0 - 7999 max. 3000	>= FC2051	0 - 7999 max. 5000	>= FC2051	0 - 7999 max. 8000	>= FC2051
DB	1 - 16000 max. 6000	>= DB2500	1 - 16000 max. 10000	>= DB2500	1 - 16000 max. 16000	>= DB2500
MB	0 - 8191	1432 – 2045 2048 – 8191	0 - 16383	1432 – 2045 2048 – 16383	0 - 16383	1432 – 2045 2048 – 16383
Timer (Times)	T0 - T2047	T0-T95 T224-T509	T0 - T2047	T0-T95 T224-T509	T0 - T2047	T0-T95 T224-T509
Counter	Z0 - Z2047	Z0 - Z2047	Z0 - Z2047	Z0 - Z2047	Z0 - Z2047	Z0 - Z2047

Table 8-2 PCU Version from V7

	S7 CPU 414		**S7 CPU 416**		**S7 CPU 417**	
Type	CPU range	User	CPU range	User	CPU range	User
FB	0 - 7999 max. 3000	>= FB3000	0 - 7999 max. 5000	>= FB3000	0 - 7999 max. 8000	>= FB3000
FC	0 - 7999 max. 3000	>= FC1001	0 - 7999 max. 5000	>= FC1001	0 - 7999 max. 8000	>= FC1001
DB	1 - 16000 max. 6000	>= DB2500	1 - 16000 max. 10000	>= DB2500	1 - 16000 max. 16000	>= DB2500
MB	0 - 8191	>= MB2048	0 - 16383	>= MB2048	0 - 16383	>= MB2048
Timer (Times)	T0 - T2047	T256-T509 >= 512	T0 - T2047	T256-T509 >= 512	T0 - T2047	T256-T509 >= 512
Counter	Z0 - Z2047	Z0 - Z2047	Z0 - Z2047	Z0 - Z2047	Z0 - Z2047	Z0 - Z2047

8.2 Unit function block

8.2.1 Unit function block

Basics

The AS system control enables simultaneous operation of up to 64 (PCU < V7) or 128 (PCU V7) SEQUENCES (units) per PCU. The instructions and the step enabling conditions which stored in the user program of the equipment operations EOP-FCs are processed recipe-controlled.

Integral components of the SEQUENCES system block are up to 20 (PCU < V7) or 24 (PCU V7) digital function modules (DFMs) plus run time monitoring, that can be configured as a timer, up counter, down counter, limit comparator, setpoint comparator, router or decoder, as a mask or as a setpoint /actual value cell. The respective setpoint or limit values are stored in the recipe lists.

User unit FB

Each unit is assigned a user FB in which the appropriate permanent conditions, interlocks, start commands, etc. are programmed by the user. These are created as an S7 FB block with the programming languages STL or SCL; the following block number bands are preset:

- with PCU < V7: FB 1001/unit 1 ... FB 1064/unit 64
- with PCU V7: FB 1001/unit 1 ... FB 1128/unit 128

The user unit FB is called by the system before and after the EOP-FC. It is possible to determine whether the current call came before or after the EOP-FC by querying a flag.

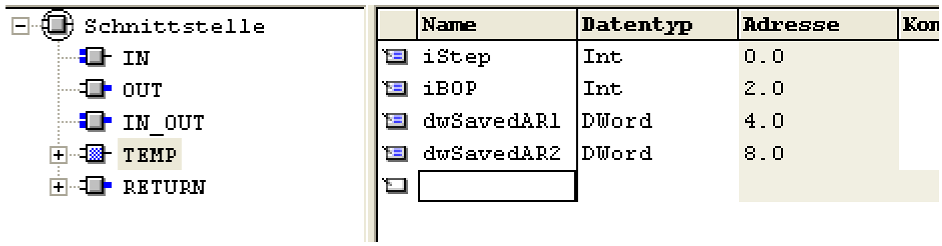
```
U      "SeqFbCalledAfterEop";   //After EOP
JC   NGOP;
      ...      //Before EOP processing
      BEU;
NGOP: ...      //After EOP processing
```

An evaluation of the system flag "SeqFbCalledAfterEop" for a sequence within the associated sequence FB is only valid if the system flag "SequenceIsRunning=1" (sequence running) is also queried in addition.

Use of the address registers AR1 and AR2

If the address registers are to be used in the user program you must make sure that they are backed up before each change and then restored:

Definition of the local variables:




```
...
...
```

... before use or block start

```
TAR1 dwSavedAR1
TAR2 dwSavedAR2
...
...
```

... after use or at block end / before system call

```
LAR1 dwSavedAR1
LAR2 dwSavedAR2
...
```

Flag interfaces between unit control and user unit FB

This flag interface is identical for the user unit FB and EOP FC and is described in section Global flag interface (Page 655).

Flag interface between unit instance and user unit FB

The sequence control has the following globally defined flags per sequence:

DB #	**Permanent condition** is generated in the user unit FB. =0 means that the sequence cannot be started	M 640.0 – M 647.7 M 648.0 – M 655.7	Control by user program / Evaluated by system
ATL #	**Operation sequence start** Is shown by the sequence control =1 the sequence is running =0 the sequence is in step=0	M 656.0 – M 663.7 M 664.0 – M 671.7	Control by system / Evaluated by user program

Assignment of the flag interfaces for TA 1...64

TA no.	DB #	ATL #	TA no	DB #	ATL #
1	M 640.0	M 656.0	33	M 644.0	M 660.0
2	M 640.1	M 656.1	34	M 644.1	M 660.1
3	M 640.2	M 656.2	35	M 644.2	M 660.2
4	M 640.3	M 656.3	36	M 644.3	M 660.3
5	M 640.4	M 656.4	37	M 644.4	M 660.4
6	M 640.5	M 656.5	38	M 644.5	M 660.5

TA no.	DB #	ATL #	TA no	DB #	ATL #
7	M 640.6	M 656.6	39	M 644.6	M 660.6
8	M 640.7	M 656.7	40	M 644.7	M 660.7
9	M 641.0	M 657.0	41	M 645.0	M 661.0
10	M 641.1	M 657.1	42	M 645.1	M 661.1
11	M 641.2	M 657.2	43	M 645.2	M 661.2
12	M 641.3	M 657.3	44	M 645.3	M 661.3
13	M 641.4	M 657.4	45	M 645.4	M 661.4
14	M 641.5	M 657.5	46	M 645.5	M 661.5
15	M 641.6	M 657.6	47	M 645.6	M 661.6
16	M 641.7	M 657.7	48	M 645.7	M 661.7
17	M 642.0	M 658.0	49	M 646.0	M 662.0
18	M 642.1	M 658.1	50	M 646.1	M 662.1
19	M 642.2	M 658.2	51	M 646.2	M 662.2
20	M 642.3	M 658.3	52	M 646.3	M 662.3
21	M 642.4	M 658.4	53	M 646.4	M 662.4
22	M 642.5	M 658.5	54	M 646.5	M 662.5
23	M 642.6	M 658.6	55	M 646.6	M 662.6
24	M 642.7	M 658.7	56	M 646.7	M 662.7
25	M 643.0	M 659.0	57	M 647.0	M 663.0
26	M 643.1	M 659.1	58	M 647.1	M 663.1
27	M 643.2	M 659.2	59	M 647.2	M 663.2
28	M 643.3	M 659.3	60	M 647.3	M 663.3
29	M 643.4	M 656.4	61	M 647.4	M 663.4
30	M 643.5	M 659.5	62	M 647.5	M 663.5
31	M 643.6	M 659.6	63	M 647.6	M 663.6
32	M 643.7	M 659.7	64	M 647.7	M 663.7

Assignment of the flag interfaces for TA 65...128 (PCU V7 only)

TA no.	DB #	ATL #	TA no.	DB #	ATL #
65	M 648.0	M 664.0	97	M 652.0	M 668.0
66	M 648.1	M 664.1	98	M 652.1	M 668.1
67	M 648.2	M 664.2	99	M 652.2	M 668.2
68	M 648.3	M 664.3	100	M 652.3	M 668.3
69	M 648.4	M 664.4	101	M 652.4	M 668.4
70	M 648.5	M 664.5	102	M 652.5	M 668.5
71	M 648.6	M 664.6	103	M 652.6	M 668.6
72	M 648.7	M 664.7	104	M 652.7	M 668.7
73	M 649.0	M 665.0	105	M 653.0	M 669.0
74	M 649.1	M 665.1	106	M 653.1	M 669.1
75	M 649.2	M 665.2	107	M 653.2	M 669.2
76	M 649.3	M 665.3	108	M 653.3	M 669.3

TA no.	DB #	ATL #	TA no.	DB #	ATL #
77	M 649.4	M 665.4	109	M 653.4	M 669.4
78	M 649.5	M 665.5	110	M 653.5	M 669.5
79	M 649.6	M 665.6	111	M 653.6	M 669.6
80	M 649.7	M 665.7	112	M 653.7	M 669.7
81	M 650.0	M 666.0	113	M 654.0	M 670.0
82	M 650.1	M 666.1	114	M 654.1	M 670.1
83	M 650.2	M 666.2	115	M 654.2	M 670.2
84	M 650.3	M 666.3	116	M 654.3	M 670.3
85	M 650.4	M 666.4	117	M 654.4	M 670.4
86	M 650.5	M 666.5	118	M 654.5	M 670.5
87	M 650.6	M 666.6	119	M 654.6	M 670.6
88	M 650.7	M 666.7	120	M 654.7	M 670.7
89	M 651.0	M 667.0	121	M 655.0	M 671.0
90	M 651.1	M 667.1	122	M 655.1	M 671.1
91	M 651.2	M 667.2	123	M 655.2	M 671.2
92	M 651.3	M 667.3	124	M 655.3	M 671.3
93	M 651.4	M 667.4	125	M 655.4	M 671.4
94	M 651.5	M 667.5	126	M 655.5	M 671.5
95	M 651.6	M 667.6	127	M 655.6	M 671.6
96	M 651.7	M 667.7	128	M 655.7	M 671.7

Standard routing between sequence control and flag interface

The sequence control does not access the global flags directly, but takes them from interface blocks. The FC700/701 programs perform the routing between the flags and interface DBs.

If the global flags are located elsewhere, the application programmer must change the above-named FC.

- DB612 → Unit permanent condition (DB # flag)
- DB613 → Unit start (ATL # flag)

DBB x	DBB x.7 ... DBB x.0	DBB x (PCU V7 only)	DBB x.7 ... DBB x.0
DBB10	Sequence 8 ... 1	DBB18	Unit 72 ... 65
DBB11	Sequence 16 ... 9	DBB19	Unit 80 ... 73
DBB12	Sequence 24 ... 17	DBB20	Unit 88 ... 81
DBB13	Unit 32 ... 25	DBB21	Unit 96 ... 89
DBB14	Unit 40 ... 33	DBB22	Unit 104 ... 97
DBB15	Unit 48 ... 41	DBB23	Unit 112 ... 105
DBB16	Unit 56 ... 49	DBB24	Unit 120 ... 113
DBB17	Sequence 64 ... 57	DBB25	Unit 128 ... 121

8.3 Create engineering operations (EOP) as function

8.3.1 General

Basics

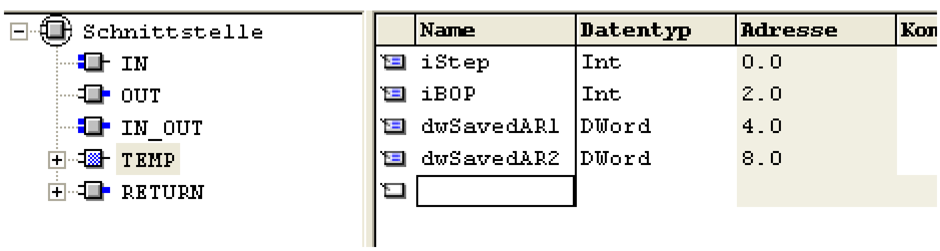
An EOP must be in the AS for this system. These are created as an S7 FC block with the programming languages STL or SCL; the following block number bands are preset:

- with PCU < V7: FC 1001/EOP 1 ... FC 1999/EOP 999
- with PCU V7: FC 1001/EOP 1 ... FC 2999/EOP 1999

Use of the address registers AR1 and AR2

If the address registers are to be used in the user program you must make sure that they are backed up before each change and then restored:

Definition of the local variables:




```
...
...
```

... before use or block start

```
TAR1 dwSavedAR1
TAR2 dwSavedAR2
...
...
```

... after use or at block end / before system call

```
LAR1 dwSavedAR1
LAR2 dwSavedAR2
...
```

8.3.2 EOP interface

Basics

An EPE has mainly the following interfaces:

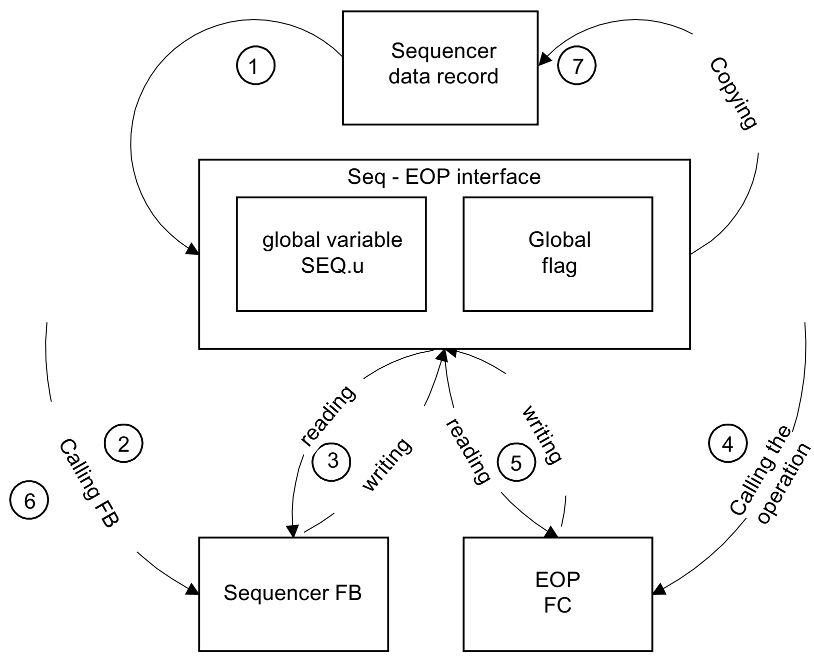
- to the Sequencer
- to the individual control modules of the unit
- to the control modules of the unit
- to the setpoints of the unit

Overview of the EPE interface

Interface	Step 7
Sequencer interface	1. Global flags 2. `"SEQ".u`
RCS interface	`"SEQ".uRCS`
Interface final control element	ICM flag + inputs
Interface Digital Values	Timer
Parameter elements	PE block
DFMs	Global flag / DS
Measurement value	AIN
Controller	PID
Job system	FC714

8.3.3 Sequencer interface

The interface to the sequencer is implemented using global flags and the variable "u" in the data block `"SEQ"` (= DB 725). This interface type makes it possible to work with the support of the STEP 7-Symbolic function. Before each EOP processing operation, the interface is built up from the stored sequencer data and the data is saved again at after processing.



1. Before the EOP is called, the global flag interface is restored from the sequencer data set. In addition, the complete data set is copied into the temporary variable `"SEQ".u` of the sequencer data building block.
2. Call of unit user FB with identifier 'Start'.
3. User program reads and writes signals of the flag interface.
4. Call of the EOP user FC using sequencer control.
5. User program in the EOP FC reads and writes signals of the flag interface.
6. Call of unit user FB with identifier "End".
7. The whole dataset of the temporary variable `"SEQ".u` is written back to the sequencer dataset after calling the EOP user FC and unit user FB. Afterwards parts of the flag interface are transferred in the sequence data set.

8.3.3.1 Global flag interface

Basics

The flag interfaces can be evaluated in the user blocks (unit user FBs and EOP-FCs).

NOTICE
Notice regarding validity and use • The global flags are only valid within the Unit user FBs belonging to the sequence instance, the EOP-FCs and the sub programs called by them (local validity). A query in a program that runs e.g. in OB1 returns an undefined result. • **Due to the local validity of the global flags, they are not suitable for use in the new (PCU-V7 or higher) system block class BLR 1...9.** Only the status bits of the sequence data set (e.g. "Sequenzen.nnn.SeqRun", etc.) may be used for this purpose in the logic operations of the BLR data source dialog.

Certain restrictions apply to the type of access to these flags. These are flag-dependent and shown in the following table.

- S1 : one-time setting
- R1: one-time resetting
- W: written and reading access
- R: only reading access

Overview global flag interface

MB	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
99				Horn-MonT-mError				
100	Alternative result **(only valid up to AS version V4.6!)**							
101	Enable-MonTm	Opera-torRe-quest	AddDevi-ceOn	SeqFb-CalledAf-terEop	Batch-End	Print-StepProt	EopDe-layTmAc-tive	Eop-MonT-mError
102	SeqMan-Mode	Seq-Perm-Cond	Disable-StepProt	Sequen-ceStart-Pulse	EopStop	EopStart	Sequen-ceIsRun-ning	Automa-ticStep-Change
103								
104								
105								
106								
107							HornGr-pError	BmRes-tart

General Signals

Symbol	Comment	Address	Access
HornMonTmError	Horn is set when the monitoring time expires. Processing and resetting by the user.	M 99.4	R1/R
EopMonTmError	Monitoring time unit Result display of the monitoring time of the unit. 0 / 1 = Time hasn't run out / Time has run out	M101.0	R
EopDelayTmActive	When an EOP is started, the wait/delay time is started. Time running: EopDelayTmActive = 0 Time expired: EopDelayTmActive = 1 The time is restarted at signal change AutomaticStepChange from 0 to 1 (display "-" to "+", from STOP to RELEASE).	M101.1	R
PrintStepProt	Start to automatic printing of the step log. When the EOP enters, PrintStepProt is always = 0.	M 101.2	S1/R1 W
SeqFbCalledAfterEop	Flag for TA-FB: SeqFbCalledAfterEop = 0: FB call before the EOP is processed Exclusively DFM-processing! SeqFbCalledAfterEop = 1: FB call after the EOP is processed	M101.4	R
AddDeviceOn	Set sequence bits of the user.	M 101.5	W
OperatorRequest	Set operator prompt	M 101.6	W
EnableMonTm	Enabling a message by exceeding monitoring time. The sequence control always sets this bit to 1. The user program (EOP) can reset it. EnableMonTm=1: By exceeding the step monitoring time, the unit control signals error (display+message). EnableMonTm=0: By exceeding the step monitoring time, the unit control doesn't signal any error.	M 101.7	W
AutomaticStepChange	Sequencer is not in hold	M 102.0	R
SequenceIsRunning	Sequencer is running. Unit-specific flag bits: • Unit 1 ... Unit 64 : M 656.0 ... M 663.7 • Unit 65 ... Unit 128 : M 664.0 ... M 671.7	M 102.1	R
EopStart	First runtime cycle of the EOP	M 102.2	R
EopStop	Last runtime cycle of the EOP	M 102.3	R
SequenceStartPulse	Start impulse Sequencer. This bit has the value 1 for the first operation, otherwise always 0.	M 102.4	R

Symbol	Comment	Address	Access
DisableStep-Prot	Block log input. DisableStepProt=1 prevents a log entry at EOP end The sequence control resets with the start of the EOP (DisableStepProt=0)	M 102.5	W
SeqPermCond	Permanent condition display Unit-specific flag bits: • Unit 1 ... Unit 64 : M 640.0 ... M 647.7 • Unit 65 ... Unit 128 : M 648.0 ... M 655.7	M 102.6	R
SeqManMode	Display of the hand operation of the sequence	M 102.7	R
HornGrpError	Horn group flag Is set at ICM-, AIN-, MSG- und Teilanl(TUE)-disturbance, processing and reset by the user	M 107.1	R1/R
M_100.x (0..7)	Alternative result **(only valid up to AS version V4.6!)**	MB 100	W

See also

Unit function block (Page 648)

8.3.3.2 Global Variable SEQ.u

Interface

The following commands can be read/written in the sequencer data building block `"SEQ"` (DB725):

Tag	Comment	Access
`"SEQ".u.CTRL.Reload_DFM`	Re-transferring of setpoints to the technical parameters (DFMs)	S
`"SEQ".u.CTRL.boHolding (V6)` `"SEQ".u.CTRL.xRemainHolding (V7)`	Retain the "Holding" state. The signal must be set by the EOP if the intermediate state "Holding" is to be retained on the "Hold" command. If the signal is withdrawn, the "Held" state is assumed.	S
`"SEQ".u.CTRL.boHeld (V6)` `"SEQ".u.CTRL.xCmdHold (V7)`	"Hold" command.	L
`"SEQ".u.CTRL.boRestarting (V6)` `"SEQ".u.CTRL.xRemainRestarting (V7)`	Retain the "Restarting" state. The signal must be set by the EOP if the intermediate state "Restarting" is to be retained on the "Restart" command. If the signal is withdrawn, the "Running" state is assumed.	S
`"SEQ".u.CTRL.boRestart (V6)` `"SEQ".u.CTRL.xCmdRestart (V7)`	"Restart" command.	L

Tag	Comment	Access
"SEQ".u.CTRL.boRunning (V6) "SEQ".u.CTRL.xIsRunning (V7)	EOP is in "Running" state.	L
"SEQ".u.CTRL.boPausing (V6) "SEQ".u.CTRL.xRemainPausing (V7)	Retain the "Pausing" state. The signal must be set by the EOP if the intermediate state "Pausing" is to be retained on the "Pause" command. If the signal is withdrawn, the "Paused" state is assumed.	S
"SEQ".u.CTRL.boPaused (V6) "SEQ".u.CTRL.xCmdPause (V7)	"Pause" command.	L
"SEQ".u.CTRL.boAborting (V6) "SEQ".u.CTRL.xRemainAborting (V7)	Retain the "Aborting" state. The signal must be set by the EOP if the intermediate state "Aborting" is to be retained on the "Abort" command. If the signal is withdrawn, the "Aborted" state is assumed.	S
"SEQ".u.CTRL.boAborted (V6) "SEQ".u.CTRL.xCmdAbort (V7)	"Abort" command.	L
"SEQ".u.CTRL.boStopping (V6) "SEQ".u.CTRL.xRemainStopping (V7)	Retain the "Stopping" state. The signal must be set by the EOP if the intermediate state "Stopping" is to be retained on the "Stop" command. If the signal is withdrawn, the "Stopped" state is assumed.	S
"SEQ".u.CTRL.boStopped (V6) "SEQ".u.CTRL.xCmdStop (V7)	"Stop" command.	L
"SEQ".u.CTRL.xSuppressOpStepChange	Preventing manual step change As long as the signal from the EOP is set, a manual step change by the operator is prevented • Process cell overview / "Select step" is prevented • Process diagrams / UnitCtrl / "Select step" is prevented • The "EOP Stop" and "EOP Cancel" commands are possible.	S

Batch and step information

The batch and step information can be read/written using the following tags in the sequencer data block.

Tag	Comment	Access
"SEQ".u.iNewStep	New step	R
"SEQ".u.byYear	Batch year	R
"SEQ".u.byRecType	Recipe type of the batch	R
"SEQ".u.iRecipe	Recipe of the batch	R
"SEQ".u.iOrder	Order number	R
"SEQ".u.iBatch	Batch number	R

Tag	Comment	Access
"SEQ".u.snBA_Name	Batch name	R
"SEQ".u.diBA_ID	Batch ID	R
"SEQ".u.diProduct_ID	Material ID for the product	R
"SEQ".u.diStep_ID	Step ID	R
"SEQ".u.byAlterResult	Alternative result/jump destination • An EOP (alternative producer) returns the numeric value of the alternative to be selected to the unit control the (number 1..255 from the recipe procedure). • The value 0 is invalid • The jump destination (derived from the recipe procedure) must be stored at this location.	W

Material sequence check

A matrix of the material sequence relationships, which is configured in the recipe editor under "Engineering/Material management/Material sequences" can be used to perform a check on the unit level to determine if the product of the source matches the product in the destination. A check for compatibility between the predecessor and successor material can be initiated via the following interface:

Tag	Comment	Access
"SEQ".u.STATUS.xMatCheckIsActive	Material compatibility check is running	W, R
"SEQ".u.CTRL.xMatCheckReset	Reset material compatibility check	W
"SEQ".u.diMatCheckPreID	Predecessor material	W
"SEQ".u.diMatCheckPostID	Successor material	W
"SEQ".u.diMatCheckCategory	(reserved)	
"SEQ".u.bMatCheckStatus	Status of the material compatibility check	R

- The user program sets the `"SEQ".u.STATUS.xMatCheckIsActive` bit in the EOP FC in order to activate the material check by the recipe server. The bit is reset by the recipe server when the material check has finished.
- The recipe server uses the `"SEQ".u.bMatCheckStatus` byte to return the result of the material check. Based on the result, a decision must be made in the EOP FC as to whether the recipe will be continued or, for example, an operation request will be issued.
  Result of the material check:
  0 = Ready
  1 → Material sequence permitted
  2 → Material sequence not permitted
  3 → Internal error
  4 → Material check is running

S88 states and their transitions are described in this chapter. Some transitions can be omitted depending on the system and customer requirements.

S88 states and transitions

Output state	Command	Result state	Notes
Ready	Start	Starting	Initial status
Starting	-	Running	The status changes from "Starting" to "Running" automatically and continues for one cycle without user intervention. The transition can be delayed if necessary via the `u.CTRL.xRemainStarting` flag.
Running	Hold	Holding	
Holding	-	Held	The status changes from "Holding" to "Held" automatically and continues for one cycle without user intervention. The transition can be delayed if necessary via the `u.CTRL.xRemainHolding` flag.
Held	Restart	Running	
Running	Pause	Pausing	
Pausing	-	Paused	The status changes from "Pausing" to "Paused" automatically and continues for one cycle without user intervention. The transition can be delayed if necessary via the `u.CTRL.xRemainPausing` flag.
Paused	Restart	Running	
Running	Stop	Stopping	
Stopping	-	Stopped	The status changes from "Stopping" to "Stopped" automatically and continues for one cycle without user intervention. The transition can be delayed if necessary via the `u.CTRL.xRemainStopping` flag. Note: If an EOP is stopped, the sequencer switches automatically to the next step. Within the stopped EOP, the "Stopped" state is only maintained for one cycle.
Running	Cancel	Canceling	
Canceling	-	Canceled	The status changes from "Canceling" to "Canceled" automatically and continues for one cycle without user intervention. The transition can be delayed if necessary via the `u.CTRL.xRemainAborting` flag. Note: If an EOP is aborted, the sequencer switches automatically to the next step. Within the canceled EOP, the "Canceled" state is only maintained for one cycle.

S88 EOP

The AS sequencer uses a finite state machine to manage S88 states and transitions internally, which can also be used by user EOPs for read-only access.

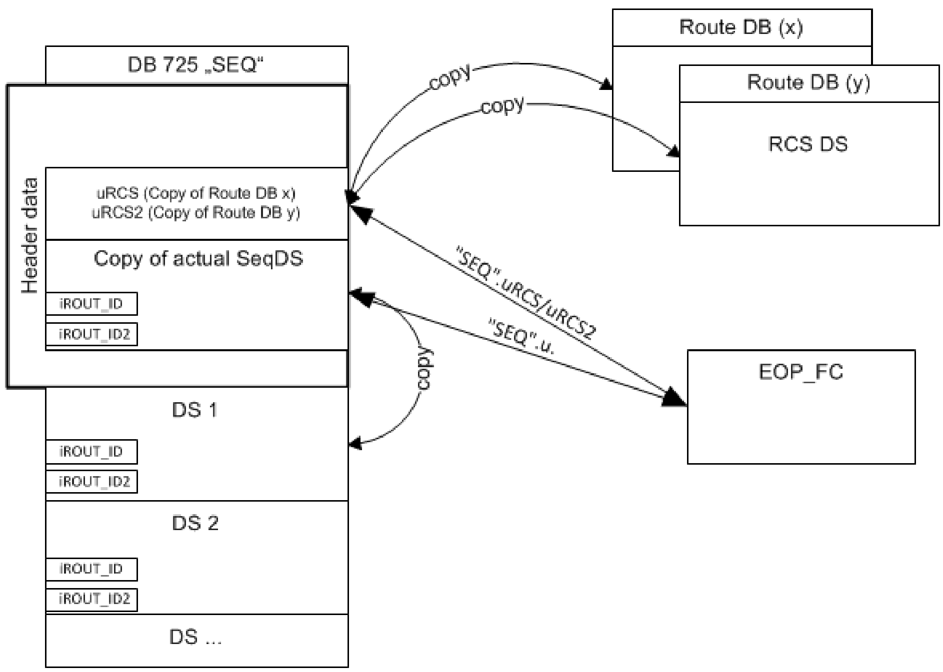
A detailed example is available in the supplier library (BM_USR: FC1002).

8.3.4 RCS interface

8.3.4.1 Overview

Structure of the sequencer data block "SEQ" (DB725) in connection with RCS.

The "SEQ".uRCS and "SEQ".uRCS2 **(PCU V7)** tags are stored in the header of the DB725 ("SEQ"). Data is exchanged with the route control via these tags.



Read data by the Route block

Prior to calling the EOP, specific data are read by the two route blocks Route DB x/Route DB y and written to the header data variables "SEQ".uRCS and "SEQ".uRCS2. The route blocks are determined through the variables "SEQ".u.iROUT_ID and "SEQ".u.iROUT_ID2. This is only the case when the respective value of the Route-ID is not zero.

Write data to the Route block

After having called the EOP, specific data from the header data variables "SEQ".uRCS and "SEQ".uRCS2 are written back in the two route blocks. The route blocks are determined through the variables "SEQ".u.iROUT_ID and "SEQ".u.iROUT_ID2. An allocation for the respective route block only takes place if

- "SEQ".uRCS.REQ or "SEQ".uRCS.RON display the value 1 → route block x
- "SEQ".uRCS2.REQ or "SEQ".uRCS2.RON display the value 1 → route block y

Route request with dynamic Route_ID allocation

The System can allocate the Route-ID to the time of the Route request. For this function, the Route-ID must have the value 0. The user program in the EOP must set the Route-ID to the value 0. The next time the EOP is called, a Route-ID is allocated by the system (see also: Dynamic Route ID allocation (Page 1084)).

Interface

uRCS / uRCS2 structure element	Description
FUNC_ID	Function-ID
TRANS_ORDER	Job number for the transport of the Route
REQ	Route Request
RON	Switch on Route
HOLD	Hold Route
IGN_ERR	Ignore error
SOLID	Route is a solid transport
ACK	Acknowledgement of time monitoring errors
GETXPE	Read External Parameter elements *)
SET_MAT	Set Material-ID
ROLLBCK_MAT	Reproduce material last used
MATCHK_DIS	Deactivate material check
SWITCH_MAT	Variant of the material order (see "RC_MAT" function)
TIME_VAL_MAT	Waiting time for OS material compatibility check
MATERIAL	ID of the transporting material
SOURCE	Source destination of the Route
VIA1 .. VIA10	Via destination of the Route
DEST	Target destination of the Route
MODE_TBL	Function catalog number of the Route
MODE_01 .. 32	Function control 1 to 32
QRET_VAL	Return value
QDIAG	Diagnosis value
QREQ_RC	Route request state
QINTERN	Internal (Automatic) External (Manual operation)
QREQ	Route request error
QON	Switch on display Route
QHOLD	Hold display Route
RESINT1	Reserved
QMODE_01 .. 32	Acknowledgement of the functions 1 to 32
QRESTPOS_01 .. 32	Position of rest error display of the functions 1 to 32
QMON_ERR_01 .. 32	Monitoring time error display of the functions 1 to 32
QFLT_ERR_01 .. 32	Error time display of the functions 1 to 32
QGRP_ERR_01 .. 32	Composite error display of the functions 1 to 32
QMON_TOU	Composite error display monitoring error times

uRCS / uRCS2 structure element	Description
QFLT_TOU	Composite error display error time
QERR	Composite error display
QMAT_ERR	Error with testing the material sequence
QMAT_OK	Test of the material sequence successful
QACTIVE	Route is activated
QXFER_OK	Material has been successfully incorporated in the connection elements
QXFER_ERR	Error when incorporating the material in the connection elements
GETXSTS	State of the external parameter elements *)
GETXDIAG	Diagnosis information for the state of the external parameter elements *)
IROUTE_STATE	Status of the Route
IRET_VAL_TRANS	Error number of the allocation block FB830
BoDATA_VALID	Allocated data are valid
SWITCH_MAT	Check/ set identification for Material
TIME_VAL_MAT	Max. runtime Material function
iRETVAL_MAT	Return value FB 836
RESINT4	Reserved
RESINT5	Reserved

Note

***) Allocating external parameter elements "ext.PE"**

The other data elements of the unit RCS interface notwithstanding, a block call "RC_IF_PE" (FC 810) must be made in order to provide the ext.PEs for each installed ext. PE!

For further details, see: RC_IF_PE (FC810) (Page 1155)

Further information on managing the RCS interface can be found in the "RCS block library" manual.

8.3.4.2 Example:

RCS unit interface

Example of an EOP-FC that requests a route including a material check:

```
UN    "EopStop"
=     "SEQ".uRCS.SET_MAT          // Enable material function with
EPE start

// parameters for the material check
L     "DFM3".au[3].rSPVal         // Setpoint for the material
T     "SEQ".uRCS.MATERIAL         // Load to the route (allow)
```

```
L     10                          // Parameters for material function
timeout
T     "SEQ".uRCS.TIME_VAL_MAT
L     4                             // Material function 4 = Check
material
                                    // and save it
T     "SEQ".uRCS.SWITCH_MAT

// Request route
SET
=     "SEQ".uRCS.REQ
=     "SEQ".uRCS.RON
=     "SEQ".uRCS.MODE_01

// Evaluate result
U     "SEQ".uRCS.QMAT_OK     // Transportation only once the material
has been saved
U     "SEQ".uRCS.QMODE_01    // and basic position OK
...
```

8.3.5 ICM interface

8.3.5.1 General

ICMs are addressed via the global flags. These flags are listed in the documentation 'Blocks S7'.

See also: section ICM - Individual Control Modules (ICM1 ICM2 ICM3 ICM4) (Page 269)

Table 8-3 There are flag interfaces for:

• CA	Control Automatic
• FOn	Feedback on
• FOff	Feedback off

8.3.6 Interface Digital Values

Digital values are addressed via the block SE-timer in the instance. This interface includes:

- Input signals
- Output signal positive
- Output signal negative

8.3.7 Interface to PE

Parameter elements can be addressed via the PE-blocks (DB97).

The allocation of the values must be made by the user program.

8.3.8 DFM Interface

Basics

The DFM setpoints are supplied by the recipe system. Process values are read via a source indication of the DFM-block. The source indication must be configured via the parameterization.

Result flag

The DFM result flags are stored in a global flag interface. These flags are described in the documentation DFM - Digital Function Modules (Page 234).

8.3.8.1 Call of the DFM processing

DFM processing must be called in the unit user FB **prior to EOP processing**. This is required, as there is only the guarantee at this location that the result flag has the correct value. This is especially important for the step change in PCU-cycle.

Note

To ensure that the DFM result is correctly established, the interface block must be called at least twice as often (in 500 ms OB) as unit user FB processing. Only then, you can guarantee that the process value in the DFM data set corresponds to the actual one at the source block.

8.3.9 Measured value interface

The measurement value interface includes:

- underride the lower limit
- override the upper limit

There is the possibility to determine the allocation of the flag in the description 'Block S7'.

See also: section AVA_PW (Page 320)

8.3.10 Controller interface

The controller interface includes:

- Correction flag YN

There is the possibility to determine the allocation of the flag in the description 'Block S7'.

See also: section PID controller (Page 329)

8.3.11 Interface to the batch system

Sequencer and EOP can influence the order system of the server via the interface FCs. Possible actions include:

- Release batch
- Disable batch
- Set batch to "ready"
- Create batch

<table>
<tr><th colspan="3">PCU < V7</th><th colspan="3">PCU V7</th></tr>
<tr><td rowspan="2">FC714</td><td rowspan="2">SEND_TEL_15</td><td rowspan="2">Combination block for all the actions named above</td><td>FC720</td><td>BmBatchChangeStateFC</td><td>Change batch status</td></tr>
<tr><td>FC721</td><td>BmBatchCreateFC</td><td>Create batch</td></tr>
</table>

Identifying a batch

A batch is uniquely identified based on the following properties:

- Batch year
- Recipe category
- Order number
- Batch number

If these properties are not fully known at runtime, the batch selection can also be dynamic by assigning the value 0 to unknown properties as parameters for the STEP 7 function. The list of possible batches resulting implicitly from this procedure can be restricted by other parameters. If several batches match the selection criteria, the batch highest in the batch list of the order system is used.

Functions and Identifiers

Function	Identifier
Release batch	50
Disable batch	51
Set batch to "ready"	52

Comparison operators

Operator	Identifier
=	1
<>	2
>	3
>=	4
<	5
<=	6

Parameter description

For PCU < V7: FC714

Parameter	Data type	Description
byYear	BYTE	Batch year
byRecType	BYTE	Recipe category
iRecipe	INT	(is not used)
iOrder	INT	Order number
iBatch	INT	Batch number
iRecord	INT	(is not used)
iNachricht	INT	Identifier of the required function (see above)
iSpaltenIdx	INT	Describes the origin of the actual value: 1-200: Actual value is the process value of the order parameter with the specified index 1000: Actual value is the line number being used otherwise: Invalid
byStartmode	BYTE	(is not used)
byCondition	BYTE	Identifier of the required comparison operator (see above)
wValue_High	WORD	Value for the comparison
wValue_Low	WORD	

For PCU V7: FC720

Parameter	Data type	Description
BatchYear:	BYTE	Batch year
RecCat	BYTE	Recipe category
Order	INT	Order number
Batch	INT	Batch number
NewState	INT	Identifier of the required function (see above)

Parameter	Data type	Description
CompSe-lector	INT	Describes the origin of the actual value: 1-200: Actual value is the process value of the order parameter with the speci-fied index 1000: Actual value is the line number being used otherwise: Invalid
CompType	INT	Identifier of the required comparison operator (see above)
CompVal-ue	DINT	Value for the comparison

8.3.11.1 Create batch

Batch data

To create a batch, the following parameters are necessary:

- Batch year
- Recipe category
- Order number
- Batch number
- Initial batch status following creation
- Start mode
- Line number (if used)

Initial batch statuses

Status	Identifier
Release	82
Ready	86
Disabled	76

Start types

Start mode	Identifier
Soon as possible	2
by time	3
by event	4
by time with automat-ic time adaptation	5

Parameter description

For PCU < V7: FC714

Parameter	Data type	Description
byYear	BYTE	Batch year
byRecType	BYTE	Recipe category
iRecipe	INT	Recipe number
iOrder	INT	Order number
iBatch	INT	Batch number
iRecord	INT	(is not used)
iNachricht	INT	60 (identifier for "Create batch")
iSpaltenIdx	INT	Line number (if no lines are used: 0)
byStart-mode	BYTE	Start type (see above)
byCondi-tion	BYTE	Initial batch status (see above)
wVal-ue_High	WORD	Start time (seconds since 1.1.1970)
wVal-ue_Low	WORD	

For PCU V7: FC721

Parameter	Data type	Description
BatchYear	BYTE	Batch year
RecCat	BYTE	Recipe category
Order	INT	Order number
Batch	INT	Batch number
Recipe	INT	Recipe number
LineNo	INT	Line number (if no lines are used: 0)
InitialState	INT	Initial batch status (see above)
StartMode	INT	Start type (see above)
StartDay	DATE	Start time Alternative format is used if StartUnixTime = 0
StartTime	TOD	
StartUnix-Time	DINT	Start time (seconds since 1.1.1970)

8.3.12 Engineering of alternative branch for recipe

To insert an alternative in the recipe procedure, a specified EOP must be engineered as alternative producer. This must be made in the recipe editor as well as in the EOP engineering.

The EOP before the recipe alternative must produce a result. As a value, the number of this alternative that was allocated in the recipe procedure is used. The value must be written to the global variable `"SEQ".u.byAlterResult (PCU < V7: DB725.DBB177, PCU V7: DB725.DBB193).`

Valid range for the alternative result: 1 - 255.

8.3.13 Engineering of jumps in the recipe

Only valid for PCUs of type Simatic S7

To insert a jump in the recipe procedure a specified EOP must be engineered as label producer. Engineering must be implemented both at the ROP in the recipe editor and in the EOP FC.

See also: Recipe system Configuration of the plant data

A jump destination must be calculated in the EOP before the recipe jump. The calculation of the result is plant specific. The result must be written to the global variable `"SEQ".u.byAlterResult`.

If no matching label to the result is found in the recipe unit procedure, the following step to the jump is executed. The jump is made in one PCU cycle.

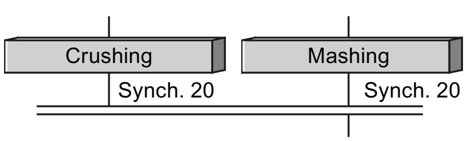
The value range for the result is 1 – 255.

8.3.14 EOP followed by an AND synchronization

Topic

If an EOP is followed by an AND synchronization, its processing may continue even though it has met the conditions for the next step.

Example:



Suppose the "Rough grinding" and "Mashing" EOPs had "A TUET" as the next step condition (continue to next step when step supervising time is reached), and five seconds was configured as the monitoring time for "Rough grinding" but only three seconds for "Mashing". "Mashing" is thus "ready" faster than "Rough grinding".

Resultant states :

	EOP "Rough grinding"			EOP "Mashing"		
Time index (sec)	Eop-Start	EopStop	S88 status	Eop-Start	EopStop	S88 status
0	1	0	Idle	1	0	Idle
1	0	0	Running	0	0	Running

	EOP "Rough grinding"			EOP "Mashing"		
2	0	0	Running	0	0	Running
3	0	0	Running	0	0	Running
4	0	0	Running	0	0	Completing
5	0	0	Running	0	0	Running
6	0 0	0 1	Completing Completing	0 0	0 1	Running Completing

As shown here, the "faster" EOP is still processed by the recipe system until the "slower" EOP is ready, which is entirely reasonable for heating operations, for example: if the temperature drops below the setpoint, heating must be restarted. The program logic of the EOP must be prepared to continue processing, regardless of whether it has met the step condition.

Note

Note: Recipe steps following a logical AND synchronization are not processed unless all EOPs participating in the synchronization process simultaneously report that they have met the conditions for the next step.

8.3.15 User step log

Core statement

The step log entries generated by the system are generated only once and permanently at the end of EOP.

It may be especially desirable for operations that run longer to record additional step log data records also during EOP runtime.

Application case:

For example, to enable a detailed evaluation of the recipe setpoints/actual values in the step report tables of a connected SQL database later, additional step telegrams can be triggered at any time by calling the "`BmTriggerUsrStepLogFC [FC 771]`" function in the EOP FCs.

Example of an additional step log entry at EOP start:

```
... EOP-code
Network 2: Trigger User Step protocol
U     "EopStart"                                   -- EOP start command
      SPBN  NWE2
      CALL  "BmTriggerUsrStepLogFC"            -- FC 771
        SubID  := 100                          -- User ID (must be > 0 !)
        RET_VAL :=#xRetVal                     -- OK: true, Not OK: false
NWE2: NOP  0
```

Example of an additional step log entry at EOP start:

```
...
Network x: Next Step Cont
...
```

Note

Call of the "BmTriggerUsrStepLogFC [FC 771]"

- **The FC 771 may only be called in the program code of EOP FC or unit FBs**.
- A call outside this program context generates step logs that can no longer be clearly assigned to the unit/an EOP.
- You call the FC 771 "edge-triggered", because a step log telegram is triggered with each processing.
- The transfer parameter `"SubID"` identifies the entry as user log within the EOP and must **be unique and > 0**, which means multiple telegrams in on EOP must receive different "SubIDs" (e.g. continuous).
- If `"SubID"` = 0 is transferred, a step log is not triggered and the return parameter `"RET_VAL"` = 0

See also

Functionality (Page 929)

8.4 User interface blocks

8.4.1 Block overview

The following blocks are specified for use by the user

PCU < V7

Block	Representation	Function
FB1001-FB1064	TA_XX_GOP_FB	User interface for unit processing
FB1200	CRST_USR_FB	User interface OB 100 (cold/warm restart)
FB1201	WRST_USR_FB	User interface OB 101 (restart)
FB1205	CAS_USR_FB	User interface for batch order start on AS Based on the transferred data set number, the user can determine which CAS and therefore for which unit data is available.
FB1220 FB1221	CYCLE_BEGIN_USR_FB CYCLE_END_USR_FB	User interface for start and end OB1 cycle
FB1222 FB1223 FB1224	TIMER_100MS_BEG_USR_FB TIMER_100MS_END_USR_FB TIMER_100MS_USR_FB	User interface for start, processing and end of the OB35 cycle. The intermediate processing follows the reading in of the process image of the inputs.
FB1225	TIMER_1S_USR_FB	User interface OB32
FB1226 FB1227 FB1228 FB1229 FB1230 FB1231 FB1232 FB1233	ESG1_BV1_128 ESG1_BV129_255 ESG2_BV1_128 ESG2_BV129_255 ESG3_BV1_128 ESG3_BV129_255 ESG4_BV1_128 ESG4_BV129_255	User interface interlock ICM
FC700	INPUT_FC	User interface patching the process inputs (I/M)
FC701	OUTPUT_FC	User interface patching the process outputs (Q/M)
FC726	ESG_IN_OUT_FC	User interface patching the ICM process inputs/outputs

PCU V7

Block	Representation	Function
FB1001-FB1128	BmUsrSeq001 - BmUsrSeq128	User interface for unit processing
FB1200	BmUsrWarmRestartFB	User interface OB 100 (cold/warm restart)
FB1201	BmUsrHotRestartFB	User interface OB 101 (restart)
FB1202	BmUsrColdRestartFB	User interface OB 102 (cold restart)

Block	Representation	Function
FB1205	BmUsrCasFB	User interface for batch order start on AS Based on the transferred data set number, the user can determine which CAS and therefore for which unit data is available.
FB1220 FB1221	BmUsrCycleBeginFB BmUsrCycleEndFB	User interface for start and end OB1 cycle
FB1222 FB1223 FB1224	BmUsr100msBeginFB BmUsr100msEndFB BmUsr100msFB	User interface for start, processing and end of the OB35 cycle. The intermediate processing follows the reading in of the process image of the inputs.
FB1225	BmUsr1000msFB	User interface OB32
 FB1226 FB1227 FB1228 FB1229	 BmUsrIcm1LockFB BmUsrIcm2LockFB BmUsrIcm3LockFB BmUsrIcm4LockFB	User interface operation interlocking ICM: BV ICM1.1 ... ICM1.256 BV ICM2.1 ... ICM2.256 BV ICM3.1 ... ICM3.256 BV ICM4.1 ... ICM4.256
FB1230	BmUsr200msFB	User interface for OB34
FB1231	BmUsr500msFB	User interface for OB33
FC700	BmUsrRouteDataInFC	User interface patching the process inputs (I/M)
FC701	BmUsrRouteDataOutFC	User interface patching the process outputs (Q/M)
FC726	BmUsrIcmRouteDataInOutFC	User interface patching the ICM process inputs/outputs

8.4.2 Allocation Blocks: FC700 FC701 FC 726

These blocks enable a free assignment of the user input/output and flag interfaces to the technological blocks.

- In the delivery state, the routing blocks FC 700 / FC 701 / FC 726 perform the default routings based on the default system allocation.
- If this default system allocation is not possible, the user must adapt the routing blocks FC 700 / FC 701 / FC 726. Thus, the user is able to connect any existing inputs/outputs of the system to the technological blocks via these free interfaces.
- The routing block FC 726 contains the routings for the ICM block. It is called by routing blocks FC 700 / FC 701 with the corresponding parameters.
- All other routings are directly contained in the networks of the FC 700 / FC 701 blocks.

Note

Address information in the system documentation

All input/output/flag information in this system documentation refers to the default allocation.

A routing is performed for the following blocks:

Name	Function	Assigned DB
SEQS	Sequence chain start block	DB 614
AOUT	Analog value output	DB 731
THRE-STEP	Three-step controller	DB 744
ICM	Individual control module (except RA)	DB 601, DB 602, DB 603, DB 604, DB 605
SeqMan-Mode	Manual release block	DB 701
MVC	Measured value control	DB 728
MSG	Message block	DB 615
AIN	Measured value recording	DB 727
MULT	Multi-functional block	DB 732
PID	Controller	DB 730
TIMER	Switch-on delay, impulse	DB 724
Sequences	Units	DB 612, DB 613

9 AS configuration with STEP 7 TIA Portal (S7-1500)

9.1 Free system resources for users

The system uses a variety of different CPU block and flag areas. Access by user programs to these and to other resources reserved for future or project-specific expansions is locked.

The tables below therefore show the resources available to the user:

Table 9-1 PCU type S7-1500

	CPU 151x	
Type	CPU range	User
FB	0 - 65.535	>= FB3000
FC	0 - 65.535	>= FC1001
DB	1 - 59.999	>= DB2500
MB	0 - 16383	>= MB2048
Timer (Times)	T0 - T2047	T256 ... T509 T512 ... T2047
Counter	Z0 - Z2047	Z0 - Z2047
Total number of elements Blocks (OB/FB/FC/DB) and UDTs	10000	...

9.2 User unit FC

Basics

The system control enables simultaneous operation of up to 128 SEQUENCES (units) per PCU. The commands and next step conditions in the user program of the technical operations (EOP-FCs) are processed recipe-controlled.

Integral components of the SEQUENCES system block are up to 24 "Digital Function Modules" (DFM), which can be configured as a timer, up counter, down counter, limit comparator, setpoint comparator, router or decoder, as a mask or as a setpoint/actual value cell. The respective setpoint or limit values are stored in the recipe lists.

Programming

A user block of Function type in which the corresponding permanent conditions, interlocks, start commands, etc., are programmed is assigned to each unit.

The user sequence FCs for all 128 units are available as a template in the delivered project in folder "...\Runtime Interfaces\Sample process cell 1\Sequence xxx - yyy\Sequence..." . The following block numbers have been specified for this:

- BmUsrSeq001FC [FC3001] / Sequence 1
- ...
- BmUsrSeq128FC [FC3128] / Sequence 128

The sequence user FC is called by the system before and after the EOP-FC using the BmUsrCallIfSeqDistributorFC [FC731] block.

Whether the current call is being made before or after the EOP-FC can be differentiated as follows:

<table>
<tr><th>Alternative 1 (default / delivered project for S7-1500):
The sequencer data record #SeqData is used to evaluate the call before or after the EOP-FC.</th><th>Alternative 2 (earlier method):
The "SeqFbCalledAfterEop" flag (=M104.4) is used to evaluate the call before or after the EOP FC.</th></tr>
<tr><td><pre>BEGIN
// Sequence permanent condition
"SEQ1-PermCond" := TRUE;
// Actions before EOP will be called
IF (#SeqData.xCallEnvBeforeEop) THEN
     ;
END_IF;
// Actions after EOP has been called
IF (#SeqData.xCallEnvAfterEop) THEN
     ;
 END_IF;
END_FUNCTION</pre></td><td><pre>BEGIN
// Sequence permanent condition
"SEQ1-PermCond" := TRUE;
// Actions before EOP will be called
IF NOT "SeqFbCalledAfterEop" THEN
     ;
END_IF;
// Actions after EOP has been called
IF "SeqFbCalledAfterEop" THEN
     ;
END_IF;
END_FUNCTION</pre></td></tr>
</table>

Call of the DFM processing

The DFM processing must be called in the user unit FC **before the EOP processing**. This is necessary because only then is it guaranteed the result flag has the current value at the time of the EOP processing. This is especially important for a step change in the AS cycle.

Note

DFM0 class

To guarantee that the DFM0 result is formed correctly, the interface block must be called at least twice as often (thus in the 500 ms OB), as the user unit FC processing. Only then can it be guaranteed that the actual value in the DFM data also corresponds to the actual value on the source block.

Flag interfaces between unit control and user unit FC

This flag interface is identical for user unit FC and EOP FC and is described in section Global flag interface (Page 655).

Flag interface between unit instance and user unit FC

The sequence control has the following globally defined flags per sequence:

SEQ#-Perm-Cond	**Permanent condition** is generated in the user unit FB. =0 means that the sequence cannot be started	M 640.0 – M 647.7 M 648.0 – M 655.7	Control by user program / Evaluated by system
SEQ#-IsRunning	**Operation sequence start** Is shown by the sequence control =1 Unit is running (step <> 0) =0 Unit is not running (step = 0)	M 656.0 – M 663.7 M 664.0 – M 671.7	Control by system / Evaluated by user program

Assignment of flag interfaces for sequence 1...64

Sequence	SEQ#-Perm-Cond	SEQ#-IsRunning	Sequence	SEQ#-Perm-Cond	SEQ#-IsRunning
1	M 640.0	M 656.0	33	M 644.0	M 660.0
2	M 640.1	M 656.1	34	M 644.1	M 660.1
3	M 640.2	M 656.2	35	M 644.2	M 660.2
4	M 640.3	M 656.3	36	M 644.3	M 660.3
5	M 640.4	M 656.4	37	M 644.4	M 660.4
6	M 640.5	M 656.5	38	M 644.5	M 660.5
7	M 640.6	M 656.6	39	M 644.6	M 660.6
8	M 640.7	M 656.7	40	M 644.7	M 660.7
9	M 641.0	M 657.0	41	M 645.0	M 661.0
10	M 641.1	M 657.1	42	M 645.1	M 661.1

Sequence	SEQ#-Perm-Cond	SEQ#-IsRunning	Sequence	SEQ#-Perm-Cond	SEQ#-IsRunning
11	M 641.2	M 657.2	43	M 645.2	M 661.2
12	M 641.3	M 657.3	44	M 645.3	M 661.3
13	M 641.4	M 657.4	45	M 645.4	M 661.4
14	M 641.5	M 657.5	46	M 645.5	M 661.5
15	M 641.6	M 657.6	47	M 645.6	M 661.6
16	M 641.7	M 657.7	48	M 645.7	M 661.7
17	M 642.0	M 658.0	49	M 646.0	M 662.0
18	M 642.1	M 658.1	50	M 646.1	M 662.1
19	M 642.2	M 658.2	51	M 646.2	M 662.2
20	M 642.3	M 658.3	52	M 646.3	M 662.3
21	M 642.4	M 658.4	53	M 646.4	M 662.4
22	M 642.5	M 658.5	54	M 646.5	M 662.5
23	M 642.6	M 658.6	55	M 646.6	M 662.6
24	M 642.7	M 658.7	56	M 646.7	M 662.7
25	M 643.0	M 659.0	57	M 647.0	M 663.0
26	M 643.1	M 659.1	58	M 647.1	M 663.1
27	M 643.2	M 659.2	59	M 647.2	M 663.2
28	M 643.3	M 659.3	60	M 647.3	M 663.3
29	M 643.4	M 656.4	61	M 647.4	M 663.4
30	M 643.5	M 659.5	62	M 647.5	M 663.5
31	M 643.6	M 659.6	63	M 647.6	M 663.6
32	M 643.7	M 659.7	64	M 647.7	M 663.7

Assignment of flag interfaces for sequence 65...128

Sequence	SEQ#-Perm-Cond	SEQ#-IsRunning	Sequence	SEQx-Perm-Cond	SEQx-IsRunning
65	M 648.0	M 664.0	97	M 652.0	M 668.0
66	M 648.1	M 664.1	98	M 652.1	M 668.1
67	M 648.2	M 664.2	99	M 652.2	M 668.2
68	M 648.3	M 664.3	100	M 652.3	M 668.3
69	M 648.4	M 664.4	101	M 652.4	M 668.4
70	M 648.5	M 664.5	102	M 652.5	M 668.5
71	M 648.6	M 664.6	103	M 652.6	M 668.6
72	M 648.7	M 664.7	104	M 652.7	M 668.7
73	M 649.0	M 665.0	105	M 653.0	M 669.0
74	M 649.1	M 665.1	106	M 653.1	M 669.1
75	M 649.2	M 665.2	107	M 653.2	M 669.2
76	M 649.3	M 665.3	108	M 653.3	M 669.3
77	M 649.4	M 665.4	109	M 653.4	M 669.4
78	M 649.5	M 665.5	110	M 653.5	M 669.5

Sequence	SEQ#-Perm-Cond	SEQ#-IsRunning	Sequence	SEQx-Perm-Cond	SEQx-IsRunning
79	M 649.6	M 665.6	111	M 653.6	M 669.6
80	M 649.7	M 665.7	112	M 653.7	M 669.7
81	M 650.0	M 666.0	113	M 654.0	M 670.0
82	M 650.1	M 666.1	114	M 654.1	M 670.1
83	M 650.2	M 666.2	115	M 654.2	M 670.2
84	M 650.3	M 666.3	116	M 654.3	M 670.3
85	M 650.4	M 666.4	117	M 654.4	M 670.4
86	M 650.5	M 666.5	118	M 654.5	M 670.5
87	M 650.6	M 666.6	119	M 654.6	M 670.6
88	M 650.7	M 666.7	120	M 654.7	M 670.7
89	M 651.0	M 667.0	121	M 655.0	M 671.0
90	M 651.1	M 667.1	122	M 655.1	M 671.1
91	M 651.2	M 667.2	123	M 655.2	M 671.2
92	M 651.3	M 667.3	124	M 655.3	M 671.3
93	M 651.4	M 667.4	125	M 655.4	M 671.4
94	M 651.5	M 667.5	126	M 655.5	M 671.5
95	M 651.6	M 667.6	127	M 655.6	M 671.6
96	M 651.7	M 667.7	128	M 655.7	M 671.7

Standard routing between sequence control and flag interface

The sequence control does not access the global flags directly, but takes them from interface blocks. The FC700/701 programs perform the routing between the flags and interface DBs.

If the global flags are located elsewhere, the application programmer must change the above-named FC.

- BmSeqPermCondDB [DB612] → Unit permanent condition
- BmSeqRunningDB [DB613] → Unit start

DBB x	DBB x.7 ... DBB x.0	DBB x	DBB x.7 ... DBB x.0
DBB10	Sequence 8 ... 1	DBB18	Unit 72 ... 65
DBB11	Sequence 16 ... 9	DBB19	Unit 80 ... 73
DBB12	Sequence 24 ... 17	DBB20	Unit 88 ... 81
DBB13	Unit 32 ... 25	DBB21	Unit 96 ... 89
DBB14	Unit 40 ... 33	DBB22	Unit 104 ... 97
DBB15	Unit 48 ... 41	DBB23	Unit 112 ... 105
DBB16	Unit 56 ... 49	DBB24	Unit 120 ... 113
DBB17	Sequence 64 ... 57	DBB25	Unit 128 ... 121

9.3 User EOP FC

Basics

- An EOP must be in the AS for this system.
- Up to 1999 EOP FCs can be defined for the maximum 128 units. Two function templates are available in the SCL programming language in delivered project "BM_S7_1500_xxxxxxxx" in folder "...\Runtime Interfaces\Sample process cell 1\!EOP template".
  - BmUsrEop0001FC [FC 1001] → SCL programming language
  - BmUsrEop0002FC [FC 1002] → STL programming language
- For all other EOPs used in the PCU, the number range [FC1001] to [FC2999] should also be used for PCU type S7-1500. However, this is not mandatory because a special jump distributor FC that represents the call interface to the system is used for this PCU type.

Overview of EOP interfaces

An EOP essentially has the following interfaces to other objects:

Interface	STEP 7
Sequencer interface	1. Global flags → see below 2. `#SeqData` → see below
Individual control modules of the unit (ICMs)	The interface to the ICMs consists of: • BA Automatic control • RE Feedback On • RA Feedback Off See also section ICM - Individual Control Modules (ICM1 ICM2 ICM3 ICM4) (Page 269)
Digital values (TIMER1/2)	The interface to the timer blocks consists of: • Input signals • Output signal positive • Output signal negative See also section TIMER - switch-on delay/impulse (Page 377)
Setpoints of the unit (DFMs)	• The DFM setpoints are supplied by the recipe system. • The actual values are read by the DFM block using a specified source. • The source specification must be configured in the parameter assignment. • The DFM result flags are stored in a global flag interface. See also section DFM1 - DFM4 - timers, limits, decoders (DINT) (Page 241) and DFM5 - DFM8 - timers, limits, decoders (REAL) (Page 254)

Interface	STEP 7
Measured values (AIN)	The interface to the measured value blocks consists of: • Lower limit fallen below • Upper limit exceeded See also section AIN - Measured value recording (Page 310)
Controller (PID)	The interface to the controller blocks consists of: • Tracking flag YN See also section PID controller (Page 329)
Order system	Sequencer and EOP can influence the order system of the server via interface FCs. 1. BmBatchCreateFC [FC721] - Create batch (see section Create batch (Page 668) 2. BmBatchChangeStateFC [FC720] - Change batch status (see section Interface to the batch system (Page 666) • Release batch • Lock batch • Set batch to "ready"
...	

Global flag interface

Symbol	Comment	Address	Access
HornMonTmError	Horn event signal Set when monitoring time expires. Handled and reset by the user	M 99.4	R1/R
EopMonTmError	Monitoring time EOP Result indicator of monitoring time of current step 0 = Time not expired 1 = Time expired	M101.0	R
EopDelayTmActive	The wait/delay time is started after EOP start 0 = Time running 1 = Time expired On a signal change from 0 to 1 (indication "-" to "+", from STOP to RELEASE), the time is retriggered.	M 101.1	R
PrintStepProt	Trigger for automatic printout of step protocols. On EOP occurrence, PrintStepProt is always = 0.	M 101.2	S1/R1 W
SeqFbCalledAfterEop	Flag for unit FB: 0 = Call of the FB before EOP processing (→ only DFM processing here!) 1 = Call of the FB after EOP processing	M 101.4	R
AddDeviceOn	Additional device On/Off	M 101.5	W
OperatorRequest	Set operation request	M 101.6	W

Symbol	Comment	Address	Access
EnableMonTm	Enable a message when step monitoring time exceeded. The sequence control always sets this bit to 1. The user program (EOP) can reset it. 1 = Indication and message when step time supervising exceeded. 0 = No indication and message when step time supervising exceeded.	M 101.7	W
AutomaticStep-Change	Sequence / automatic step change is enabled	M 102.0	R
SequenceIsRun-ning	The sequence is running. Sequence-specific flag bits: • Seq 1 ... Seq 64 : M 656.0 ... M 663.7 • Seq 65 ... Seq 128 : M 664.0 ... M 671.7	M 102.1	R
EopStart	First processing cycle of an EOP	M 102.2	R
EopStop	Last processing cycle of an EOP	M 102.3	R
SequenceStart-Pulse	Start pulse sequencer. This bit has value 1 the first time the sequence is passed through; otherwise the bit is always 0.	M 102.4	R
DisableStepProt	Disable report entry. 1 = No report entry at EOP end The sequence control DisableStepProt = 0 when the EOP starts	M 102.5	W
SeqPermCond	Permanent condition indication Sequence-specific flag bits: • Seq 1 ... Seq 64 : M 640.0 ... M 647.7 • Seq 65 ... Seq 128 : M 648.0 ... M 655.7	M 102.6	R
SeqManMode	Indication of manual mode of the sequence	M 102.7	R
HornGrpError	Horn event signal group flag is set when an ICM, AIN, MSG and sequence error occurs, handled and reset by the user	M 107.1	R1/R

Sequencer interface general

Tag	Comment	Access
`#SeqData.CTRL.boReload_DFM`	Re-transferring of setpoints to the technical parameters (DFMs)	S
`#SeqData.CTRL.xRemainHolding`	Retain the "Holding" state. The signal must be set by the EOP if the intermediate state "Holding" is to be retained on the "Hold" command. If the signal is withdrawn, the "Held" state is assumed.	S
`#SeqData.CTRL.xCmdHold`	"Hold" command.	L
`#SeqData.CTRL.xRemainRestarting`	Retain the "Restarting" state. The signal must be set by the EOP if the intermediate state "Restarting" is to be retained on the "Restart" command. If the signal is withdrawn, the "Running" state is assumed.	S

Tag	Comment	Access
SeqData.CTRL.xCmdRestart	"Restart" command.	L
#SeqData.CTRL.xIsRunning	EOP is in "Running" state.	L
#SeqData.CTRL.xRemainPausing	Retain the "Pausing" state. The signal must be set by the EOP if the intermediate state "Pausing" is to be retained on the "Pause" command. If the signal is withdrawn, the "Paused" state is assumed.	S
#SeqData.CTRL.xCmdPause	"Pause" command.	L
#SeqData.CTRL.xRemainAborting	Retain the "Aborting" state. The signal must be set by the EOP if the intermediate state "Aborting" is to be retained on the "Abort" command. If the signal is withdrawn, the "Aborted" state is assumed.	S
#SeqData.CTRL.xCmdAbort	"Abort" command.	L
#SeqData.CTRL.xRemainStopping	Retain the "Stopping" state. The signal must be set by the EOP if the intermediate state "Stopping" is to be retained on the "Stop" command. If the signal is withdrawn, the "Stopped" state is assumed.	S
#SeqData.CTRL.xCmdStop	"Stop" command.	L
#SeqData.CTRL.xSuppressOpStepChange	Preventing manual step change As long as the signal from the EOP is set, a manual step change by the operator is prevented • Process cell overview / "Select step" is prevented • Process diagrams / UnitCtrl / "Select step" is prevented • **The "EOP Stop" and "EOP Abort" commands are still possible!**	S

Batch and step information

The batch and step information can be read/written using the following tags in the sequencer data block.

Tag	Comment	Access
#SeqData.iNewStep	New step	R
#SeqData.byYear	Batch year	R
#SeqData.byRecType	Recipe type of the batch	R
#SeqData.iRecipe	Recipe of the batch	R
#SeqData.iOrder	Order number	R
#SeqData.iBatch	Batch number	R
#SeqData.snBA_Name	Batch name	R
#SeqData.diBA_ID	Batch ID	R

Tag	Comment	Access
#SeqData.diProduct_ID	Material ID for the product	R
#SeqData.diStep_ID	Step ID	R
#SeqData.byAlterResult	For an alternative or a jump in the recipe, an EOP must be configured as an alternative or jump destination producer. This configuration must be carried out on the ROP in the recipe editor and implemented in the EOP-FC. **Alternative result application case:** • An EOP (alternative producer) supplies the unit control the numerical value of the alternative to be selected (number 1 ... 255 from recipe procedure). • The value range for the result is 1 ... 255 (value 0 is invalid). **Jump destination application case:** • The jump destination (from recipe procedure) must be entered here. • If no suitable jump destination is found in the unit procedure, the step following the jump is executed. The jump is executed in one AG cycle. • The value range for the result is 1 ... 255 (value 0 is invalid).	W

Material sequence check

A matrix of the material sequence relationships, which is configured in the recipe editor under "Engineering/Material management/Material sequences" can be used to perform a check on the unit level to determine if the product of the source matches the product in the destination. A check for compatibility between the predecessor and successor material can be initiated via the following interface:

Tag	Comment	Access
#SeqData.STATUS.xMatCheckIsActive	Material compatibility check is running	W, R
#SeqData.STATUS.xMatCheckReset	Reset material compatibility check	W
#SeqData.diMatCheckPreID	Predecessor material	W
#SeqData.diMatCheckPostID	Successor material	W
#SeqData.diMatCheckCategory	(reserved)	
#SeqData.bMatCheckStatus	Status of the material compatibility check	R

- The user program sets the `#SeqData.STATUS.xMatCheckIsActive` bit in the EOP FC in order to activate the material check by the recipe server. The bit is reset by the recipe server when the material check has finished.
- The recipe server uses the `#SeqData.bMatCheckStatus` byte to return the result of the material check. Based on the result, a decision must be made in the EOP FC as to whether the recipe will be continued or, for example, an operation request will be issued.
  Results of the material check:
  0 = Ready
  1 → Material sequence permitted
  2 → Material sequence not permitted
  3 → Internal error
  4 → Material check is running

EOP creation with S88 states

In this section, S88 states and their transitions are described. Depending on the process cell and customer requirements, some transitions can be omitted.

Initial status	Command	Result status	Remarks
Idle	Start	Starting	Initial state
Starting	-	Running	The state change from "Starting" to "Running" occurs automatically and is present for one cycle without a user intervention. The `#SeqData.CTRL.xRemainStarting` signal can be used to delay the transition if necessary.
Running	Hold	Holding	
Holding	-	Held	The state change from "Holding" to "Held" occurs automatically and is present for one cycle without a user intervention. The `#SeqData.CTRL.xRemainHolding` signal can be used to delay the transition if necessary.
Held	Restart	Running	
Running	Pause	Pausing	
Pausing	-	Paused	The state change from "Pausing" to "Paused" occurs automatically and is present for one cycle without a user intervention. The `#SeqData.CTRL.xRemainPausing` signal can be used to delay the transition if necessary.
Paused	Restart	Running	
Running	Stop	Stopping	
Stopping	-	Stopped	The state change from "Stopping" to "Stopped" occurs automatically and is present for one cycle without a user intervention. The `#SeqData.CTRL.xRemainStopping` signal can be used to delay the transition if necessary. Note: If an EOP is stopped, the sequencer changes automatically to the next step. The "Stopped" state is present for only one cycle within the stopped EOP.

Initial status	Command	Result status	Remarks
Running	Abort	Aborting	
Aborting	-	Aborted	The state change from "Aborting" to "Aborted" occurs automatically and is present for one cycle without a user intervention. The `#SeqData.CTRL.xRemainAborting` signal can be used to delay the transition if necessary. Note: If an EOP is aborted, the sequencer changes automatically to the next step. The "Aborted" state is present for only one cycle within the aborted EOP.

EOP before AND synchronization

For more information, see section EOP followed by an AND synchronization (Page 670)

User step protocol

For more information, see section User step log (Page 671)

9.4 Call system for user unit/EOP FCs

Basics

For PCU type S7-1500, the calls of user unit and EOP FCs must each be handled in a call block to be configured by the user.

The "...\Runtime interfaces\03_Miscellaneous\" folder of the delivered project "BM_S7_1500_Vxxxxxxxx" contains the function templates described below:

Configuration of unit and EOP call functions

The following functions must be placed in the "Program blocks" folder of each PCU S7-1500 and supplemented as appropriate:

<table>
<tr><th>BmUsrCallIfEopDistributorFC [FC730]
- Call block for user EOP FCs</th><th>BmUsrCallIfSeqDistributorFC [FC731]
- Call block for user unit FCs</th></tr>
<tr><td>

Source code delivery state:

```
//V7.5.0.0
   VAR_INPUT
      EopFcNo : UInt;
   END_VAR
   VAR_OUTPUT
      NextStepCondition : Bool;
   END_VAR
   VAR_IN_OUT
      SeqData : "BmSeqUDT";
   END_VAR
BEGIN
 #NextStepCondition := FALSE;
 IF (#EopFcNo = 0) THEN
     RETURN;
 END_IF;
 CASE (#EopFcNo) OF
     997:
         "BmSeqLabelFC"(NextStepCondition =>
#NextStepCondition, SeqData := #SeqData);
     998:
         "BmSeqJumpFC"(NextStepCondition =>
#NextStepCondition, SeqData := #SeqData);
     999:
         "BmSeqNopFC"(NextStepCondition =>
#NextStepCondition, SeqData := #SeqData);
     1001:
         "BmUsrEop0001FC"(NextStepCondition
=> #NextStepCondition, SeqData := #SeqData);
 END_CASE;
END_FUNCTION
```

</td><td>

Source code delivery state:

```
//V7.5.0.0
   VAR_INPUT
      SeqNo : UInt;
   END_VAR
   VAR_IN_OUT
      SeqData : "BmSeqUDT";
   END_VAR
BEGIN
 // Illegal sequence numbers will fall
through
 CASE (#SeqNo) OF
     1:
         "BmUsrSeq001FC"(SeqNo := #SeqNo,
                        SeqData := #SeqData);
     2:
         "BmUsrSeq002FC"(SeqNo := #SeqNo,
                        SeqData := #SeqData);
     3:
         "BmUsrSeq003FC"(SeqNo := #SeqNo,
                        SeqData := #SeqData);
     4:
         "BmUsrSeq004FC"(SeqNo := #SeqNo,
                        SeqData := #SeqData);
     5:
         "BmUsrSeq005FC"(SeqNo := #SeqNo,
                        SeqData := #SeqData);
     6:
         "BmUsrSeq006FC"(SeqNo := #SeqNo,
                        SeqData := #SeqData);
     7: ...
     64:
       "BmUsrSeq064FC"(SeqNo := #SeqNo,
                        SeqData := #SeqData);
 END_CASE;
END_FUNCTION
```

</td></tr>
<tr><td colspan="2">Additions by the user (example):
The user FCs used in the project must be incorporated in the "CASE" instruction</td></tr>
</table>

<table>
<tr><th>BmUsrCallIfEopDistributorFC [FC730]
- Call block for user EOP FCs</th><th>BmUsrCallIfSeqDistributorFC [FC731]
- Call block for user unit FCs</th></tr>
<tr><td><pre>
     ...
     1005:
         "BmUsrEop0005FC"(NextStepCondition
=> #NextStepCondition, SeqData := #SeqData);

     1011:
         "BmUsrEop0011FC"(NextStepCondition
=> #NextStepCondition, SeqData := #SeqData);

     1014:
         "BmUsrEop0014FC"(NextStepCondition
=> #NextStepCondition, SeqData := #SeqData);

     1025: .....
</pre></td><td><pre>
     ...
     2:
         "BmUsrSeq002FC"(SeqNo := #SeqNo,
                         SeqData := #SeqData);
     3:
         "BmUsrSeq003FC"(SeqNo := #SeqNo,
                         SeqData := #SeqData);
     4:
         "BmUsrSeq004FC"(SeqNo := #SeqNo,
                         SeqData := #SeqData);
     7: ...
</pre></td></tr>
<tr><td>Important:
The input parameter <code>#EopFcNo</code> checked with the CASE instruction has a usable value range of 1001 ... 2999
The function range BmUsrEop0001FC [FC1001] - BmUsrEop1999FC [FC2999] is available for the EOP FCs.</td><td>Important:
The input parameter <code>#SeqNo</code> checked with the CASE instruction has a usable value range of 1 ... 128
The function range BmUsrSeq0001FC [FC3001] - BmUsrSeq128FC [FC3128] is available for the user unit FCs.</td></tr>
</table>

9.5 User interface blocks

9.5.1 Block overview

The following blocks are specified for use by the user

Folder \ Block	Function
...\01_Environment\BmUsrCallIf100msBeginFC [FC902] ...\01_Environment\BmUsrCallIf100msEndFC [FC904] ...\01_Environment\BmUsrCallIf100msFC [FC903]	User interface for start, processing and end of the OB35 cycle. The intermediate processing follows the reading in of the process image of the inputs.
...\01_Environment\BmUsrCallIf200msFC [FC905]	User interface for OB34
...\01_Environment\BmUsrCallIf500msFC [FC906]	User interface for OB33
...\01_Environment\BmUsrCallIf1000msFC [FC907]	User interface OB32
...\01_Environment\BmUsrCallIfCycleBeginFC [FC900] ...\01_Environment\BmUsrCallIfCycleEndFC [FC901]	User interface for start and end OB1 cycle
...\01_Environment\BmUsrCallIfIcmRouteDataInOutFC [FC726]	User interface for routing of ICM process inputs/outputs
...\01_Environment\BmUsrCallIfRestartFC [FC908]	User interface for OB 101 (restart)
...\01_Environment\BmUsrCallIfRouteDataInFC [FC700]	User interface for routing of process inputs (I/M)
...\01_Environment\BmUsrCallIfRouteDataOutFC [FC701]	User interface for routing of process outputs (Q/M)
...\02_Control modules\... BmUsrCallIfIcm1LockFC [FC910] BmUsrCallIfIcm2LockFC [FC911] BmUsrCallIfIcm3LockFC [FC912] BmUsrCallIfIcm4LockFC [FC913]	User interface for interlock ICM: BV ICM1.1 ... ICM1.256 BV ICM2.1 ... ICM2.256 BV ICM3.1 ... ICM3.256 BV ICM4.1 ... ICM4.256
...\03_Miscellaneous\BmUsrCallIfAinConvValueFC [FC732] ...\03_Miscellaneous\BmUsrCallIfAoutConvValueFC [FC733]	These functions are used as a template for the process value conversions to be made by the user in connection with the MESS/AIN and ANAU/AOUT classes.
...\03_Miscellaneous\BmUsrCallIfEopDistributorFC [FC730] ...\03_Miscellaneous\BmUsrCallIfSeqDistributorFC [FC731]	Configuration of unit and EOP call functions

Folder \ Block	Function
...\03_Miscellaneous\BmUsrBosFB [FB1205]	User interface for batch order start in the AS. Based on the transferred data set number, the user can determine which CAS and therefore for which unit data is available.
...\03_Miscellaneous\BmCurveSpTa-bleGroup1 [DB50]	Parameter assignment of the curve interpolation points This is done using the IOS application 'Curve entry' on the 'Configuration' tab of the main menu.

9.5.2 Allocation Blocks: FC700 FC701 FC 726

These blocks enable a free assignment of the user input/output and flag interfaces to the technological blocks.

- In the delivery state, the routing blocks BmUsrCallIfRouteDataInFC [FC700] / BmUsrCallIfRouteDataOutFC [FC701] / BmUsrCallIfIcmRouteDataInOutFC [FC726] perform the routings based on the default system allocation.
- If this default system allocation is not possible, the user must adapt these routing blocks. Thus, the user is able to connect any existing inputs/outputs of the system to the technological blocks via these free interfaces.
- The routing block BmUsrCallIfIcmRouteDataInOutFC [FC726] contains the routings for the ICM block. It is called by routing blocks BmUsrCallIfRouteDataInFC [FC700] / BmUsrCallIfRouteDataOutFC [FC701] with the corresponding parameters.
- All other routings are directly contained in the networks of the BmUsrCallIfRouteDataInFC [FC700] / BmUsrCallIfRouteDataOutFC [FC701] blocks.

Note

Address information in the system documentation

All input/output/flag information in this system documentation refers to the default allocation.

A routing is performed for the following blocks:

Block	Function
AOUT [DB731]	Analog value output
TSC [DB744]	Three-step controller
BmIcmCmdDB [DB601] BmIcmLockDB [DB602] BmIcmFbk1DB [DB603] BmIcmFbk0DB [DB604] BmIcmCtrlDB [DB605]	Individual control module: Command automatic ... Operation interlock ... Feedback On ... Feedback Off ... Load output
SYS [DB701]	Manual release block
MVC [DB728]	Measured value control
BmMsgCmdDB [DB615]	Message block

Block	Function
AIN [DB727]	Measured value recording
MULT [DB732]	Multi-functional block
PID [DB730]	PID controller
TIMER1 [DB724] TIMER2 [DB745]	ON delay pulse
BmSeqPermCondDB [DB612] BmSeqRunningDB [DB613]	Unit permanent condition (DB flag) Unit start (ATL flag)
BmSeqStartCmdDB [DB614]	Sequences: Sequence start

Batch processing 10

10.1 Definitions

10.1.1 Recipe

Recipe categories

The recipe system structures the master recipes in recipe categories. It is possible to create one order type for any recipe category. Basic recipes of the same category have the same process parameter definition and the same order type definition.

Master recipe

The master recipe is the description of how to produce a product. This description consists of header data, process parameters, and process input materials. The master recipe has a reference on a recipe procedure.

Process input list

All process inputs that are necessary for this recipe are defined in the list. Only the substance and the quantity are specified in the list. The storage location is defined by the time of the batch generation via the storage location administration.

Process parameters

All master recipes of a category have the same process parameter definition.

Recipe procedure

The units that are necessary for production and the sequence of steps are defined in the recipe procedure.

Recipe unit procedure

Description of the step sequence within a unit.

10.1.2 Batch

Basics

A batch is the conversion of a master recipe.

Batch list

Sequence of batches that shall be processed.

Control recipe

For the batch, a copy is generated from the recipe procedure. This copy is known as the control recipe.

Process parameters

The process parameters are not copied or stores into the batch data. They can serve as default values for the order parameters as well.

Process input list

When creating the batch the component list is copied in the control recipe. The storage locations are determined by the storage location administration and entered in the control recipe at this time. There is a process input for each batch that can be modified individually until the batch processing of this batch starts.

Order parameters and batch parameters respectively

The values of the order parameter can be entered in the dialog for creating batches. The order parameters can be preset with either fixed numerical values or process parameters.

Control recipe for recipe unit procedure

The control recipe is created on a unit-granular basis. This means that the control recipe is created for each recipe unit procedure at that time of starting the sequence.

All formal parameters of the recipe unit procedure are substituted by their actual values at this time. These are process parameters, order parameters, and process inputs.

Dosing

Dosing operations for all materials from the master recipe’s process input list are generated from the Dosing-ROPs when generating the control recipes.

10.1.3 Batch data

The data that is generated when creating and processing batches The following data is logged charge-related:

- Batch header data (number, basic recipe, point of time)
- Batch parameters
- Batch-related process input
- Batch history (start and end time of the batch)
- Step report (for every step: Start time, duration, all setpoints and actual values
- Messages
- Curve recording
- Plant-specific protocols of the batch

10.2 Introduction

10.2.1 Description of processing sequence

In the IOS a list of batch orders is stored.

One recipe and one line are assigned to each batch order so that a starting sequence is determined implicitly.

The processing of the order begins with starting this sequence. The start is implemented via the CAS-block (‘ChargenAuftragsStart’, the German word for Batch Order Start) in the PCU. This CAS-block has a data record. The order system supplies the data records of the CAS.

As soon as the CAS has started the start sequence, a message is sent to the order system. If more batch orders are available for this start sequence, the new job is loaded immediately into the CAS-block. The CAS waits until the sequence is released and starts the next order.

As long as the batch order is loaded into the CAS and the batch hasn't been started yet, the batch is in the status 'Ready to start'.

There are different starting-modes for the starting of batches:

- ‘As soon as possible’ (start as soon as sequence is released)
- ‘By time’ (start after time runs out and sequence is released)
- ‘By time with automatic time correction’ (delay of a previous batch is considered in the start times of the sequential batches)
- 'By event' (an event is created in the SIMATIC);
  for S7 the user function block CAS_USR_FB (FB1205) is called in CAS_FB; in this FB, the user can determine the event; when RLO = 1, the sequence is started

A file listing all batch orders (`<proj-path>\Bali\Bali.dbf`) is stored on the IOS. The processing occurs in the sequence of input and by starting time. Locked batches are simply skipped.

A batch order stays listed as long as it has not been processed. The user program in the PCU must create the batch end message. This happens automatically when processing the ‘END’-ROP.

Note

The batch process values can be sent to the IOS with the help of ‘free protocols’. As they are stored as dBase-files, a corresponding program, e.g., MS-Access, should be used for the evaluation.

10.2.2 Connection Recipe - Order

Core statement

The order system has the task of processing orders with one or more batches.

Order types are defined, whereby each order type is assigned to a definite recipe type.

Orders can then be entered for the configured order types.

The purpose of the order is to produce a product according to a specified recipe.

Parameters can also be defined for the order and incorporated into the recipe.

Master recipe

This is the aggregation of the

- **Process parameter list**
  A list of parameters specific for each master recipe that influence the execution of a control recipe.
- **Process input list**
  A list of materials that are necessary for producing the recipe; this list can be different from batch to batch.
- **Recipe procedure**
  Consists of the assigned RUPs that in turn consist of sequences of recipe operations with setpoints that are necessary for producing the recipe on a technical plant. In the recipe procedure setpoints can be defined as placeholders for values that are filled from the process parameter list, order/batch parameters or from the process input list. Furthermore setpoints can be configured to be scaled with the batch size.
  - One recipe procedure can be assigned to many master recipes of the same recipe category.
  - If a master recipe is selected for the production of a batch, always the combination of master recipe header data, process parameters, process input list, and recipe procedure is meant.
  - The recipe procedure is transformed into a batch-dependent control recipe at runtime. Due to the recipe request (order number, batch number, sequence number) of the PCU:
  - The corresponding recipe procedure will be determined.
  - The order system is queried for the batch size.
  - All setpoints in the recipe procedure which are to be scaled with the batch size are converted.
  - All setpoints which are supposed to be substituted by a batch parameter are processed.
  - All setpoints which are supposed to be substituted by a process parameter are processed.
  - All dosing-ROPs are expanded into the appropriate number of dosing and emptying operations for the materials found in the sequence.
  - The recipe which is adapted in the described way is called ‘control recipe’ and transferred into the PCU where it is executed.

10.2.3 Batch generation on the IOS

The order system generates the batches on the IOS.

Order and batch numbers must be unique within a year and recipe type. Due to this fact batch numbers cannot be allocated unrestrictedly by the user.

Before the first batch is generated, the order system must determine the number of batches that is necessary and which batch numbers are allocated.

Batch start

The starting mode for the batches is determined, too.

The starting modes that can be used with a certain order type are selected when the order type is configured. The checkbox for the starting mode of an order type can be found in the dialog ‘Project planning > Order type’ by pressing the ‘Start mode’-button.

Taking over batch parameters

With the generation of the batches all order parameters with the exception of the sum parameters are copied into the batch data.

A sum parameter is a part of an order parameter that is proportional to the size of a single batch. It is calculated following the given equation:

- Parameter (batch) = parameter(order) / size(order) * size(batch)
- An order parameter is configured as a sum parameter when the order type is defined. The sum parameter checkbox can be found in the dialog “Project planning > Order types’ by pressing the ‘Order parameters’-button.

10.2.4 Order parameters

Every batch order can include parameters.

The number of the parameters and their definitions need to be configured.

The configuration can be different for every order type.

10.2.5 Job list

There is no independent order list on the IOS-level.

However, an order view based on the batch list is offered.

In this deduced order list all batches of the same order number are summarized and represented in a line.

10.2.6 Batch start list

The batch start list determines the start sequence for the different lines (the entries are line-related).

Every line has a start sequence. The data is stored on parameterized disk drives redundantly.

The data is accessed via a DLL (BALIDATA) with the following functions:

- The interfaces for plant-specific expansions are exposed.
- The corresponding CAS and the start sequence are determined from the line number and the recipe when an order is created.

10.2.7 Batch monitoring

The batch monitoring displays the batch. The state is determined from the IOS-copies of the CAS and the sequence image.

In every line a batch is shown with the sequences that have already been running, the sequences that are running, and the following sequences according to the unit recipe (in different colors).

The sequences are combined in the application batch monitoring. The sequences processing-sequence is stored line-dependently.

The sequence image is updated via a general request at system startup.

The sequence control sends the current copy for all sequences after the general request has been received. Otherwise a telegram will be sent to the IOS only if a sequence’s state changes.

10.2.8 Batch parameters

The batch parameters are transferred in the control recipes instead of the placeholders in the recipe procedure (substitute by batch parameter).

In the “Batch management” application, you can display batches and their parameters in the form of a table. Each line represents one batch and its associated parameters. Consequently, it is only possible to display batches of the same type. The editor does not distinguish between future, present or past.

10.3 Procedure processing

10.3.1 Division between PCU and IOS

10.3.1.1 IOS-Functions

- Creates control recipes for recipe unit procedures and loads them into the PCUs
- Executes synchronizations
- Executes alternatives
- Sequence image
- Processes and stores step protocols

10.3.1.2 PCU-Functions

- Processes sequences
- Controls technical operations
- Controls the transitions in a PLC-cycle
- Executes jumps
- Updates sequence images
- Operating modes

10.3.1.3 Sequence image

Basics

When processing the recipe procedures, any change of a sequence’s state is signaled to the IOS. A copy of the image data is kept by the application ‘PCU_SERVER’. At system start the server requests the data by sending a ‘general request’.

The copy is updated by the PCU. The PCU sends telegrams of type 13 to signal a change in state.

The IOS recipe control uses the sequence image.

Visualization

The sequence image can be monitored in the recipe server monitoring window of the PCU_SERVER.

(See also: PCU Server dialog 'Recipe control' (Page 948))

10.3.2 Synchronizations

10.3.2.1 Processing synchronization

Processing in the PCU

If the step enabling conditions are met for the processing of an operation in the PCU (RLO=1) and the next step is a synchronization operation, then:

- The transition to the next operation is delayed; this is shown in the sequencer data set,
- a sequence state (type 13) is sent to the IOS,
- the operation preceding the synchronization is processed despite RLO=1 as long as the synchronization has not been met
- The message 'Synchronization achieved' can also be withdrawn again (operation is not exited with RLO=1). The sequence is described in section Step condition of an operation before the synchronization (Page 706).

Processing in the IOS

The recipe control creates a synchronization object after receiving this telegram.

This object is identified by synchronization number, year, recipe type, order, and batch number. Any further messages for the batch and the synchronization are stored and evaluated in this object.

With termination of the batch all synchronization objects are deleted for this batch.

In the object the states of the sequence that are waited for and that should be started are stored and evaluated.

If the synchronization is completed (all sequences that are waited for have reached their synchronization, all sequences that are to be started are free), all participating sequences are released. Sequences that need to be started with the synchronization are started by the recipe control.

Visualization

In the sequence control application the state 'Sequence waits at synchronization' is displayed as '='.

As an expanded possibility of diagnosis the state of the synchronization can be visualized via the application 'diagnosis batch control'.

Messages

If the synchronization contains a sequence that cannot be started, a message will be created (exception: The sequence is busy with another batch).

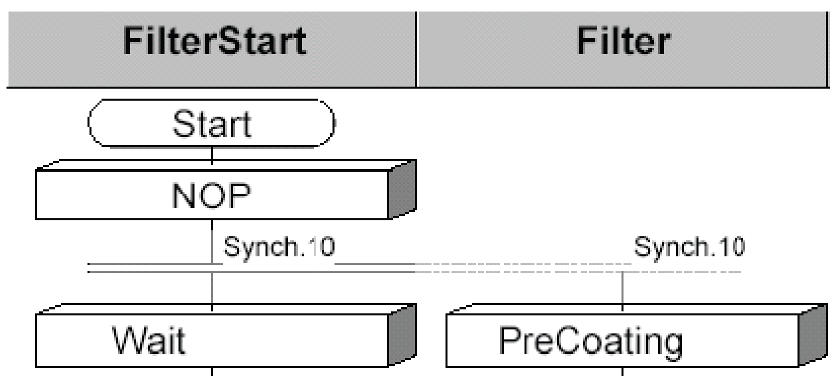
10.3.2.2 Completing a synchronization

If all sequences which are to be started have sent positive feedback, the transition is completed. The transitions of all sequences which participate in the synchronization are set. This is displayed in the sequence data record (relative address in the record D 28.13 with SIMATIC S5, D 52.5 with SIMATIC S7). The sequence processing continues and executes the next step.

10.3.2.3 Non-blocking synchronization

Non-blocking synchronization can only be used for starting sequences.

Function



If the synchronization 10 is met by processing the sequence 'FilterStart', it will be attempted to start the sequence filter. If this sequence is busy, in manual mode or the preselection hold is active, the start will be stored and executed later. The processing of the sequence 'FilterStart' is continued.

The processing at the synchronization 10 would stop and wait for the sequence 'filter' for a blocking synchronization.

All stored starts are deleted with batch end independent of whether they have been executed or not.

Note

To be able to use the non-blocking synchronizations, these must be enabled explicitly in "`...<proj-path>\recipe\project\plant.ini`" by setting "EnableNonBlockingSync=1" in the section "[Version4.60]".

10.3.2.4 Step condition of an operation before the synchronization

Step condition, "Standard"

With two successive EOPs ("EOP A" and "EOP B") the FCs are executed in the PCU as follows:

- *Cycle n:*
  Start processing EOP A ("EopStart" is set)
  ...
- *Cycle m:*
  Processing of EOP A, step enabling condition met ("RLO"=1)
  Last cycle of EOP A ("EopStop" is set)
  First processing of EOP B ("EopStart" is set)
  ...
- *Cycle m+1:*
  Processing of EOP B

Step sequence with synchronization

When the EOP is followed by a synchronization, the operating principle is the same, though the last EOP cycle with set "EopStop" is not executed unless the subsequent synchronization condition is met.

If the EOP step condition is not met within the time the synchronization condition is met, the synchronization is discarded!

- *Cycle n:*
  Start of EOP A processing ("EopStart" is set)
  ...
- *Cycle m:*
  Processing of EOP A, step enabling condition met ("RLO"=1)
  Verification of the synchronization condition by the recipe control function of the IOS
  ...
- *Cycle m+k:*
  Synchronization fulfilled (**this also means**: Step condition of EOP A is still met)
  Last cycle of EOP A ("EopStop" is set)
  Start of EOP B processing ("EopStart" is set)

Note:

In the second example above, if the step condition for EOP A is no longer valid after cycle m, the synchronization condition is also discarded, execution of the EOP is not run through and terminated with "EopStop" .

If technological aspects allow the step enabling condition to be met by one positive signal edge (e.g. temperature setpoint reached once, step enabling can also take place when the process value of the temperature drops below the setpoint again), the step enabling condition should be first saved to a flag bit, for example, and then reset with "EopStop".

10.3.3 Alternatives

10.3.3.1 Processing alternatives

Processing in the PCU

The operation preceding the alternative (alternative producer) transfers the alternative result in the global "SEQ".u.byAlterResult variable ((PCU < V7: DB725.DBB177, PCU V7: DB725.DBB193)).

If the transition (RLO=1) is met and if the next step is an alternative operation then:

- the transition is delayed; this is displayed in the sequence data record (relative address in the data record D 28.13 with SIMATIC S5, D 52.5 with SIMATIC S7),
- a sequence state (type 13) is sent to the IOS,
- the recipe operation before the alternative is processed, in spite of RLO=1 as long as the alternative hasn't been completed.

Processing in the IOS

When the telegram type 13 is received, an alternative object is created in the recipe control (IOS).

This object is identified by its alternative number, year, recipe type, order, and batch number. Any further messages for the batch and alternative are stored and evaluated in this object.

If the ‘End’-operation is processed in the PCU, all alternative objects of this batch are deleted.

Visualization

In the sequence control application the state 'Sequence waits at alternative' is shown as '_'.

As an expanded possibility of diagnosis the state of the alternative can be visualized via the application 'diagnosis batch control'.

See also section: Diagnosis recipe control 'RecContr' (Page 945)

Messages

If the alternative contains a sequence that cannot be started, a message will be created (exception: The sequence is busy with another batch).

10.3.3.2 Completing an alternative

If the sequence which is to be started has sent positive feedback, the transition is continued. Consequently, the step condition for the two units involved in the alternative are released again, this is displayed in the unit data set.

The sequence processing in the PCU goes on to the next step.

10.3.4 Jump Operation

The operation preceding the jump determines a jump destination before a jump is processed. This has to be implemented in the equipment operation. In the recipe editor the possible jump destinations are configured. The jump destination needs to be transferred in flag byte MB100 as well as in the global variable `"SEQ".u.byAlterResult (PCU < V7: DB725.DBB177, PCU V7: DB725.DBB193).`

Jumps in the recipe unit procedure can only be inserted following a jump destination producer.

If a jump is processed, the RUP is searched for a corresponding jump destination. Searching starts at the first step and is conducted sequentially.

If a corresponding jump destination is found, execution is branched to this step. If no corresponding destination is found, the sequence processing proceeds as if no jump operation was met.

The step is changed in the PLC-cycle.

10.3.5 General description of processing

The task of the recipe system can be divided into three units.

These are:

- the recipe-load-function,
- the recipe control
- and the recipe editor.

The PCU processes the sequence of operations. If a synchronization or alternative is reached, this will be signaled and the processing remains in this position. The processing will be enabled from the recipe control again, if the conditions are met.

Via synchronization any further sequences are started. Every sequence requests a recipe from the recipe-load-function before the processing starts.

The recipe processing ends with the last equipment operation of the last sequence. This is a special operation of ‘End’-type usually and it signals ‘End of batch’ to the IOS.

10.3.5.1 Recipe load function

- In the IOS a list of batch orders is stored. One recipe and one line are assigned to each batch order The start sequence can be determined from the recipe procedure header due to that.
- A batch order is loaded from the order control in the SeqStart block of the PCU. This block starts the sequence.
- Before the processing of the sequence starts, it requests a control recipe from the IOS. In the IOS the master recipe is determined by the recipe header.
- Order parameters, process parameters, and dosing operations are incorporated in the recipe procedure.

- This batch-dependent recipe procedure (=control recipe) is transferred to the PCU. The recipe control in the PCU processes the sequence of unit operations.
- If there is a dosing-operation in the recipe procedure, it will be inserted as often as materials from the process input list can be dosed by this unit. Dosing operations are inserted only if the sequence ID (weigher-unit) and the dose group (order of dosing) correspond.
- If the process input list yields an empty sequence input list, the system usually omits the resulting empty sequence, but a different behavior is possible if desired.

10.3.5.2 Determine start sequence

When an order is entered, it is determined with which master recipe and on which line the order should be processed.

The order control needs the starting sequence for the CAS in order to start a batch job.

The recipe procedure number is found in the master recipe header data.

The starting sequence is determined from the recipe procedure header.

If the line has the identification 'reference line', the determined sequence is the start sequence.

If the line has the identification 'parallel line', the start sequence must be determined via the sequence assignment table.

10.3.5.3 Connection of Process Input List and Recipe

In the process input list materials are stored for an individual master recipe.

A process input list is always needed if a recipe procedure contains dosing operations.

This list can come from a CIS. For every batch a list exists. The list (from the CIS) consists of CIS-process-input-ID and quantity.

In the IOS the CIS input list is supplemented with

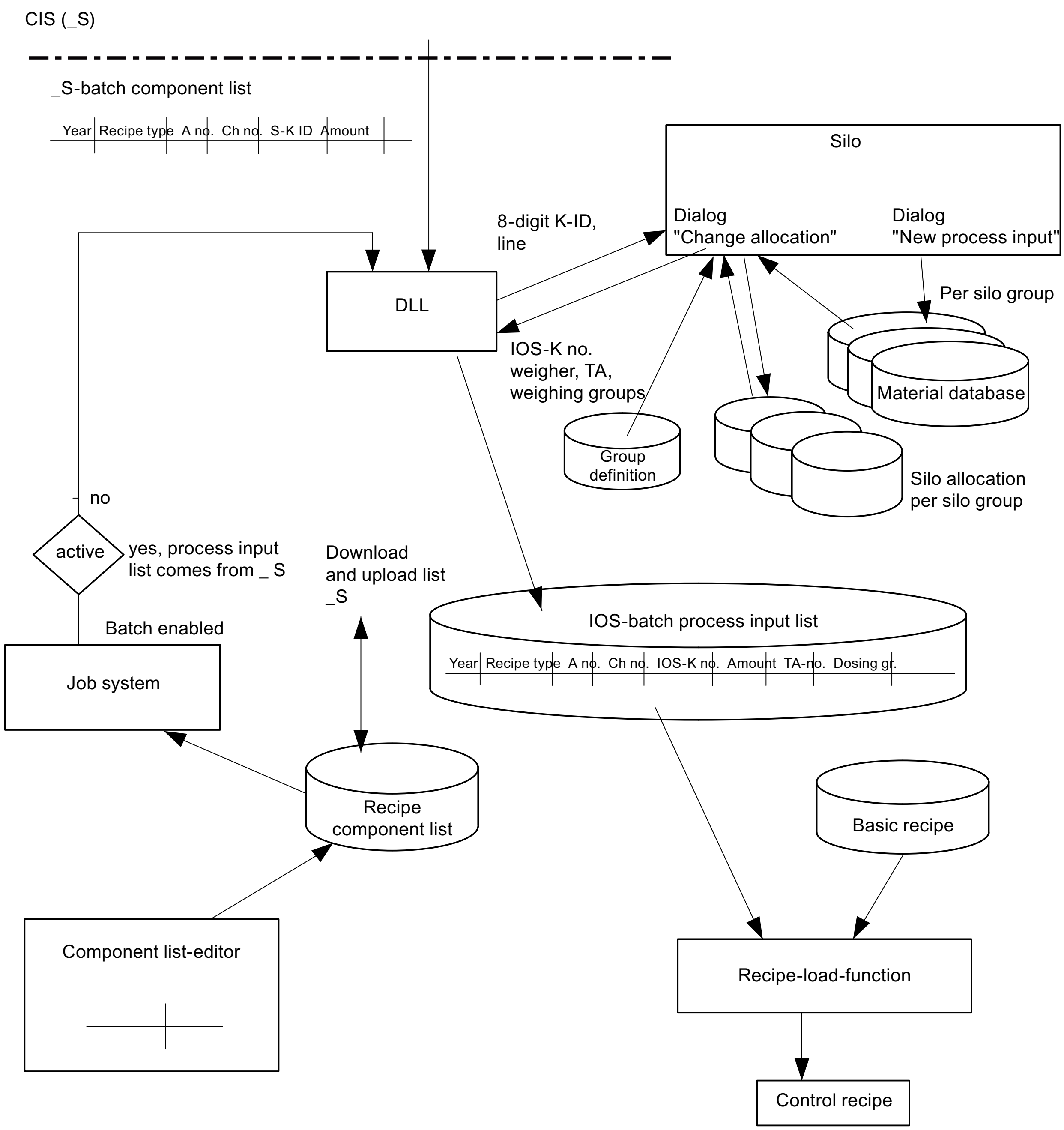
- IOS – process-input-ID,
- Sequence number (weigher),
- Dosing group (to determine the dosing-sequence)

and yields the IOS process input list

To supplement the CIS-process-input list the system needs to access the data stored in the material database (tank silo management, see also: Dosing and Weighers). All materials that are used must be in the IOS material database.

Without a superpositioned CIS the master recipe process input list is transformed into the IOS batch process input list.

All materials are scaled to the actual batch size before they are stored in the batch list. The list has to be supplemented with the data from the material database as well.


CIS (_S)
_S-batch component list
Year Recipe type A no. Ch no. S-K ID Amount
Silo
Dialog
"Change allocation"
Dialog
"New process input"
8-digit K-ID,
line
DLL
IOS-K no.
weigher, TA,
weighing groups
Per silo group
Material database
Group
definition
Silo allocation
per silo group
no
active
yes, process input
list comes from _ S
Download
and upload list
_S
IOS-batch process input list
Year Recipe type A no. Ch no. IOS-K no. Amount TA-no. Dosing gr.
Batch enabled
Job system
Recipe
component list
Basic recipe
Component list-editor
Recipe-load-function
Control recipe

10.4 Batch list editor 'BaliEdit'

10.4.1 General

Short description

The order system has the task of processing a list of batches in cooperation with the recipe control on the IOS and subordinated PCUs. This list can be created by the order system on the IOS or by a higher-level system (PMS) and the batch sequence and list content can be changed later.

Furthermore, the order system makes it possible to track and monitor the processing of batches.

The order system is called from the main menu on the "Production" / "Batch list" tab.

Note

The order system and the recipe system share a common database that contains data such as recipe headers and similar items. After you modify a recipe (e.g. the header), you need to restart the batch list program in order to refresh the program with the data which were written to interim memory for reasons of performance.

10.4.2 Functions

10.4.2.1 Order and recipe system redundancy

The following conditions must be fulfilled so that the redundancy of the order and recipe system works correctly:

- For the AREA, two redundant IOS servers must be configured in the "SiteCfg" site configuration tool.
- By activating the recipe server, the application synchronization must run.
- At the time of the recipe server failure, the reserve recipe server must have been started.
- The startup as active recipe server must have been completed without any interruption.

10.4.2.2 Order types

- Up to 255 order types can be configured.
- There is a 1:1 relation between recipe category and order type
- One different list of order parameters can be configured per order type
- The adjustments for batch input, as well as order and batch number allocation can be set up for each order type specifically.

10.4.2.3 Batch list

- In one list, all planned and running batches are held.
- Ready batches are removed from the list.
- There is no limitation to the list. It should not exceed, however, the length of 1000 batches.
- The displayed columns can be configured.
- The status of the batches is displayed.
- In the list, batch parameters can be displayed.
- The sequence of the batches in the list can be changed.
- The batches can have a line reference, if the recipe was configured as a line recipe.
- A global or order type-specific view can be selected.
- The start sequence depends on the start mode, the start time or the event. For simultaneously authorized starts, the batch that stands at the top of the list is started.
- The view is updated automatically in case of changes.

10.4.2.4 Order list

- In an order list, orders (batches with the same batch number) are displayed.
- The displayed dates are created from the batch list.

10.4.2.5 Batch monitoring

- In the view ‘batch monitoring’, the necessary batches of the list are displayed for all batches of the list.
- The status of the unit in reference to the batch is displayed in color.
- The batch monitoring displays the batch progress from the figures of the unit.
- In each line one batch is displayed with the units which have already finished, the running and following (conforming to the recipe) units (in different colors).
- In the list any further data for the batch can be displayed beside the units.

10.4.2.6 History

- In this view all finished batches are displayed.
- The start and end time of the batches are displayed

Basics

- The following start modes are available: ‘as soon as possible’, ‘by event’, ‘after absolute time’, ‘after time with automatic adaptation’.
- The possible start modes can be configured order type-specific.

Soon as possible

The batch is started as soon as the required start sequence is free, in automatic mode, not stopped and not disabled. If several batches are released and are waiting for the same sequence the batch highest in the list is started.

by event

The batch is loaded in the CAS block of the S7 as soon as this is free. There, the system waits until a bit in the CAS dataset is set by the user (event). The bit enters the name boFreeMode4 in UDT 718.

The sequence (and with it the batch) is only started if the sequence is free.

Note

If this property is used, the application developer is responsible for assigning the order and batch numbers. If a batch with the same order and batch number is started several times, this can lead to incorrect blocking of batch processing.

Starting a batch that is to be produced with a stream or line recipe is then only possible in the "by event" start mode if the batch was generated and set to the "Ready" status with the batch list application since the system is otherwise not capable of determining the correct stream or line number.

---

by time

The batch is started when the batch start time is reached or exceeded. The batch is only started if the CAS required for the sequence and the sequence itself are free.

Automatic batch start time adaptation

The start is as with the 'by time' start mode. However with this start mode, the start times of the follow-on batches are automatically adapted if there are delays in starting the batch. This adaptation is only for batches with this start mode and batches that have the same start sequence.

Manual change to the start time / start mode

If the dialog in the order system in the batch list under 'Edit', 'Change start time' is closed with 'OK', the procedure is as follows depending on the selected start mode:

- **Start mode: 'soon as possible':**
  There is no adaptation of the other batches.
- **Start modes: 'By time' and 'By time /Auto':**
  All following batches with the mode "by time / Auto' are adapted to the start cycle.
  The adaptation ends at the first batch with the 'by time' mode. This allows gaps to be created (for example weekend).
  Batches with the mode 'soon as possible' or the status 'Ready.', 'Started', 'Deleted' and 'Finished' are skipped.

As soon as a batch changes to the 'Started" status, the start times of the remaining batches with the same start unit are triggered.

Initially the current time is entered as the start time in the started batch.

Then all batches with the mode "by time / Auto' are adapted to the start cycle.

To prevent constant adaptation of the start times after a batch has started, a tolerance limit can be set for checking the start times.

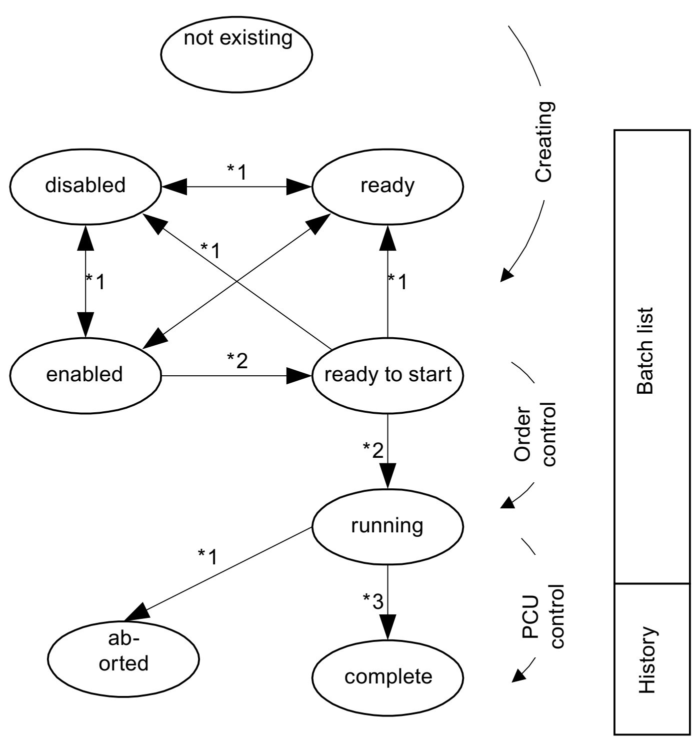
The tolerance limit can be set in the file bali.ini in the 'wincds\sys' directory in balidata TimeCorrTolerance=60 .

TimeCorrTolerance=60 is the default value (in seconds) and means that there will only be a reaction if the start time is exceeded or underrun by more than 60 seconds.

10.4.2.7 Batch status

Conditions and transitions

Basics



*1	Operation
*2	Batch control
*3	PCU control

Table 10-1 Notes

Code	Description
*1	Status change by operator
*2	Status change by OS order control
*3	Status change by AS order control
Hatched area	Here the operator can change to the status "Deleted"

Table 10-2 Description of the possible status transitions

Status	Possible change to	By
Disabled	Ready	Operation
	Enabled	Operation
	Deleted	Operation
Ready	Disabled	Operation
	Enabled	Operation
	Deleted	Operation
Enabled	Disabled	Operation
	Ready	Operation
	Deleted	Operation
	Ready-to-start	OS order control
Ready-to-start	Disabled	Operation
	Deleted	Operation
	Started	PCU control
Started	Canceled	Operation
	Done	OS order control
Error n (n = error number)	Disabled	Operation (if error corrected)
	Ready	Operation (if error corrected)
	Enabled	Operation (if error corrected)
	Deleted	Operation
Done	---	---
Deleted	---	---
Canceled	---	---

Table 10-3 Meaning of error numbers in batch state:

Error 1	Recipe header could not be read.
Error 2	Conversion of the process input list failed.
Error 3	Process input list user function delivers FALSE
Error 4	Writing of batch status into bali.dbf failed
Error 5	Loading of batch in the CAS of start batch failed
Error 6	no CAS copy for the batch
Error 7	no unit sequence for the batch

Error 8	Invalid batch values on enabling: • Y<0 (Year) • RT<=0 (Recipe type) • ONr.<=0 (Order number) • BNr.<=0 (Batch number) • LNr.<=0 (Line number)
Error 9	For user function saves

Error 1: Recipe header could not be read.

Incomplete or incorrect recipe engineering is present on this machine.

Error 2: Handling of errors in the process input list

First, a check must be performed to ascertain whether a process input list is used for the corresponding order type.

If this isn't the case, the option **process input list required** can be deselected under the adjustment **order input**.

If a process input list is used, there will be a basic error in the input configuration or material and storage location administration.

Error 3: Process input list user function delivers FALSE

refer to user interface documentation (not part of the system)

Error 4: Writing of batch status into bali.dbf failed

The writing of the file bali.dbf can't be performed without errors. The reason can be a full hard disk. If the file is located on a network drive, a missing network connection can be the reason.

Error 5: Loading of batch in the CAS of start batch failed

The reason is usually a communication problem with the PCU.

Error 6: no CAS copy for the batch

An attempt was made to start a batch which is using a start unit without having an entry in the CAS image. This can only happen when the number of CAS entries is lower than the number of sequencers.

Error 7: no unit sequence for the batch

Incomplete or incorrect recipe engineering is present on this machine.

Error 8: Invalid batch values on enabling

The key of the batches is not valid.

10.4.2.8 Order parameters

Basic principle

- For every order type up to 220 parameters can be assigned.
- All batches of an order type have the same order parameter definition.
- The parameters can be used in operations of the recipe procedure

Parameter template

- DFMs (digital function modules) can be referenced from all PCUs. Input is the same (upper limit, lower limit, decimal point, dimension,) as defined in the process cell configuration.

Parameter type

- Normal
  The entry and display is the same (upper limit, lower limit, decimal point, dimension,) as defined in the process cell configuration.
- Summation parameter
  For batch order as 'normal'. For the input of an order, the input refers to the amount of batches, i.e. for the batch generation the value is divided into individual batches (see transfer of the batch parameter).
- Free parameter
  A free parameter it is a 16 bit integer. whose value is not checked as in the case of a DFM.

Acceptance of batch parameters

- By creating batches, all order parameters, with the exception of sum parameters, are copied to the batches.
- The sum parameters refer to the number of batches and are calculated for the individual batches as follows: parameter (batch) = parameter(order) / size(order) * size(batch)
- A sum parameter must be defined during configuration under the application 'order system', 'configuration', 'order type', 'order parameter'.

Preset order parameters

- The parameters can be preset with values of the recipe process parameter.

Basics

The following types are possible:

- Start with 1 per order
- Continuously for year and order type
- Order number as calendar week

The order numbers are proposed by the order system and assigned consecutively.

The user has the possibility to overwrite this preset. An inspection occurs in this case to ascertain whether the input is permissible with regard to the explicitness of the order and batch numbers. The input of an order number is only allowed if it has been enabled under 'order type', 'configuration', 'order type', 'order input', 'allowed inputs' for this order type.

Order and batch number range

The assignment of order and batch numbers is governed by the following rules:

- The next free number is always searched for in ascending order starting at 1.
- The maximum order and batch number is 32767.

Order and batch numbers higher than 32767

The current version of the system allows order and batch numbers higher than 32767 to be assigned. For reasons of downward compatibility, this function must be activated with the "ExtendedNumbers=1" entry in the "Balidata" section of the "<proj-path>\sys\bali.ini" file. The "ExtendedNumbers=0" entry restores the default setting and prevents order and batch numbers higher than 32767.

Start with 1 per order

- Order numbers are assigned consecutively per year and order type. As default, the batch numbers begin at 1 for every order. The other batches are numbered consecutively.
- The first batch number can be changed to any other number by the user (the option of making the change must be enabled in the configuration of the order type).
- If the user enters the order number, this is checked to ensure it is unique and if it is not, the next valid number is proposed.

Continuously for year and order type

- The batch numbers are assigned consecutively per year and order type. The first batch receives the batch number of the last batch of the preceding order + 1. All further batches are numbered consecutively.
- As default, the order number is set to equal the first batch number and is saved in a file to avoid conflicts if the assignment mode is changed.

- The order number can be changed to any other order number by the user since it is always unique due to the batch number.
- If the user enters the batch number, this is checked to ensure it is unique and if it is not, the next valid number is proposed.

The method of assigning the batch number is selected in the application 'Order system', 'Engineering', 'Order type', 'Batch generation'.

User intervention in the automatic assignment of batch numbers means that consecutive numbering of the batches per order can no longer be guaranteed.

The uniqueness of the batch numbers is, however, always ensured.

Order number as calendar week

Here, there are still two sub variants possible:

- Batch number running per calendar week
- Batch number running per calendar year

Order number as value plus calendar week

The calendar week can also be expanded by the order type and recipe number.

The formula is:

CW = CW + (RTyp + RTypOffset)*RTypFactor + (RecipeID + RecipeOffset)*RecipeFactor

The factors and offsets must be entered in the `otypes.ini` configuration file.

File:	`...<proj-path>\bali\otypes.ini`
Key:	`[ordertypex]` x stands for the order number
Entry:	`RecipeFactor=0`
Entry:	`RecipeOffset=0`
Entry:	`RTypFactor=0`
Entry:	`RTypOffset=0`

Reaching the maximum order and batch number

If the maximum order or batch number has been reached (=32767), new orders can only be entered again if the user deletes the relevant number file for the order type.

It is assumed that the user will then also backup the corresponding batch data in FRPROT because this will be overwritten with the new batches.

The number files are stored as follows:

- Order numbers in the '<proj-path>\bali\used_nr\onr_xx.yyy' files,
- batch numbers in the '<proj-path>\bali\used_nr\bnr_xx.yyy' files

Here,

- xx is the year and
- yyy is the order type number.

Files with'~' added to the name are backup files.

In the 'calendar week' mode, the hist_jj.dbf file must be deleted after first backing it up.

Release of numbers when deleting batches

Batch number, 'Always start with 1'

- Since in this case, the numbers are not stored anywhere, they are not really released.
- However when appending batches to an existing order, the highest existing batch number is always determined and the next number is used for the new batch.
- If the batch with the highest number is deleted, this number will be used for a new batch of this order.
- If, however, a batch in between is deleted, its number is not used again.

Batch number, 'consecutive numbering'

The number of a deleted batch is always released and used again.

Order number

The number is only released when no batch from the order has run so far; in other words 'Batch number' equals 'Residual batches' and the entire order is deleted.

Order number as calendar week

The precise deleted combination of order and batch number is released and can be used again.

10.4.2.9 Batch input

- A total amount can be set. This one is then automatically distributed to individual batches.
- A number of batches can be set.
- A total amount and a number of batches can be set. The amount is allocated to the required number of batches.

10.4.2.10 Batch generation

There are two methods of generating batches.

- The allocated amount of the order is distributed equally to the number of batches.
- Batches with the maximum size are set up in order to reach the allocated amount. The remaining quantity is set up as a remaining batch.

Determine batch count → Order size in batches

If the order size is indicated in batches, the number of batches is determined.

The nominal batch size is taken as a batch size, except for if

- the nominal size is smaller than the minimum batch size => minimum batch size is used
- the nominal size is larger than the maximum batch size => maximum batch size is used

Determine batch count → Order size as amount

If an order size is allocated as amount, the following can be selected from two methods for determining the batch number and size:

- All batches have the same size. The number of required batches is calculated by the maximum batch size (rounded up to a whole number). Note: Nominal batch size doesn't need to be between the minimum and maximum batch sizes. If the batch size is lower than the minimum, the user is informed and must make a decision.
- n batches with the maximum possible batch size and one remaining batch.
  Note: If the remaining batch size is lower than the minimum, the user is informed and must make a decision.

The selection of the method for determining the batch number is executed under the application 'order system'->'configuration'->'order type'->'batch generation'.

10.4.2.11 Batch sequence

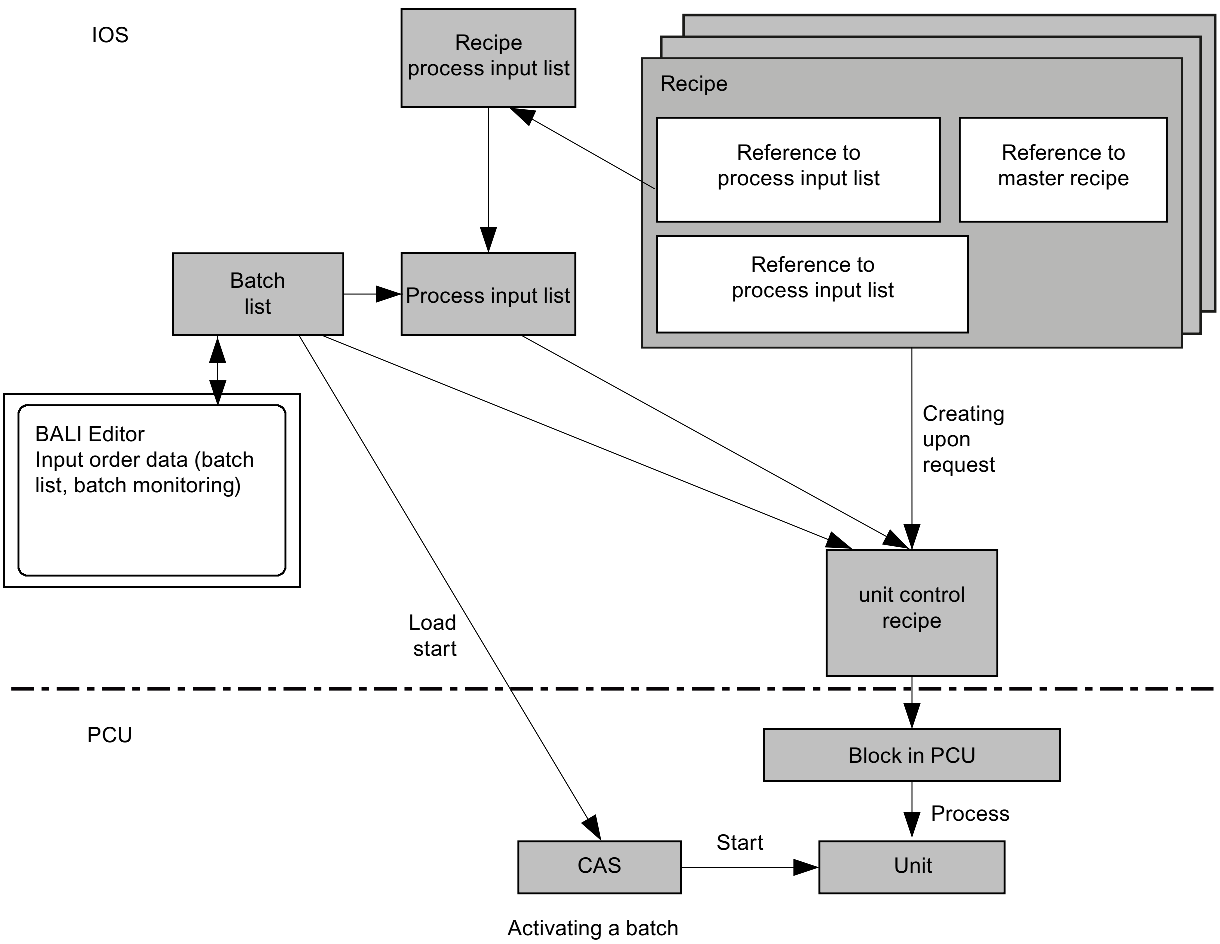
The batch is started as soon as the start conditions have been met for the batch. Once the batch has started, the recipe for this batch is requested by the PCU. The recipe load function in the IOS determines which recipe procedure is required based on the request.

The batch-dependent setpoints are scaled in this recipe procedure and selected setpoints are replaced by batch parameters or by values from the process parameters. This batch-dependent recipe procedure (name: control recipe) is loaded to the PCU and is executed there. If there are 'dose equipment operations' in the recipe procedure, the data from the process input list is incorporated into the control recipe.

The control recipes are created:

- from data from the recipe procedure (held by IOS)
- from data from the basic recipe (held by IOS)
- from data from the batch order (entered via IOS Editor)
- and from data from the component list (not batch-related, created in IOS)

"Starting a batch" process description:



Following the order entry with the Bali Editor IOS application, the batches are formed and incorporated into the batch list. The processing is performed in the sequence of the entries (and according to start mode).

Each recipe unit procedure is requested by the PCU before processing starts. The IOS converts the request in a batch-related control recipe. Order parameters, process parameters and data of the process input list flow into the control recipe.

10.4.2.12 IOS - SQL server database coupling

The "Free protocols" recorded during the batch process can be automatically transferred to an MS SQL Server database in the higher-level DB host system.

The unique key for identification within PCU and IOS is year, recipe category, order number (16 bits) and batch number (16 bits).

The batch parameters can therefore include a string which is the batch name. This name has no meaning for the PCU.

10.4.3 Configuration

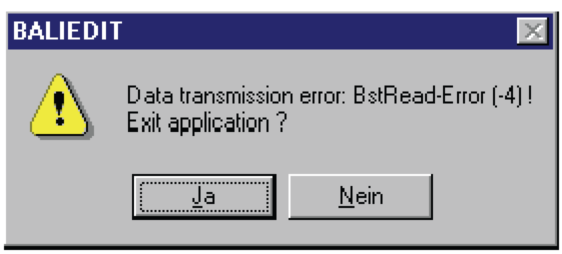
10.4.3.1 Configuration

The order system is configured in 3 steps.

- Step 1
  Configuration of the connections and communication paths
- Step 2
  See also: Start and server switchover
- Step 3
  Configuration of the order type. The basis for the order types is a current recipe configuration. See also: Recipe system

10.4.3.2 Connection to the server

For the start of the application 'order system', one connection to the recipe server is used. If no connection can be established, the following message appears:



This message always appears if no recipe server is active in the system.

First of all only the respective appropriation of the order/recipe system can be considered.

In order to use the order/recipe system, there are still some very system-specific configurations necessary, which can't be known during the installation phase of their system IOSs.

The following points should be considered during installation of the order/recipe system:

- The order/recipe system requires at least one, and with a redundant plant concept two, so-called recipe servers. A recipe server is an IOS that links directly to the PCUs with a network interface module.
- The process cell configuration must have parameters assigned correctly with the "SiteCfg" tool.
- In the PCU server of the IOS that should become the active recipe server (with a redundant system design in the PCU servers of the two IOSs which are enabled as recipe servers) the coupling option 'recipe server' must be activated.
- Then the IOS that should become the active recipe server in the PCU server is switched on under the program 'Activate recipe servers' or with the help of the corresponding button in the toolbar.
- The recipe server only starts up if it receives acknowledgment from all PCUs that were set in the parameters in the process cell configuration. If one of the parameterized PCUs does not report itself, it remains in the standby state. This restriction can be switched off.

10.4.3.3 File Synchronization

The restart of an IOS as a recipe server or a recipe server switchover automatically activates the 'Synchronization' application (synchro.exe).

This application ensures that the newest data is available on the IOSs. It compares files on the IOSs and copies the newest files.

In the event of a recipe server failover the synchronization is set in the `'...<proj-path>\sys\recipe.ini'` file.

`[StandByControl]`

`StartFileSync=1`

`AbortWaitForSync=300`

If StartFileSync = 1, the file synchronization is automatically activated by the redundancy failover. If no automatic file synchronization is selected ('StartFileSync=0=0'), it will wait for a maximum of 300 seconds, then the start as recipe server is aborted.

10.4.3.4 Monitoring the recipe server

Basics

The monitoring of the recipe server can be set in the `'<proj-path\sys\recipe.ini'` file.

If the monitoring process is parameterized, at the parameterized time intervals, telegrams are triggered in all the PCUs participating in the recipe control process.

These telegrams are of type 16 and are received by both the active recipe server and the standby recipe server.

If a recipe server failure is detected, the message 'Recipe Server Failure' is output in all IOSs of the plant.

Settings in the recipe.ini file

[ServerSupervising]

- Enable=1
  If Enable =1 is set, the monitoring of the recipe server is enabled.
  If Enable = 0 is set, the recipe server is not monitored. In this case, the message 'Recipe Server Failure' is therefore not generated. The automatic recipe server switchover can only run if the monitoring process has been enabled.
- CycleTime=30
  If CycleTime = 30, a message is triggered in all the PCUs participating in the recipe control process every 30 seconds.
- ReactionCycle=24
  For ReactionCyle = 24, a message "Recipe server failed" is triggered after 24 messages have failed. This parameterizes a reaction time of CycleTime x ReactionCycle = 24 x of 30 Sec.
- RepeatTime=300
  After RepeatTime=300, the message "Recipe server failed" is repeated

- WaitTime=240
  After booting the PCU server, only after the time WaitTime=240 seconds is the monitoring of the PCU started.
- AutomaticServerActivation=0
  In the event of a server failure or if the system is powered up without a clear server configuration in the assigned PCUs, user intervention is required.
- AutomaticServerActivation=1
  In the event of a server failure, the system switches over to the spare recipe server automatically. For the function of this switchover, the monitoring of the recipe server [ServerSupervising] Enable=1 must be switched on.

  Note

  Power-up when there is no clear configuration in the PCUs

  If no clear recipe server configuration can be detected in the assigned PCUs during power-up (e.g. when AS is reloaded on a PCU), the 1st server in the AREA automatically powers up as the recipe server.

- Automatic server switchover is enabled by default (in the delivery state) (AutomaticServerActivation=1).

10.4.3.5 Batch start cycle time processing

Criteria for the batch start

The CAS status of all start units of the released batches is checked cyclically. In addition to the cyclic search, a search is triggered with a status change to 'Released', with the release of a CAS dataset and with the creation of a new batch.

The cycle can be set in the `'<proj-path\sys\bali.ini'` in...

- `[balidata]`
  `SearchTime=80`

. The value corresponds to the number of the PCU server timer calls (=200ms) in this case.

- This means: 80 corresponds to 80* 200 ms = 16 seconds.

A search is performed for batches which are to be started as soon as the CAS of the start unit of at least one enabling batch is free. The batch list is searched from the top to the bottom.

The first batch with start mode 'as soon as possible' is started.

From the batches with mode 'after time' or 'after time/auto' the one with the shortest start time starts, regardless of its position in the batch list.

10.4.3.6 Order types

Order types

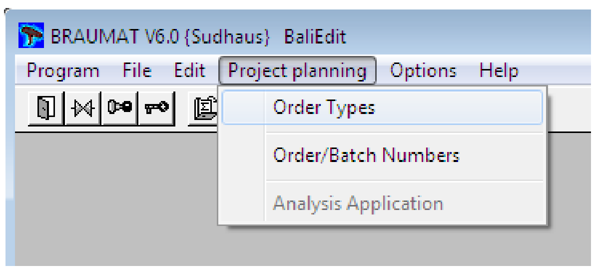
Order types must be defined before entering orders, whereby each order type is assigned to one recipe category. There is a 1:1 relation between order type and recipe category. The order type number is identical to the number of the recipe category.

The new order/recipe system isn't preconfigured, i.e, no order types are preset.

Configuration of the order types only makes sense if recipe categories, master recipes, recipe procedures, process parameters, process input lists, etc. were created.

See also: Recipe system 3.1 Creating a simple recipe.

Configuration dialog selection

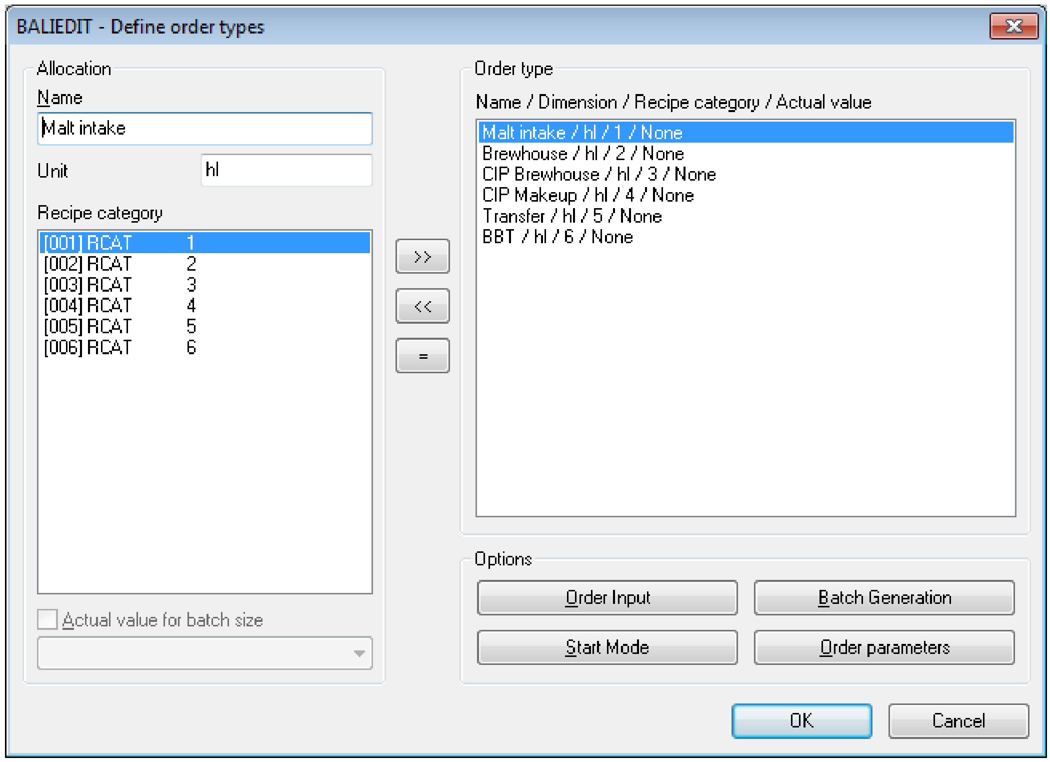


If the menu item 'configuration' isn't available, the current view with the menu item 'file' > 'close' must be closed.

Dialog 'Define order types'

To select the dialog box:

- Click "Configuration" -> "Order types"



Name

- Name → The name of the job type can be entered here.
- Dimension name→ A dimension name can be assigned. The dimension is indicated during the order input.

Actual value 'batch size'

Here one parameter from the free protocols can be selected, which can be indicated as the actual value for the batch size in the batch list. For the function, a configuration of free protocols is necessary.

Assignment

In the list, a recipe category can be selected to which the order type is assigned. If a selection is made in the list of existing order types, the currently assigned recipe category is displayed.

List 'name / dimension / ...

The order types which have already been configured are displayed in the list. The following information: Name, dimension, recipe category number and actual value of the batch size with diagonal stroke is displayed separately. For the selection of an order type, the input fields with the configured data are set.

If an order type is supposed to be edited, it must be selected in this list box. Then, a new definition can be selected. With the key '=', the changes are accepted.

Order input

With this button, you reach the dialog 'define order input'.

Batch generation

With this button, you reach the dialog 'define batch generation'.

Start mode

With this button, you reach the dialog 'define start mode'.

Order parameters

With this button, you reach the dialog 'define order parameter'.

Button '>>'

A new order type is included in the list. In this way, the name is loaded in the field 'name' and the selected recipe category is accepted and checked.

Button '<<'

In this way, the order type which has just been selected is deleted.

Button '='

In this way, the definition of the selected order type with the current inputs is overwritten.

Button 'OK'

File definition and dialog end.

Button 'Abort'

Do not save file definition and dialog end.

Create new order type

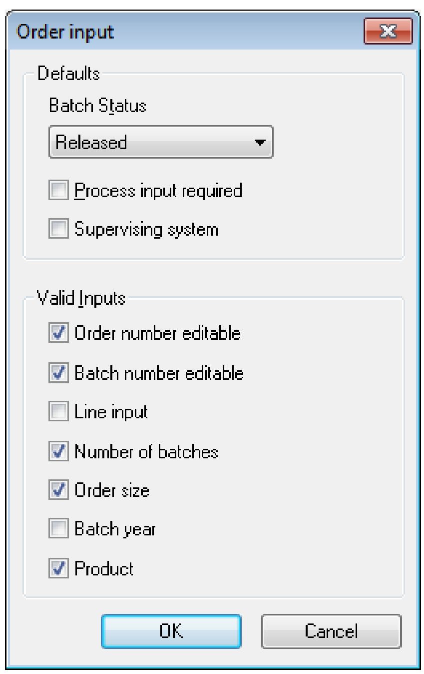
Procedure when creating a new order type:

- In the 'recipe category' window, one recipe category is selected (click the requested recipe category).
- After the selection the desired name for the order type is entered in the window 'name'.
- With the key '>>', the recipe category is accepted in the list and is visible in the window "Name\Dimension\Recipe category\Process value".

Dialog define batch generation

To call the dialog box

- Select "Edit"
- Select "Order types"
- Click "Starting mode"



Preset 'Batch state'

This selection determines with which status the batches are created after creation. It is possible to select between 'disabled', 'enabled' or 'ready'.

See also: Batch status (Page 714)

Preset 'necessary to process input list'

Here is determined whether the batches of the order type use a process input list. The deselection of the function has advantages in terms of performance.

Preset 'Higher-level system'

Here it is determined whether a higher-level system (MES) is responsible for this order type. If the higher-level system is active, the batch input for this order type is locked in the "Batch Manager" application. This lock can be removed temporarily, i.e. until the next activity of the MES system, via the menu point "Options/Disable MES". The operator permission "BALI_PROJ" is required for this.

Allowed inputs

This selection determines which input possibilities are enabled by creating batches.

- 'Order number editable'
  The order number can be changed. A test to ascertain explicitness is executed.
- 'Order number editable'
  The first batch number of an order can be changed. A test to ascertain explicitness is executed.
- 'Path input'
  Path can be selected. If this option is deselected, the reference line of the assigned recipe is always decisive.
- 'Batch number'
  Input possibility for the number of batches per order.
- 'Order size'
  Input possibility for the size of each order.
  If neither 'batch number' nor 'order size' is selected, 1 is always accepted as a number of batches per order.
- 'Batch year'
  The year of the batch can be changed manually. The default setting is the current calendar year.
- 'Product'
  When a new order is created, a product/another product can be selected from the material list. In principle, only the product defined in the basic recipe is used.

Button 'OK'

The changes are accepted temporarily. A final acceptance only occurs when the OK button of the dialog 'define order types' is pressed.

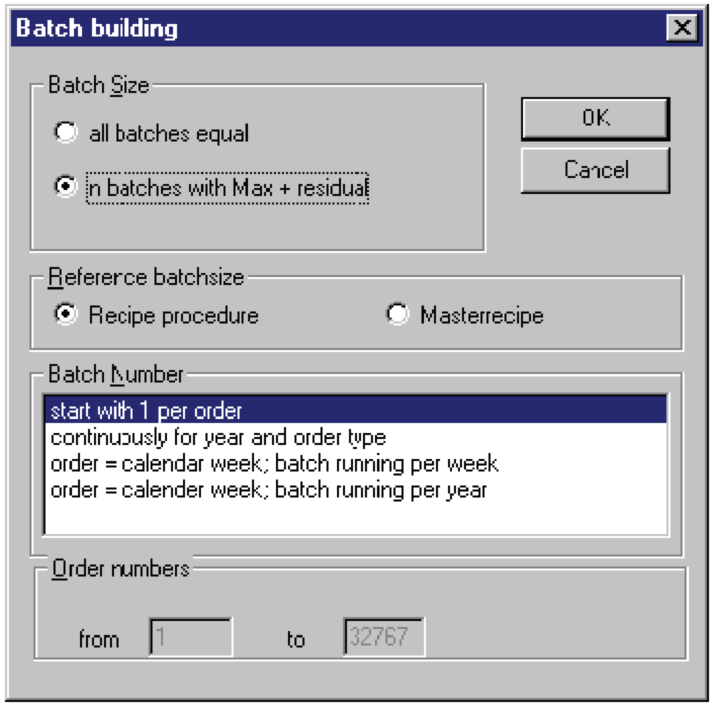
'Abort' button

The changes are not accepted.

Dialog 'Defining batch building'

To call the dialog box

- Click "Configuration" -> "Order types"
- Click "Batch building"



Batch size

Here a selection can be made between the two types of batch building.

Reference batch size: Recipe procedure

The nominal batch size of the recipe procedure is used as default batch size. The limits to the recipe procedure are used.

Reference batch size: Master recipe

The batch sizes for the batch input are calculated as follows:

Default value:	= nom. MR
minimum value:	= min. RP / nom. RP * nom. MR
maximum value:	= min. RP / nom. RP * nom. MR

(RP = recipe procedure; MR = master recipe)

Batch numbers

Here, a selection can be made between the types of batch number allocation.

Order numbers

In the fields, the limiting values of the order numbers are indicated.

Button 'OK'

The changes are accepted temporarily. A final acceptance only occurs when the OK button of the dialog 'define order types' is pressed.

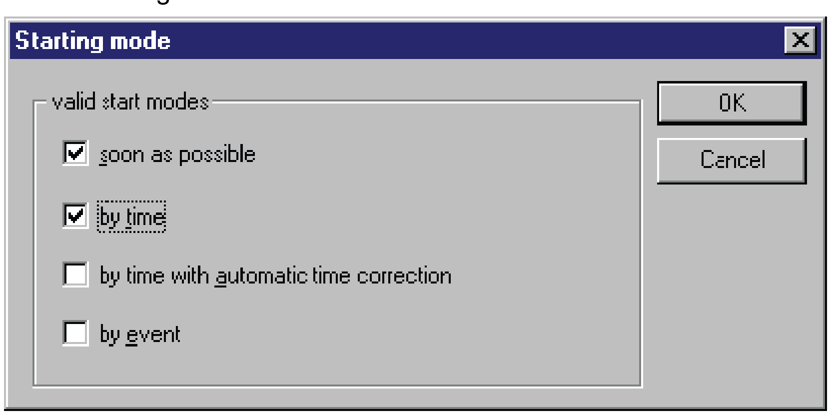
Button 'Abort'

The changes are not accepted.

Dialog 'Define start mode'

To call the dialog box

- Click "Configuration" -> "Order types"
- Click "Starting mode"



possible start modes

Here the possible start modes are defined, which are possible with the batch input

Button 'OK'

The changes are accepted temporarily. A final acceptance only occurs when the OK button in the dialog 'define order types' is pressed.

Button 'Abort'

The changes are not accepted.

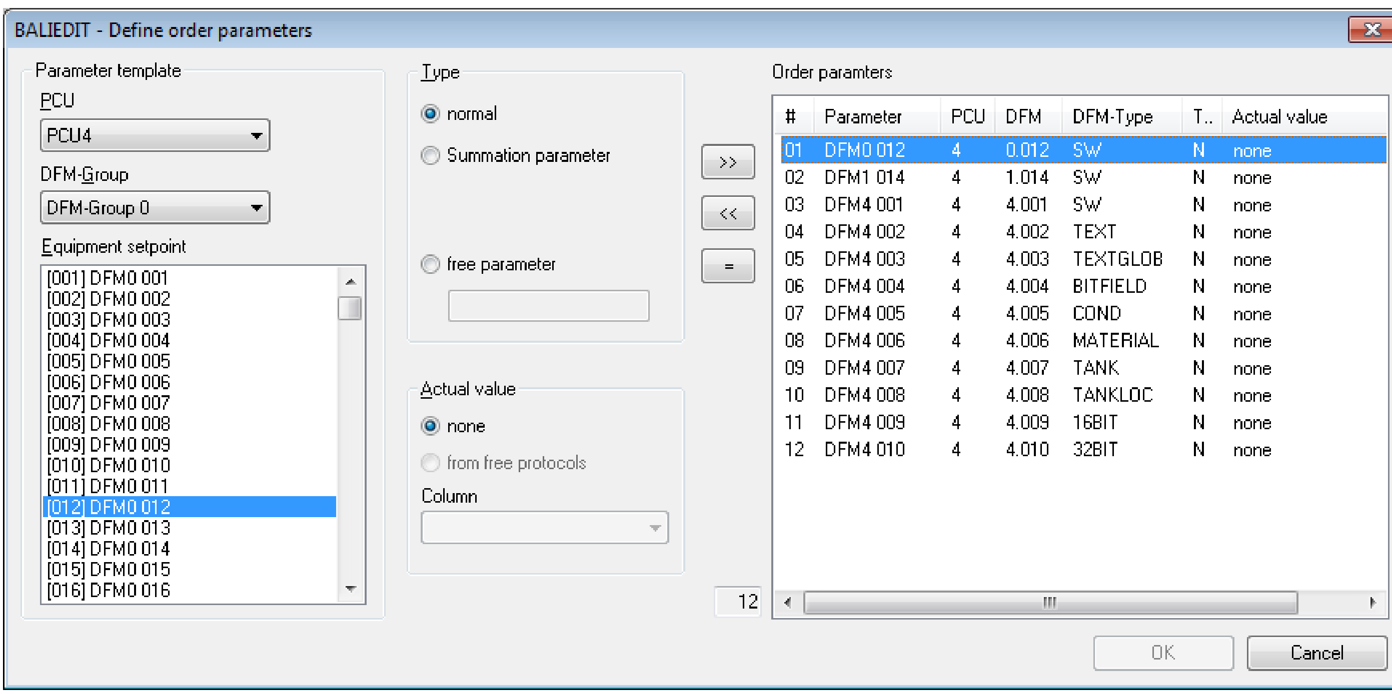
Dialog 'Define order parameter'

To call the dialog box

- Click "Configuration" -> "Order types"
- Click "BALIEDIT"

If the batch list is not completely empty, initially a message appears indicating that the order parameters cannot be changed.

Dialog



Parameter template

- PCU, DFM group and DFM
  The DFM instance that is used as the parameter template for the order parameters is selected

Type

- normal
  With the selection 'normal', the parameter is used for input and output as was determined in the setpoint definition.
- Summation parameter
  With the selection '**sum parameter**', the input/output is the same as the selection of 'normal', with the difference that the parameter is distributed to the individual batches.
  With the selection of 'normal', 'route control' and 'sum parameter' the EPARs are automatically selected.
- Free parameter
  With the selection '**free parameter**', an order parameter is defined that is not a DFM. For this parameter, only one name can be allocated.

Actual value

- One actual value can be assigned to each parameter. Actual values are columns from the dates of the free protocols.
- With the combo box, the column is selected in the dBase file.
- For the function, a configuration of free protocols is necessary.

'Order parameters' list

In this list box, all order parameters and their definitions are shown.

'>>' button

With this button, an order parameter is added to the list.

'<<' button

With this button , an order parameter is deleted from the list.

'=' button

This button applies the definition of the selected DFM in the selected order parameters.

'OK' button

The changes are accepted temporarily. A final acceptance only occurs when the OK button of the dialog 'define order types' is pressed.

'Cancel' button

The changes are not accepted.

10.4.3.7 Reconfiguration of order parameters

Requirements

The reconfiguration can occur only under specific conditions.

- Batches may not be available in the batch list. This test is executed by the application.
- All current batch archives must be paged out and deleted.

Note!

If there is a reconfiguration without saving the 'old' archives, no further batches can be created.

Note!

Current order parameters are assigned to their corresponding formal parameters (the placeholders in the recipe procedure in which these parameters are inserted by substitution), based on the line number of the order parameter in the order parameter definition list of this dialog box. When the arrangement of parameters in the list of this dialog is changed, the assignments in the recipe procedures are not automatically updated when order parameters are substituted in the recipe procedures. You must definitely take this point into account when you modify the parameter list. You can always append new parameters to the list, without having to adapt existing recipe procedures. All other modifications, for example deleting parameters or changing their position in the list, may cause an undefined behavior of the recipe procedures!

10.4.3.8 Preset order parameters

Predefining order parameters

The order parameters in the "New Order" dialog can be configured in file '...<proj-path>\bali \oparadef.ini'.

In this file, there is an area for every order type

```
[OrderType#] # stands for order type
```

Under this area, the values are listed for every order parameter.

```
Parameter#= # stands for the number of the order parameter (starting
with 1)
```

Fixed value

A fixed value is written directly behind the sign '='. Numbers with decimal places must be displayed with any points.

Examples:

```
Paramater1=5
```

```
Parameter2=6.67
```

Value from process parameter of the recipe of the batch

The order parameter can be preset with a value from the process parameters of the recipe. In this case, the number of the parameter (Index of the list beginning at 1) must be indicated with a leading '%' sign.

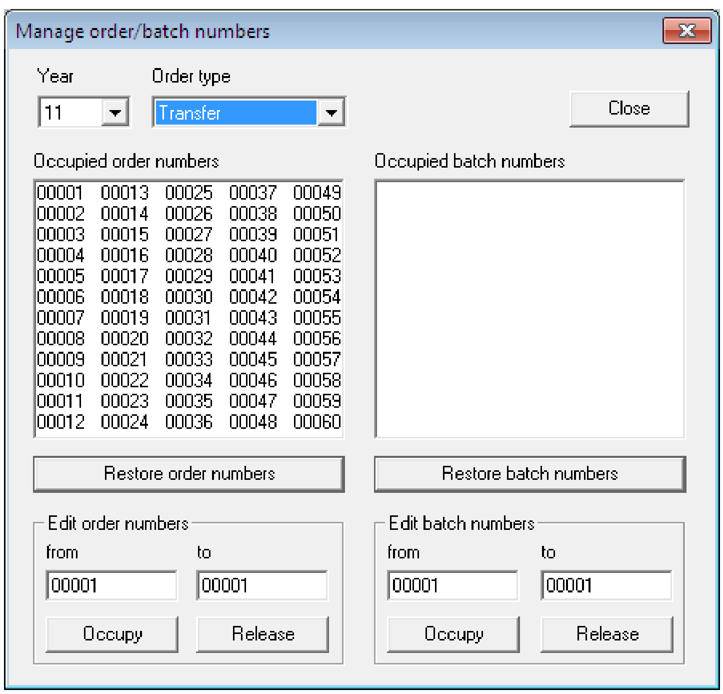
Example:

```
Parameter1=%5
```

10.4.3.9 Allocating/releasing order/batch numbers

Calling the dialog:

- Click Engineering/Order numbers/Batch numbers



This dialog allows you to reconstruct order/batch numbers from the batch history in the system allocation list or to allocate/release number ranges.

- First, you have to select the "Year" and order type
  The "Occupied order numbers/batch numbers" list fields show the current system allocation lists for the selection you have made.
- "Restore order numbers"/"Restore batch numbers" buttons
  The batch history in the ,…...<proj-path>\Bali\hist_yy.dbf' files is read, and the order/batch numbers for year and order category are entered in the system lists under ,…...<proj-path> \Bali\Used_No' .
- "Edit order numbers/Edit batch numbers" fields
  Here, you can pre-assign or release the relevant number ranges.
  – Enter "from"/"to" and press "Occupy" button → The number range is pre-assigned or reserved
  – Enter "from"/"to" and press "Release" button → The number range is released or deleted

⚠ CAUTION
Data loss as a result of incorrect use
Once the order/batch numbers have been released, they are reused for any new orders or batches that are created. As a result, any batch reports that are left over get overwritten.
In addition to causing data loss, the incorrect use of this function can also result in incorrect assignments between steps and batches in reports/MIS/reporting.
Access to this dialog is protected by means of a dedicated program level: ,BALI_PROJ'.

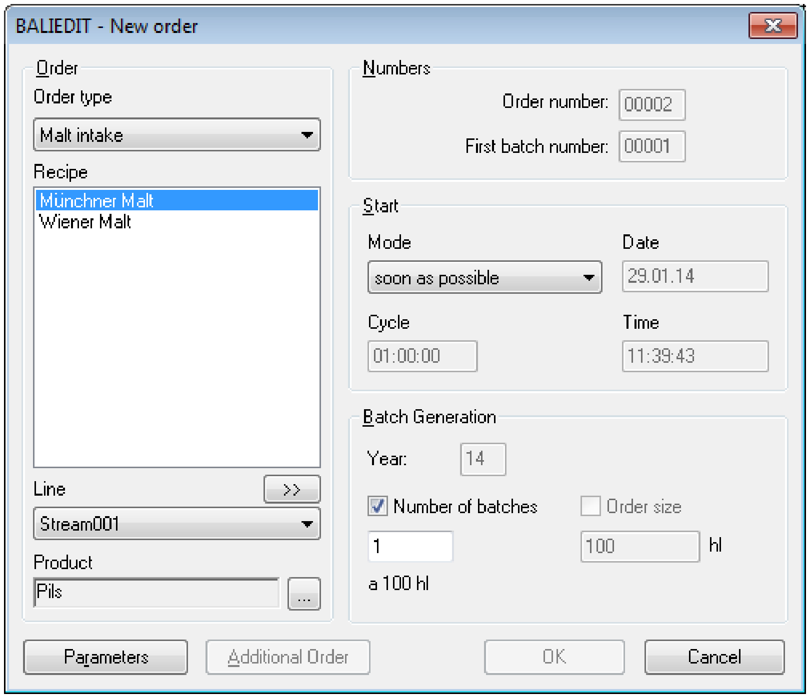
10.4.4 Menu dialogs

10.4.4.1 Dialog 'New Order'

Dialog

Selecting the dialog

- Closing the order or batch list view
- Select "Edit / New Order" in the menu



With this dialog new orders with their parameters can be created.

The input of batch number or order size can be disabled.

Numbers

Here the order and the batch number of the 1st batch of the new order are entered. The numbers are suggested by the system. The number can be changed if this is enabled. If the order consists of more than one batch, the order numbers will be allocated by the system, using the defined algorithm.

Order

The order type, the recipe and the line for the new order are queried. When opening, type, recipe and line are preset automatically.

Combo box 'line'

With this it can be determined on which line the order is supposed to run. It can be configured whether a line input is possible. By opening the dialog, the line is preset with the reference line of the current recipe. With the line selection, the start unit is determined implicitly (via recipe header and assignment table). The permissible lines are stated per recipe.

Start

The following start modes (like CAS modes) can be selected:

- 'as soon as possible'
- 'after time'
- 'after event' and
- 'after time/auto'
- For each order type the selection of the mode can be limited.

The indication of 'date' and 'time' refers to the mode 'after time'. The cycle determines the time difference between the individual batch starts.

With the selection of the recipe, the cycle and the start time is preset with the current values for the selected recipe. However, these values can be overwritten.

Order size

Here, either only the number of batches for an order or the size of the order or both are entered. When leaving this field the batches are generated according to the method configured for this order type.

'>>' button

Has no importance for this version.

Key 'Parameter'

Call of the dialog 'order parameter'. This call is absolutely necessary if parameters were defined for this order type. Only then are the keys 'OK' and 'further order' enabled.

Key 'Additional order'

Acceptance of the current order and input of a further order.

Key 'OK'

Accept new order data and dialog end. After that, it will be automatically branched out to the batch list (view: all batches of this order).

Key 'Abort'

Do not accept new order data and dialog end.

See also

Batch input (Page 720)

Error report at order input

Message Box 'Invalid Recipe Header'

The dialog starts with a presetting that indicates a recipe for which no valid recipe header is configured. A selection of different order types and recipes with valid data is possible.

Message Box: 'Invalid Input!'

If there is an attempt to enter new orders, a

- Message Box: 'Invalid Input!'
  is output, one of the values that follow is implausible for the required master recipe (see master recipe header), in this case <= 0:
  - nominal batch size
  - minimum batch size
  - maximum batch size

In this case no new orders can be entered. The values have to be > 0.

Creating orders for recipe procedures with stream control

Basic principle

The order system integrates the use of recipe procedures with path changeover.

When you program a new order, you can select the batch production line from the "Path" combo box on the bottom left. You can click ">>" to expand the "New order" dialog box to open the bitmap diagram for the currently selected path.

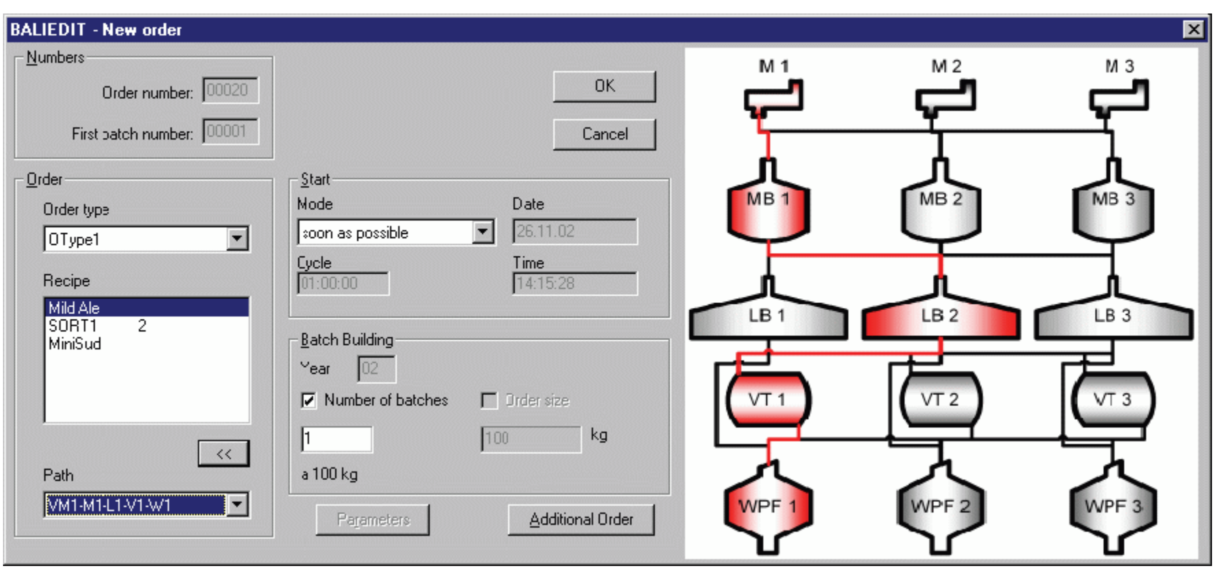


Figure 10-1 "New order" dialog box, showing the bitmap diagram of the path set in the combo box on the bottom left.

You can always modify the stream while the batch process is running. However, the system imposes a physically required restriction on your hot-change options:

- i.e. you can select only the streams which contain at the time of change all the units to be used.

To reconfigure the stream while the batch process is running, double-click the path name in the order list (default line name "LinieName", reconfiguration possible by means of the "Options->Layout" command is possible) to open the "Change stream" dialog box.

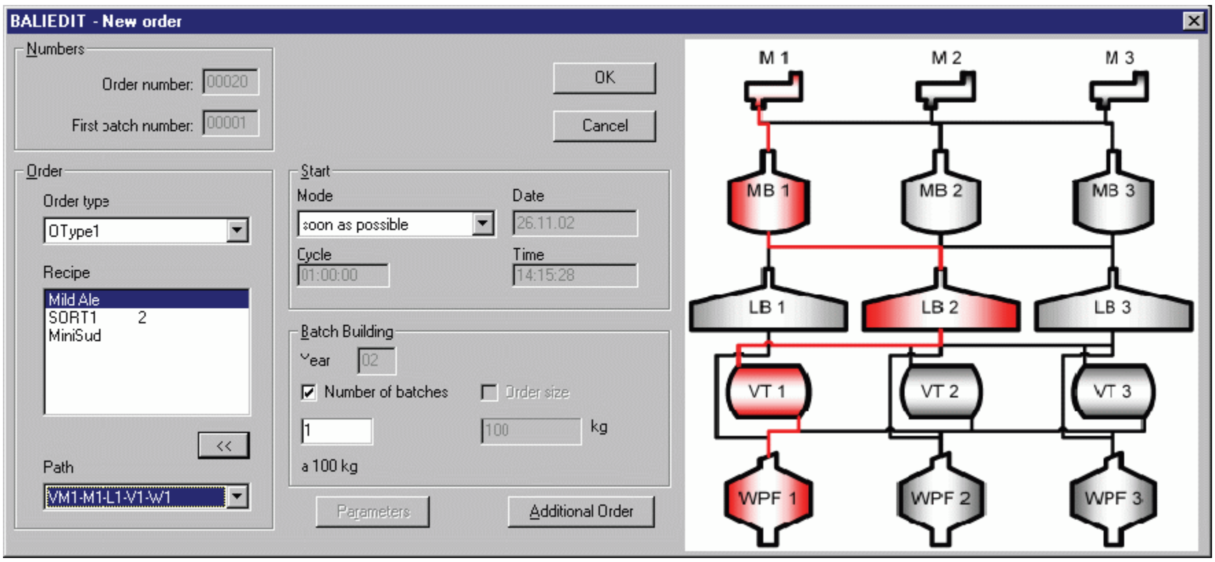


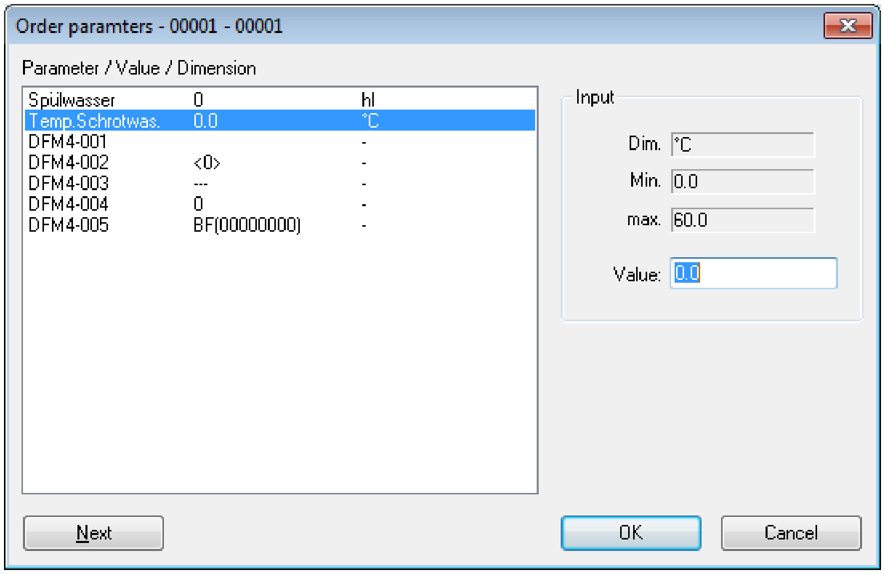
Figure 10-2 Dialog box for changing the stream of an active batch production in the ‘Order list’ program.

This dialog box is opened by double-clicking the name in the LinieName column.

10.4.4.2 Dialog 'order parameter'

Selecting the dialog box

- Closing the order or batch list view
- Select "Edit / New Order" in the menu
- Click the "Parameter" button



The window is used to enter or select order parameters. Depending on the parameter type, a new value can be entered in the "Value" box or can be selected from the list (for example in special list dialogs).

The dialog can also be opened for an existing batch by double-clicking on a parameter box.

In the header, it is indicated for which batch the parameters are displayed and changed.

'Next' button

The system jumps to the next parameter and the input focus is set in the field 'value'.

"OK" button

The parameters are accepted and the window 'order parameter' is closed.

"Cancel" button

The window 'order parameter' is closed.

"Setpoint" parameter type

The input box appears as in the dialog above. The dimension, the minimum and the maximum limit value are displayed.

"Text" parameter type

The selection dialog shows the configured text file in the project directory "`<proj-path>\Pcu.nnn\Texte.x`" (nnn= PCU no.; x= language number).

If you select 'delete', the value "#" (no setpoint change) is set.

"Text (global)" parameter type

The selection dialog shows the configured text file in the project directory "`<proj-path>\Texte.x`" (x= language number).

If you select 'delete', the value "#" (no setpoint change) is set.

"Text (Bitfield)" parameter type (PCU from V7)

- The setpoint is defined by a bit pattern (32 bits).
- The setpoint is entered in the parameter dialogs using a separate dialog in which the 32 bit positions can be set/reset using select boxes.

"Text (indirect)" parameter type

- The setpoint is represented by text in various text lists.
- The function is comparable with TEXT type but is implemented in 2 stages. It makes it possible to select elements from groups again.
- See chapter Setpoint definition type ‘COND’ (Page 1229) for further information about the configuration and function of the setpoint type.

"Material" parameter type

An additional selection dialog opens. This allows the selection of a material from the full list (material group "*") or from a defined material group.

"Tank" or "Tank location" parameter type (PCU as of V7)

Clicking on an order parameter of the type "Tank" or "Tank location" opens the tank selection dialog.

(description see section 'Tank selection' dialog (Page 744))

"Bit pattern (16 Bit)" parameter type

- The 16 lower bits of the setpoint are shown in binary.
- The entry is made in the box in binary format (0 and 1).

"Bit pattern (32 Bit)" parameter type

- The setpoint is represented as an 8-digit hexadecimal number.
- The entry is made in the "Value" box in 'hex' format.

See also

Material and material groups (Page 585)

10.4.4.3 'Tank selection' dialog

"Tank" or "Tank location" parameter type (PCU as of V7)

Here, an online view shows the available tanks and their attributes that are currently important for the selection. By selecting a line in the tank list and confirming with "OK", the corresponding tank number (PCU-related) or the location (AREA-related) is entered in the order parameters and the dialog is closed.

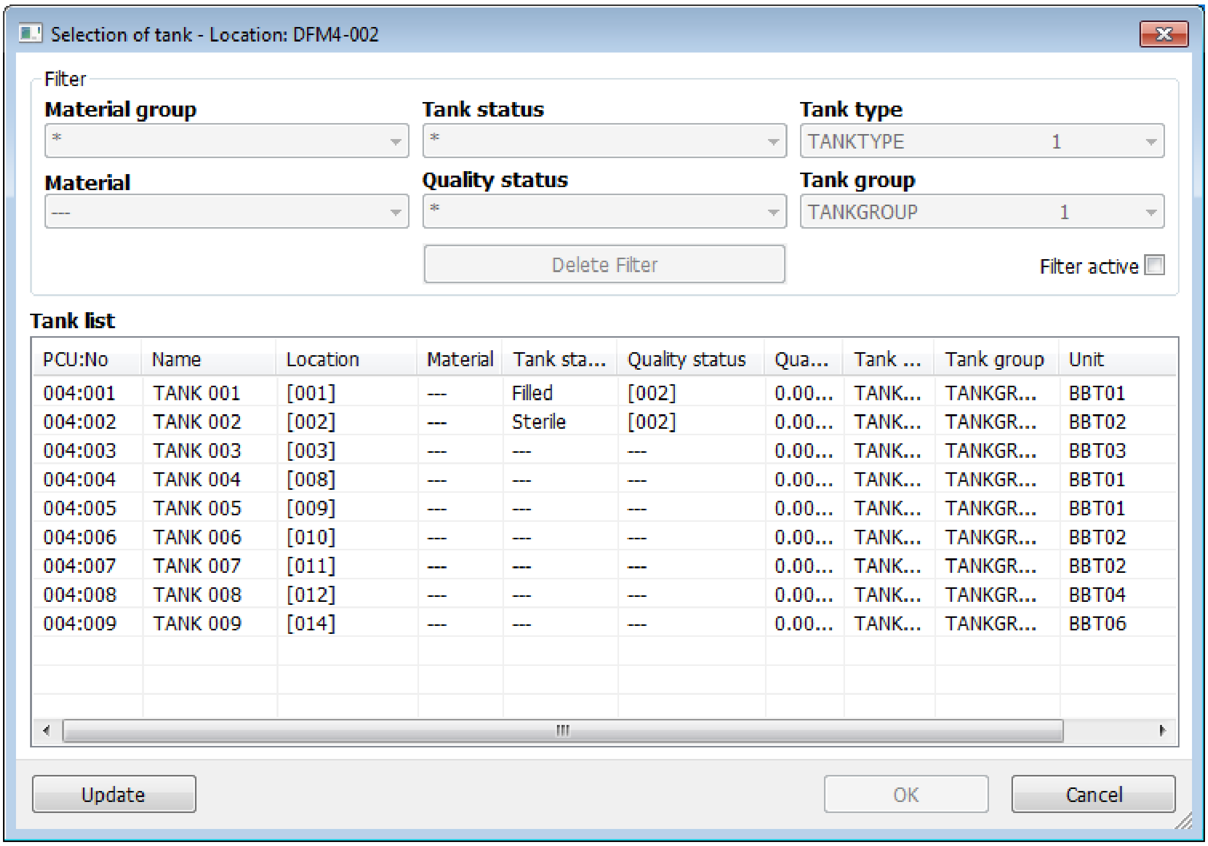
Note

The tank list is always only filled during runtime by means of so-called "tank inventory telegrams". The telegram type 22 used here must also be entered in the process cell configuration / PCU settings / FIFOs / FIFO 1+4!

The display of the various attributes in the tank list assumes that these were correctly and completely configured during parameter assignment of the TANK block instances. Parameter assignment with regard to trigger event and acquisition cycle also takes place there (for more details see section TANK - functions (Page 371)).

- With the type "Tank", the tank list shows only the tanks of the relevant PCU (according to the DFM assignment).
  - The first column shows the tank number
  - The "Location" column is missing
- With the type "Tank location", all the tanks of this AREA are displayed.
  - The first column shows the combination "PCU number:Tank number"
  - The "Location" column is also shown
  - To use this function, a location must be assigned to every TANK when assigning parameters for the TANK block instances.
  - The list of locations (file "`<proj-path>\texte.x\rcs_loc.txt`") is automatically generated from the RCS offline engineering when the "Route Control System" option is used. If RCS is not used, this file must be edited using the text parameter assignment.

Tank selection dialog and filter options



"Filter" area

By selecting an element in the list box, the corresponding filter is activated. The list boxes contain the elements used in the visible/filtered tank list. All tanks are displayed only when the filtered criteria are not active or have been deleted.

- Material group list box → only tanks with materials of the selected material group are displayed
- Material list box → only tanks with the selected material/product are displayed
- Tank status list box → only tanks with the selected tank status are displayed
- Tank type list box → only tanks of the selected tank type are displayed
- Tank group list box → only tanks of the selected tank group are displayed
- Quality status list box → only tanks with the selected quality status are displayed
- "Delete Filter" button → The selected filter criteria are reset to "*" = all

---

Note

When the tank selection dialog is opened, filtering is activated and the material of the current batch is preselected in the "Material" list box.

"Tank list" area

With the "Filter active" check box, the filtering of the tank list can be deactivated/activated. When it is activated, the table contains only the tanks that correspond to the previously described filter criteria.

The contents of the boxes are derived from the current online attribute values of the PCU block instances at the time the dialog was opened. The screen can be updated at any time using the "Update" button.

10.4.4.4 Dialog 'delete order'

Selecting the dialog box

This dialog box is only available after you have preselected the order list.

- Select the File / Open order list menu
- Select the Edit / Delete Order menu

'Order' dialog section

The order number and the number of batches are displayed.

The 'Delete' button deletes all the batches that have not yet started. Batches which have already been started cannot be deleted.

10.4.4.5 Dialog 'Move order'

Selecting the dialog box

- Select the File / Open order list menu
- Select the Edit / Move Order menu
- It is simpler to click on the row number and drag it to the new row ("drag and drop").

In this window the production sequence can be changed.

Order

The order number that is supposed to be displaced is indicated.

Move

The line positions are indicated from which the order is supposed to be displaced.

Key 'OK'

Displace and dialog end.

Key 'Abort'

Do not displace and dialog end.

10.4.4.6 Dialog 'Change order state'

Selecting the dialog box

- Select the File / Open order list menu
- Select the Edit / Order Status menu

Current status

Signal of the current order state. In the header line of the dialog the order number is displayed.

New state

A new state can be selected from the list.

- Disabled → All batches of this order that are ready, released or ready to start are disabled.
- Ready → All batches of this order that are disabled or released are set to ready.
- Released → All batches of this order which are disabled or ready are released.
- Deleted → All batches of this order that are ready, released, ready to start or disabled (no running batches) are deleted.

Key 'OK'

New order status accepted and dialog end.

Key 'Abort'

Do not accept new order state and dialog end.

10.4.4.7 Dialog 'Increase number of batches'

Selecting the dialog box

- Select the File / Open Batch List menu
- Select the Edit / Number of Batches menu

In this window the, the number of batches for the order can be increased. If the number of batches is supposed to be reduced, the batches must be deleted via the change of batch state in the batch list.

Selection 'add to order'

The new batches are appended directly behind the last batch of the order in the order database.

Selection 'add to list'

The new batches are attached to the end of the customer sequence database.

10.4.4.8 Dialog 'Change Batch state'

Selecting the dialog box

- Select the File / Open Batch List menu
- Select the Edit / Batch Status menu

The dialog header displays the order and batch number for which the change will be made.

Current status

Signal current status of the batch

New state

A new state can be selected from the list.

Selection

It can be determined for which batches the change is supposed to be valid:

- 'this batch only' → only for the current batch
- 'all batches of order' → for all batches of the order in the current row that have not yet been started.

Key 'OK'

Accept new batch state and dialog end.

Key 'Abort'

Do not accept new batch state and dialog end.

10.4.4.9 Dialog 'Change batch size'

Selecting the dialog box

- Select the File / Open Batch List menu
- Select the Edit / Batch Size menu

With this dialog the batch size can be determined.

- The nominal, minimum and maximum batch size is taken from the recipe procedure header.
- There are also different sizes possible. If the selected size isn't between the minimum and maximum size, an additional dialog appears in which this must be confirmed.

Key 'OK'

Accept new batch size and dialog end.

Key 'Abort'

Do not accept batch size and dialog end.

10.4.4.10 Dialog 'Move batch'

Selecting the dialog box

- Select the File / Open Batch List menu
- Select the Edit / Move Batch menu

This window can be opened using the menu or by clicking on the row number and dragging it to the new row ("drag and drop").

With this window the production sequence can be changed.

Batch

In the fields, the order number and batch number of the displacing batch are indicated.

Move

The line positions are displayed.

Key 'OK'

Displace and dialog end.

Key 'Abort'

Do not displace and dialog end.

10.4.4.11 Dialog 'Change start data'

Selecting the dialog box

- Select the File / Open Batch List menu
- Select the Edit / Change Starting Time or Change Starting Mode menu

Start

- Mode
  Here the start mode of the batch can be changed.
  In the selection, only the modes which are permitted in the order type configuration are offered.
- Date
  Input box for the date in the format day.month.year
  An entry is only possible in the start mode 'by time' and 'by time/auto'.
- Time
  Input box for the time in the format hour:minute:second
  An entry is only possible in the start mode 'by time' and 'by time/auto'.

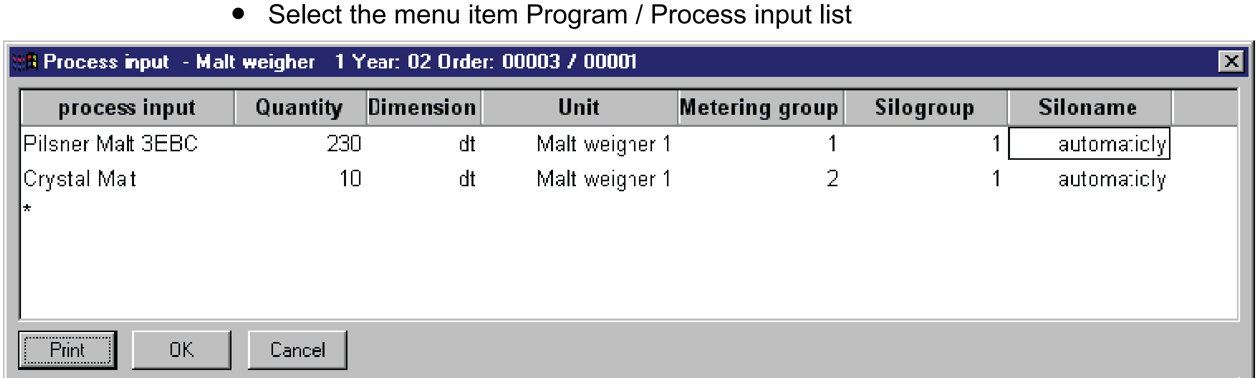
10.4.4.12 'Batch process input list' dialog box

Basics

The dialog is used to process materials if a process input list has been defined for the job type.

Selecting the dialog box

- Select the File / Open Batch List menu item
- Select the menu item Program / Process input list



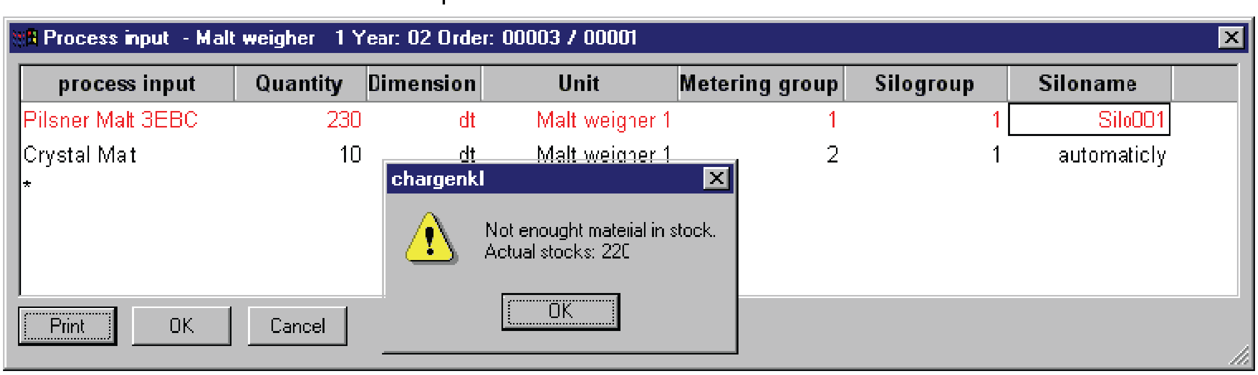
This dialog is only available if you have set "Process input required" in your configuration of the order type of the batch.

You can modify the values in the following columns of the process input list, provided the current batch status is "locked":

- Volume
  The material metering volume in this line
- Metering group
  The material metering group in this line (metering sequence).

Notes:

- In contrast to this, the ability to make changes can always be forced, i.e., regardless of the batch state, with the following switch:
- In file: ..<proj-path>\sys\BALIEDIT.INI
  [Settings]
  ...
  CompListEditAlways=1
  This corresponds to the old behavior (to V4.6)
- Metering processes with the selected metering group must be defined in the recipe procedure in order to ensure weighing of the material.
- After you have enabled stock management and tank selection, you can select a tank from the "Siloname" column by means of a text list selection dialog, and thus override the "automatic" or preset selection.



If the material volume in this tank volume is too low, a corresponding message pops up, and the text of the corresponding process input line is marked in red color. You can only select a tank that contains the corresponding material.

You can always return to "automatic" mode.

10.4.5 Views

10.4.5.1 View 'order list'

In the order list all current or future orders can be displayed. You can update the displayed batches/orders manually by reselecting the current view in the menu or by using the button in the toolbar.

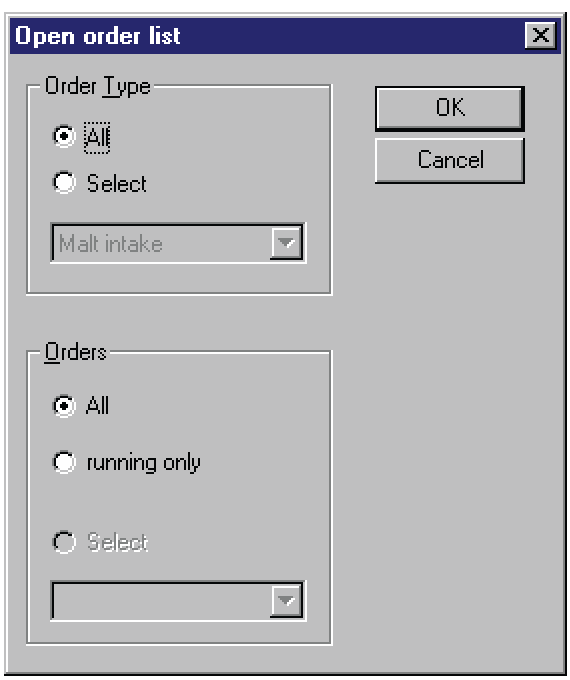
To select the dialog box:

- From the main menu, select "Process monitoring" -> "Batch list"
- Select "File"
- Select "Open order list"

Dialog 'open order list'

With this dialog, a view of the order list is opened. Depending on the selected option, all or only partial amounts of the running and future orders are displayed.

The dialog can be opened via the menu item 'file' – 'open order list' or via the button  of the button bar.



Order type

- All
  All orders of different order types are displayed.
- Selection
  Only 1 order type is selected for which the orders are displayed.

Orders

- All
  All orders, i.e. current and future orders, are displayed.
- Running only:
  Only orders where at least one batch runs are displayed.
- Selection:
  Only batches of the order selected in the combo box are displayed.

Key 'OK'

The order list is opened according to the selected options.

Key 'Abort'

The selection is aborted and it is returned to the old view.

View

The column sequence (Layout) can be determined per order type. Individual columns can also be dropped completely. For the order list, several layouts can be determined.

- For the selection 'all order types'
- for every order type
  For every order type, type-specific parameters can be accepted in the layout for every order.

See also

Layout adjustments (Page 766)

Operation with the mouse

- Double-clicking on the status of an order opens the dialog 'change order status'
- A double-click on the size or the batch number opens the dialog 'Increase batch number'.
- A click on the row number and dragging opens the 'Move Order' dialog
  (with the row number of the click as 'from' and the row number when releasing the mouse button as 'to')

Menu bar

Program

- This menu area is filled according to the entries in 'Menuappl.ini'.

Form editor [Ctrl+F7]

- Call of the application 'Recipe editor' with the current master recipe

Unit overview [Ctrl+F8]

- Call the application 'sequence control' for the selected order.

File

Open order list [F2]	Branch out to the dialog 'order list'
Open batch list [F3]	Branch out to the dialog 'batch list'
Open batch monitoring [F5]	Branch out to the dialog 'batch monitoring'
Open batch monitoring history [F6]	Branch out to the dialog 'History'
Batches for order [Shift+F3]	For the current line the batch list is called.
Batch monitoring [Shift+F5]	For the current line the batch monitoring is called.
Last view [F9]	The currently selected view is closed and the view which has been selected before is opened.
Print... [F4]	Branch out to the dialog 'print'
Close [Ctrl + F4]	The current selection is completed and the engineering view appears.

Process

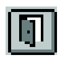
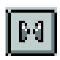
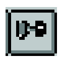
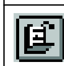
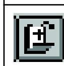
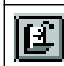
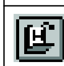
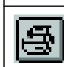
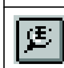
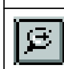
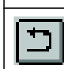
New order	Branch out to the dialog 'New Order'
Delete order	The batches of the order of the current line are deleted.
Move order	An order is displaced to a different position.
Order state	For the order of the current line the order state can be changed.
Batch number	For the order of the current line the number of batches can be increased. If the number of batches is supposed to be reduced, this must be executed by changing the batch state.

Options

Button bar	Switch button bar on/off
Status line	Switch status line on/off
Function keys	Switch function key marking on/off
Layout	Create/change layout for the table.

Colors order state	Define colors for the order state
Disable MES	Temporarily remove the order input lock initiated by the MES (see Dialog define batch generation (Page 729))

Button bar

	Close Application
	Acknowledge ICM error
	Acknowledge horn
	Open order list
	Open batch list
	Open batch monitoring
	Open history
	Print
	Open batch list for current order
	Open batch monitoring for current order
	Back to the last view

Status line

1. Field:	Current line / maximum number of lines
2. Field:	Order number of the current line
3. Field:	unused
4. Field:	Additional notes
5. Field:	unused
6. Field:	Total number of data records in the batch database bali.dbf

10.4.5.2 View 'batch list'

View 'batch list'

To select the dialog box:

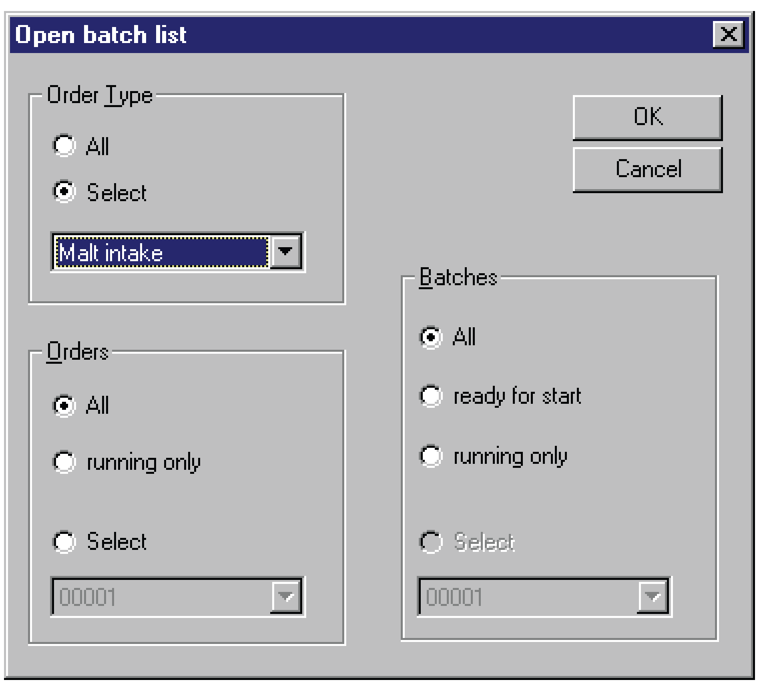
- From the main menu, select "Process monitoring" -> "Batch list"
- Select "File"
- Click "View batch list"

In the batch list all current or future batches can be displayed.

Dialog 'Open batch list'

With this dialog, a view of the batch list is opened. Depending on the selected options, all or only partial amounts of the current or future batches are displayed.

The dialog can be opened by selecting "File" > "Open batch list" or by pressing the button on the toolbar.



Order type

- all
  All orders of different order types are permitted. The batches are displayed without any additional parameters, as these ones are order type-dependent.
- Selection
  Only 1 order type is selected. The display is executed with additional batch parameters, as projected for this order type.

Orders

- all
  All orders, i.e. current and future orders are selected.
- running only:
  Selection of orders where at least one batch runs.
- Selection:
  Only batches of the order selected in the combo box are selected.

Batches

- All
  All batches of the selected order are displayed.
- In the mode 'ready-to-start'
  Only the batches that have already been loaded into the PCU (CAS) for the next start of a unit are displayed.
- In the mode 'running'
  Only batches of the selected orders which have already been started in the PCU (PS) are displayed.
- Selection
  The batch selected in the combo box is displayed.

Key 'OK'

The batch list is opened according to the selection.

Key 'Abort'

The selection is aborted and it is returned to the old view.

View

The column sequence (Layout) can be determined per order type. Individual columns can also be dropped completely. Several layouts can be determined for the batch list.

- For the selection 'all order types'
  In this layout no order parameters can be included
- for every order type
  Type-specific parameters can be included for every order type in the layout.

See also

Layout adjustments (Page 766)

Operation with the mouse

- Double-clicking on the status of an order opens the dialog 'change order status'
- Double-clicking on the start time or the start mode opens the dialog 'change start dates'.
- A click on the row number and dragging opens the 'Move Batch ' dialog
  (with the row number of the click as 'from' and the row number when releasing the mouse button as 'to')

Menu bar

Program

This menu area is filled according to the entries in "Menuappl.ini".

Control recipe [F7] 'Control recipe'	Call of the application 'Recipe editor' with the current master recipe
Process input list [F8]	Opens the process input list for the selected batch.
Formula Editor [Ctrl+F7]	Call of the application 'Recipe editor' with the current master recipe
Unit overview [Ctrl+F8]	Call of the application 'sequence control' for the selected batch.

File

Open order list [F2]	Branch out to the dialog 'order list'
Open batch list [F3]	Branch out to the dialog 'batch list'
Open batch monitoring [F5]	Branch out to the dialog 'batch monitoring'
Open batch monitoring history [F6]	Branch out to the dialog 'History'
Batches for order [Shift+F3]	For the current line the batch list is called.
Batch monitoring [Shift+F5]	For the current line the batch monitoring is called.
Last view [F9]	The currently selected view is closed and the view which has been selected before is opened.

Print... [F4]	Branch out to the dialog 'print'
Close	The current selection is completed and the engineering view appears.

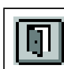
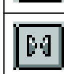
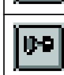
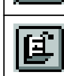
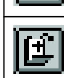
Process

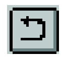
New order	Branch out to the dialog 'New Order'
Add batch	For the order of the current line, a batch is added.
Move batch	Branch out to the dialog 'Move batch'
Batch state	Branch out to the dialog 'Change batch states'
Batch size	Branch out to the dialog 'batch size'
Change start time	Change start time of the batch
Change start mode	Change start mode of the batches

Options

Button bar	Switch button bar on/off
Status line	Switch status line on/off
Function keys	Switch function key marking on/off
Parameter window	Opening of the window 'batch parameter'
Layout	Create/change layout for the table
Colors batch state	Change the colors for the batch condition
Disable MES	Temporarily remove the order input lock initiated by the MES (see Dialog define batch generation (Page 729))

Button bar

	Close
	Acknowledge ICM error
	Acknowledge horn
	Open order list
	Open batch list
	Open batch monitoring
	Open history
	Print
	Open batch list for current order

	Open batch monitoring for current order
	Back to the last view

Status line

1. Field:	Current line / maximum number of lines
2. Field:	Order number of the current line
3. Field:	Batch number of the current line
4. Field:	Additional notes
5. Field:	unused
6. Field:	Total number of data records in the batch database bali.dbf

10.4.5.3 View batch monitoring

View batch monitoring

To select the dialog box:

- From the main menu, select "Process monitoring" -> "Batch list"
- Select "File"
- Click "Open batch tracking"

All running or future batches can be displayed in the batch monitoring.

As the menus, symbols and dialogs of the batch monitoring correspond to those of the batch list, only the deviations are described here.

Dialog 'Open batch monitoring'

The open dialog is the same as used in batch list view.

See also

View 'batch list' (Page 756)

View

The column sequence (Layout) can be determined per order type. Individual columns can also be dropped completely. For the batch monitoring, several layouts can be determined.

- For the selection 'all order types'
  In this layout no order parameters can be included
- for every order type
  Type-specific parameters can be included for every order type in the layout.

See also

Layout adjustments (Page 766)

Menu bar

See also: View 'batch list' (Page 756)

The dialog of the same name can also be called up with the 'Options' / 'Recipe unit procedure status colors' menu option.

10.4.5.4 History view

History view

To select the dialog box:

- From the main menu, select "Process monitoring" -> "Batch list"
- Select "File" >Click "View history"

In the history, all started, running, aborted and deleted batches can be displayed.

A batch appears in the history as soon as it has been started or deleted. The sequence of the batches is determined via the start/delete time whereby the batch with the most current time stands at the beginning of the list.

Dialog 'Open history'

Basics

With this dialog, a view of the history is opened. Depending on the selected options, either all or only partial amounts of the batches from the history are displayed. In general, the batches are always displayed pages by page due to the high amount. The amount of batches per page can be allocated in 'Options' 'adjustments' in the configuration view.

Year

- The year determines the archive file from which the batches are displayed.
- There is a separate file for each year called "`<proj-path>\BALI\hist_xx.dbf`" (xx=year). i.e. the history can only be moved within one calendar year.
- This file is always completely read in. It contains all batches that were created in the batch year. The start time also contained in the file can differ!

Month, Day

- These two indications determine the starting point in the history.
- The page is displayed which contains the first batch with the appropriate start/delete time. These indications do not result in any restrictions in the selection.
- This page can also contain batches with other start times (pages are fixed and are always completely indicated). From there, one can then move again in the whole file

Order type

- All
  All orders of different order types are permitted.
- Selection
  Only 1 order type is selected.

Orders

- All
  All orders i.e. current and future orders are selected.
- Selection:
  Only batches of the order selected in the combo box are selected.

Batches

- All
  All batches of the selected orders are displayed.
- Selection
  The batch selected in the combo box is displayed.

Key 'OK'

The history is opened according to the selection.

Key 'Abort'

The selection is aborted and it is returned to the old view.

View

The column sequence can be determined. Individual columns can also be dropped completely.

The definition of the layout and the color of the batch states are identical to the batch list.

Functions for processing, creating, displacing or deleting batches can't be executed here. For displaying the batch parameters, the parameter window is to be opened from the menu 'options'.

Operations with the mouse

No operations with the mouse are possible in the list.

Menu bar

Program

This menu area is filled according to the entries in 'Menuappl.ini'.

File

Open order list [F2]	Branch out to the dialog 'order list'
Open batch list [F3]	Branch out to the dialog 'batch list'
Open batch monitoring [F5]	Branch out to the dialog 'batch monitoring'
Open batch monitoring history [F6]	Branch out to the dialog 'History'
Last view [F9]	The currently selected view is closed and the view which has been selected before is opened.
Print... [F4]	Branch out to the dialog 'print'
Close	The current selection is completed and the engineering view appears.

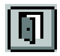
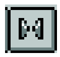
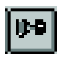
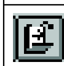
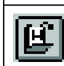
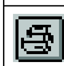
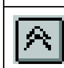
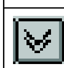
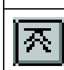
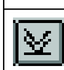
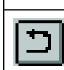
View

Page forward [F7]	Page forward by one page (in the future)
Page down [F8]	Page down by one page (in the past).
First page [Shift+F7]	Jump to the first (most current) page of the history
Last page [Shift+F8]	Jump to the last (oldest) page of the history

Options

Button bar	Switch button bar on/off
Status line	Switch status line on/off
Function keys	Switch function key marking on/off
Parameter window	Opening of the window 'batch parameter'
Layout	Create/change layout for the table
Colors batch state	Change the colors for the batch condition

Button bar

	Close
	Acknowledge ICM error
	Acknowledge horn
	Open order list
	Open batch list
	Open batch monitoring
	Open history
	Print
	Page forward
	Page down
	Jump to the first page
	Jump to the last page
	Back to the last view

Status line

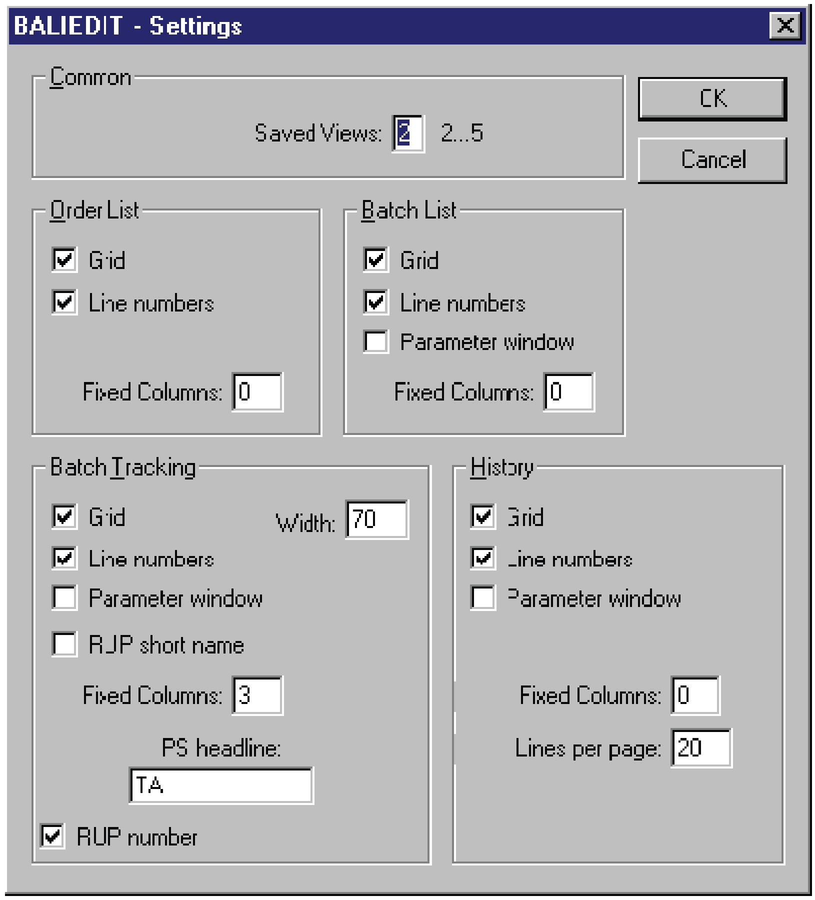
1. Field:	Current page/maximum number of pages
2. Field:	Order number of the current line
3. Field:	Batch number of the current line
4. Field:	Additional notes
5. Field:	unused
6. Field:	Total number of data records in the selected history database hist_xx.dbf

10.4.6 Adjustments

To select the dialog box:

- From the main menu, select "Process monitoring" -> "Batch list"
- Select "File"
- Click "Close"

- Select "Options"
- Select "Settings"



With this dialog the basic settings for the editor and four views can be made.

General

'Filed Views'	Number of views which are held in the page file.

Order list

Adjustments for the order list view.

'Grid'	Indicate table grids.
'Line numbers'	Indicate column for line numbers.
'Fixed Columns'	Number of fixed columns on the left table edge. These are not moved by the vertical scrolling.

Batch list

'Grid', 'line number' and 'fixed columns' as for order list	
'Parameter window'	View parameter windows automatically by opening the view.

Batch monitoring

'Grid', 'line number', 'fixed columns' and 'parameter window' as with batch list	
''Width''	Width in display units (Pixel) for the Seq columns.
'Seq.-short name'	Use unit short names in the Seq columns.
'Seq.-heading'	Heading for the Seq columns.
'Seq.-number'	Indicate unit number in the Seq column heading in addition.

History

'Grid', 'line number', 'fixed columns' and 'parameter window' as with batch list	
'Lines per page'	Number of lines which are displayed per page in the history.

10.4.6.1 Layout adjustments

To select the dialog box:

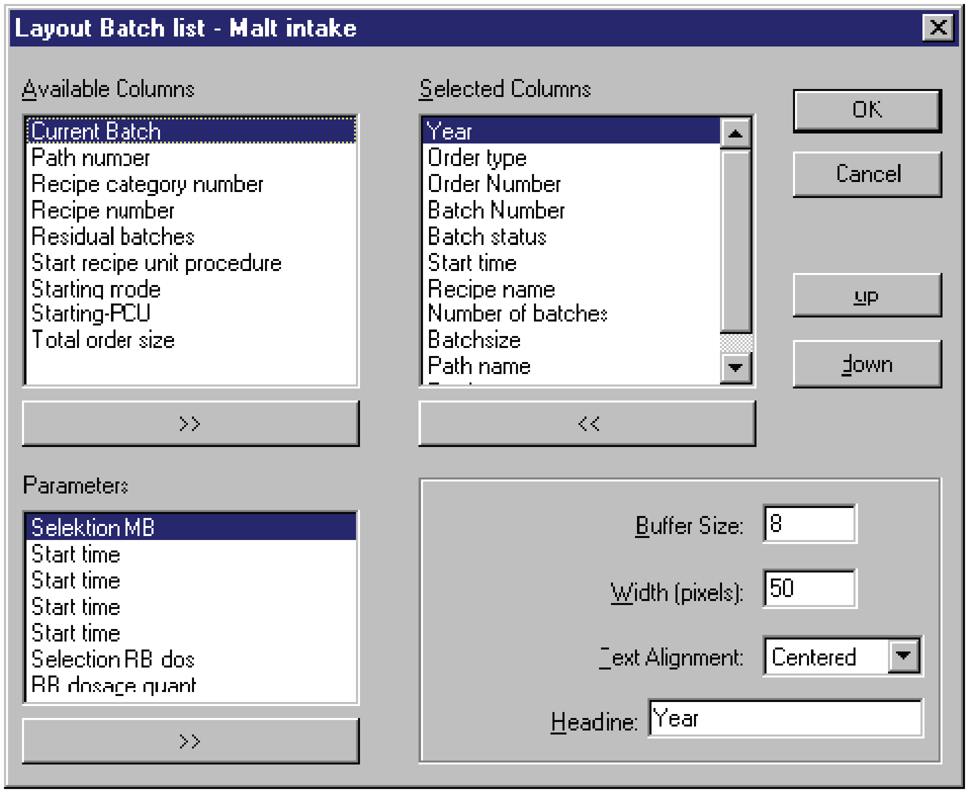
- From the main menu, select "Process monitoring" -> "Batch list"
- Select "File"
- Select "Open order list" or "Open batch list", or click "Open batch tracking"
- Select "Options"
- Select "Layout"

The layout adjustments can be adjusted for order, batch and batch tracking separately.

For every order type, specific layout adjustments can be executed.

For the view 'all order types', one layout adjustment can be executed.

The adjustments are changed for the view which has just been displayed. The view is displayed in the header line of the dialog.



Available columns

In this list, all remaining columns that aren't used in the view are displayed.

Selected columns

In this list, all columns of the view are indicated in the sequence from left to right.

Parameter

In this list, the parameters of the order type are indicated. The list is empty in the layout dialog of the order list or batch list in which all types have been selected.

Key '<<'

With this key, the selected column is removed from the list.

Key '>>'

With this key columns are added in the view.

Note

Parameters can't be displayed in the order list as they are batch-related!

In the batch list view with the display of all batches, no parameter can be indicated.

Key 'up'

With this key, the selected column is moved upwards by one position.

Key 'down'

With this key, the selected column is moved downwards by one position.

Input field 'buffer size'

Input of the maximum character count for the selected column (8-32).

Input field 'Width in pixel'

Input of the column width in display units (pixel) (25-250).

Input field 'text alignment'

The text can be aligned in the left, right or middle of the column.

Input field 'Headline'

Input of the column header.

10.4.6.2 Batch/order and sequence status colors dialog box

To select the "Batch/order status colors" dialog box:

- Select "File"
- Select "Open order list" or "Open batch list", or click "Open batch tracking"
- Select "Options"
- Select "Order status colors"

To select the sequence status dialog box:

- Select "File"
- Click "Open batch tracking"

- Select "Options"
- Select "Batch/order status colors"

The foreground and background colors indicating the status can be set individually.

Key 'Changing'

Open the Windows Standard dialog 'Define colors' from which the requested color can be selected for the front and background.

10.4.6.3 Password protection

All operations in the application 'BaliEdit' are protected by a password. You divide between various password levels.

- Change parameters
- Change, enable, disable status of a batch
- Abort batch
- Input, delete order
- Add and delete batch
- Configuration

10.4.7 Example

Chapter 3 of the "*Recipe system*" manual contains a simple recipe sample. This recipe controls the draining of caustic cleaning solutions, and shall now be started by the order system.

In this example, the master recipe "Cleaning" is assigned to the recipe category "Cleaning processes" (CIP).

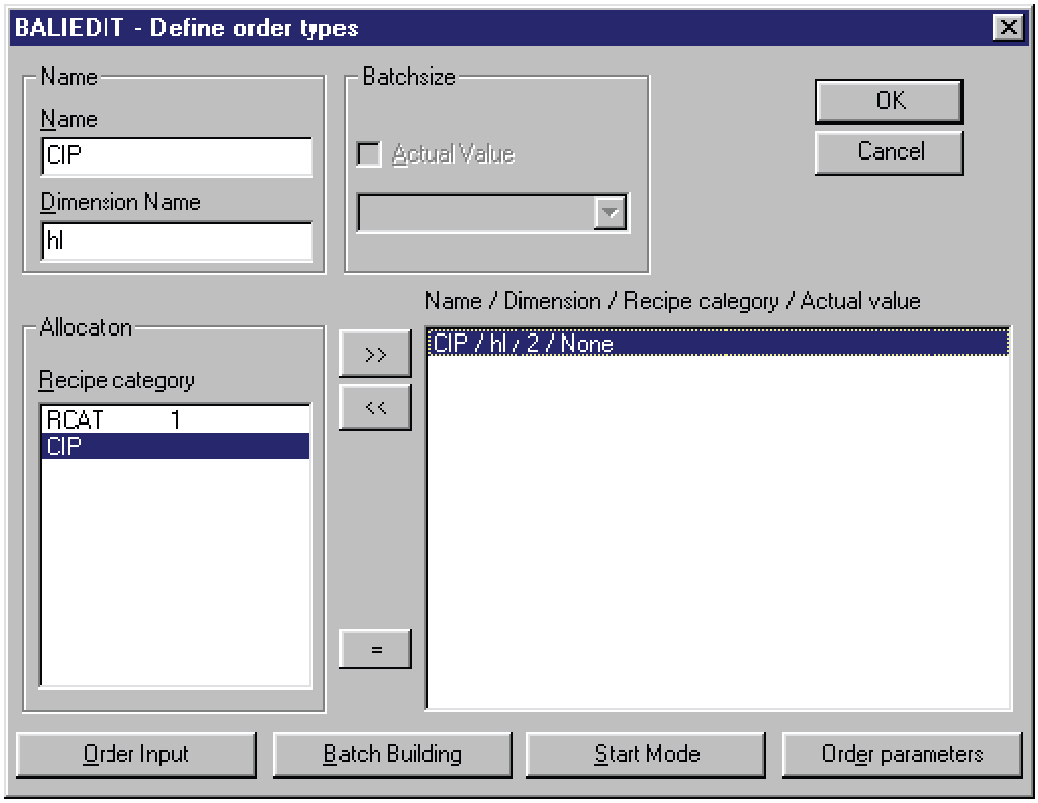
Starting the order system

- From the main menu, select "Process monitoring"
- Double-click "Batch list"

To set up the order type for the order system

- From the main menu, select "File"
- Click "Close"
- From the menu, select "Configuration"

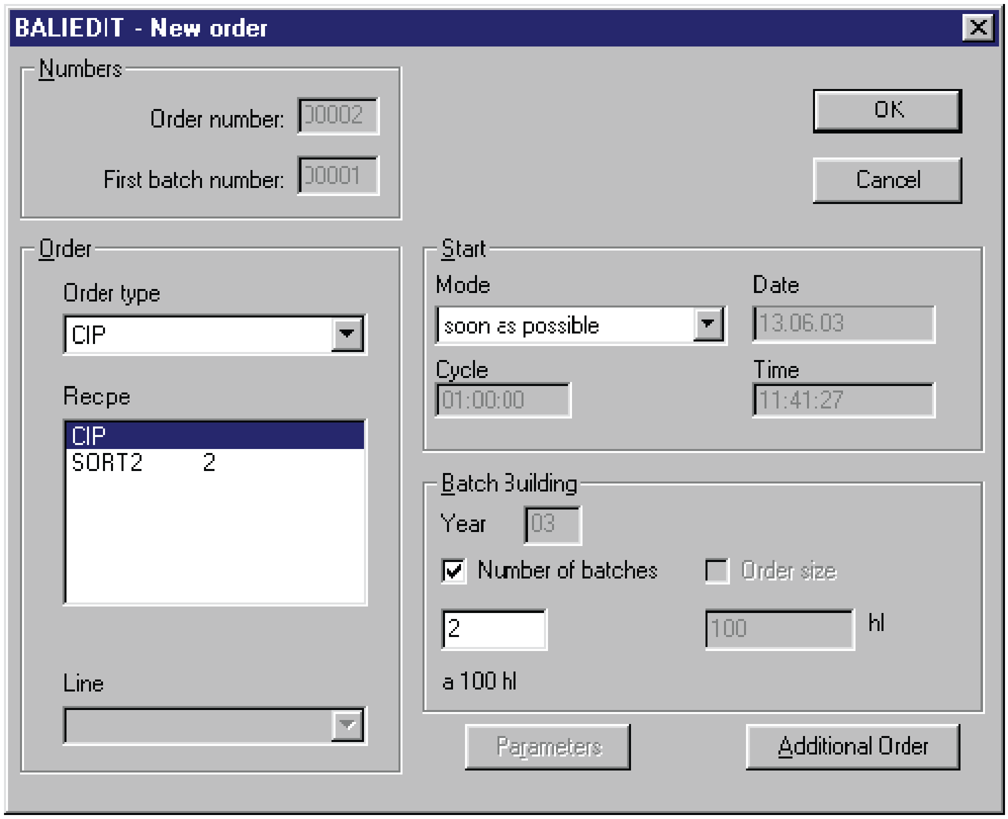
- Select "Order types"



- Enter the name (CIP) and the dimension name (hl) in the dialog box. Apply the cleaning recipe category by double-clicking its name. For this example you can accept the batch building and order input defaults. Order parameters are not required, because the recipe does not contain any parameters.
- Click the button. In the dialog box, set the start mode "soon as possible" and "time controlled". A description of this setting is found in chapter "History (Page 712)". You can now confirm all settings with OK.

Creating a new order

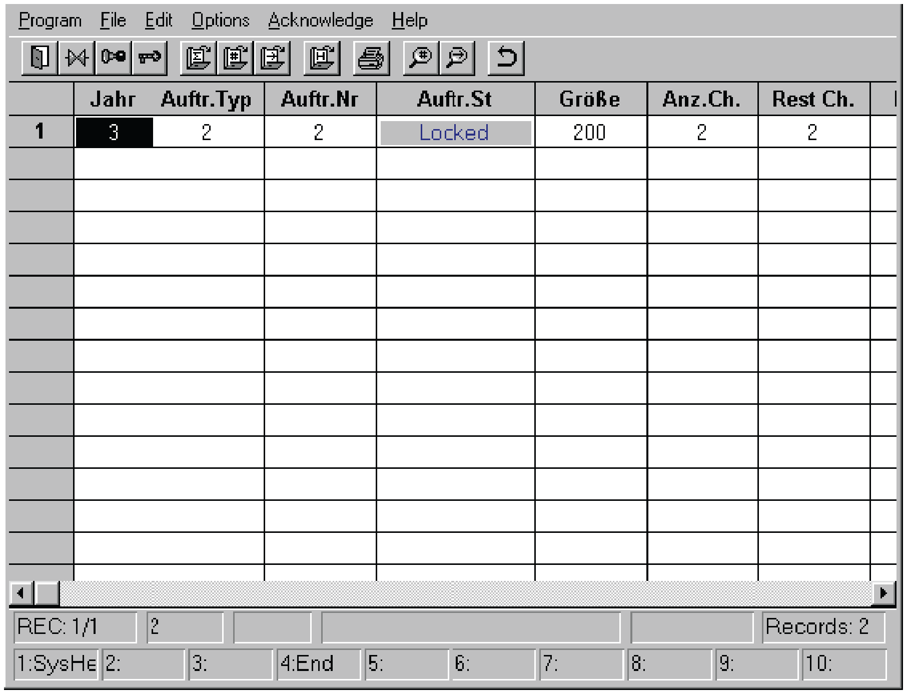
- Click "File" -> "Close"
- Click "Edit" -> "New order"



In this dialog box, you can increase the number of batches. Click OK to confirm your entries and to close the dialog box.

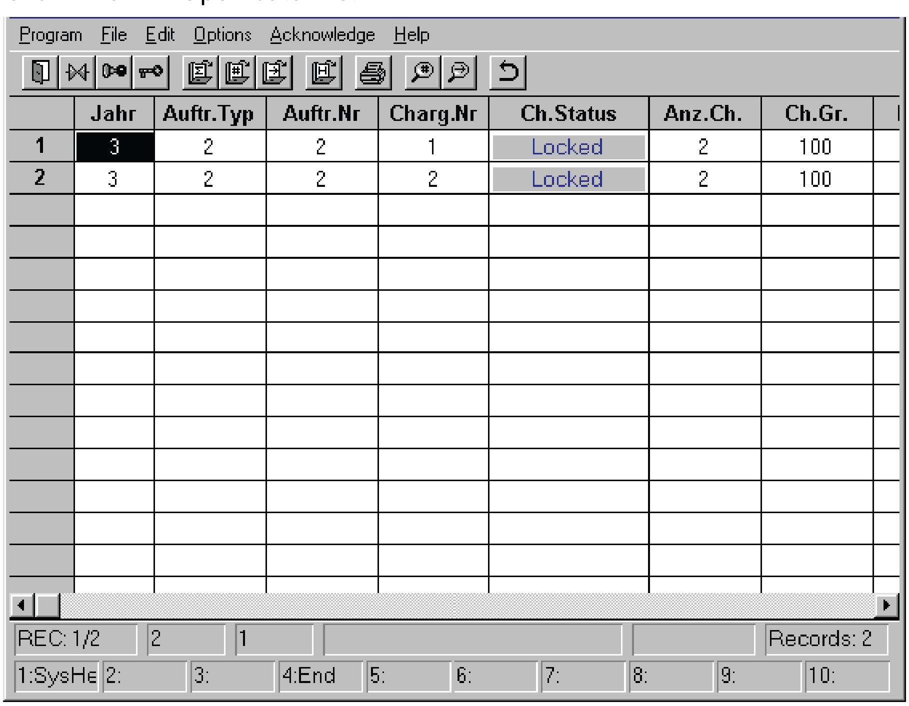
The order list should now contain an order.

- Click "File" -> "Open order list"



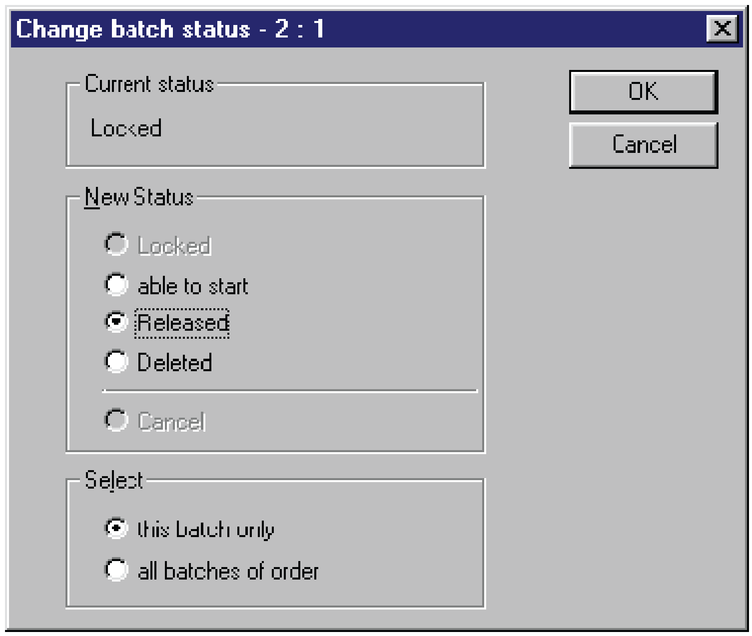
The batch list should contain two batch orders, because this order consists of two batches.

- Click "File" -> "Open batch list"



The order is locked and cannot yet be processed, which means you need to enable batch processing.

- Click "File" -> "Open order list"
- Click "Edit" -> "Order status"
- In the dialog box, set "Released", and then close the dialog box with OK



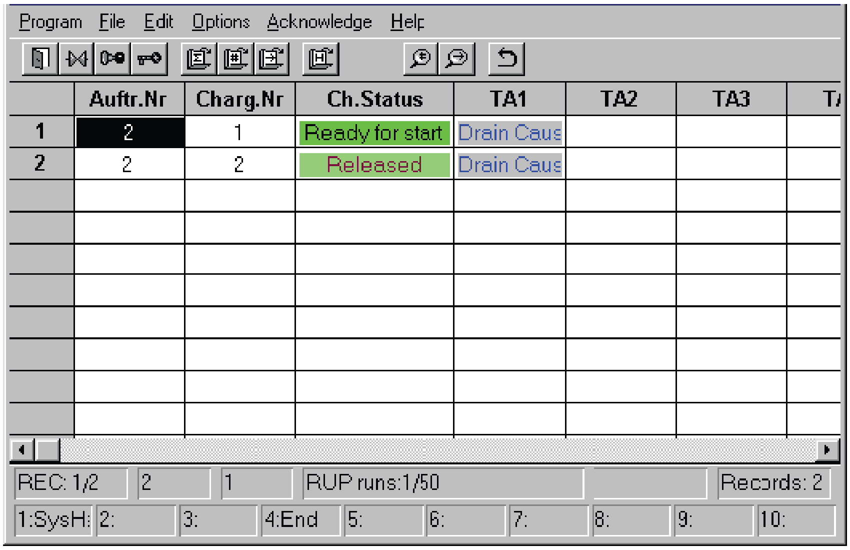
The order status is briefly set to "enabled" and then changes to "Started".

In the plant overview, you can monitor the recipe sequence of both batches (see section 2). The "Batch tracking" feature also allows you to monitor the batch process.

After processing of both batches is completed, the order status changes to "Done".

Batch tracking

- Click "File -> Open batch tracking"



10.5 ‘Batch management’ application

10.5.1 General

Short description

The order system has the task of processing a list of batches in cooperation with the recipe control on the IOS and subordinated PCUs. This list can be created by the order system on the IOS or by a higher-level system (PMS) and the batch sequence and list content can be changed later.

Furthermore, the order system makes it possible to track and monitor the processing of batches.

The order system is called from the main menu on the "Production" / "Batch list" tab.

Note

The order system and the recipe system share a common database that contains data such as recipe headers and similar items. After you modify a recipe (e.g. the header), you need to restart the batch list program in order to refresh the program with the data which were written to interim memory for reasons of performance.

10.5.2 Functions

10.5.2.1 Order and recipe system redundancy

The following conditions must be fulfilled so that the redundancy of the order and recipe system works correctly:

- For the AREA, two redundant IOS servers must be configured in the "SiteCfg" site configuration tool.
- By activating the recipe server, the application synchronization must run.
- At the time of the recipe server failure, the reserve recipe server must have been started.
- The startup as active recipe server must have been completed without any interruption.

10.5.2.2 Order types

- Up to 255 order types can be configured.
- There is a 1:1 relation between recipe category and order type
- One different list of order parameters can be configured per order type
- The adjustments for batch input, as well as order and batch number allocation can be set up for each order type specifically.

Basics

- The following start modes are available: ‘as soon as possible’, ‘by event’, ‘after absolute time’, ‘after time with automatic adaptation’.
- The possible start modes can be configured order type-specific.

'as soon as possible' start mode

The batch is started as soon as the required start sequence is free, in automatic mode, not stopped and not disabled. If several batches are released and are waiting for the same sequence the batch highest in the list is started.

'by event' start mode

The batch is loaded in the CAS block of the S7 as soon as this is free. There, the system waits until a bit in the CAS dataset is set by the user (event). The bit enters the name boFreeMode4 in UDT 718.

The sequence (and with it the batch) is only started if the sequence is free.

Note

If this property is used, the application developer is responsible for assigning the order and batch numbers. If a batch with the same order and batch number is started several times, this can lead to incorrect blocking of batch processing.

Starting a batch that is to be produced with a stream or line recipe is then only possible in the "by event" start mode if the batch was generated and set to the "Ready" status with the batch list application since the system is otherwise not capable of determining the correct stream or line number.

'by time' start mode

The batch is started when the batch start time is reached or exceeded. The batch is only started if the CAS required for the sequence and the sequence itself are free.

'after time/auto' start mode

The start is as with the 'by time' start mode. However with this start mode, the start times of the follow-on batches are automatically adapted if there are delays in starting the batch. This adaptation is only for batches with this start mode and batches that have the same start sequence.

Manual change to the start time / start mode

The start time / start mode is changed in the batch list by selecting batches and then calling the menu or shortcut menu commands:

- "Edit / Change start time"
- "Edit / Start mode / <new start mode>"

If the dialog "New start time" is closed with 'OK', the procedure is as follows depending on the selected start mode:

- **Start mode: 'as soon as possible':**
  There is no adaptation of the other batches.
- **Start mode: 'By time' and 'By time /Auto':**
  All following batches with the mode "by time / Auto' are adapted to the start cycle.
  The adaptation ends at the first batch with the 'by time' mode. This allows gaps to be created (for example weekend).
  Batches with the mode 'as soon as possible' or the status 'Ready.', 'Started', 'Deleted' and 'Finished' are skipped.

As soon as a batch changes to the 'Started" status, the start times of the remaining batches with the same start unit are triggered.

Initially the current time is entered as the start time in the started batch.

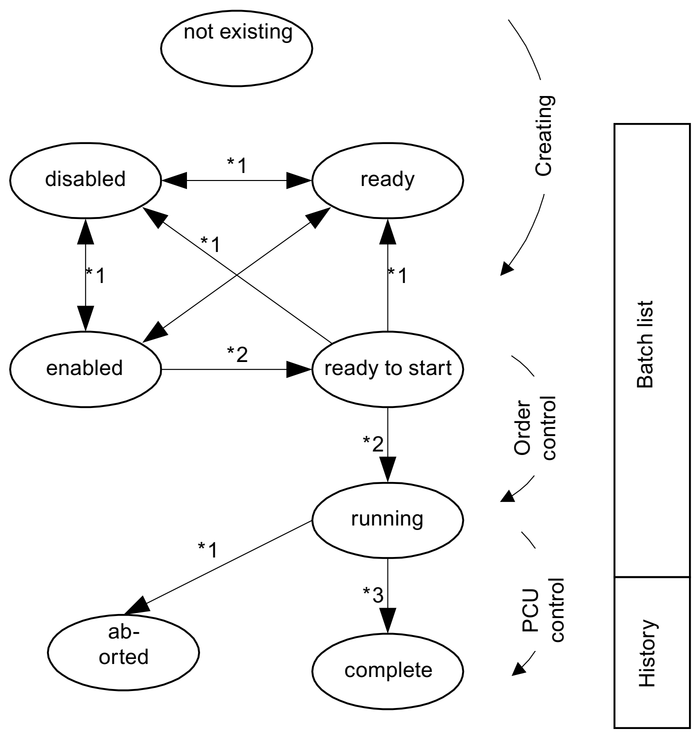
Then all batches with the mode 'by time / Auto' are adapted to the start cycle.

To prevent constant adaptation of the start times after a batch has started, a tolerance limit can be set for checking the start times.

- The tolerance limit can be set in the configuration file "<proj-path\\sys\bali.ini" in the section [balidata] TimeCorrTolerance=xx .
- TimeCorrTolerance=60 is the default value (in seconds) and means that there will only be a reaction if the start time is exceeded or underrun by more than 60 seconds.

10.5.2.3 Conditions and transitions

Basics



*1 Operation
*2 Batch control
*3 PCU control

Table 10-4 Notes

Code	Description
*1	Status change by operator
*2	Status change by OS order control
*3	Status change by AS order control
Hatched area	Here the operator can change to the status "Deleted"

Table 10-5 Description of the possible status transitions

Status	Possible change to	By
Disabled	Ready	Operation
	Enabled	Operation
	Deleted	Operation
Ready	Disabled	Operation
	Enabled	Operation
	Deleted	Operation

Status	Possible change to	By
Enabled	Disabled	Operation
	Ready	Operation
	Deleted	Operation
	Ready-to-start	OS order control
Ready-to-start	Disabled	Operation
	Deleted	Operation
	Started	PCU control
Started	Canceled	Operation
	Done	OS order control
Error n (n = error number)	Disabled	Operation (if error corrected)
	Ready	Operation (if error corrected)
	Enabled	Operation (if error corrected)
	Deleted	Operation
Done	---	---
Deleted	---	---
Canceled	---	---

Table 10-6 Meaning of error numbers in batch state:

Error 1	Recipe header could not be read.
Error 2	Conversion of the process input list failed.
Error 3	Process input list user function delivers FALSE
Error 4	Writing of batch status into bali.dbf failed
Error 5	Loading of batch in the CAS of start batch failed
Error 6	no CAS copy for the batch
Error 7	no unit sequence for the batch
Error 8	Invalid batch values on enabling: • Y<0 (Year) • RT<=0 (Recipe type) • ONr.<=0 (Order number) • BNr.<=0 (Batch number) • LNr.<=0 (Line number)
Error 9	For user function saves

Error 1: Recipe header could not be read.

Incomplete or incorrect recipe engineering is present on this machine.

Error 2: Handling of errors in the process input list

First, a check must be performed to ascertain whether a process input list is used for the corresponding order type.

If this isn't the case, the option **process input list required** can be deselected under the adjustment **order input**.

If a process input list is used, there will be a basic error in the input configuration or material and storage location administration.

Error 3: Process input list user function delivers FALSE

refer to user interface documentation (not part of the system)

Error 4: Writing of batch status into bali.dbf failed

The writing of the file bali.dbf can't be performed without errors. The reason can be a full hard disk. If the file is located on a network drive, a missing network connection can be the reason.

Error 5: Loading of batch in the CAS of start batch failed

The reason is usually a communication problem with the PCU.

Error 6: no CAS copy for the batch

An attempt was made to start a batch which is using a start unit without having an entry in the CAS image. This can only happen when the number of CAS entries is lower than the number of sequencers.

Error 7: no unit sequence for the batch

Incomplete or incorrect recipe engineering is present on this machine.

Error 8: Invalid batch values on enabling

The key of the batches is not valid.

10.5.2.4 Order parameters

Basic principle

- For every order type up to 220 parameters can be assigned.
- All batches of an order type have the same order parameter definition.
- The parameters can be used in operations of the recipe procedure

Parameter template

- DFMs (digital function modules) can be referenced from all PCUs. Input is the same (upper limit, lower limit, decimal point, dimension,) as defined in the process cell configuration.

Parameter type

- Normal
  The entry and display is the same (upper limit, lower limit, decimal point, dimension,) as defined in the process cell configuration.
- Summation parameter
  For batch order as 'normal'. For the input of an order, the input refers to the amount of batches, i.e. for the batch generation the value is divided into individual batches (see transfer of the batch parameter).
- Free parameter
  A free parameter it is a 32 bit integer. whose value is not checked as in the case of a DFM.

Acceptance of batch parameters

- By creating batches, all order parameters, with the exception of sum parameters, are copied to the batches.
- The sum parameters refer to the number of batches and are calculated for the individual batches as follows: parameter (batch) = parameter(order) / size(order) * size(batch)
- A sum parameter must be defined during configuration under the application 'Order system' > 'Configuration' > 'Order type' > 'Order parameter'.

Preset order parameters

- The parameters can be preset with values of the recipe process parameter.

Basics

The following types are possible:

- Start with 1 per order
- Continuously for year and order type
- Order number as calendar week

The order numbers are proposed by the order system and assigned consecutively.

The user has the possibility to overwrite this preset. An inspection occurs in this case to ascertain whether the input is permissible with regard to the explicitness of the order and batch numbers. The input of an order number is only allowed if it has been enabled under 'order type', 'configuration', 'order type', 'order input', 'allowed inputs' for this order type.

Order and batch number range

The assignment of order and batch numbers is governed by the following rules:

- The next free number is always searched for in ascending order starting at 1.
- The maximum order and batch number is 32767.

Order and batch numbers higher than 32767

The current version of the system allows order and batch numbers higher than 32767 to be assigned. For reasons of downward compatibility, this function must be activated with the "ExtendedNumbers=1" entry in the "Balidata" section of the "<proj-path>\sys\bali.ini" file. The "ExtendedNumbers=0" entry restores the default setting and prevents order and batch numbers higher than 32767.

Start with 1 per order

- Order numbers are assigned consecutively per year and order type. As default, the batch numbers begin at 1 for every order. The other batches are numbered consecutively.
- The first batch number can be changed to any other number by the user (the option of making the change must be enabled in the configuration of the order type).
- If the user enters the order number, this is checked to ensure it is unique and if it is not, the next valid number is proposed.

Continuously for year and order type

- The batch numbers are assigned consecutively per year and order type. The first batch receives the batch number of the last batch of the preceding order + 1. All further batches are numbered consecutively.
- As default, the order number is set to equal the first batch number and is saved in a file to avoid conflicts if the assignment mode is changed.
- The order number can be changed to any other order number by the user since it is always unique due to the batch number.
- If the user enters the batch number, this is checked to ensure it is unique and if it is not, the next valid number is proposed.

The method of assigning the batch number is selected in the application 'Order system', 'Engineering', 'Order type', 'Batch generation'.

User intervention in the automatic assignment of batch numbers means that consecutive numbering of the batches per order can no longer be guaranteed.

The uniqueness of the batch numbers is, however, always ensured.

Order number as calendar week

Here, there are still two sub variants possible:

- Batch number running per calendar week
- Batch number running per calendar year

Order number as value plus calendar week

The calendar week can also be expanded by the order type and recipe number.

The formula is:

CW = CW + (RTyp + RTypOffset)*RTypFactor + (RecipeID + RecipeOffset)*RecipeFactor

The factors and offsets must be entered in the `otypes.ini` configuration file.

File:	`...<proj-path>\bali\otypes.ini`
Key:	`[ordertypex]` x stands for the order number
Entry:	`RecipeFactor=0`
Entry:	`RecipeOffset=0`
Entry:	`RTypFactor=0`
Entry:	`RTypOffset=0`

Reaching the maximum order and batch number

If the maximum order or batch number has been reached (=32767), new orders can only be entered again if the user deletes the relevant number file for the order type.

It is assumed that the user will then also backup the corresponding batch data in FRPROT because this will be overwritten with the new batches.

The number files are stored as follows:

- Order numbers in the '<proj-path>\bali\used_nr\onr_xx.yyy' files,
- batch numbers in the '<proj-path>\bali\used_nr\bnr_xx.yyy' files

Here,

- xx is the year and
- yyy is the order type number.

Files with'~' added to the name are backup files.

In the 'calendar week' mode, the hist_jj.dbf file must be deleted after first backing it up.

Release of numbers when deleting batches

Batch number, 'Always start with 1'

- Since in this case, the numbers are not stored anywhere, they are not really released.
- However when appending batches to an existing order, the highest existing batch number is always determined and the next number is used for the new batch.
- If the batch with the highest number is deleted, this number will be used for a new batch of this order.
- If, however, a batch in between is deleted, its number is not used again.

Batch number, 'consecutive numbering'

The number of a deleted batch is always released and used again.

Order number

The number is only released when no batch from the order has run so far; in other words 'Batch number' equals 'Residual batches' and the entire order is deleted.

Order number as calendar week

The precise deleted combination of order and batch number is released and can be used again.

10.5.2.5 Batch generation

There are two methods of generating batches.

- The allocated amount of the order is distributed equally to the number of batches.
- Batches with the maximum size are set up in order to reach the allocated amount. The remaining quantity is set up as a remaining batch.

Determine batch count → Order size in batches

If the order size is indicated in batches, the number of batches is determined.

The nominal batch size is taken as a batch size, except for if

- the nominal size is smaller than the minimum batch size => minimum batch size is used
- the nominal size is larger than the maximum batch size => maximum batch size is used

Determine batch count → Order size as amount

If an order size is allocated as amount, the following can be selected from two methods for determining the batch number and size:

- All batches have the same size. The number of required batches is calculated by the maximum batch size (rounded up to a whole number). Note: Nominal batch size doesn't need to be between the minimum and maximum batch sizes. If the batch size is lower than the minimum, the user is informed and must make a decision.
- n batches with the maximum possible batch size and one remaining batch.
  Note: If the remaining batch size is lower than the minimum, the user is informed and must make a decision.

The selection of the method for determining the batch number is executed under the application 'order system'->'configuration'->'order type'->'batch generation'.

10.5.2.6 Batch sequence

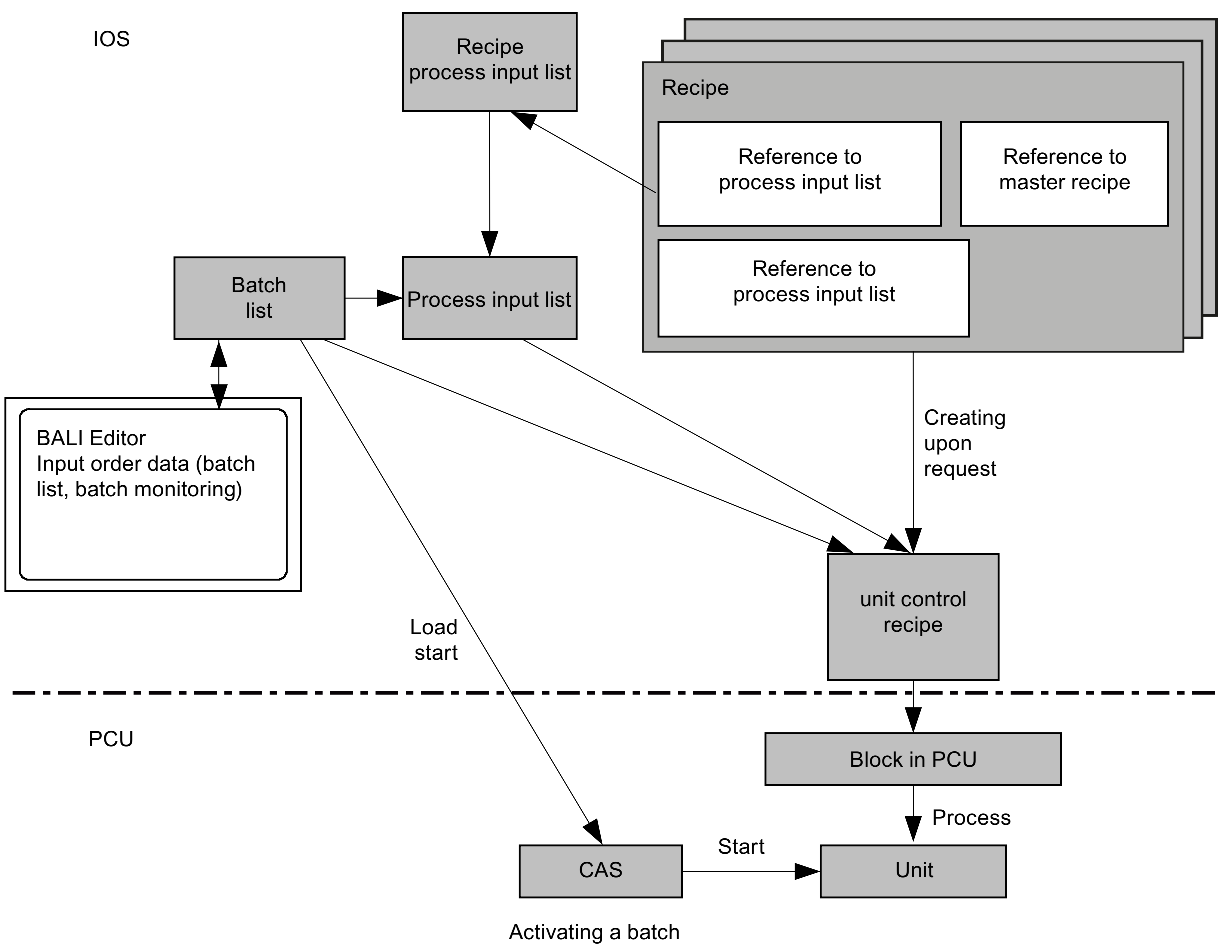
The batch is started as soon as the start conditions have been met for the batch. Once the batch has started, the recipe for this batch is requested by the PCU. The recipe load function in the IOS determines which recipe procedure is required based on the request.

The batch-dependent setpoints are scaled in this recipe procedure and selected setpoints are replaced by batch parameters or by values from the process parameters. This batch-dependent recipe procedure (name: control recipe) is loaded to the PCU and is executed there. If there are 'dose equipment operations' in the recipe procedure, the data from the process input list is incorporated into the control recipe.

The control recipes are created:

- from data from the recipe procedure (Recipe editor application)
- from data from the master recipe (Recipe editor application)
- from data from the batch order (Batch management application)
- and from data from the component list (not batch-related, Recipe editor application)

"Starting a batch" process description:



Following the order entry with the "Batch management" application, the batches are formed and incorporated into the batch list. The processing is performed in the sequence of the entries (and according to start mode).

Each recipe unit procedure is requested by the PCU before processing starts. The IOS converts the request in a batch-related control recipe. Order parameters, process parameters and data of the process input list flow into the control recipe.

10.5.2.7 Monitoring and diagnostics functions

Connection to the server

The following points should be considered during installation of the order/recipe system:

- The order/recipe system requires at least one, and with a redundant plant concept two IOS servers with enabled "Recipe control" function.
- In the PCU server of the IOS that is to become the active recipe server, you must call the menu command "Program / Activate recipe server".
- The recipe server only starts up if it receives acknowledgment from all PCUs that were set in the parameters in the plant configuration. If one of the parameterized PCUs does not report itself, it remains in the standby state. This restriction can be switched off.

If a connection to the recipe server cannot be established when the order system starts, a corresponding message appears.

File synchronization

The restart of an IOS as a recipe server or a recipe server switching automatically activates the 'Synchronization' application (synchro.exe).

This application ensures that the newest data is available on the IOSs. It compares files on the IOSs and copies the newest files.

In the event of a recipe server switching the synchronization is set in the `'...<proj-path>\sys\recipe.ini'` file.

```
[StandByControl]
StartFileSync=1
AbortWaitForSync=300
```

When StartFileSync = 1, the file synchronization is automatically activated by the redundancy switching. If no automatic file synchronization is selected ('StartFileSync=0'), it will wait for a maximum of 300 seconds, then the start as recipe server is aborted.

Monitoring the recipe server

Fundamentals

The monitoring of the recipe server can be set in the `'<proj-path\sys\recipe.ini'` file. If the monitoring process is parameterized, at the parameterized time intervals, telegrams are triggered in all the PCUs participating in the recipe control process. These telegrams are of type 16 and are received by both the active recipe server and the standby recipe server. If a recipe server failure is detected, the message 'Recipe Server Failure' is output in all IOSs of the plant.

Settings in the recipe.ini file

[ServerSupervising]

- Enable=1
When Enable =1 is set, the monitoring of the recipe server is enabled.
When Enable = 0 is set, the recipe server is not monitored. In this case, the message 'Recipe Server Failure' is therefore not generated. The automatic recipe server switching can only work when the monitoring process has been enabled.
- CycleTime=30
If CycleTime = 30, a message is triggered in all the PCUs participating in the recipe control process every 30 seconds.
- ReactionCycle=24
For ReactionCyle = 24, a message "Recipe server failed" is triggered after 24 messages have failed. This parameterizes a reaction time of CycleTime x ReactionCycle = 24 x of 30 Sec.
- RepeatTime=300
After RepeatTime=300, the message "Recipe server failed" is repeated.
- WaitTime=240
After booting the PCU server, only after the time WaitTime=240 seconds is the monitoring of the PCU started.
- AutomaticServerActivation=0
In the event of a server failure or if the system is powered up without a clear server configuration in the assigned PCUs, user intervention is required.
- AutomaticServerActivation=1
In the event of a server failure, the system switches over to the spare recipe server automatically. For this switchover to work, the monitoring of the recipe server [ServerSupervising] Enable=1 must be switched on.

  Note

  Power-up when there is no clear configuration in the PCUs

  If no clear recipe server configuration can be detected in the assigned PCUs during power-up (e.g. when AS is reloaded on a PCU), the 1st server in the AREA automatically powers up as the recipe server.

- Automatic server switching is enabled by default (in the delivery state) (AutomaticServerActivation=1).

Cycle time batch start processing

Criteria for the batch start

The CAS status of all start units of the released batches is checked cyclically. In addition to the cyclic search, a search is triggered with a status change to 'Released', with the release of a CAS dataset and with the creation of a new batch.

The cycle can be set in the `'<proj-path\sys\bali.ini'` file in...

- `[balidata]`
`SearchTime=80`

. The value corresponds to the number of the PCU server timer calls (=200ms) in this case.

- This means: 80 corresponds to 80* 200 ms = 16 seconds.

A search is performed for batches which are to be started as soon as the CAS of the start unit of at least one released batch is free. The batch list is searched from the top to the bottom.

The first batch with start mode 'as soon as possible' is started.

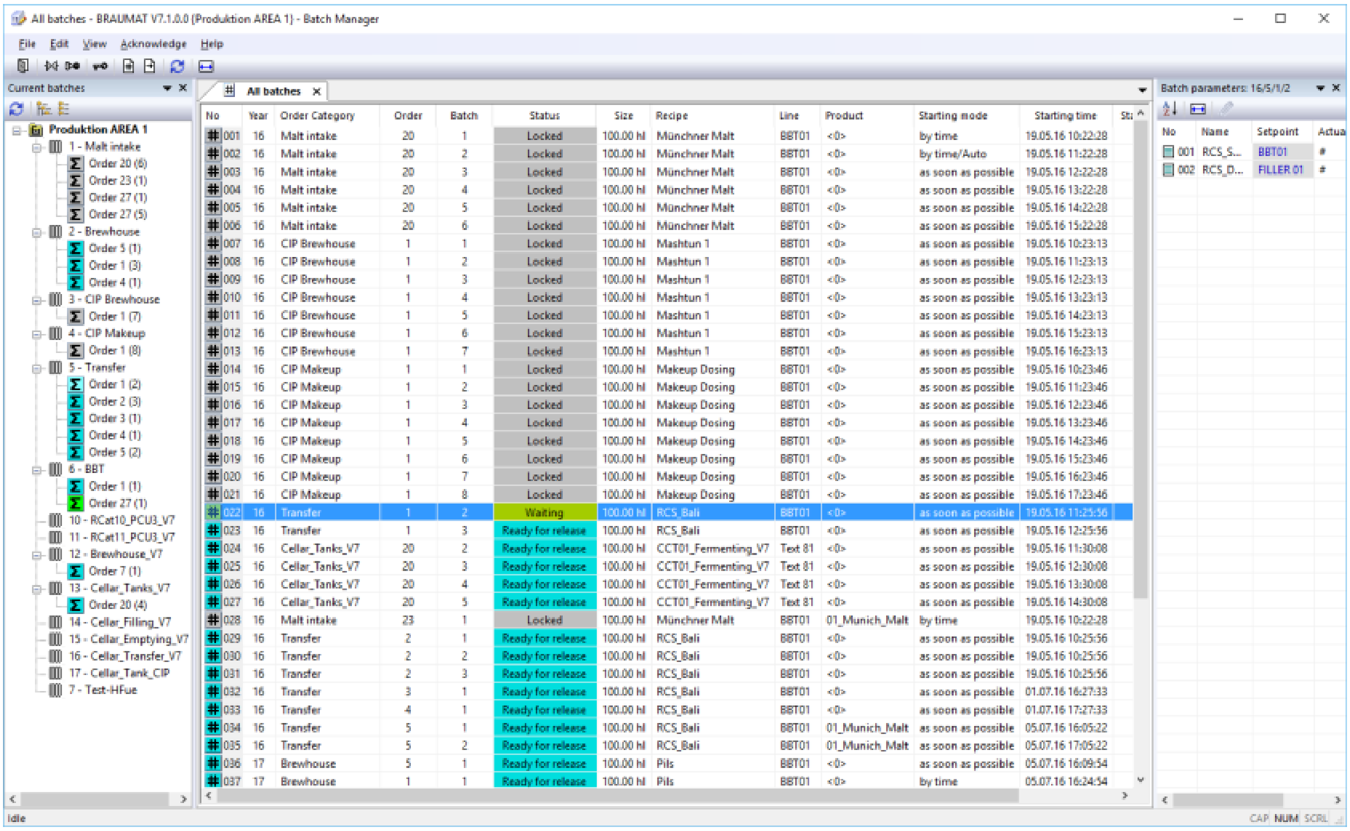
From the batches with 'by time' or 'by time/auto' mode, the one with the shortest start time starts, regardless of its position in the batch list.

10.5.3 Working with the application

10.5.3.1 Structure of the user interface

Basics

The figure below shows the structure of the batch management user interface:



① Menu bar:

You can find all the commands needed to operate the batch management in the menu bar.

② Toolbar:

The toolbar provides you with buttons for frequently used commands.

③ Navigation area:

In the navigation area, you can access the following views of the connected AREA via two tabs:

1. Order categories and current orders
2. Order archive sorted according to production year

This view can be fixed ("docked") or moved freely ("undocked").

④ Working area:

The batches filtered in the navigation area are displayed in the working area. The working area has a table structure. Each batch occupies one row. Multiple worksheets can be opened. Multiple selection of batches for a subsequent menu or shortcut menu operation is possible.

⑤ Batch parameters:

The batch parameters of the selected batch are displayed and input here. This view can be fixed ("docked") or moved freely ("undocked").

⑥ Status bar:

General project information is displayed in this area.

Positioning of navigation area and batch parameters

By dragging the window with the mouse to one of the four direction arrows, it can be permanently positioned at the margin (top, bottom, right, left = docked). Alternatively, the window can be freely positioned in the application (= undocked). The last configuration is saved when you exit batch management.

Description of the panes

- Navigation area / tree view
  In the first tab, the tree view shows the order categories as well as the respective current orders that are assigned.
  The second tab contains the order history, with the levels production year, order category, and order. This is always sorted in descending chronological order from top to bottom.
  You specify the batch list filters in the current worksheet by clicking on a node. An order can be created or batches can be added with the shortcut menu.
- Working area / batch list
  The window can contain multiple worksheets. A worksheet shows the filtered batch list or the unit allocation to the batches as per the selection in the navigation area.
  - Selection root node "Production <AREA x>" → all batches of this AREA are displayed
  - Selection order category → all batches/all orders of this order category are displayed
  - Selection order → all batches of this order/these orders are displayed
- Batch parameters
  The window shows the list of batch parameters for the selected batch in the working area.
  - The header includes the following information in short form <year>/<order category no>/<order no>/batch no>.
  - Depending on the parameter type, a new value can be entered in the "Value" field, or it can be selected in a type-specific selection list or in a filter dialog.
    The possible parameter types are described in the section "Recipe system" / 'Edit DFM definitions' dialog (Page 581).

Menu bar

- "File" menu
  The following table shows the commands in the "File" menu:

Command	Explanation
New batch list	A new "Batch list" worksheet is opened in the working area. The filtered batch list can be displayed by selecting the order category or the order. The following information columns are displayed: No., Year, Order category, Order, Batch, Status, Size, Recipe, Line, Product, Start mode, Start time, Start PCU, Start TA.
New unit allocation list	A new "Unit allocation" worksheet is opened in the working area. The filtered batch unit allocation can be displayed by selecting the order category or the order. This view shows a batch in each line with the units that have already run, that are running and that are coming up (according to the recipe). The status of the unit with regard to the batch is shown in color. This means batch tracking shows the batch progress from the process images of the unit.
Close <Ctrl+F4>	The active worksheet is closed.
Close all <Shift+F4>	All worksheets in the working area are closed.
Process parameter editor <Ctrl+F7>	Opens the master recipe view of the recipe editor for the selected batch.
Unit overview <Ctrl+F8>	Opens the plant overview with the units referenced in the recipe for the selected batch.
Control recipe <F9>	Opens the control recipe view for the selected batch
Exit	Exits the batch management

- "Edit" menu (→ only visible when worksheets are open / see "File / New" menu)
  The following table shows the commands in the "Edit" menu:

Command	Explanation
New order <Ctrl+N>	Creating a new order (see 'New order' dialog (Page 794))
Lock	Set selected batch(es) to "Locked" status
Provide	Set selected batch(es) to "Ready for release" status
Release	Set selected batch(es) to "Released" status
Abort	Set selected batch(es) to "Aborted" status
Delete <Del>	Set selected batch(es) to "Deleted" status
Start mode / ... • As soon as possible • By time • By time / auto • By event	Change start mode for selected batch(es) (release depending on basic order category see Dialog 'Define order types' (Page 798) / Start mode button)
Change start time	Change start time for selected batch(es) (release only when selected batches have start mode "by time" or "by time/auto")
Change batch size	Change batch size for selected batch(es) (release depending on basic order category / Dialog define batch generation (Page 800))

Command	Explanation
Change product	Change product for selected batch(es) (release depending on basic order category / Dialog define batch generation (Page 800)) • The "Material selection" list box opens.
Add batches	Additional batches can be created for this order through a dialog.
Process input list <F8>	See 'Batch process input list' dialog box (Page 808) The menu item can only be selected if the ‘Process input list required’ setting has been activated in the applicable order category.

- "View" menu
  The following table shows the commands in the "View" menu:

Command	Explanation
Toolbars and windows / ... • Toolbar • Current orders • Archive • Batch parameters	Displays and hides the toolbars and docking windows • Show/hide toolbar • Show/hide “Current orders” tree view in the navigation area • Show/hide “Order archive” tree view in the navigation area • Show/hide “Batch parameters” window
Status bar	Displays and hides the status bar at the very bottom of the window
Optimize column width	Optimizes the column width of the working area to the available window width.
Refresh <F5>	Refreshes all panes in the application

- "Configuration" menu (→ only visible when no worksheet is open any longer / see "File / Close" menu)
  The following table shows the commands in the "Configuration" menu:

Command	Explanation
Order categories	Opens the Dialog 'Define order types' (Page 798).
Order/batch numbers	Opens the Allocating/releasing order/batch numbers (Page 807) dialog

- “Acknowledgment” menu
  The following table shows the commands in the "Acknowledgment" menu:

Command	Explanation
Acknowledge ICM errors	Acknowledgement of pending ICM errors
Acknowledge horn	Acknowledgement of pending horn signals
User logoff	Reset of logged-in user and his/her authorizations
Disable MES	Temporarily remove the order input lock initiated by the MES (see “Specifications/Higher-level system” section in Dialog define batch generation (Page 800))

- "Help" menu
  The following table shows the commands in the "Help" menu:

Command	Explanation
System <Alt+F5>	Opens the online help for the overall system
Batch management <F1>	Opens the online help for batch management
About ...	You can use this menu to access information about the installed version of the batch management.

Toolbar

The following table shows the commands that are displayed in the toolbar. This is only visible if the standard view is activated in the View/Toolbars and windows menu.

Symbol	Explanation
Exit	Exits the batch management
Acknowledge ICM errors	Acknowledgement of pending ICM errors
Acknowledge horn	Acknowledgement of pending horn signals
User logoff	Reset of logged-in user and his/her authorizations
New batch list	Creates a new “Batch list” document in the working area
New unit allocation	Creates a new “Unit allocation of the batches” document in the working area
Refresh	Refreshes all panes in the application
Optimize column width	Optimizes the column width of the working area to the available window width.

Extra functions

Shortcut menu

Right-clicking on a previously selected element in the navigation or working area opens a shortcut menu with all possible editing and view commands (for descriptions see “Edit” and “View” menus). Multiple selection is also possible in the working area. This is helpful for batch operations.

Move batch

By dragging (left mouse button) a selected batch to a new table position, a dialog opens which presents the source and target position. By confirming the query, the selected batch is moved and the processing order will be changed.

10.5.3.2 'New order' dialog

Basics

A new order as well as one (multiple) associated batch(es) can be created with this dialog.

Note

Default settings for order data when dialog is opened

The default settings for recipe, line, product as well as batch parameters come from the previously selected batch in the working area or the selected order category in the navigation area.

If a batch does not exist in the working area yet, you must first select an order category.

Recipe data

- Recipe
  The default setting can be changed through a recipe selection dialog ("..." button).
- Line
  With this it can be determined on which line the order is supposed to run. It can be configured whether a line input is possible. By opening the dialog, the line is preset with the reference line of the current recipe. With the line selection ("..." button), the start unit is determined implicitly via recipe header and assignment table.
- Product
  The default setting can be changed through the material selection dialog ("..." button).

Template for parameter values

- Order category
  The values of the batch parameters are derived from the default values of the order category.
- Batch
  The values of the batch parameters are derived from the previously selected batch. If a batch does not exist in the working area yet, this option is locked.

Batch ID

- Batch year
  The default value (current calender year) can be changed if this is enabled.
- Order number and first batch number
  The order and batch numbers of the 1st batch of the new order is entered here. The numbers are suggested by the system. The number can be changed if this is enabled. If the order consists of more than one batch, the order numbers will be allocated by the system, using the defined algorithm (see the section Order parameters (Page 780)).

Start

The following start modes (like CAS modes) can be selected:

- 'as soon as possible'
- 'by time'
- 'by time/auto'
- ‘by event’

For each order type the selection of the modes can be limited. The indication of 'date' and 'time' refers to the mode 'after time' or ‘after time/auto’. The cycle determines the time difference between the individual batch starts.

With the selection of the recipe, the cycle and the start time is preset with the current values for the selected recipe. However, these values can be overwritten.

Batch generation

Here the number of batches for an order and the order size is entered. The line beneath shows the calculated breakdown for checking purposes. The batches are built according to the method configured for this order type (see the section Batch generation (Page 784)).

‘Create order’ button

Accept new order data and dialog end. The confirmation for creating the batch(es) will be requested afterwards with the "Create batches" dialog.

'Cancel' key

Do not accept new order data and dialog end.

'Invalid Recipe Header' message box

The dialog starts with a default that points to a recipe for which no valid recipe header is configured. A selection of different order types and recipes with valid data is possible.

Message box: 'Invalid Input!'

If 'Invalid Input!' message box is output when creating a new order, one of the following values is implausible for the requested master recipe, in this case <= 0:

- nominal batch size
- minimum batch size
- maximum batch size

In this case no new orders can be entered. The values have to be > 0 (see master recipe header).

10.5.3.3 'Tank selection' dialog

"Tank" or "Tank location" parameter type (PCU as of V7)

Here, an online view shows the available tanks and their attributes that are currently important for the selection. By selecting a line in the tank list and confirming with "OK", the corresponding tank number (PCU-related) or the location (AREA-related) is entered in the order parameters and the dialog is closed.

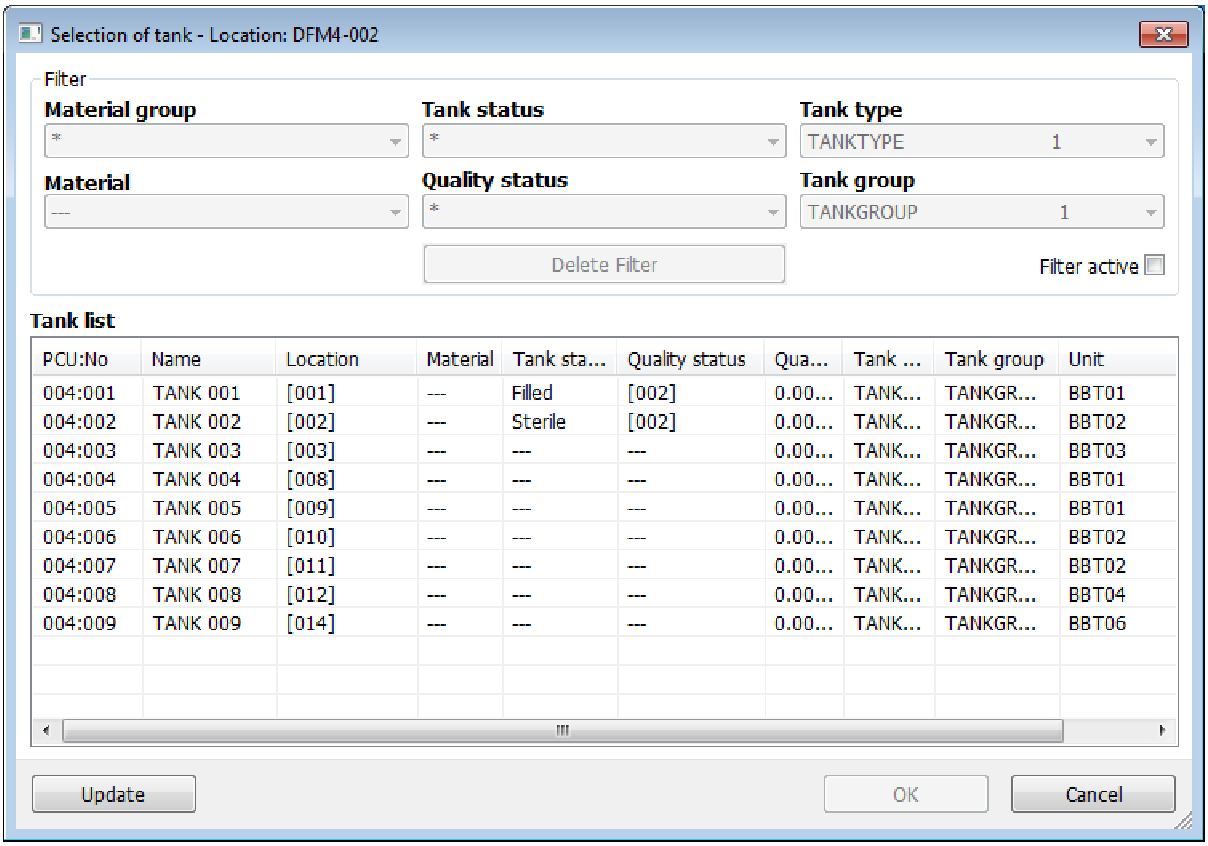
Note

The tank list is always only filled during runtime by means of so-called "tank inventory telegrams". The telegram type 22 used here must also be entered in the process cell configuration / PCU settings / FIFOs / FIFO 1+4!

The display of the various attributes in the tank list assumes that these were correctly and completely configured during parameter assignment of the TANK block instances. Parameter assignment with regard to trigger event and acquisition cycle also takes place there (for more details see section TANK - functions (Page 371)).

- With the type "Tank", the tank list shows only the tanks of the relevant PCU (according to the DFM assignment).
  - The first column shows the tank number
  - The "Location" column is missing
- With the type "Tank location", all the tanks of this AREA are displayed.
  - The first column shows the combination "PCU number:Tank number"
  - The "Location" column is also shown
  - To use this function, a location must be assigned to every TANK when assigning parameters for the TANK block instances.
  - The list of locations (file "`<proj-path>\texte.x\rcs_loc.txt`") is automatically generated from the RCS offline engineering when the "Route Control System" option is used. If RCS is not used, this file must be edited using the text parameter assignment.

Tank selection dialog and filter options



"Filter" area

By selecting an element in the list box, the corresponding filter is activated. The list boxes contain the elements used in the visible/filtered tank list. All tanks are displayed only when the filtered criteria are not active or have been deleted.

- Material group list box → only tanks with materials of the selected material group are displayed
- Material list box → only tanks with the selected material/product are displayed
- Tank status list box → only tanks with the selected tank status are displayed
- Tank type list box → only tanks of the selected tank type are displayed
- Tank group list box → only tanks of the selected tank group are displayed
- Quality status list box → only tanks with the selected quality status are displayed
- "Delete Filter" button → The selected filter criteria are reset to "*" = all

---

Note

When the tank selection dialog is opened, filtering is activated and the material of the current batch is preselected in the "Material" list box.

"Tank list" area

With the "Filter active" check box, the filtering of the tank list can be deactivated/activated. When it is activated, the table contains only the tanks that correspond to the previously described filter criteria.

The contents of the boxes are derived from the current online attribute values of the PCU block instances at the time the dialog was opened. The screen can be updated at any time using the "Update" button.

10.5.3.4 Configuration of order types

Order types

Core statement

Order types must be defined before entering orders, whereby each order type is assigned to one recipe category. There is a 1:1 relation between order type and recipe category. The order type number is identical to the number of the recipe category.

The new order/recipe system isn't preconfigured, i.e, no order types are preset.

Configuration of the order types only makes sense if recipe categories, master recipes, recipe procedures, process parameters, process input lists, etc. were created.

Selection of configuration dialog

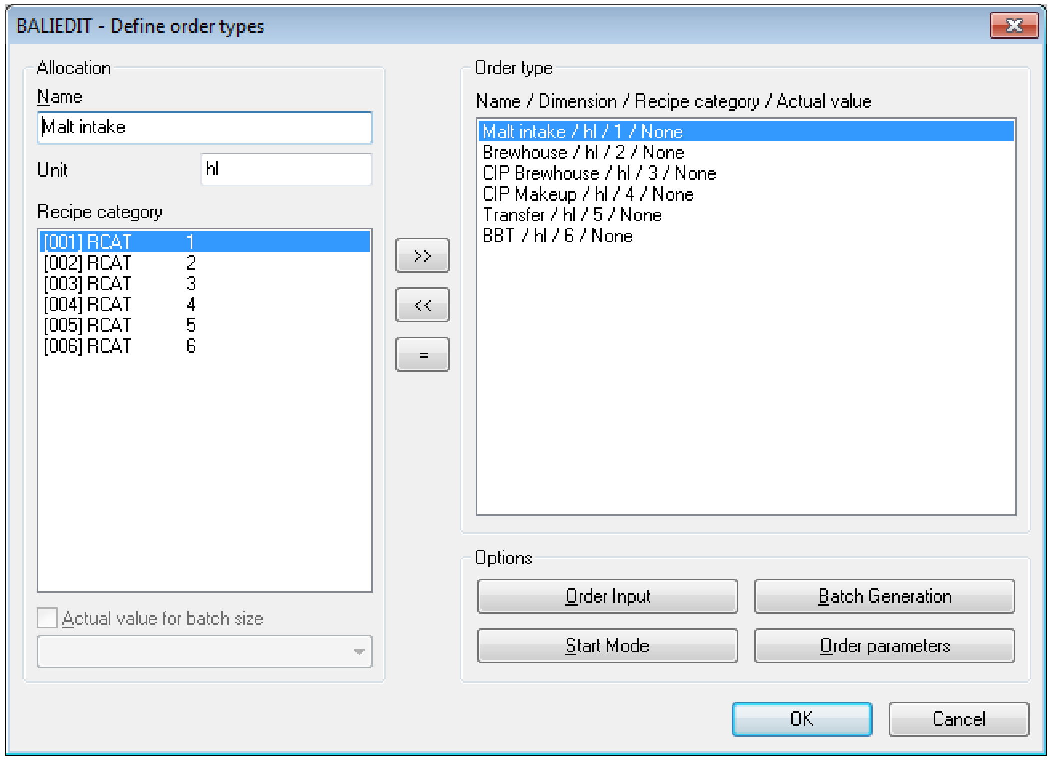
Call the menu item "Configuration / Order categories". The menu item 'Configuration' can only be selected when the batch list is closed.

Dialog 'Define order types'

Create order types:

Procedure for creating a new order category:

- Select the required recipe category under 'Recipe category'.
- After the selection, enter the required name of the order type and the name of the unit. The name is used during order input.
- The recipe category is applied to the list with the '>>' button and displayed in the field 'Name \Dimension\Recipe category\Actual value'.



"Order type" area

The order types which have already been configured are displayed in the list. The following information: Name, dimension, recipe category number and actual value of the batch size with diagonal stroke is displayed separately. For the selection of an order type, the fields on the left are set with the configured data under "Allocation".

If an order type is supposed to be edited, it must first be selected in the list. Then the new name, unit and recipe category can be specified in the 'Allocation' area.

With the key '=', the changes are accepted in the selected order type.

Actual value batch size

Here one parameter from the free protocols can be selected, which can be indicated as the actual value for the batch size in the batch list. For the function, a configuration of free protocols is necessary.

"Order Input" option

With this button, you reach the 'Order Input' dialog.

"Batch Generation" option

With this button, you reach the 'Batch Generation' dialog.

"Start Mode" option

With this button, you reach the 'Start Mode' dialog.

"Order parameters" option

With this button, you reach the dialog 'define order parameter'.

Button '>>'

A new order type is included in the list. In this way, the name is loaded in the field 'name' and the selected recipe category is accepted and checked.

Button '<<'

In this way, the order type which has just been selected is deleted.

Button '='

In this way, the definition of the selected order type with the current inputs is overwritten.

Button 'OK'

File definition and dialog end.

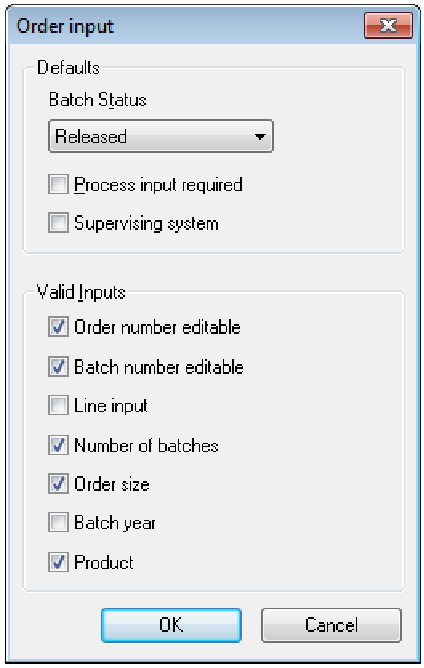
Button 'Abort'

Do not save file definition and dialog end.

Dialog define batch generation

To call the dialog box

- Select "Edit"
- Select "Order types"
- Click "Starting mode"



Preset 'Batch state'

This selection determines with which status the batches are created after creation. It is possible to select between 'disabled', 'enabled' or 'ready'.

See also: Batch status (Page 714)

Preset 'necessary to process input list'

Here is determined whether the batches of the order type use a process input list. The deselection of the function has advantages in terms of performance.

Preset 'Higher-level system'

Here it is determined whether a higher-level system (MES) is responsible for this order type. If the higher-level system is active, the batch input for this order type is locked in the "Batch Manager" application. This lock can be removed temporarily, i.e. until the next activity of the MES system, via the menu point "Options/Disable MES". The operator permission "BALI_PROJ" is required for this.

Allowed inputs

This selection determines which input possibilities are enabled by creating batches.

- 'Order number editable'
  The order number can be changed. A test to ascertain explicitness is executed.
- 'Order number editable'
  The first batch number of an order can be changed. A test to ascertain explicitness is executed.

- 'Path input'
  Path can be selected. If this option is deselected, the reference line of the assigned recipe is always decisive.
- 'Batch number'
  Input possibility for the number of batches per order.
- 'Order size'
  Input possibility for the size of each order.
  If neither 'batch number' nor 'order size' is selected, 1 is always accepted as a number of batches per order.
- 'Batch year'
  The year of the batch can be changed manually. The default setting is the current calendar year.
- 'Product'
  When a new order is created, a product/another product can be selected from the material list. In principle, only the product defined in the basic recipe is used.

Button 'OK'

The changes are accepted temporarily. A final acceptance only occurs when the OK button of the dialog 'define order types' is pressed.

'Abort' button

The changes are not accepted.

Dialog 'Defining batch building'

To call the dialog box

- Click "Configuration" -> "Order categories"
- Click "Batch building"

Batch size

Here a selection can be made between the two types of batch building.

- All batches the same
- n batches with Max + Rest

Reference batch size: Recipe procedure

The nominal batch size of the recipe procedure is used as default batch size. The limits to the recipe procedure are used.

Reference batch size: Master recipe

The batch sizes for the batch input are calculated as follows:

Default value:	= nom. MR
minimum value:	= min. RP / nom. RP * nom. MR
maximum value:	= min. RP / nom. RP * nom. MR

(RP = recipe procedure; MR = master recipe)

Batch numbers

Here, a selection can be made between the types of batch number allocation.

- Start with 1 per order
- Continuously for year and order type
- Order = calender week; batch seq. per week
- Order = calender week; batch seq. per year

Button 'OK'

The changes are accepted temporarily. A final acceptance only occurs when the OK button of the dialog 'define order types' is pressed.

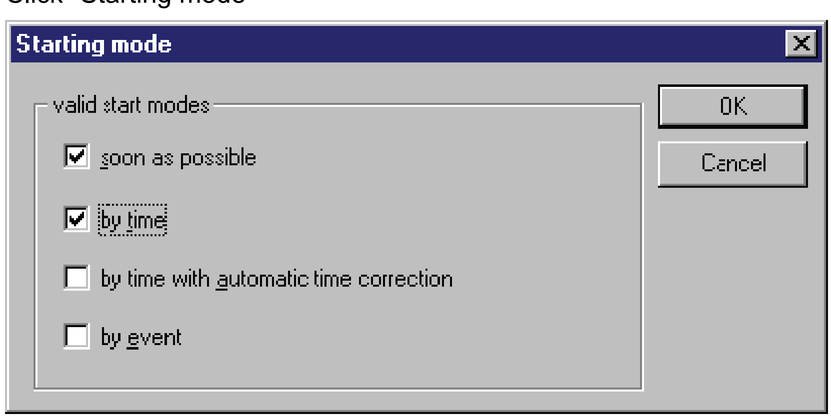
'Abort' button

The changes are not accepted.

Dialog 'Define start mode’

To call the dialog box

- Click "Configuration" -> "Order types"
- Click "Starting mode"



possible start modes

Here the possible start modes are defined, which are possible with the batch input

Button 'OK'

The changes are accepted temporarily. A final acceptance only occurs when the OK button in the dialog 'define order types' is pressed.

Button 'Abort'

The changes are not accepted.

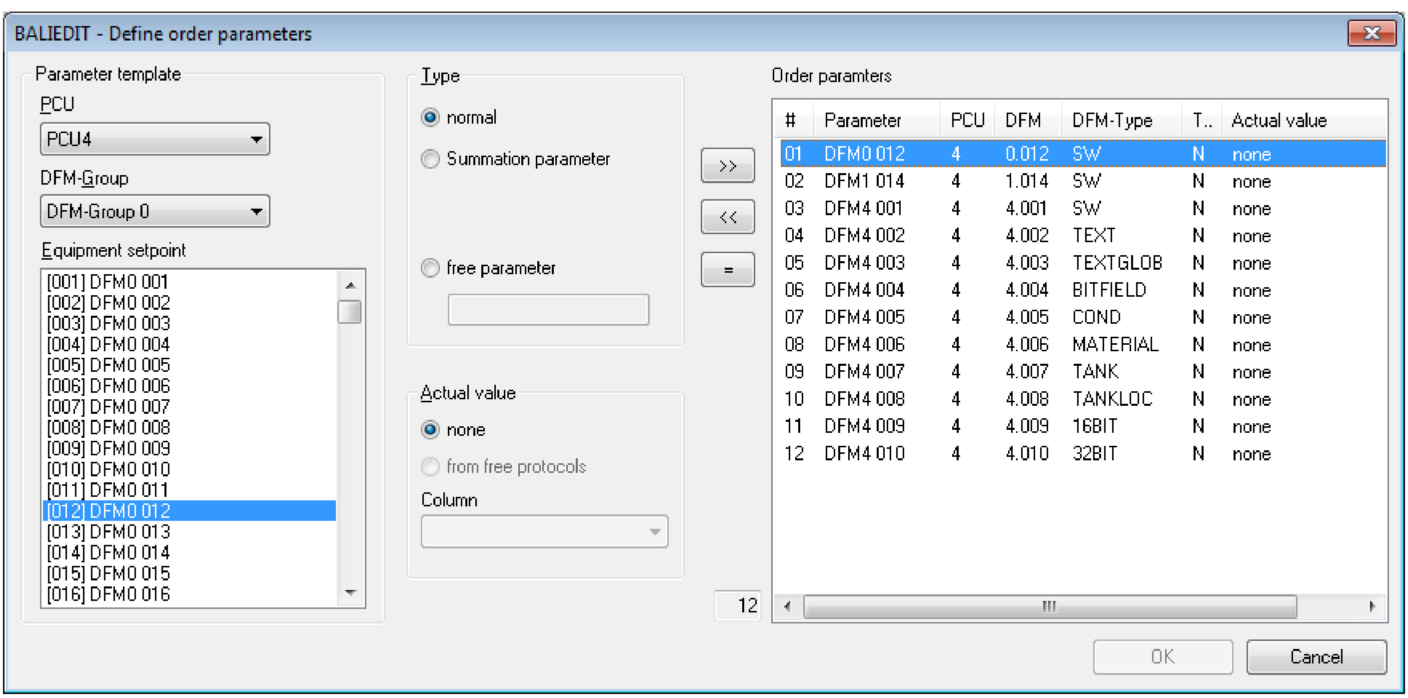
Dialog 'Define order parameter'

To call the dialog box

- Click "Configuration" -> "Order types"
- Click "BALIEDIT"

If the batch list is not completely empty, initially a message appears indicating that the order parameters cannot be changed.

Dialog



Parameter template

- PCU, DFM group and DFM
  The DFM instance that is used as the parameter template for the order parameters is selected

Type

- normal
  With the selection 'normal', the parameter is used for input and output as was determined in the setpoint definition.
- Summation parameter
  With the selection **'sum parameter'**, the input/output is the same as the selection of 'normal', with the difference that the parameter is distributed to the individual batches.
  With the selection of 'normal', 'route control' and 'sum parameter' the EPARs are automatically selected.
- Free parameter
  With the selection **'free parameter'**, an order parameter is defined that is not a DFM. For this parameter, only one name can be allocated.

Actual value

- One actual value can be assigned to each parameter. Actual values are columns from the dates of the free protocols.
- With the combo box, the column is selected in the dBase file.
- For the function, a configuration of free protocols is necessary.

'Order parameters' list

In this list box, all order parameters and their definitions are shown.

'>>' button

With this button, an order parameter is added to the list.

'<<' button

With this button , an order parameter is deleted from the list.

'=' button

This button applies the definition of the selected DFM in the selected order parameters.

'OK' button

The changes are accepted temporarily. A final acceptance only occurs when the OK button of the dialog 'define order types' is pressed.

'Cancel' button

The changes are not accepted.

Reconfiguration of order parameters

Requirements

The reconfiguration can occur only under specific conditions.

- Batches may not be available in the batch list. This test is executed by the application.
- All current batch archives must be paged out and deleted.

Note!

If there is a reconfiguration without saving the 'old' archives, no further batches can be created.

Note!

Current order parameters are assigned to their corresponding formal parameters (the placeholders in the recipe procedure in which these parameters are inserted by substitution), based on the line number of the order parameter in the order parameter definition list of this dialog box. When the arrangement of parameters in the list of this dialog is changed, the assignments in the recipe procedures are not automatically updated when order parameters are substituted in the recipe procedures. You must definitely take this point into account when you modify the parameter list. You can always append new parameters to the list, without having to adapt existing recipe procedures. All other modifications, for example deleting parameters or changing their position in the list, may cause an undefined behavior of the recipe procedures!

Preset order parameters

Predefining order parameters

The order parameters in the "New Order" dialog can be configured in file '...<proj-path>\bali \oparadef.ini'.

In this file, there is an area for every order type

```
[OrderType#] # stands for order type
```

Under this area, the values are listed for every order parameter.

```
Parameter#= # stands for the number of the order parameter (starting with 1)
```

Fixed value

A fixed value is written directly behind the sign '='. Numbers with decimal places must be displayed with any points.

Examples:

```
Paramater1=5
Parameter2=6.67
```

Value from process parameter of the recipe of the batch

The order parameter can be preset with a value from the process parameters of the recipe. In this case, the number of the parameter (Index of the list beginning at 1) must be indicated with a leading '%' sign.

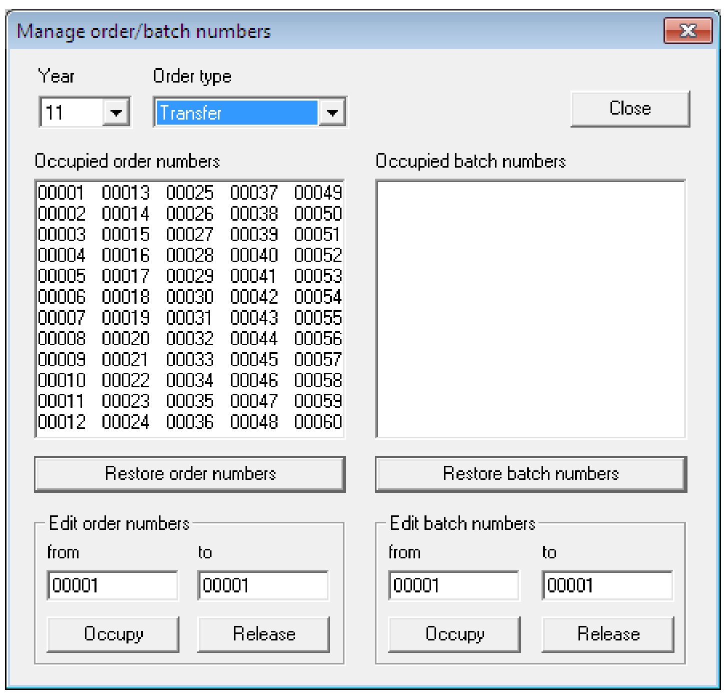
Example:

```
Parameter1=%5
```

10.5.3.5 Allocating/releasing order/batch numbers

Calling the dialog:

- Click Engineering/Order numbers/Batch numbers



This dialog allows you to reconstruct order/batch numbers from the batch history in the system allocation list or to allocate/release number ranges.

- First, you have to select the "Year" and order type
  The "Occupied order numbers/batch numbers" list fields show the current system allocation lists for the selection you have made.
- "Restore order numbers"/"Restore batch numbers" buttons
  The batch history in the ‚…...<proj-path>\Bali\hist_yy.dbf’ files is read and the order/batch numbers for the year and order category are entered in the system lists under ‚…...<proj-path>\Bali\Used_Nr’ .
- "Edit order numbers/Edit batch numbers" fields
  Here, you can pre-assign or release the relevant number ranges.
  – Enter "from"/"to" and press "Occupy" button → The number range is pre-assigned or reserved
  – Enter "from"/"to" and press "Release" button → The number range is released or deleted

⚠ CAUTION

Data loss as a result of incorrect use

Once the order/batch numbers have been released, they are reused for any new orders or batches that are created. As a result, any batch reports that are left over get overwritten.

In addition to causing data loss, the incorrect use of this function can also result in incorrect assignments between steps and batches in reports/MIS/reporting.

Access to this dialog is protected by means of a dedicated program level: ‚BALI_PROJ’.

10.5.3.6 'Batch process input list' dialog box

Basics

The dialog is used to process materials if a process input list has been defined for the job type.

Selecting the dialog box

This dialog is only available if you have set "Process input required" in your configuration of the order type of the batch.

You can modify the values in the following columns of the process input list, provided the current batch status is "locked":

- Volume
  The material metering volume in this line
- Metering group
  The material metering group in this line (metering sequence).

Notes:

- In contrast to this, the ability to make changes can always be forced, i.e., regardless of the batch state, with the following switch:
- In file: `..<proj-path>\sys\BM_BATMAN.INI`
  `[Settings]`
  `...`
  `CompListEditAlways=1`
  This corresponds to the old behavior (to V4.6)
- Metering processes with the selected metering group must be defined in the recipe procedure in order to ensure weighing of the material.
- After you have enabled stock management and tank selection, you can select a tank from the "Siloname" column by means of a text list selection dialog, and thus override the "automatic" or preset selection.
- If the material volume in this tank volume is too low, a corresponding message pops up, and the text of the corresponding process input line is marked in red color. You can only select a tank that contains the corresponding material.
- You can always return to "automatic" mode.

10.5.3.7 Password protection

All operations in the application are protected by password. You divide between various password levels.

- Bali_Chgpa = order system - change batch parameters, status, properties
- Bali_Newor = order system - enter order
- Bali_Proj = order system- change configuration
- Bali_Relba = order system - change status
- BALI_INSBA = order system - enter, attach, move batch
- Batch_PIL_Chg = order system - change process input list

10.6 System overview 'SeqCtrl'

10.6.1 Functionality

Basic principles

In the current version every unit can have 1 sequence. Therefore, the terms unit and sequence are used synonymously in the description.

The application 'Sequence control' is used for the representation of the current state of the process cells. In addition, it is possible to select and to operate the individual units. With this, the user can intervene in the current process and influence the different processes. In the context of IAS-S88, units are defined units of the entire system that can work partly independently.

Program start

The application is opened in the basic menu / Production tab → System overview.

10.6.2 Configuration

10.6.2.1 Configuration

You can call the ‘Process cell’ command from the menu to select a PCELL or an overview of the sequencers.

Note

- The area described here is not the same as the term "Area" in the configuration tool 'SiteCfg'.
- and can be interpreted and configured as a process cell. In its simplest context, the process cell covers a complete area.
- The names and the units shown in the overview can be customized by users.

Determine views

Basic principle

The names of the system areas stored in the file `<proj-path>\TEXTE.x\BEREICH.TXT`. The ‘Functions > Edit process cell’ command in the menu starts the editor notepad.exe for editing the file.

Note:

It is possible to create a subdirectory on the server for every client (IOS.XXX), in which client-specific settings can be stored.

Example:

```
<proj-path>\IOS.021\etc\kpos001.ini
<proj-path>\IOS.021\texte.0\bereich.txt
```

As soon as a client directory exists on the server under `<proj-path>`, the settings are read from this directory, otherwise the old behavior applies. Through this, the settings can be continuously managed/stored centrally on the server. In addition, a Multi-Client automatically receives its own settings for every area!

The new process cell names are only valid for the next start of the application.

There are two different methods of building up the files.

Flat structure

Example:

- Raw material acceptance
- Mixing plant
- Filling
- Cleaning

The line position corresponds to the number of the "kpos" file (raw material acceptance -> kpos001.ini ... CIP -> kpos004.ini for the example above).

Hierarchical structure

In the file Bereich.txt a hierarchy can be indicated. By entering keywords 'Begin' and 'End' a menu structure can be configured. Only leading blanks or tabs may be inserted.

The text following the double slash is only an informative comment and may not be part of the code.

Determine units per area

- Up to 64 units can be represented (from different PCUs).
- In the file `<proj-path>\etc\kposxxx.ini` it is defined which unit is displayed in which line.
- xxx corresponds to the line number in file `<proj-path>\TEXTE.x\BEREICH.TXT`.

Structure of the File 'Bereich.txt':

Example:

```
Brewhouse // Title for main menu
```

```
Start // Key word "Start submenu"
Brewing line1 @1 // Text input for 1. Input in submenu
Brewing line2 @2 // Text input for 2. Input in submenu
End // Key word "End submenu"
Fermenting cellar // Title for the second main menu
Start // Key word "Start submenu"
Vertical Tanks1 @10 // Text input for 1. Input in submenu
Vertical Tanks2 @11 // Text input for 2. Input in submenu
End // Key word "End submenu"
```

Explanations:

- The @ sign with the following number points to the 'Kposxxx.ini' file there.
- Example: @12 is a reference to 'kpos012.ini'.
- If no @ sign is inserted, the units ('Kposxxx.ini' file) are assigned to the areas after the line number in the 'Bereich.txt' file.
- xxx corresponds to the process cell number.
- Example:
  ```
  Unit // Line1
  Begin // Line2
  Unit1 // Kpos003.ini Line3
  Unit2 @2 // Kpos002.ini Line4
  End // Line5
  ```

Structure of the files 'kposxxx.ini':

<Blank>

<PCU Number><blank><unit number>

<PCU Number><blank><unit number>

etc.

Example:

```
<...blank...>
1 1
1 2
1 10
2 1
```

Explanations:

- A blank line or the entry of PCU no = 0 and unit no = 0 in the file results in an empty line on the screen.
- The same unit number can be registered again.

- In the application 'plant overview', it is possible to access the notepad editor for adapting the corresponding KPOS file via the menu item 'function' -> 'edit process cell view'.
- The new adjustments are only valid after an area change.

Configure message windows

A message window can be shown in the lower section of the system overview. In the window, messages relating to the selected sequence are displayed.

File `<proj-path>\sys\seqctrl.ini`:

```
[Mainwin]
```

```
ShowMessage=1 // 1: Das Meldefenster wird angezeigt.
(Defaulteinstellung) 0: Das Meldefenster wird nicht angezeigt.
```

The column widths of the message window must be moved once to the correct width. The adjustment is filed and used by the next start.

Setpoints for units

Up to 24 setpoints (DFMs) can be displayed for each unit in the "Sequence related setpoints" pane of the process cell overview. This can be configured during equipment engineering in the recipe editor. Select the unit, followed by the "Properties" shortcut menu.

Recipe category for the start

Basic principles

For the start of recipes in the system overview, the recipe category must be configured for each unit. This can be configured during equipment engineering in the recipe editor. Select the unit, followed by the "Properties" shortcut menu (see Dialog 'Configure sequences' (Page 575)).

More than one recipe category can be displayed. The entries must be separated by comma.

Reduce selection of recipes

For the given recipe category, all recipes are displayed in the recipe selection dialog. This selection can be reduced.

For the reduction, the allowed recipes must be engineered behind the recipe category in brackets. The following statements are allowed:

- to set single recipe numbers
- to set ranges of recipe numbers

Enabling step operation

The general step operation must be enabled. A value must be entered in the file 'seqctrl.ini'. The value is evaluated in binary form. The value 255 enables all operations.

EOP Start	1
EOP Pause	2
EOP Hold	4
EOP Stop	8
EOP Abort	16
Leave intermediate state	32

The value is the equivalent of the sum of the individual states to be enabled.

Example: To enable EOP Start and EOP Stop:

Value = 1 + 8 = 9

The calculated value must be entered in the file `<proj-path>\sys\seqctrl.ini` as follows:

```
[MainWin]

EOPAdmin=Wert
```

The "Plant overview" application must be restarted when this value is changed.

Note

Requirements for activating the operating buttons:

- The step operating buttons are only enabled with PCUs as of "Recipe version V5", that is, not for PCUs with "Recipe version V3".
- When the enable signals for step operations are value-controlled, the corresponding buttons are only enabled when the unit is running and the current status permits step control. For example: Actuation of HOLD enables RESTART and interlocks HOLD.

10.6.2.2 Command line parameter

The application seqctrl.exe can be started by using a command line parameter. The parameter indicates the number of the kpos file. With the call a further application is started (see Multi-instance adjustment (Page 814)).

10.6.2.3 Multi-instance adjustment

Basic principles

With this adjustment in the file `<proj-path>\sys\seqctrl.ini`, the application can start only one time. A second start brings the already started instance into the foreground.

```
[App]

EnableMultiInst=
```

EnableMultiInst=0

A started application is set in the foreground. The plant combination is transferred if a parameter was sent in the start call.

EnableMultiInst=1

A further instance of the application is started. If a start parameter was sent the plant combination is opened. If no parameter was sent the combination which was opened when the application was last closed is opened.

10.6.2.4 Configure colors

The colors can be configured in the view. The indication is entered as RGB values (red, green, blue). Every color value can be between 0 and 255.

The table below shows examples of the RGB values for some of the primary colors.

	Bright	Dark
White	255,255,255	
Gray	200,200,200	
Red	255,0,0	125,0,0
Blue	0,0,255	0,0,125
Green	0,255.0	0,125,0
Yellow		255,255,0
Black	0,0,0.	

This can be adjusted in the file <proj-path>\sys\seqctrl.ini under [MainWin].

Colors in the window sequencer

TAListFGCol=0,0,0	// foreground color
TAListBGCol=255,255,255	// background color
CursorBGCol=128,128,128	// cursor background color
CursorFGCol=192,0,0	// cursor foreground color
TAFontName=MS Sans Serif	// font
TAFontSize=12	// character size
TAListGFGCol=0,0,64	// foreground color just lines
TAListGBGCol=0,64,0	// background color just lines

Colors of the status indicator display

The colors of the status indicator can be configured.

StatusAFGCol=255,255,255	// foreground color of the "A" symbol
StatusABGCol=0,0,0	// background color of the "A" symbol

StatusHFGCol=255,255,255	// foreground color of the "H" symbol
StatusHBGCol=0,0,0	// background color of the "H" symbol
Status-FGCol=255,255,255	// foreground color of the "-" symbol
Status-BGCol=0,0,0	// background color of the "-" symbol
Status+FGCol=255,255,255	// foreground color of the "+" symbol
Status+BGCol=0,0,0	// background color of the "+" symbol
StatusBFGCol=255,255,255	// foreground color of the "B" symbol
StatusBBGCol=0,0,0	// background color of the "B" symbol
StatusSFGCol=255,255,255	// foreground color of the "S" symbol
StatusSBGCol=0,0,0	// background color of the "S" symbol
Status#FGCol=255,255,255	// foreground color of the "#" symbol
Status#BGCol=0,0,0	// background color of the "#" symbol
Status0FGCol=255,255,255	// foreground color of the "0" symbol
Status0BGCol=0,0,0	// background color of the "0" symbol
Status1FGCol=255,255,255	// foreground color of the "1" symbol
Status1BGCol=0,0,0	// background color of the "1" symbol
StatusRFGCol=255,255,255	// foreground color of the "R" symbol
StatusRBGCol=0,0,0	// background color of the "R" symbol
StatusGFGCol=255,255,255	// foreground color of the Sync symbol
StatusGBGCol=0,0,0	// background color of the Sync symbol
Status_FGCol=255,255,255	// foreground color of the Alternative symbol
Status_BGCol=0,0,0	// background color of the Alternative symbol
StatusEFGCol=255,255,255	// foreground color of the Error symbol
StatusEBGCol=0,0,0	// background color of the Error symbol
StatusPFGCol=255,255,255	// foreground color of the Process symbol
StatusPBGCol=0,0,0	// background color of the process symbol
StatusCFGCol=255,255,255	// foreground color of the RCS symbol
StatusCBGCol=0,0,0	// background color of the RCS symbol

Colors of the setpoint/process values column

The colors of the setpoint/process value can be configured.

SollFGCol=100,80,200	// foreground color of the setpoint column
IstFGCol=200,80,100	// foreground color of the process value column
SPListFontName=MS Sans Serif	// font
SPListFontSize=12	// character size
SollFGCol=0,0,192	// reference value type color
SollGBGCol=0,192,0	// reference value background even lines
SollUBGCol=192,0,0	// reference value background odd lines
IstFGCol=0,0,64	// process value type color
IstGBGCol=0,64,0	// process value background even lines
IstUBGCol=64,0,0	// process value background odd lines
SWGBGCol=64,64,0	// setpoint window background even lines

SWGFGCol=0,64,64	// setpoint window document even lines
SWUBGCol=192,192,0	// setpoint window background odd lines
SWUFGCol=0,192,192	// setpoint window document odd lines
SWDefaultBGCol=200,200,200	// setpoint window background color preset

10.6.3 View

10.6.3.1 Division and printout

The view is divided into four regions:

- Viewing section for units
- Display section for unit-related setpoints
- Display section for step-related setpoints and corresponding current process values.
- Display section for unit-related messages (only reasonable in the type based on S7)

Printing out the process cell view

The current window of the process cell view can be printed out using the integrated print function (see Hardcopy (Page 64)).

Column 1:	**No.**
Number of units. Adding empty lines (blanks) is possible.	
Column 2:	**Sequence (Unit)**
Specifies the symbolic unit name. The unit name is allocated with the text parameterization.	
Column 3:	**'Seq.Status'**
Column 4:	**'Display'**
See also: Status indicators (Page 818)	
Column 5:	**Step**
Numeric signaling of the currently active step of the corresponding unit.	
Column 6:	**EOP**
Number of the currently active equipment operation (EOP)	
Column 7:	**EOP Name**
EOP name that describes the currently running equipment operation	
Column 8:	**EOP Status**
Current status of the EOP	
Column 9:	**Time**
Indication of step running time of the current step.	
Column 10:	**Recipe category**
Indication of current recipe category	
Column 11:	**Recipe**
Indication of current recipe name	
Column 12:	**Order number**
Indication of current order number.	

Column 13:	Batch
Indication of current batch number.	

10.6.3.2 Status indicators

Indications for usage are displayed as a current status of the individual units.

Table 10-7 In the 'Display' column, the following 7 icon positions can be occupied from left to right:

Symbol 1		Unit is in automatic mode
		Unit is in manual mode
Symbol 2		Unit switches to next step after actual step is finished
		Unit is stopped after the current step is finished.
Symbol 3		Permanent condition for this step missing
		Unit is shown at a synchronization / Synchronization not fulfilled
		Unit is shown at a synchronization / synchronization met
		Unit is shown at an alternative / alternative not met
		Unit is shown at an alternative / alternative met
Symbol 4		Security EOP running
Symbol 5		The user flag of the unit is inactive
		The user flag of the unit is active
Symbol 6		Recipe load error (e.g. the unit recipe is missing)
		EOP monitoring time expired
Symbol 7	flashing	Operation acknowledgment for unit required
	flashing	Error message, unit is defective.
	flashing	Warning entered for unit
	flashing	Message entered for unit

10.6.3.3 Multi-client function

When you start an application on a client that is configured for operation in a multiclient architecture, the view selection dialog box opens, showing the enabled views. After you have selected a view, the current instance of the control recipe visualization is automatically linked to this view. This view assignment is maintained throughout the life cycle of this instance.

To access further areas, an additional instance of the plant overview is started. Visualization in other views is not possible in one instance of the control recipe visualization!

10.6.3.4 Sequencer setpoints and process values

Core statement

- The actual setpoint and process values of the currently selected sequence are visualized in a separate setpoint/process value window.
- Depending on the parameter type, the setpoints can be edited or selected in special selection dialogs.
- The special features of the individual parameter types are described in the section 'Edit DFM definitions' dialog (Page 581).

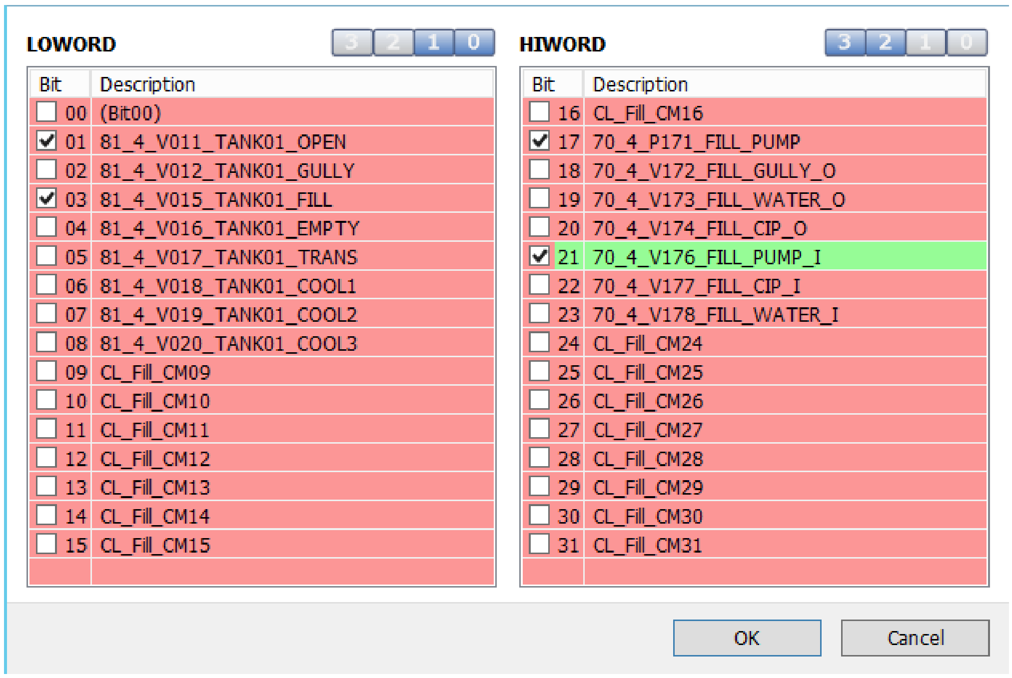
Columns displayed:

- Name (name of the DFM)
- Dimensions
- Setpoint
- Actual value
- Delta (= setpoint – process value)

.

"Text (Bitfield)" parameter type (PCU from V7)

- The setpoint is defined by a bit pattern (32 bits).
- The display uses a separate dialog in which the 32 bit positions can be set/reset using select boxes.



Note

Update and representation of the setpoints as well as the associated actual values

The displayed bit information from the PCU DFM object is formed when the dialog opens.

- Check mark is set → setpoint bit = true
- Check mark is not set → setpoint bit = false
- Line green → actual actual value = setpoint
- Line red → actual value ≠ setpoint

Caution:

The statuses are not updated when the dialog is open. The statuses visible here are written to the DFM object in case of changes to the setpoint bits and confirmation with the OK button, regardless of whether the statuses have changed in the PCU in the meantime (e.g. through step forwarding).

See also

Structure of the user interface (Page 788)

10.6.3.5 Step-related setpoints

In the window, the step-related setpoints of the running step are displayed and these can be changed.

As a first setpoint, the step running time is indicated.

Otherwise, the same is valid for the input and the view as for the setpoint and process values of the unit.

10.6.3.6 Working with the application

View description

The meaning of the individual parts of the view is described under 'format' (Chap. "Division and printout (Page 817)").

The display of the units is in blocks of a maximum of 16 units. However, it is possible to page back and forth line- and page-wise with the cursor keys. The selected unit is marked in color. The setpoints and process values are displayed for the currently selected unit in the lower operating area. It is also possible to select the unit by clicking the unit name with the mouse.

Selection of the process cell

For a plant-crossed operation the system which needs to be operated can be selected via the menu 'Anlage' 'process cell'. The displayed data and manual operations of units refer exclusively to the selected process cell.

Selection of the unit

The colored background of the unit name indicates the currently selected unit.

There are two possibilities for selecting the unit:

- Cursor keys
- Click the unit names with the mouse

Selection of a sequence step

The sequence step can be selected in 3 ways:

- By selecting the requested unit and clicking the **'Step'** icon in the toolbar
- Selecting the requested unit. After that the menu 'function' is selected via the menu item **'step'.**
- Click the column '**step'** for the requested unit with the mouse. The selection of the unit follows automatically after that.

The dialog box **'step selection'** follows. This sets the 'plant overview' application to the waiting state (model dialog) as long as the operation with the input of one value between 0 and 255 and return is closed.

With the dialog box the current step is displayed. This can be confirmed by a return or overwritten by entering a different value and return. When entering a negative value the **plant overview** message box appears with the note "Value too small!"; with the input of a value more than 255 the **input** message box "Value too large!" appears. If you do not enter a number, the **Plant overview** message box pops up the with the information: "Please enter an integer number! “. Both message boxes are modal and must be operated with the OK confirmation. The cursor is in the input field of the step selection again, and the primary value is displayed. After a successful step selection the entered value appears in the step column of the selected unit.

The input of the value "0" causes a stop of the unit. With the input of the value "1" the unit is started. It is also possible to start the unit via the menu item **'Start sequence'** in the menu **'functions'**. With a standing unit (Step 0) the input of recipe and batch is possible.

Note

- The step selection dialog reads the recipe structure from the offline data management. If a recipe procedure is enabled for editing and changed while batches based on it are still running, if the step selection dialog is opened later, there may be a deviation compared with the currently running batch.
- **Changes to recipes should therefore only be made when no more batches are running with this recipe procedure.**

Selection of the recipe, order and batch number

The selection is only possible if the corresponding unit is not started (current step 0). The selection can take place in several ways:

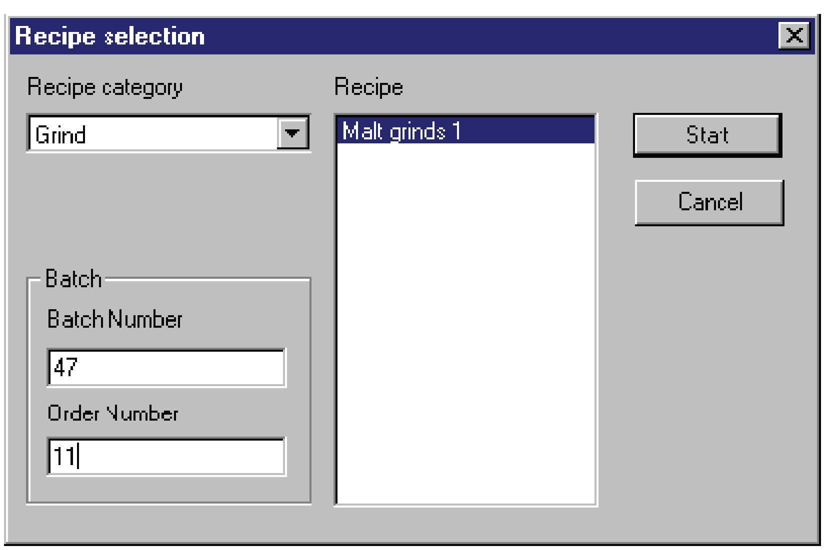
- Selection of the requested unit. After that the menu 'function' is selected via the menu item **'step'**..
- Click the column **'Recipe' 'Batch'** or **'OrderID'** for the requested unit with the mouse. The selection of the unit follows automatically after that.

The dialog box **'recipe select'** follows. This sets the application **'plant overview'** in holding position until the operation is confirmed with OK or abort.

Dialog 'Recipe selection'

Basic principle

The indication 'recipe selection' for the recipe, order and batch number are preset via a dialog.



Batch

Input fields for the order and batch number.

Recipe

Signaling recipes for the selected recipe category.

OK

The values are accepted and transmitted to the controls.

Input of setpoints

The displayed setpoints and actual values in the parameter window always refer to the current step of the selected unit. After the continuation to the next step or after the selection of another unit the setpoints and actual values are updated automatically. In addition, the actual values are updated automatically during the processing of a step. The setpoints are accepted from the corresponding recipe; the actual values come from the process directly.

The setpoints for a sequence step can always be changed. However, the change is only effective for this individual step; no parameter transfer occurs in the corresponding recipe.

For the input of setpoints the following operation is necessary:

- Move the cursor with the mouse into the corresponding input field of the setpoints and click on the field. The current value in the field is selected.
- It is possible to change to different input fields with the mouse.
- The high and low limits for setpoint input are monitored. Values are only accepted if they lie within range.
- The setpoint value input is closed with the return key.

Starting a unit

Basic principle

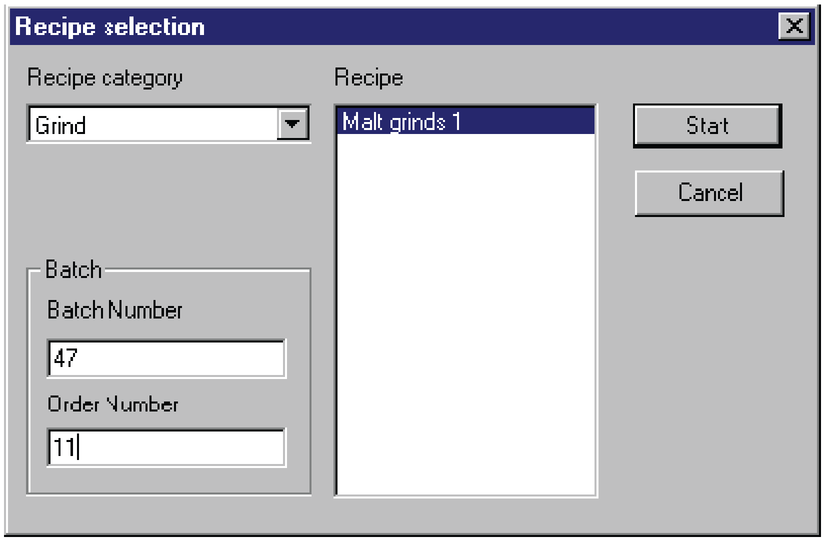
The start of a unit occurs

- via the 'Start' menu item of the menu 'Functions',
- with the input of step 1 for the corresponding unit (no further dialog is shown)
- or by clicking the start icon  in the button bar.

Before the start of the sequence, a dialog which indicates the current order and batch number and the selected recipe is displayed.

After the start, the corresponding setpoints are fetched from the recipe list for every step and displayed in the unit parameter window.

Dialog



Batch

Input fields for the order and batch number.

Recipe

Signaling recipes for the selected recipe category.

Start

The values are accepted and transmitted to the SIMATIC. The sequence is started.

Aborting a sequence

The unit can be aborted at any time

- via the menu item **'Sequence stop'** in the menu **'functions'**,
- with the input of step 0 for the corresponding unit
- or by clicking the icon  **'Sequence stop'** in the button bar.

Enable step switchover

The automatic incrementing into the next step of the current unit can be enabled with the operating mode selection "**+**":

- Menu item **'enabling'** in the menu **'functions'**
- Click the icon **'enabling'** in the button bar

The unit is held after closing any step by the operation type selection:

- Menu item **'Hold sequence'** in the menu **functions**
- Click the icons **'Hold sequence'** in the button bar

Errors of a unit

An error in the unit is displayed by a flashing icon in the 'status indicators' column of the corresponding unit. In the event of an error a horn can be controlled, the signal for which is acknowledged with the symbol bar icon 'acknowledge horn' or with **F12**.

Errors in units can have the following causes:

- The monitoring time expires without reaching the requested setpoint.
- The monitoring time expires and the step enabling condition for the next step is missing (e.g. acknowledgments).
- The monitoring time expires and the sequencer is in stop mode.

After removing errors, the unit can be started with the next step or the unit switches automatically to the next step.

The sequence steps of a unit can be continued in spite of present errors. You continue to the next step of the unit via the manual step selection. In this way it is possible to avoid errors in the unit. The user must be aware of the possible effects.

Additional flag On/Off

This function enables a data bit to be set and reset in the data record of the unit that is user-defined.

User request

Basic principle

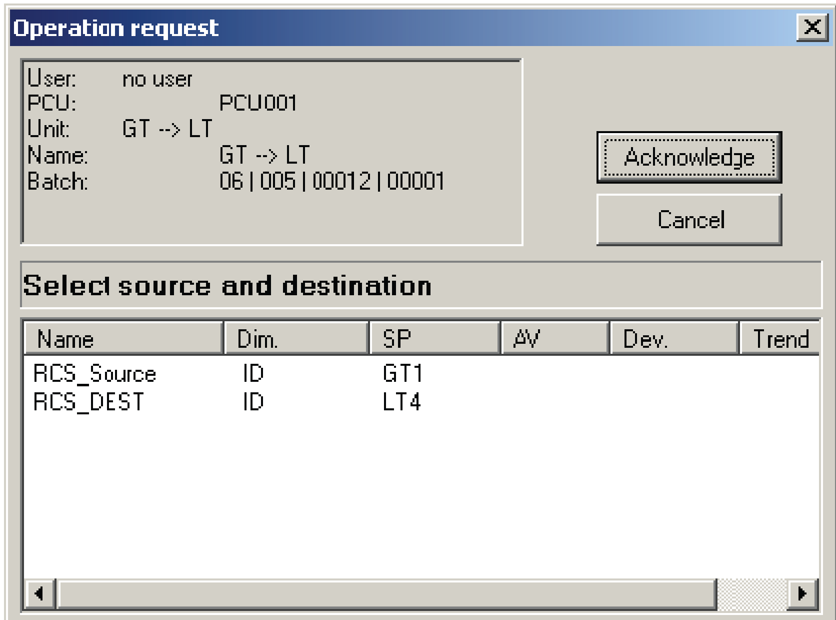
Active operator requests marked by a flashing in the **'Status indicators'** column can be acknowledged

- with the menu item **'Acknowledge->User acknowledge'**
- by using the function key **F8**
- or by clicking the start icon in the button bar.

acknowledged.

Operation request dialog

After activating the button, a dialog is displayed.



Operating request with value input

The operating request can also be submitted with a value input.

- Depending on the parameter type, the setpoints can be edited or selected in special selection dialogs.
- The special features of the individual parameter types are described in the section Dialog 'order parameter' (Page 741).
- The size and column width of the setpoint windows can be changed. The affected settings are stored.

(See also: Dialog 'Project equipment operation' (Page 577))

Sequence-related message window

The pending errors, warnings and process messages are removed from the view by activating the corresponding buttons in the button bar.

	Remove messages which are configured as message class 'Error'
	Remove messages which are configured as message class 'Warning'
	Remove messages which are configured as message class 'Process messages'

Note function

Open a text editor for entering a note by the user. The text can be changed, added and filed in any way.

- The note is processed via the Windows editor.
- A note for the unit or a note for the current recipe can be written or called by the user.
- The recipe note is selected via the '**recipe note**' menu item in the menu 'functions' or via the function keys '**Shift+F10**'.
- The note for the unit is selected via the menu item **'note'** in the menu '**functions**' or via the function key '**F10**'.

Selection unit process images

Basic principle

Process images of the individual units can be selected directly by the menu item '**unit process figure'** in the menu **'program'** or via the function key '**F2'.** The process figure which is associated with the selected unit is displayed.

Check your configuration if the message "Image not found" appears. The image names for all 128 units are stored in the file `<proj-path>\pcu.xxx\texte.x\kettbld.txt`.

Example:

```
\Sudhaus\MALT-1.bik
\Gärkeller\FLOTAT1.bik
\Gärkeller\PROPAGA.bik
\Gärkeller\PROPAGA.bik
\Filtration\TFF.bik
\Filtration\WA-YEAST.bik
\Filtration\TFF-CIP.bik
KETTBLD   08
...
```

The stated subfolders belong in the folder `<proj-path>\bilder\....`

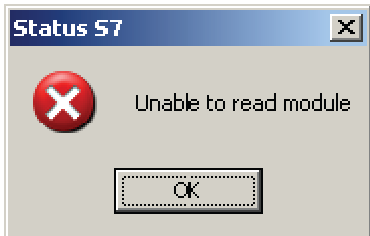
Selection Status

The status of the current unit is displayed after the call of the menu item **'status'** in the menu '**program'** or via the function key **'F4'.**

When the unit is not running, the system displays corresponding permanent conditions in FB1001 to FB1128.

When the unit is running, the system displays Network 1 of the currently active EOP (FC1001 to FC2999).

The following message box indicates that the block was not found in the controls



For a description of application 'Status', see chapter General (Page 982).

Status of start and permanent condition

The permanent condition of the current unit is displayed after calling the **'start and permanent condition'** in the **'program menu'** or via the function key **Shift+F4.**

One of the following possibilities is displayed, depending on the parameterization of the object attribute "Sequenzen\PermCondRef" (source for the permanent condition / from PCU Version V7):

- PermCondRef = not activated → the application "Status_S7.exe" is opened with the permanent condition in the associated unit FB (FB1001 ... FB1128)
  See chapter Status for Simatic S7 via ‘Status_S7.exe’ application (Page 982)
- PermCondRef = activated to object "Binary Logic Result" (BLR) → The application BLREdit.exe is opened with the online logic display of the referenced BLR instance
  See chapter BLR diagnosis/Online (Page 232)

Selection Status step on condition

The permanent condition of the current unit is displayed after calling the **'start and permanent condition'** in the **'program menu'** or via the function key **Shift+F4.** Depending on the unit state and the selected EOP configuration of the current step in the recipe editor, one of the following displays is opened:

- Case 1: The unit is not running:
  In this case, the associated sequence permanent condition is displayed (see section above).
- Case 2: The unit is running:
  In this case, one of the following diagnostics applications is opened depending on the EOP configuration (see section Dialog 'Project equipment operation' (Page 577)) of the control recipe currently running:
  – Step condition "via RLO of the EOP FC"
    The network 5 "Step condition" of the EOP FC (FC1001 to FC2999) currently running is displayed via the application "Status_S7.exe"→ see also section Status for Simatic S7 via ‘Status_S7.exe’ application (Page 982).
  – Step condition via BLR object
    The application BLREdit.exe is opened with the online logic display of the referenced BLR instance → see also section BLR diagnosis/Online (Page 232).

Selection diagnosis of the routes

The diagnosis function of the route control for the current unit can be called directly via the menu item '**route'** in the menu **'program'** or via the function key **'Alt+F8'.**

Selection batch list

Via the menu item '**batch list'** in the menu **'program'** or via the function key **'Ctrl+F8'** the batch list is called for the current unit.

Show control recipe

Via the key **'F9'** the actual control recipe of the batch of the selected sequence is opened.

Via the key **'CTRL + F9'** the control recipe data block of the PCU is opened (Control recipe PCU) and the control recipe unit procedure is shown in the list view.

Selection 'Edit process cell'

Via the menu item **'edit process cell' under 'functions'** you reach an ASCII editor with a loaded file "Area.txt". The individual unit areas are entered here.

Selection 'Edit process cell view'

Via the menu item **'edit process cell' under 'functions'** you reach an ASCII editor with a loaded file "KPOSxxx.txt". Here, the units which are part of one area are entered whereby "xxx" corresponds to the area number.

10.7 Control recipe display 'ControlRec'

10.7.1 System-relevant properties

Note

- The function for the visualization of control recipes does not archive data of expired batches. This archiving function is covered by the step protocol functions. This means that only the currently active batches can be displayed.
- For reasons found in the architecture and utilization of the system, integration of the control recipe display function in the recipe process is based on sourcing methods, instead of using the PCU datagrams as in the step protocols. For this reason, it may seem that a step is skipped when the system is operating at higher step rates and when certain steps are performed (Start EOP and End EOP). In this case, the step protocol provides reliable offline information on all processed operations.
- The graphic control recipe view reads the recipe structure from the offline data management. If a recipe procedure is enabled for editing and changed while batches based on it are still running, if the control recipe view is opened later, there may be a deviation compared with the currently running batch.
- **Changes to recipes should therefore only be made when no more batches are running with this recipe procedure.**

10.7.2 Short description

The task of the control recipe visualization function is to visualize the processing steps of control recipes which were generated by the recipe system and downloaded to the PCU. Based on the recipe procedures, the recipe system generates the control recipes by substituting the process/order parameters with the current master recipe/order parameters. When the recipe procedure contains weighing recipe operations, the control recipe dynamically generates weighing and interim draining operations, depending on the volume of process input materials to be weighed. The completed control recipes are then downloaded as unit recipes for processing on the relevant PCUs.

10.7.3 Starting control recipe visualization

10.7.3.1 Multi-client function

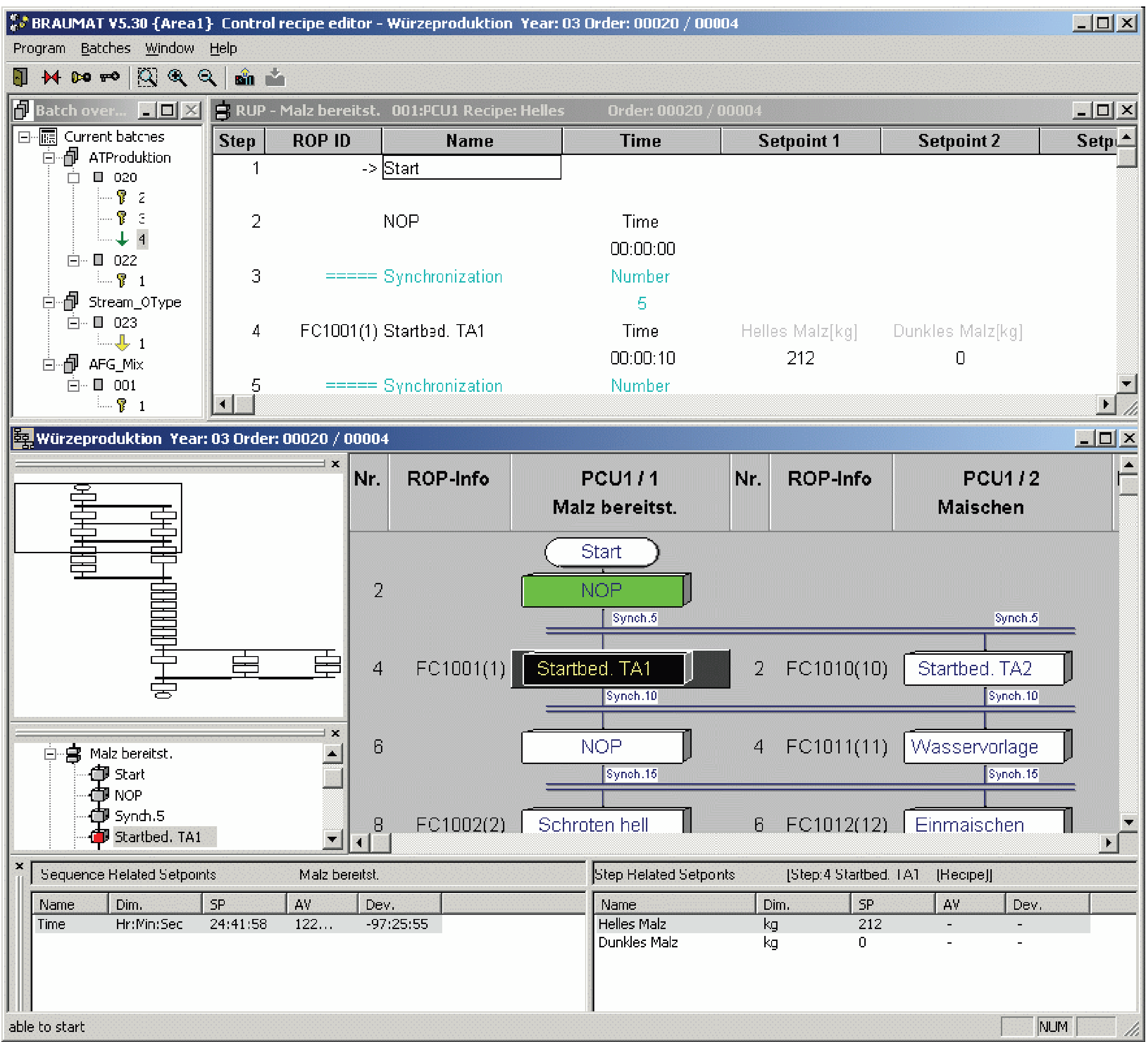
When you start an application on a client that is configured for operation in a multiclient architecture, the view selection dialog box opens, showing the enabled views. After you have selected a view, the current instance of the control recipe visualization is automatically linked to this view. This view assignment is maintained throughout the life cycle of this instance.

Visualization in other views is not possible in one instance of the control recipe visualization!

10.7.3.2 Multiple-instance capability and number of concurrently monitored control recipes

For reasons of resource utilization, it is not possible to open multiple instances of the control recipe view simultaneously. This also applies in multiclient mode of an IOS station. Operators who want to monitor control recipes located in the inactive area of the current control recipe visualization must restart the application and then select the relevant view from the area selection dialog box.

10.7.4 Overview – Views of the control recipe visualization



Currently active recipes are visualized in the control recipe view in two different modes. The program can generate the structure of a control recipe, based on the offline recipe configuration, and it can visualize a batch process in this control recipe structure, based on the unit image it fetches from the PCU server. The graphic view of the control recipes serves this purpose. The step and sequencer reference values, such as the setpoint for the step monitoring time, can be modified in the setpoint list windows of the graphic control recipe view.

The control recipe view can also fetch the current recipe DB from the automation device and visualize it as unit procedure control recipe. In this visualization mode, the operator can modify the setpoint values and monitoring times of the unit recipe, and then perform a delta download to the automation device.

The batch overview shows all active batches, sorted in a group according to their order type and number, and it is used to select a control recipe for visualization in the graphic view.

10.7.5 Operator control elements common for all views

10.7.5.1 Menu commands

Program

- This menu area is filled according to the entries in "Menuappl.ini".

Batches

- Update Refreshes the structure view of the batch overview, and removes expired batches (status = "Done") from the list
- Fetch TRP from PCU Fetches the current unit recipe from the PCU. When the batch overview is active, the system shows a dialog window for selecting the sequence. When the graphic control recipe view is active, the system loads the recipe unit procedure which is indicated in the column at the cursor position.
- Order parameters Opens the order parameters list of the batch that is selected from the batch overview or opened in the graphic view. This menu item is disabled if you have not selected a batch in the batch overview.
- Process input Opens the process input list of the batch that is selected from the batch overview or opened in the graphic view. This menu item is disabled if you have not selected a batch in the batch overview.
- Close Closes the control recipe view.

Window

- Cascade
- Tile horizontally
- Tile vertically Arranges the opened and not maximized windows accordingly.

Help

- System Opens the system Online Help.
- System index Opens the index of the system Online Help.
- Control recipe Opens the Online Help for the control recipe view.
- Control recipe index Opens the index of the Online Help for the control recipe view.
- About Opens a dialog that shows Copyright and Version information.

10.7.5.2 Dialog boxes used in all views

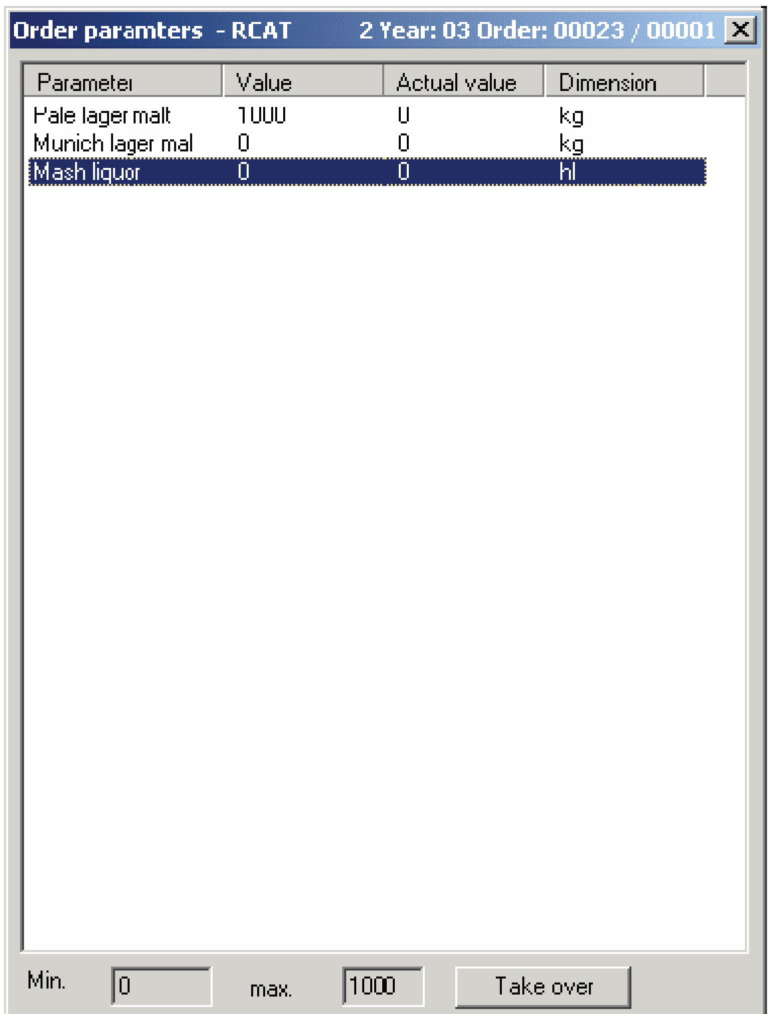
Toolbar buttons used in all views

	Exit program. Closes the control recipe view
	Acknowledge ICM errors
	Acknowledge audible messages
	Reset password immediately
	Download recipe unit procedure from PCU

This toolbar button is assigned various functions, depending on which window is currently active in the control recipe view: When the batch overview is active, the function opens a sequence selection dialog window for selecting the sequence (= recipe DB) of which the recipe unit procedure is to be read. When the graphic view is active, the sequence is determined by the cursor position, i.e. the function reads the recipe unit procedure of the sequence on which the cursor is positioned.

'Order parameters' dialog box

Basics



You can open this dialog box by calling the "Batch/order parameters" command in the batch overview and in the graphic control recipe view, or the "Order parameters" shortcut command for a batch in the batch structure view.

Order parameters list

The order parameters of an inactive batch can be modified in the values column of the order parameters list. The dialog box is used for monitoring the order parameters of active batches. However, users can explicitly enable CIR for the order parameters of an active batch. This is done by setting the "ChangeRunningBatches" switch in the "Settings" section of the file "`...<proj-path>\sys\baliedit.ini`" to the value "1".

Min. and Max.

The two read-only text boxes show the minimum and maximum values of the selected order parameter.

Apply

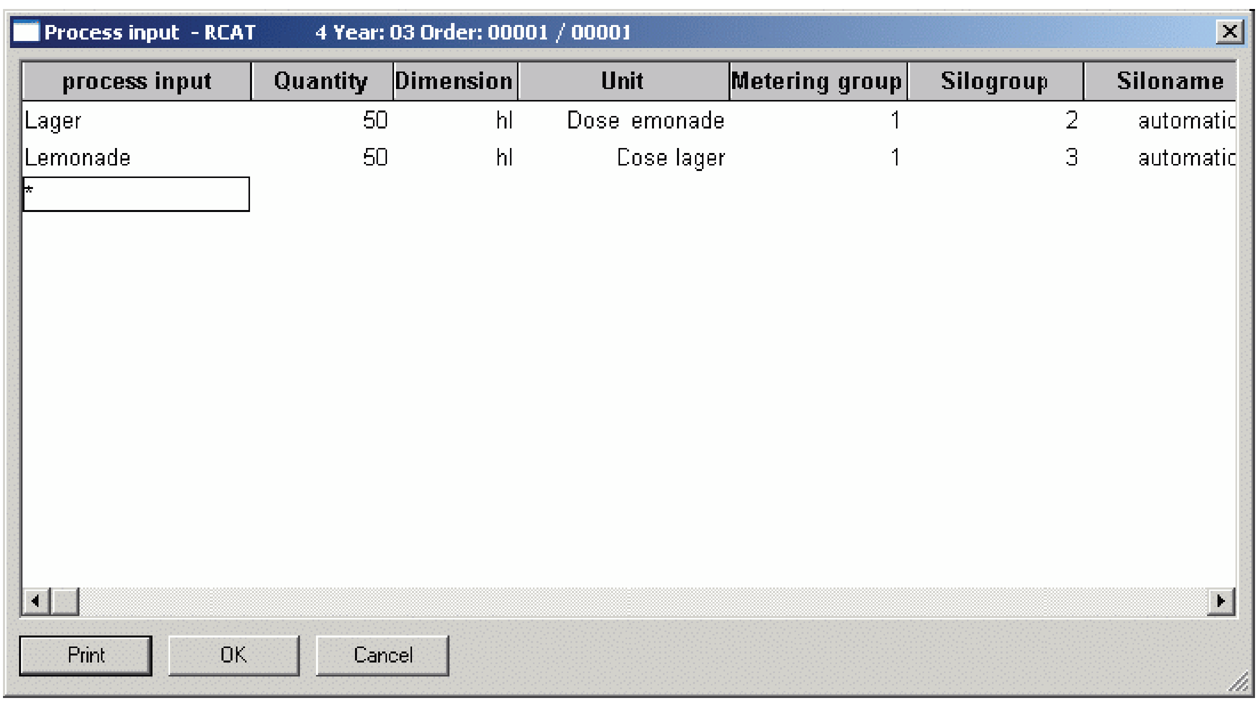
Downloads the modified order parameters to the PCU.

Close order parameters list

This dialog box is closed either by pressing the RETURN key or by clicking the exit button ("x") on the upper right.

'Process input list' dialog box

Shows the process input list for the selected batch.



The error message "Could not open batch process input list!" is output instead of the dialog box when a request to read the batch process input list fails, or an attempt is made to open a process input list, which does not contain weighing procedures.

The control recipe visualization program utilizes the batch process input list for the purpose of monitoring. The process input list cannot be modified in this application.

Print

Opens the standard Windows® printer dialog box, and outputs the batch process input list to the printer.

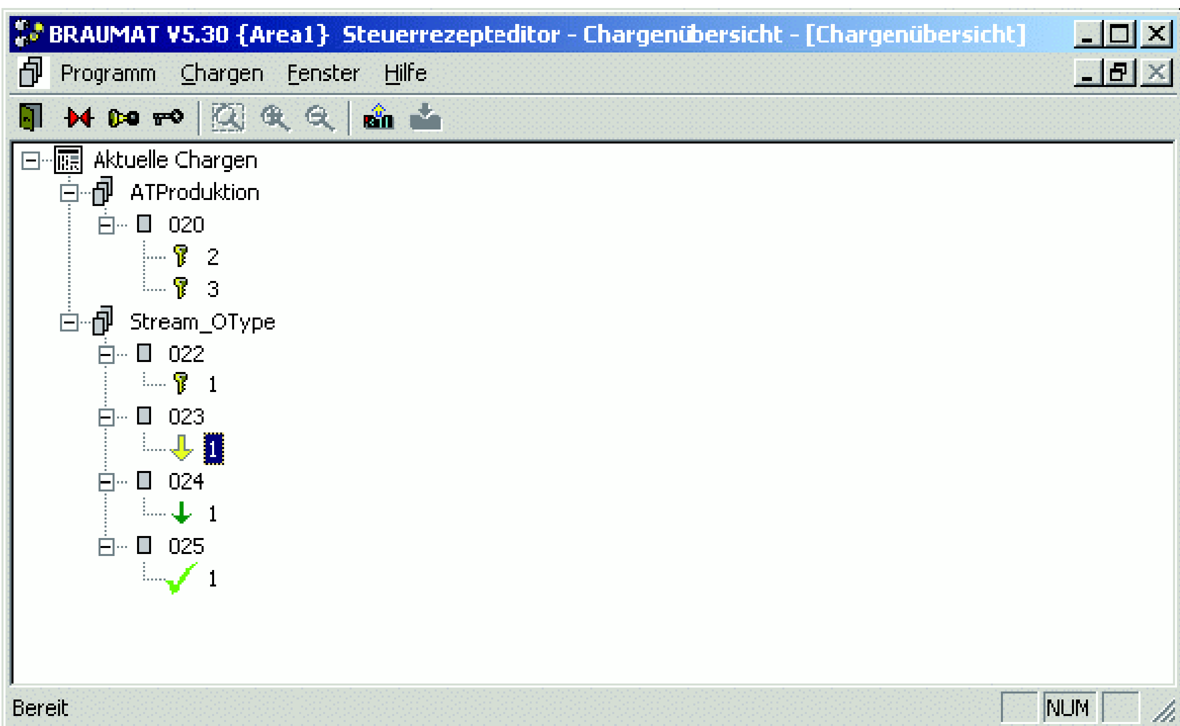
OK

Closes the dialog box.

Cancel

Closes the dialog box

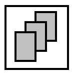
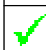
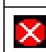
10.7.6 Batch overview



The batch overview represents the primary view of the control recipe visualization program. It opens immediately after the application has started and cannot be closed. This tree structure shows all active batches, sorted by their order types and orders. The control recipe visualization program does not access system archives, and does not provide internal batch history data. Hence, the application shows only the batches which are in the "locked", "ready" or "started" state. The program shows all states of the batches opened in the structure view while the control recipe view is active, i.e. it also indicates the status transition to "done". However, batches which have acquired the "done" status are deleted from the structure when the batch overview is updated.

10.7.6.1 Symbols of the structure view

The icons shown in the table below are used to indicate the batch status and the structure in the overview.

Icon	Meaning
	Root element of the Batch overview, "Current batches"
	Order type
	Order
	Batch in state "Locked"
	Batch in status "Ready for release"
	Batch in state "Released"
	Batch in state "Started"
	Batch in state "Released"
	Batch in status "Aborted"

10.7.6.2 Menu commands of the batch overview

In addition to the menu commands which are available in all views, a shortcut menu is provided for the batches in the structure view (open with right-click). The commands are:

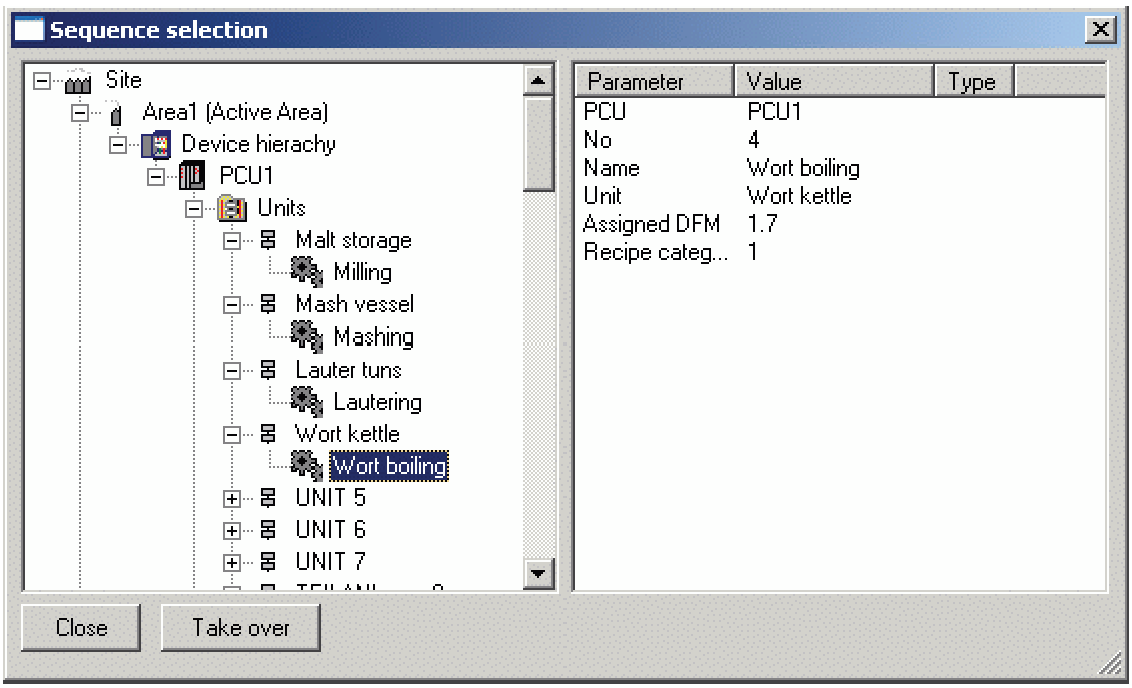
Batch shortcut menu

- Open control recipe
  Opens the graphic view of control recipes
  See also: Chap. "Graphic view of control recipes (Page 839)"
- Order parameters
  Shows the order parameters list of the selected batch.
  See also: Chap. "'Order parameters' dialog box (Page 834)"
- Process input list
  Shows the process input list for the selected batch.
  See also: Chap. "'Process input list' dialog box (Page 835)"

10.7.6.3 Batch overview dialog boxes

'Sequence selection' dialog box

Before you can upload a control recipe list from the PCU by means of the batch overview, you need to select a sequence from the "Sequence selection" dialog box in order to determine which recipe DB the system has to fetch.



Structure view of the plant overview

Provides a system component tree and is used to select a sequence. The system reads the recipe unit procedure of the sequence from the corresponding PCU, and interprets and visualizes it in a list.

Properties list

Improves orientation by showing a list of the system components.

Close

Closes the dialog box, without selecting a recipe unit procedure.

Apply

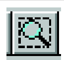
Closes the dialog box and reads the selected recipe unit procedure from the PCU. This button is only enabled when a sequence is selected.

10.7.7 Graphic view of control recipes

10.7.7.1 Title bar information

The title bar of the window, or of the application with maximized window, shows the name of the recipe category, the order year and the order/batch number.

10.7.7.2 Toolbar commands

	User-specific zoom	
Allows the user to set the zoom factor. In the graphic recipe procedure, the operator can hold down the left mouse button to mark an area, and then drag it to the full size of the window.		
	Zoom in	
Increases the zoom factor by one step.		
	Zoom out	
Decreases the zoom factor by one step.		

10.7.7.3 User interfaces of the graphic control recipe view

The "Recipe procedure overview" and "Recipe tree" windows are a navigation tool,

- which the user can switch on and off, dock onto other windows, and use for quick navigation and orientation in complex recipe procedure structures.
- In contrast to the navigation tools of the recipe editor, data in the windows of the graphic control recipe view are read-only and thus cannot be edited.
- In contrast to the recipe editor, the graphic recipe procedure view of the control recipe view contains a setpoint window, which can be docked and switched on and off.

10.7.7.4 Status indication

Basics

The processing states of technical operations are indicated in the graphic control recipe view by means of configurable character/background colors, and using the icons next to the operations.

Color settings

The colors of the graphic control recipe view can be customized to suit user requirements in "`...<proj-path>\recipe\project\plant.ini`". If the user does not write data to this file or deletes data, the system uses the default values shown in the table below.

The colors are coded according to the RGB model, i.e. the color scheme is an additive generated from the red, green and blue element. The intensity of an element is described by a value between 0 (no color) and 255 (maximum of the element). Thus, "0,0,0" black, "255, 255, 255" white, "255, 255, 0" yellow, "255, 0, 255" violet and "0, 255, 255" turquoise.

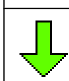
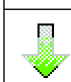
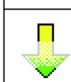
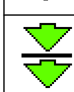
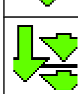
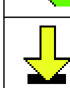
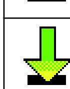
The following table lists all entries and the accompanying default values:

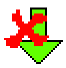
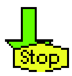
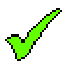
Section	Entry	Standard RGB value	Comment
Grafic-Onl	StdBk	255, 255, 255	Background color of the drawing area
Grafic-Onl	GopColorStd	0, 0, 0	Standard character color for foreground
Grafic-Onl	GopColorStdBk	255, 255, 255	Standard character color for background
Grafic-Onl	GopColorIdle	0, 0, 0	Color of characters in "Idle" state
Grafic-Onl	GopColorIdleBk	255, 255, 255	Background color in "Idle" state
Grafic-Onl	GopColorRunning	0, 0, 128	Character color in "Running" state
Grafic-Onl	GopColorRunningBk	0, 255, 0	Background color in "Running" state
Grafic-Onl	GopColorStarting	0, 0, 0	Character color in "Starting" state
Grafic-Onl	GopColorStartingBk	228, 128, 32	Background color in "Starting" state
Grafic-Onl	GopColorRestarting	0, 0, 0	Color of characters in "Restarting" state
Grafic-Onl	GopColorRestartingBk	228, 128, 32	Background color in "Restarting" state
Grafic-Onl	GopColorPaused	0, 0, 0	Character color in "Paused" state
Grafic-Onl	GopColorPausedBk	32, 128, 128	Background color in "Paused" state
Grafic-Onl	GopColorPausing	0, 0, 0	Character color in "Pausing" state
Grafic-Onl	GopColorPausingBk	32, 128, 128	Background color in "Pausing" state
Grafic-Onl	GopColorHeld	0, 0, 0	Character color in "Held" state
Grafic-Onl	GopColorHeldBk	255, 255, 0	Background color in "Held" state
Grafic-Onl	GopColorHolding	0, 0, 0	Character color in "Holding" state
Grafic-Onl	GopColorHoldingBk	255, 255, 0	Background color in "Holding" state
Grafic-Onl	GopColorAborted	0, 0, 0	Character color in "Aborted" state
Grafic-Onl	GopColorAbortedBk	128, 255, 128	Background color in "Aborted" state

Section	Entry	Standard RGB value	Comment
Grafic-Onl	GopColorAborting	0, 0, 0	Character color in "Abort-ing" state
Grafic-Onl	GopColorAbortingBk	0, 255, 0	Background color in "Abort-ing" state
Grafic-Onl	GopColorStopped	0, 0, 0	Character color in "Stop-ped" state
Grafic-Onl	GopColorStoppedBk	128, 255, 128	Background color in "Stop-ped" state
Grafic-Onl	GopColorStopping	0,0,0	Character color in "Stop-ping" state
Grafic-Onl	GopColorStoppingBk	0,255,0	Background color in "Stop-ping" state
Grafic-Onl	GopColorComplete	0,0,0	Character color in "Com-plete"
Grafic-Onl	GopColorCompleteBk	128,255,128	Background color in "Com-plete" state
Grafic-Onl	GopColorOpReq	0,0,0	Character color for queued operator request
Grafic-Onl	GopColorOpReqBk	0,255,0	Background color for queued operator request
Grafic-Onl	GopColorTimeOut	0,0,0	Character color for timeout
Grafic-Onl	GopColorTimeOutBk	0,255,0	Background for timeout

Status icons

The actual state of the operations is shown on the graphic recipe procedure view by means of a symbol on the right beside the operation.

Icon	Status
No icon	Idle
	Running
	Starting
	Restarting
	Paused
	Pausing
	Held
	Holding

Icon	Status
	Canceled
	Aborting
	Stopped
	Stopping
	Complete
	Queued operator request
	Timeout

10.7.7.5 Setpoint/process value window

The graphic recipe procedure view has a setpoint/process value window in which the setpoint and process values of the sequence are shown in the left half and the setpoint and process values of the steps are shown in the right half.

Sequencer setpoints and process values

The actual setpoint and process values for the 13 parameters of the currently selected sequence are visualized in a separate setpoint/process value window.

Sequencer setpoints are configured in "sequence.ini".

See also: Setpoints for units

The first position in the list of sequencer setpoints shows the step monitoring time of the active step of the sequence, followed by the sequencer setpoints/process values. The setpoints/process values are hidden when the cursor is positioned in the column of an inactive sequence. After the sequence is completed, the window shows the step monitoring time setpoint/process value of the step which was last executed.

You can edit the setpoint column directly in the window. To edit text-based setpoints, a separate dialog box is opened when you select a setpoint cell.

Columns displayed:

- Name (name of the DFM)
- Dimensions
- Setpoint
- Actual value
- Delta (= setpoint – process value)

.

Step-related setpoints

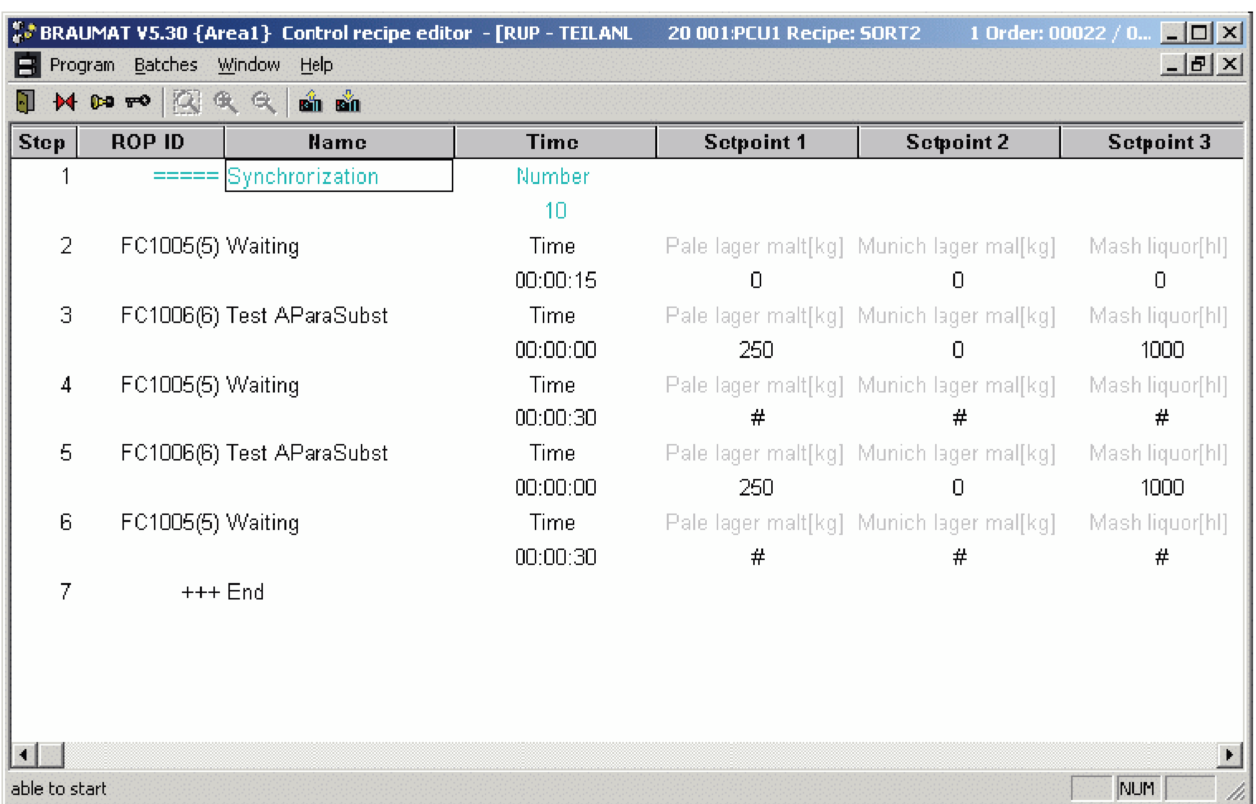
The step setpoints list shows the setpoints and process values of the DFM when the step is active, and the setpoints of the recipe DB when the step is inactive. This difference is indicated by the list header. The text "[DFM]" is appended to the setpoint and process values of the DFM, otherwise the text "[recipe]".

Otherwise, the same applies to the input and view as for the setpoints and actual values of the sequence.

Visualization of setpoints in the "Tooltip" window of the graphic tree view

The "Tooltip" window of the graphic control recipe view always shows the offline recipe setpoints in the step Tooltip.

10.7.8 Recipe unit procedures list



The recipe unit procedures list is read online from the PCU, and represents a recipe unit procedure in the form of a unit control list for one sequence. Setpoints and step monitoring times can be modified in the recipe unit procedures list, and can be written back to the PCU by means of the toolbar/menu command "Write RUP list to PCU".

10.7.8.1 Title bar information

The following information is shown in the title bar of the recipe unit procedure window, or, when the window is maximized, in the title bar of the control recipe view:

- Name of the sequence
- PCU number
- PCU name
- Name of the master recipe
- Order number
- Batch number

.

10.7.8.2 Toolbar commands

Write RUP list to PCU Downloads the recipe unit procedure which was modified in the control recipe view to the PCU.

10.7.8.3 Menu commands

Batches

- Write RUP list to PCU Downloads the recipe unit procedure which was modified in the control recipe view to the PCU.

10.7.8.4 Recipe unit procedure list

Special operations such as "Synchronization" or "Alternative" are shown in the list window of the recipe editor.

See also: Recipe system The recipe procedure views/windows/recipe unit procedure list

In this list you can modify only the setpoints and monitoring times. The recipe structure is read-only. Setpoints can neither be substituted with order or process parameters, nor can they be scaled.

11 Batch curves

11.1 Basic principles

11.1.1 Overview

Functions

- The system records batch-oriented measured value curves. The triggering process happens in the PCU. It triggers groups of curves rather than individual measured values. Measured values can be divided into up to a maximum of 240 curve groups per server / server pair.
- If measured values are recorded by a server, the number of variables is reduced to the quantity that the server is able to provide at any given moment.
- More than 1 server can record curves in the plant.
- A compression procedure is used during recording.
- A list describing the measured values needs to be created. The list describes all measured values to be recorded. For each server this list may contain up to 1200 entries for analog measured values and 2400 entries for digital measured values.
- The digital measured values can be derived from all the analog measured values that are available.
- Besides batch trends, the system also handles weekly archives and short-time recording.
- Curve visualization is provided by a client application, which can run on each system.
- The client can visualize measured values from different curve servers.
- An aggregation of several values is called a "curve picture“.
- It is possible to put several curves on top of each other and move and stretch them for the comparison.

Configurations

- Single-user configuration
  Curve record as well as curve visualization run on the same OS.
- Client - server configuration
  There is a curve record on one or two redundant servers and the curve visualization is available on all clients.
- Multi client - server configuration
  On the server as well as on the clients, measured values from several servers are visualized.

11.1.2 Measured value type

A measured value must be allocated to one of the following archive types:

- 1: Short-term curve
  The measured value can only be used in short-term archives.
  They run automatically, i.e. do not need a curve trigger from the control system.
- 2: Batch archive
  The measured value can be used in batch archives and in weekly archives.
  They do not run automatically, i.e. they need a start trigger and a stop trigger from a control system.
- 3: Weekly archive
  This measured value can only be used in weekly archives.
  These run automatically when a curve group > 240 is used. If curve groups 1-240 are used, a trigger from a control system is required.

11.1.3 Curve group

A curve group contains several values, which are related in a technological manner. Recording is enabled or stopped for the whole group, whereas **at most 240 groups** can be configured.

Grouping of curves is performed during engineering of the description list mentioned above.

11.1.3.1 Status of the curve groups

The PCU software includes a **trigger data block**, which aggregates the actual information relating to all curve groups (status list). The data record of a curve group includes four data words

(DB689 in SIMATIC S7).

Structure of the data block

Address (in DW)	Meaning	
Offset	Offset	Has the value 2
Offset + (n-1)*4	Job number	n = Group number
Offset + (n-1)*4 + 1	Batch number	n = Group number
Offset + (n-1)*4 + 2	Year/Recipe type	n = Group number
Offset + (n-1)*4 + 3	Recipe number	n = Group number

11.1.4 Triggering curves

Triggering (enabling/stopping) data collection within the PCU is performed by a function block trigger **function block (FC695).** Batch information as well as the curve group number is provided for this block.

The function block is called by recipe procedures (Unit FB or basic operation).

11.1.5 Compare curves

Basics

The Option “Compare curves“ provides the possibility to view several batch or weekly archives simultaneously for comparison purposes. In this way you can distinguish between two functions.

Free selection

Function 1:
You can open up the curves of up to 8 batches or weeks for a configured picture (selection of measured values). The graphs can be moved in order to match them for an easier comparison.

Function 2:
By selecting freely, you can select curves from different batches. This selection is used as a reference (Step 1 of Selection) for comparison with another batch.

Compare curves

In order to determine the quality of a produced batch, the curve graphs of one batch or selection (step ‘Free Selection’) can be compared with the corresponding curves of another batch.

Note

If digital values are configured in the opened curve picture too, the comparison feature is not selectable.

Enable

In general, the function must be enabled.

File	`…<proj-path>\sys\kurven.ini`
Input	[Select]
Key	CmpOptionEna=1

11.2 AS configuration

11.2.1 How 'triggering' works

- Batch information is transferred into the trigger data block by starting the trigger.
- The Trendmanager is informed about the start of a record for the curve group by an AS → OS message.
- The Trendmanager reads the batch data and starts by recording the corresponding curve group.
- By stopping records an AS → OS message is sent to the Trendmanager, which stops the recording of measured values for this group.

11.2.2 Batch reference of trigger data block

- Direct specification of the batch reference (RecCatNo, BatchYear, OrderNo, BatchNo, RecipeNo)
  `BatchInfoFromSeq = 0`
  `SeqNo = 0`
- Indirect specification via sequencer number
  `BatchInfoFromSeq := 0`
  `SeqNo > 0`
  The batch reference is read from the sequencer data record, The sequencer data record is specified via the input 'SeqNo' .
- Indirectly via currently processed sequence
  `BatchInfoFromSeq = 1`
  `SeqNo = 0`

11.2.3 Input parameter of the trigger function block

CurveGroupID (1 – 240)

The curve group number must be provided manually and must be the same as configured in the measured value description list.

RecCatNo, BatchYear, OrderNo, BatchNo, RecipeNo

The batch reference can be provided directly via these inputs. The parameters are transferred into the trigger DB when starting the group.

Run (Bool)

A rising edge at this input parameter starts recording values for this group. As the batch identifier the values currently applied at the inputs RecCatNo, BatchYear, OrderNo, BatchNo and RecipeNo are taken.

BatchInfoFromSeq (Bool)

The parameter has the effect that the batch reference is adopted from the currently processed seqence.

SeqNo (Integer)

This input parameter references the sequencer (number) from which the batch reference should be taken.

11.2.4 Call examples

Direct input of the batch reference

The start and the batch reference can be provided independently by the sequencer with this call:

```
CALL  "BmTriggerCurveGroupUsrFC"
       Run              := TRUE
       BatchInfoFromSeq:= FALSE
       CurveGroupID     := nnn
       SeqNo            := 0
       RecCatNo         := B#16#1
       BatchYear        := B#16#1
       OrderNo          := 4711
       BatchNo          := 1
       RecipeNo         := 1
```

Indirect indication by Sequencer number

This call is necessary for background tasks of a sequence, which run independently of the sequence process itself but should trigger the curves anyway.

```
CALL  "BmTriggerCurveGroupUsrFC"
       Run              := TRUE
       BatchInfoFromSeq:= FALSE
       CurveGroupID     := nnn
```

```
        SeqNo           := 10
        RecCatNo        := B#16#0
        BatchYear       := B#16#0
        OrderNo         := 0
        BatchNo         := 0
        RecipeNo        := 0
```

Indirectly via currently processed sequence

The call must be executed in an EOP/EPH (FC1001 – FC1999) or in the sequencer function block (FB1001 – FB1128).

When the call is made, `"BatchInfoFromSeq":=TRUE` must be set and the curve group supplied. all other parameters are irrelevant.

```
CALL  "BmTriggerCurveGroupUsrFC"
        Run             := TRUE
        BatchInfoFromSeq:= TRUE
        CurveGroupID    := nnn
        SeqNo           := 0
        RecCatNo        := B#16#0
        BatchYear       := B#16#0
        OrderNo         := 0
        BatchNo         := 0
        RecipeNo        := 0
```

11.2.5 Telegram for triggering

Basic principle

Buffers (FIFOs) must be configured for the trigger telegrams The messages are type 18. The message type must be entered into one of the three FIFOs for each server. If curves are recorded in parallel, two FIFOs must be configured - FIFO 3 or FIFO 6 should be used as a default configuration.

Direct input

The input of the message type can be performed directly via the parameterization tool. However, you can use the automatic configuration during server startup.

Server startup

You can let the system parameterize the FIFOs during the startup of a server; This automatic parameterization must be activated via the application 'SiteCfg' in the PCU settings / tab 'FIFOs'.

- The option "Parameterization in server startup" must be activated here.
- The message type 18 has to be entered for FIFOs 3 (Server 1) and 6 (Server 2).
- The standard message types for the recipe system and route control are configured automatically and do not need to be preselected.

11.3 OS configuration

11.3.1 Configuring recording

Basics

- A configuration of configuration file "Trendman.ini" is generally no longer required for recording batch trends!
- The existing measured value trend servers are specified in the system configuration → IOS Settings → Enable 'Trend manager'
- The data recording is only possible on IOS servers.
- Trendman.exe starts automatically → Configuration for ProSched no longer required!
- The default location of the path for the curve data is now 'BM_ARCH' (like MSG, STEP and GlobLog)
  This path can be adjusted in SiteCfg → IOS Settings → Remote paths (should only be necessary exceptional cases!)
- The curve record can be redundant. This means that the value record runs on two servers, whereby both records run independently of each other.

For a few special cases there are settings in the configuration file "Trendman.ini" described below:

StoreOnExit

Save data before closing the application ON (default) / OFF.

Entry:	[App]
Key:	StoreOnExit=1

STBufferSize

Size of the buffer for short-term curves. The default buffer size setting is 150. The value can be changed within the limits of 60 to 1800.

Entry:	[App]
Key:	STBufferSize=150

StartMark

Enables (1, default) or disables (0) saving additional user information, when start trigger rises.

Entry:	[App]
Key:	StartMark=1

StartMark=0 (Default)

When trend recording is switched off by the curve trigger function or by closing Trenman.exe and then switched on again, the off value is linked to the start value, i.e. the user does not see that trend recording was interrupted.

Startmark=1

This sets a start flag at 80000000 hex in the curve archive each time trend recording is started. This value is not shown as a curve value, but is only an identifier for the curve output. The end value is thus not linked to the start value in the curve display, i.e. the curve is interrupted at this point, and the user can detect this interruption.

FlushCycle

Measured values are recorded in intervals according to the configuration in the measured value description list. In order to reduce the amount of hard disk accesses, the measured values are collected in the main memory of the server and written on the disk after a certain time. The cycle for storing can be defined by this indication (in seconds, default is 15 seconds). If there is a failure on the server, the values which are held in the main memory will be lost.

Entry:	[App]
Key:	FlushCycle=15

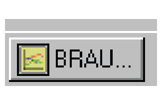
DiffTime

- Measured values are stored as compressed values.
- For the compression, a method is applied that records a value only if the slope change exceeds a certain hysteresis value.
- Therefore it could be possible that a value will not be recorded for a very long time. This could cause disadvantages in regard to data security.
- The user can force writing the value onto disk at least after a given period of time (in seconds, the default value is 3600 seconds (60 minutes)).

Entry:	[App]
Key:	DiffTime=3600

11.3.2 Trendmanager window

The Trendmanager has no operator interface in the standard operation, runs as process on the server and is displayed as an icon in the toolbar. The window cannot be opened directly.



Pressing the 'ALT‘ key and clicking on the icon at the same time will open the window, which lists all curve groups and their actual batch information:

```
The window can be opened automatically when the program starts by
means of the following settings in '<proj-path>\sys\trendman.ini'.
```

Input	[Window]
Key	Show=1

11.3.3 Where the data is located

Note

- As previously, the value description and pictures are in the project path defined via 'BM_PROJ'.
- The curve data is in the new trend path, which by default points to 'BM_ARCH'.
- The following files are required for recording batch curves:

'<proj-path>\trend\ini\measdesc.dbf'

Analog measured value description lists are stored into this file (dBase format).

'<proj-path>\trend\ini\digidesc.dbf'

Digital measured value description lists are stored within this file (dBase format).

'<proj-path>\trend\picture'

Configured pictures are stored in this sub folder, where three different file types exist (different archives):

- *.st (short time) Short-time archive
- *.ltb (long time batch) batch archive
- *.ltw (long time week) weekly archive

.

'<trend-path>\trend\data.xxx'

The recorded values are stored into this directory. In order to reduce the amount of files per sub folder, the system creates a sub folder for every 50 measured values, e.g. the sub folder “data.050” contains records of the measured values # 1 up to # 50.

11.4 Measured value description list

11.4.1 General

Basics

- All measurement values to be recorded must be configured in a measured value description list.
- In principle, there is a distinction between analog measured values and their derived digital values. Each of these groups is saved in a separate dBase file. These two lists are combined via the application "Measured value description list editor" (MWBL editor) on one screen.
- Multiclient function: The application has an area selection dialog.
- Changes are now always saved automatically to both servers of an area!
  Upgrading: If previously 2 different MWBL files (measdesc.dbf, digidesc.dbf) existed on both servers of an area, a decision has to be made as to which are the 'Master' lists before editing with MeasEdit.exe!

Note

- A maximum of 1200 active, meaning to be recorded, measured values is possible.
- The measured value description list, however, can include a maximum of 9999 data sets. Measured value definition files can be standardized this way and then activated selectively.
- The default is 'active=1'. The measured value is only taken into account by the trend manager when the active flag is set and transferred with "Sistar_adp.exe" to the SQL DB host computer, if necessary.
- During upgrading, the new column is automatically inserted when the measured value description list is opened for the first time, and all existing measured values are set to 'active=1'.
- The number of configurable measured values and thus also the maximum measured values ID was increased to 9999. However, a maximum of 1200 active measured values is still possible. A corresponding note will appear in the MWBL editor when the maximum number of active measured values has been reached.
- The number of configured and active measured values is displayed permanently in the status bar of the editor.

Multiclient function:

Note the following points when operating a client in a multiclient configuration:

- When "Opening" the area, to whose standard server the application connects, is determined via the area selection dialog.
- When "Saving", the measurement value description lists are redundantly saved on both servers. Any different content on the backup servers is overwritten.

Working with the application

- The call is made in the "Application Center" under Engineering / Trending definition
- The system can only record a limited number of scanning instants. Under the assumption that all scanning instances are saved, the following interrelation can be used to calculate the smallest possible sampling cycle:
  min. sampling cycle = max. recording time in seconds/number of points
- The practical high limit (with acceptable performance) of the number of scanning instants is approx. 10,000, although the theoretical high limit is a rate of approx. 1,000,000.

Note

To apply the deltas in a measured value description list, you need to restart the TREND-MANAGER.

11.4.2 Analog values

11.4.2.1 Attributes

Basic principle

An analog measured value is characterized by the following attributes:

Active

For activating/deactivating the measured value.

A maximum of 1200 active, meaning to be recorded, measured values is possible per AREA (displayed in status bar and monitored by the system).

ID (Identification)

System-wide unique identifier of the measured value.

The derived digital measured values and the curves based on this measured value are directly dependent on this.

Permitted number of entries: 1 to 9999.

Measured value description

The name of the measured value and the caption for the curve.
The maximum length of the name is 32 characters and it may not contain any special characters.

Addressing

Analog measured values can be addresses in two ways:

1. By direct definition of an absolute address (in the editor: "Absolute")
2. By an attribute of a system (or user) object (in the editor: "System")

PCU

The measured value must originate from a controller. Here, the number of the PCU is expected as assigned in the system control.

Class

Block class of the measured value to be recorded (only for "System" source variant)

Instance

Instance number of the measured value to be recorded (only for "System" source variant)

Attribute

Attribute of the measured value to be recorded (only for "System" source variant)

Absolute address

Relevant for addressing type “Absolute” only:

- Absolute address of the measured value to be recorded in STEP 7 syntax (e.g. DB712.DBW18)
- Measured values from the input, output and flag areas in STL syntax (e.g. I35.7, M99.1, MW 132, QB22)

Archive type

Every analog measured value has to be assigned one of these four possible archive types:

- Short-time archive
- Batch archive
- Weekly archive
- Manual archive
  → Only display for MWBL from previous version, no support for manual data archives from Version V7

Curve group

Number of the curve group to which this value should belong.

Refresh

Recording cycle, that is timing interval between two scans in seconds. Valid range is 2 to 32767 seconds. The real cycle depends on the hysteresis and the slope of the curve.

Lower limit

Minimum value of the scaling in the picture.

Upper limit

Maximum value of the scaling in the picture.

Hysteresis

For long-term curves: Tolerance band where a value is stored if it is not within this range.

The hysteresis is not used for short-time archives.

Unit

Unit of the measured value (e.g. °C, min, …)

This appears below the measured value name in the form of a legend in the curve picture.

Decimal point

Number of digits for the axis marking.

11.4.2.2 Data types and limits

Note

For data types and limit values there are several unusual features to take into consideration:

For historic reasons of downward compatibility, we have restricted the field length of the high and low limit attributes to six characters. While 16-bit integer values do not represent a problem, the representation of a 32-bit integer or real value (DINT, REAL) would require a significantly higher number of characters.

With this restriction, the numeric ranges shown in the following tables apply to the usable limit values and therefore also to the measured values.

Addressing type "Absolute"	Interpretation	Lower limit	Upper limit
DBB, EB, AB, MB	unsigned 8-bit integer	0	255
DBW, EW, AW, MW	signed 16-bit integer	-32 768	32 767
DBD, ED, AD, MD	signed 32-bit integer	-2 147 483 648	2 147 483 647

Addressing type "System"	Interpretation	Lower limit	Upper limit
BYTE WORD DWORD	Bit pattern *1)	fixed, cannot be changed 16#00 16#0000 16#00000000	fixed, cannot be changed 16#FF 16#FFFF 16#FFFFFFFF
USINT	unsigned 8-bit integer	0	255
INT	signed 16-bit integer	-32 768	32 767
DINT	signed 32-bit integer	-2 147 483 648	2 147 483 647
REAL	32-bit floating-point number *2)	-2 147 483 648	2 147 483 647

Note

1. Measured values of the data type "bit pattern" (BYTE, WORD, DWORD) cannot be used in analog curves. In the measured value description list, they serve as "containers" for the digital measured values that need to be defined additionally.
2. System attributes of the REAL data type are converted to the DINT format and stored taking into account the decimal places in the "Decimal point" column. The measured values in the curve picture are always represented with these decimal places. The same also applies for the upper / lower limits; in other words, any decimal places entered are shortened when the value is applied. Only the value specified in the "Decimal point" column serves as the number of decimal places.

11.4.3 Digital values

In addition to analog values you can also record and display digital values. Digital measured values are not inserted directly, but are derived from **recorded analog** measured values.

- Up to **2400** digital measured values are available per curve server
- These can be derived from the individual bits or bit ranges of the assigned analog measured values (of which there can be up to **1200**)

A digital measured value is defined by the following attributes:

Associated analog value

The digital value is assigned to the analog measured value in the editor using ID 1 ... 1200. This is done by selecting it and calling the "Derive new digital measured value" function. This assignment is retained even if you change the ID of the linked analog value.

Measured value description

Name of the measured value as it will appear as a caption in the trend visualization. It may only be a maximum of 32 characters and must not contain any special characters.

Start bit

First bit within the measured value for a bit range that is going to be evaluated

End bit

Last bit within the measured value for a bit range that is going to be evaluated

If only one individual bit of a measured value is going to be evaluated, the start and end bit must be identical.

Combination type

If multiple bits of a measured value are going to be evaluated for the purpose of a digital measured value (start and end bit are different), the type of logic operation has to be defined. You can select either an AND or an OR logic operation.

If the start and end bits are identical, this attribute is irrelevant.

Note

The bit range that can be configured for digital measured values is always 0 to 15 (maximum), even if the underlying measured value is a double word (i.e. consists of 32 bits).

11.4.4 Handling the editor

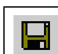
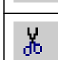
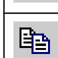
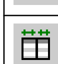
You can edit analog measured values and their derived digital values in the MVDL Editor. The user interface consists of two table views which are handled in accordance with the standard Windows features.

11.4.4.1 Elementary usage

Shortcuts

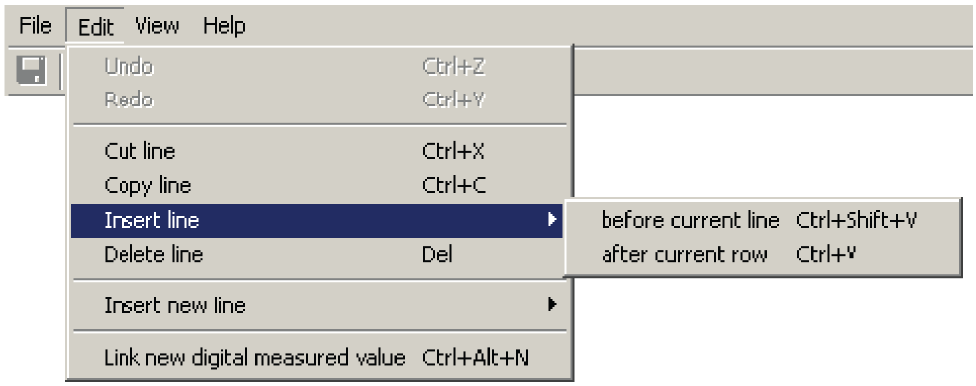
Program	
<Ctrl> S	Save trending definition
<Ctrl> P	Print analog or digital trending definition
<Ctrl> <Shift> P	Print preview for analog or digital trending definition
Edit	
<Ctrl> Z	Undo
<Ctrl> Y	Redo
<Ctrl> X	Cut current line
<Ctrl> C	Copy current line to clipboard
<Ctrl> V	Insert line from clipboard after current line
<Ctrl> <Shift> V	Insert line from clipboard before current line
<Del>	Delete current line
<Ctrl> N	Insert new line after current line
<Ctrl> <Shift> N	Insert new line before current line
<Ctrl> <Alt> N	Derive a new digital measured value from current analog measured value
Views	
<Ctrl> <TAB> <F6>	Toggle between views of analog and digital measured values

Toolbar buttons

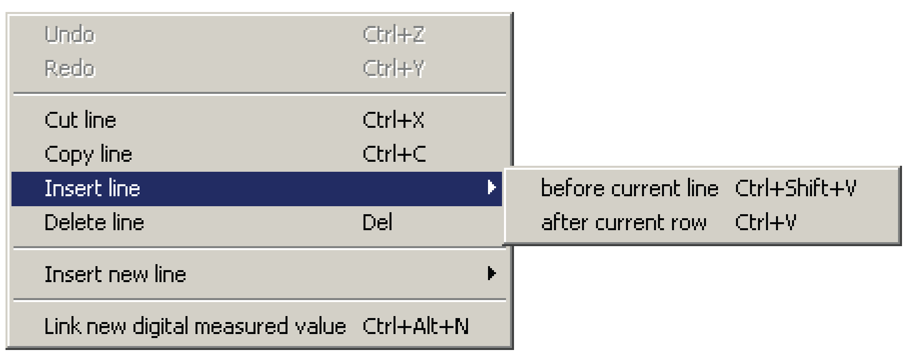
	Save trending definition
	Print analog or digital trending definition
	Print preview for analog or digital trending definition
	Cut current line
	Copy current line to clipboard
	Insert line from clipboard after current line
	Undo
	Redo
	Optimize column widths for current view
	Toggle: Show all digital measured values or only those which are derived from current analog measured values

11.4.4.2 Create new analog measured values

Alternative 1: the main menu



Alternative 2: By means of the shortcut menu in the "Analog values" view (right-click on the table view)



Alternative 3: keyboard shortcuts

As indicated above, the keyboard shortcut is a faster means of inserting a new definition:

- <Ctrl>+<Shift>+N New definition line before current line
- <Ctrl>+N New definition line after current line

11.4.4.3 Create new digital measured values

When creating digital measured values, you can also choose from three methods:

- the main menu
- the shortcut menu
- a keyboard shortcut (<Ctrl> <Alt> N).

Note

Digital measured values always have to be derived from analog measured values.

11.4.4.4 Editing measured values

Notes on the input of values

With the Editor, you have a highly efficient tool at hand for editing attribute values.

You can basically select any value for editing by means of the shortcut key <F2>, with <ENTER>, or by double-clicking with the left mouse button.

After you have edited a value, you can apply the changes and move to the next column by pressing the <TAB> key (does not apply to the pop-up selection boxes).

Editing possibilities

Table 11-1 Analog values

Column	Description
Active	Selection box for activating/deactivating the measured value.
ID	Unique identification. Input of an existing identification is rejected. <range 1...9999>
Measured value's desc.	Name of the measured value
Addressing	<F2>, <ENTER>, double-click show a selection box, shortcut: "s" for "System", "a" for "Absolute"
PCU	<F2>, <ENTER>, a double-click opens a selection box, Shortcut: PCU number
Class Instance Attribute	<F2>, <ENTER>, a double-click opens a selection box, Shortcut for an instance: instance number. (if you prefer, you can also enter the class and attribute in text format. The text is not case-sensitive.)
Absolute address	STEP 7-AWL-Syntax

Column	Description
Archive type	<F2>, <ENTER>, a double-click opens a selection box, Short input by means of the numbers known in previous system versions: • 1 (short-term archive) • 2 (batch archive) • 3 (weekly archive) • 4 (manual archive) → **Only display for MWBL from previous version, no support for manual data archives from Version V7**
Curve group Update Lower limit Upper limit Hysteresis Unit Decimal point	(no special considerations for the input)

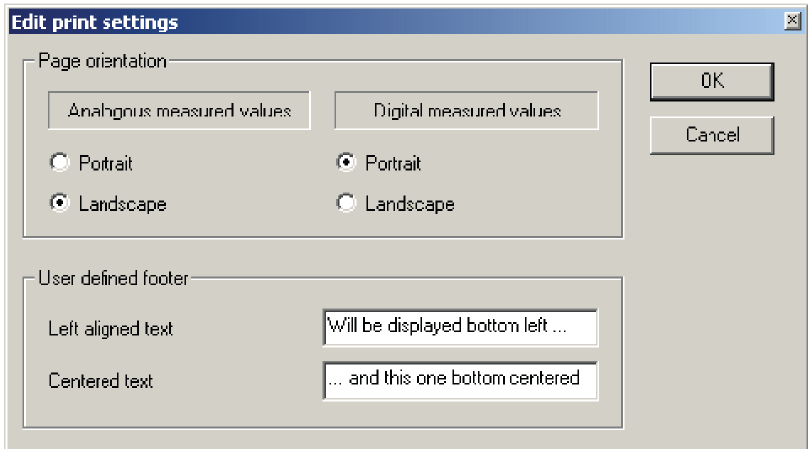
Table 11-2 Digital values

Column	Description
Linked analog value	(information only; cannot be edited)
Measured value's desc. Start bit End bit	(no special considerations for the input)
Logical connection	<F2>, <ENTER>, double-click show a selection box, shortcut: "a" for "AND", "o" for "OR"

Note

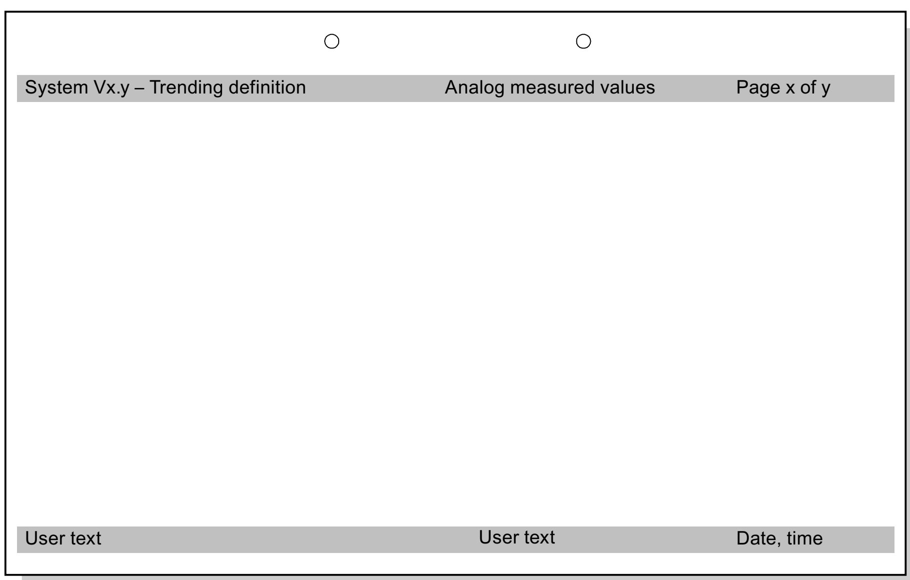
All clipboard operations (Cut, Copy, ...) and the "Delete" command always relate to a complete measurement value line.

11.4.4.5 Print settings



Here you can set some of the printer defaults: separate page layout for analog and digital values and an adaptable footer.

The figure below shows you the layout of a printed page (portrait pages are shown in analog format):



In the user text you can define the customer or the system, for example.

11.5 Curve visualization

11.5.1 Overview

Basics

The application 'batch curves' is used for a graphic output of measured values. It enables the time display of archived or running curve values as diagrams on the screen. Measurement value curves can be loaded from different area servers via an area selection dialog (Multiclient function).

Display of analog curves

You can assemble up to a maximum of eight curves. Each individual curve can be hidden. Furthermore the user can print a screenshot of the curves. The user can zoom into a graph and get any absolute value from the archive with the corresponding point of time additionally.

The scale, unit and archive maintenance of the curve values are defined in the measured value description list.

Display of digital properties

Digital curve properties of individual bits are displayed as follows:

- Condition "0" (graph with gaps)
- Condition "1" (colored graph)

Display

The following picture combinations are possible:

- Picture with analog curves only: up to 8 analog graphs
- Picture with digital curves only: up to 32 digital graphs
- Mixed picture: up to 8 analog and up to 16 digital graphs

Trendmanager

The curve visualization requires a connection to the area server's Trendmanager.

It provides access to the curve archives, which are configured in the measured values description list. Graphs showing currently running batches are updated cyclically.

Location of the curve pictures

The previous configuration switch 'SynchonizeFiles' no longer applies. The curve diagrams are always saved to both area servers (SynchonizeFiles=1)!

11.5.2 Creating pictures

Three types of pictures are possible

- Pictures consisting of values from the 'short-term archive‘.
- Pictures consisting of values from the 'weekly archive‘.
- Pictures consisting of values from the 'batch archive‘.

Create a new picture by selecting “file/new” from the menu, a dialog appears, where the user can enter the name of the picture and select a curve type.

- The user can select the short-term archive, batch archive or weekly archive.
- After selecting an archive, first a dialog appears "edit analog measured values", then "edit digital measured values".

11.5.3 Dialog 'edit analog measured values'

The dialog opens automatically when creating new pictures. If a picture is already open, select 'File' → 'Edit' → 'analog measured values' in order to open this dialog. In this dialog, the user defines the values to be displayed

- First of all, the IOS, value group (* = all) and PCU are selected.
- The measurement values to be displayed are selected via the two list boxes "Measurement value" and "Measurement values in the diagram" at most 8 values per picture can be selected.
- In addition one out of **16 colors** can be chosen,
- Pressing the "**>>**" button assigns the selected value to the current picture.
- Pressing the "<<" button deletes the selected value from the current picture in the list box measured value from the current picture.
- Only those values which could be added to a picture (analog/digital values) are shown.

11.5.4 Dialog 'edit digital measured values'

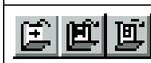
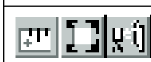
The dialog opens automatically when creating new pictures. If a picture is already open, select 'File' → 'Edit' → 'digital measured values' in order to open this dialog. In this dialog the user defines the digital values of a picture

The function is similar to the "Editing analog values" dialog.

- First of all, the IOS is selected.
- The measurement values to be displayed are selected via the two list boxes "Measurement value" and "Measurement values in the diagram" Up to 16/32 values per picture are possible.
- In addition, the **color display** of the measurement value is selected from one of 16 colors.
- Pressing the "**>>**" button assigns the selected value to the current picture.

- Pressing the "**<<**" button deletes the selected value from the current picture in the list box measured value from the current picture.
- Only those values which could be added to a picture (analog/digital values) are shown.

11.5.5 Meaning of the tool bar

Icon	Topic	Function
	Standard buttons	• Exiting the application • Quitting ICM errors (only necessary for "BRAUMAT based on S7") • Quitting horn • Resetting password
	Open pictures	• Open pictures based on the short-term archive • Open pictures based on the weekly archive • Open pictures based on the batch archive
	Print a picture	
	Show/hide additional information	• Ruler • Show complete graphs • Show all measured values (show hidden values)

11.5.6 Open pictures

11.5.6.1 Open pictures

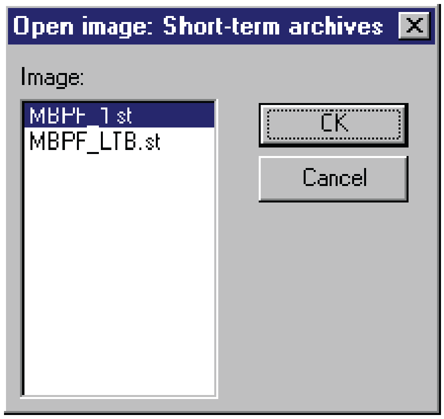
Possibilities:

- Select “File/Open” and click on the desired curve type in the corresponding sub menu.
- Click on the icon in the button bar.

A dialog appears, select a picture.

Depending on the selected curve type (short-term, batch, weekly archive) more dialogs appear.

11.5.6.2 Open a short-term archive



For a short-term archive no additional selection is necessary.

11.5.6.3 Open a batch archive

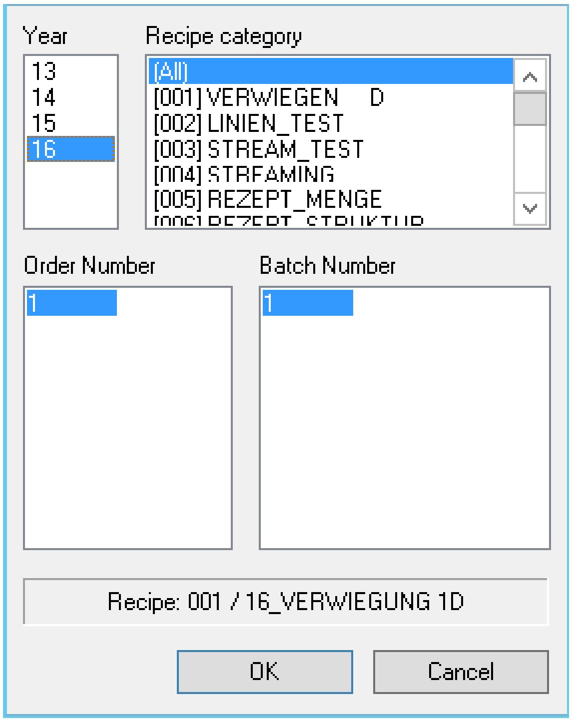
Via 'Batch selection' the area of the corresponding curve archive is selected by the selection of year, job number and batch number.

Batch archives provide the following possibilities:

- Standard view
  The measured values of the selected image are related to one batch, the user has to select the batch ID in the second step.
- Free selection
  In this way it is possible to take the measured values of the selected image from different batches. In a second step up to 8 batches are selected for the curves. The selection is measured value-specific. This selection is used as a reference for the selection 'compare curves'.
- Compare curves
  With this option the image curves of a freely selectable batch can be set in relation to an already selected reference arrangement. Here it is also necessary to select a batch (ID) in a second step.

11.5.6.4 Selecting batches

This dialog is used for the "Normal view" and the "Reference curve" option.



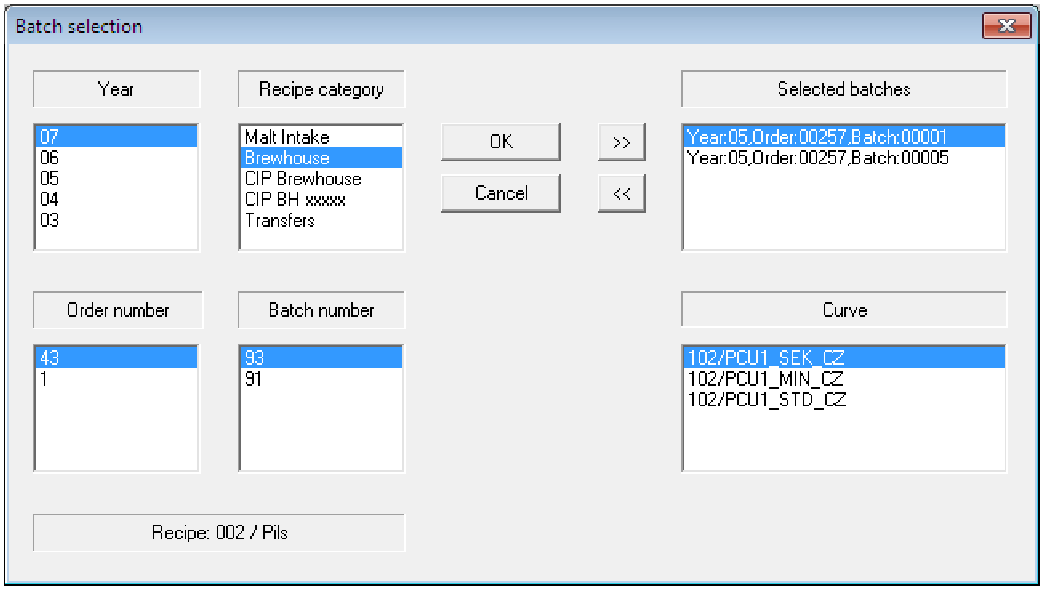
- When changing batch, only the batches for which the data of at least one measured value has been recorded are displayed.
- A batch is selected by choosing the year/recipe category/order number/batch number

---

Note

The number of displayed batch datasets is limited internally to 3248 entries.

11.5.6.5 Batches “Free selection”



In this dialog the user can select graphs of different batches,

which are used later as a reference for the option “reference curves”.

Step 1: Curve

This list contains the configured curves of the earlier selected picture. Adding or removing takes effect on the selected measured value.

Step 2: Year/Recipe category/Job number/Batch number

Use these fields to select a batch for a curve.

Step 3: Button '>>'

Add batch to the selected curve.

Step 3: Button '<<'

Remove batch from selected curve.

Selected batches

This list shows the selected batch or batches of the selected curve.

11.5.6.6 Open weekly archive

For weekly archives, the user can select between:

- Normal view
- Free selection
- Compare curves

Please refer to “batch archives” for function description and selection sequence. Weeks are compared instead of batches here. After selecting an image for weekly archives (*.ltw), the week has to be selected in the next dialog.

11.5.6.7 Selecting the week

Weekly selection

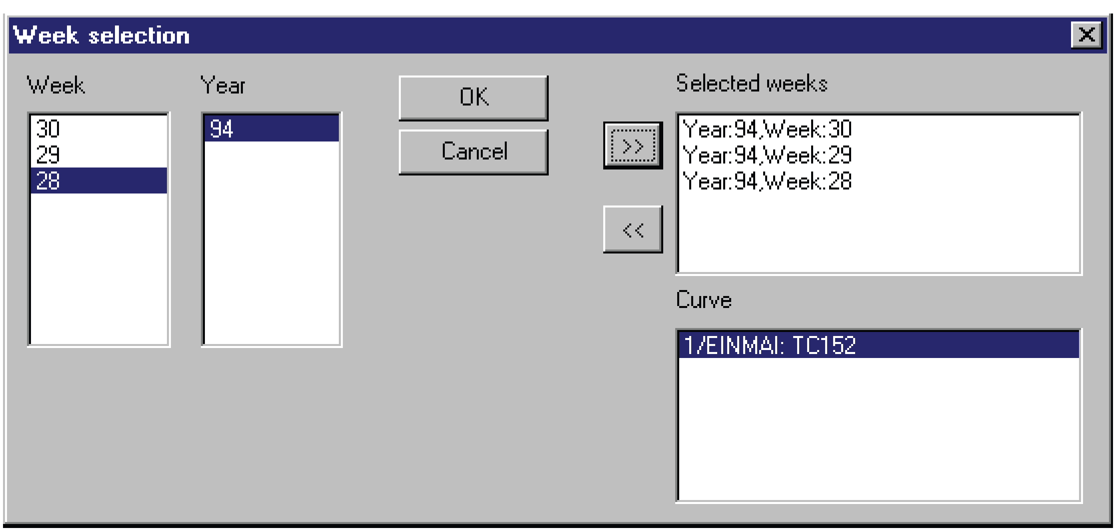
The dialog is used for the 'normal view' as well as for the option “reference curve”.

Week/Year

Only those weeks are displayed for which data has been recorded for at least 1 measured value.

11.5.6.8 'Week' selection for ‘Free selection’

After selecting the picture, the following dialog appears and the user can choose the week archives for the curves



Step 1: Selecting the curve

This list contains the measured values that have been configured for the picture. The buttons '<<‘ and '>>‘ and the 'Selected weeks‘ list are related to the selected curve.

Step 2: Selecting the week/year

Select the week and a year. Only those weeks are shown for which data has been recorded for at least one measured value.

Step 3: Button '>>'

This button adds a selected archive (list 'year' and 'week') to a list of weeks of a selected measured value.

Step 3: Button '<<'

This button removes a selected week from a list of weeks of a selected measured value.

Selected weeks

This list shows only those weeks related to the value, selected in the list “curve”.

11.5.7 Printing pictures

Basic principle

An existing picture can be printed out by selecting "Print" from the “File” menu or pressing the button "Print" on the button bar.

The following dialog appears after the "Print" function is selected:

Selection

- All
  prints all digital and analog curves of a picture
- Analog curves
  prints all analog curves of a picture
- Digital curves
  prints all digital curves of a picture

Background

- White
  The background color of the picture is white
- Normal
  The background color of the picture corresponds to the color of the screen (Windows settings)

Part of the picture

- Current sub picture
  prints the actual selected part of a picture
- All
  prints all curves that are displayed on the screen (full time axis and y-axis).

Button "configuration"

After pressing the button, the Windows standard dialog "Print setup" appears.,

The printers installed on the PC and the associated print options can be set using this dialog.

11.5.8 Hide/Unhide trend curves

11.5.8.1 Hide curves

Curves shown in a picture can be hidden or shown by pressing the button with name of the measured value.

The scale is being hidden according to its curve.

11.5.8.2 Unhide curves

Pressing the button again shows the curve. By pressing the 'Show all analog curves' button

or by selecting 'Functions → All curves’ in the menu, all hidden curves are visible again.

11.5.8.3 Unhide only one curve

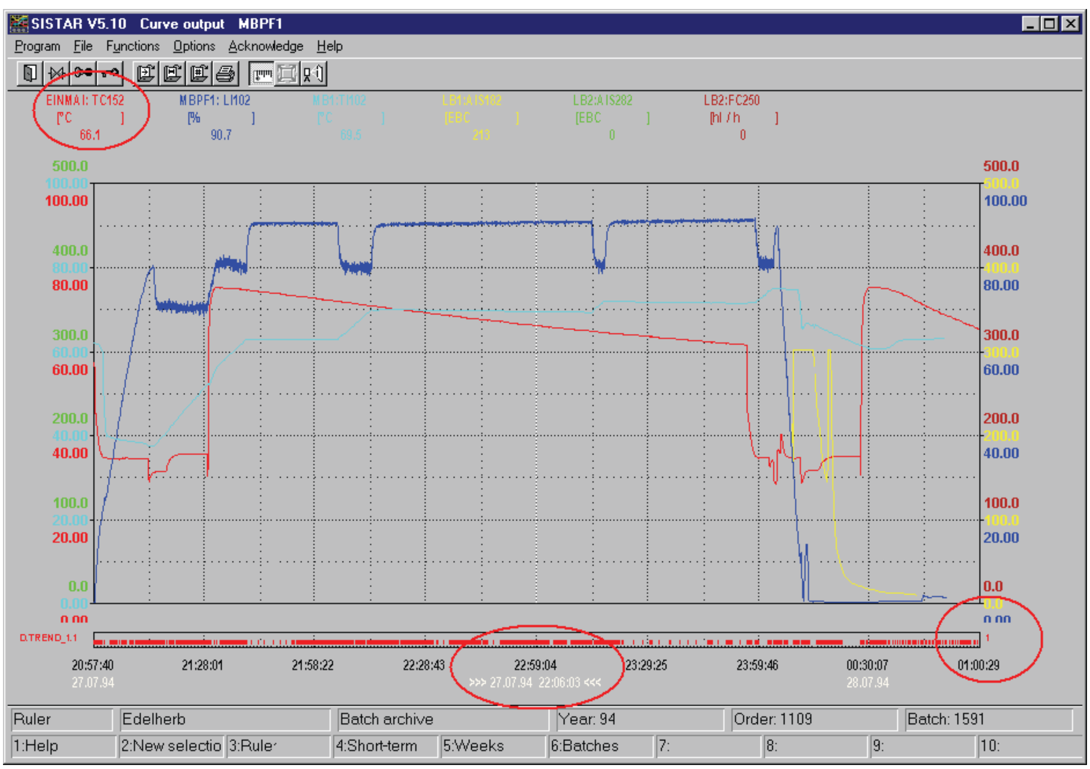
By clicking on a single curve within the picture (left mouse button), all other graphs are hidden and only the selected curve will be displayed.

11.5.8.4 Hide/Unhide scaling

The scaling can be unhidden and hidden by pressing the button that is labeled with the dimension.

11.5.9 Ruler

11.5.9.1 Ruler

<table>
<tr><td></td><td>Button of the toolbar</td></tr>
<tr><td colspan="2">Select ‘Functions/marker’ or press the F3 function key or click on button “marker” on the symbol bar, the picture shows a white vertical line, called "marker line".</td></tr>
<tr><td></td><td>Showing the values</td></tr>
<tr><td colspan="2">In the upper part of the picture the absolute values of the graphs (crossing point with the marker line) together with its time base are shown.
For digital values the state value (0 or 1) is given on the right side of the digital bar.</td></tr>
<tr><td></td><td>Move marker line</td></tr>
<tr><td colspan="2">The marker can be moved horizontally over all graphs with the cursor while pressing the left mouse or trackball button. The time and y-axis value are updated automatically. By pressing F3 again or clicking the button “marker line” of the button bar the application switches back to the normal mode.

</td></tr>
</table>

11.5.9.2 Difference between two marker lines

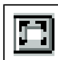
Select "difference marker lines" from the menu “Functions” or via Shift + F3, provided the function "marker line" has been activated.

The picture shows a second vertical, white marker line, called "difference marker line". Within the upper part of the picture the difference between the curves (cross point of marker lines with a curve) between marker line 1 and 2 is shown. The marker can be moved horizontally over all graphs with the cursor while pressing the left mouse or trackball button. The difference and time values are updated automatically. Pressing Shift+F3 again resets the application to normal mode.

11.5.9.3 Zooming

Zooming into parts of a picture can be performed by pressing the left mouse button and drawing a rectangle within the picture (upper left to lower right corner). A white outlined rectangle marks the selected area, and after releasing the left mouse button, the new zoomed in part will be displayed.

11.5.9.4 Rezoom

	**Button of the toolbar**
Select "Rezoom" from the menu “Functions” or press Shift + F5 or click the button "Rezoom" on the button bar. The scales are reset to their original values (full scale) and revert to normal format.	

11.5.9.5 Redraw curves

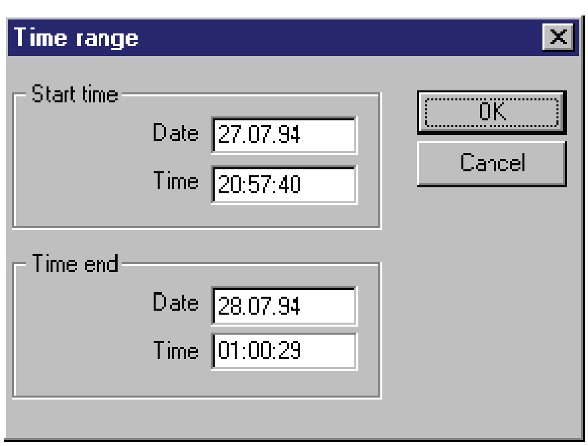
Selecting "Functions/Redraw" redraws all curves of a picture, often used for reference curves. Moving graphs sometimes causes parts of curves to be invisible.

11.5.9.6 Select new week or batch

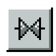
Selecting "functions/new" or pressing F2 depending on the last opened archive type opens the dialog **batch archive** or **weekly archive**.

11.5.9.7 Display range

By selecting "functions/range", the user can select a range of time.



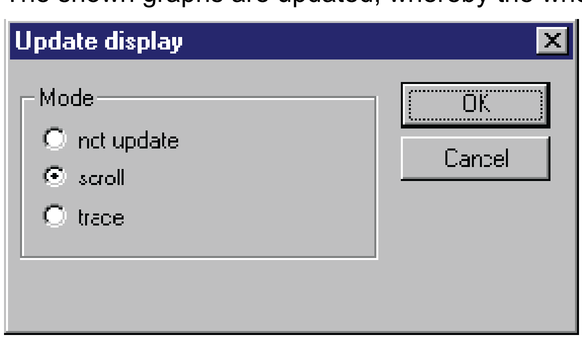
11.5.9.8 Standard functions

• Button bar	
Selecting "options/button bar" unhides and hides the button bar.	
• Status bar	
Selecting "options/status line" unhides or hides the status bar.	
• Function keys	
Selecting "options/function keys" unhides or hides the meaning of the functions keys.	
• Standard size	
Selecting "options/standard size" unhides the whole picture on the screen.	
• Quit ICM	
Selecting "quit/ICM error" or pressing the button "quit all ICM errors" on the button bar or pressing F11 quits all ICM errors.	
• Quitting horn	
By selecting the function "Acknowledge horn" or by clicking the icon "Acknowledge the signal horn" in the button bar or by activating the function key F11 the signal horn can be acknowledged.	
• Resetting password	
By selecting the function "Reset password" or by clicking the icon "Reset validity of the password" in the button bar the validity of the password is reset.	

11.5.9.9 Changing the display mode

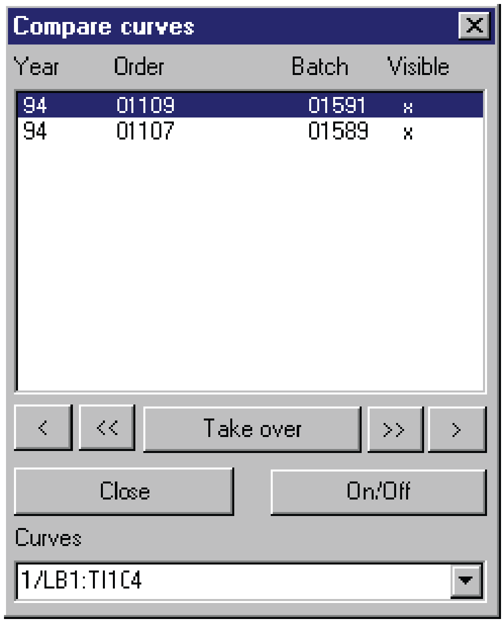
Selecting "options/viewing mode" opens a dialog **update view** with the following options:

- Don’t update
  The shown graphs are not updated.
- Scroll
  The shown graphs are updated, whereby only a part of the time scale (and the curve) is visible.
- Trace
  The shown graphs are updated, whereby the whole time scale (and the curve) is visible.



11.5.9.10 Compare curves

The following dialog is called via menu ‘Options → Compare curves’



This window can be opened as an additional (modeless) dialog on top of the curve window and is used to select certain curves, whereby the selected curve is shown in black color.

A selected curve can

- be moved (time) left by pressing '<‘ and '<<‘.
- be moved (time) right by pressing '>‘ and '>>‘.
- be left in its current position by ‘take over’ and affects only the view not the archive file.
- be switched on (visible) or off (hidden).

11.5.9.11 Help

Menu input “Help” offers assistance for:

• System	ALT+F1
• System index	
• Curves	F1
• Curves index	
• Information about the application (version).	

11.5.10 Examples of comparison curves

The system provides the option of comparing saved curves of a curve picture.

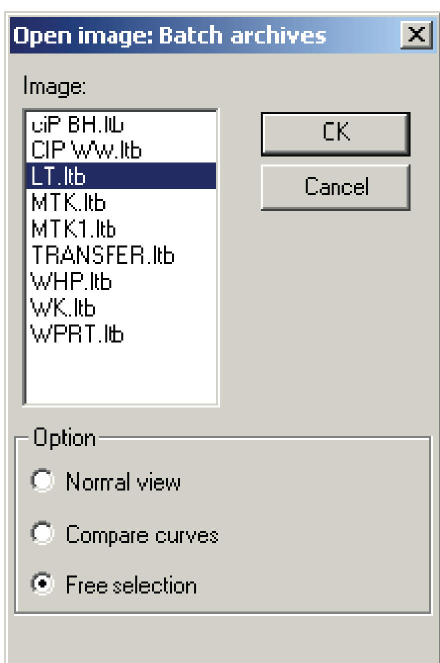
For weekly curves, the curves of different calendar weeks can be compared with one another; for batch curves, the curves of different batches can be compared with one another.

An example of the selection of batch trends for comparison is provided below. Weekly trends are selected analogously, except that for weekly trends, "Year/Calendar week" is used instead of "Year/Order number/Batch number".

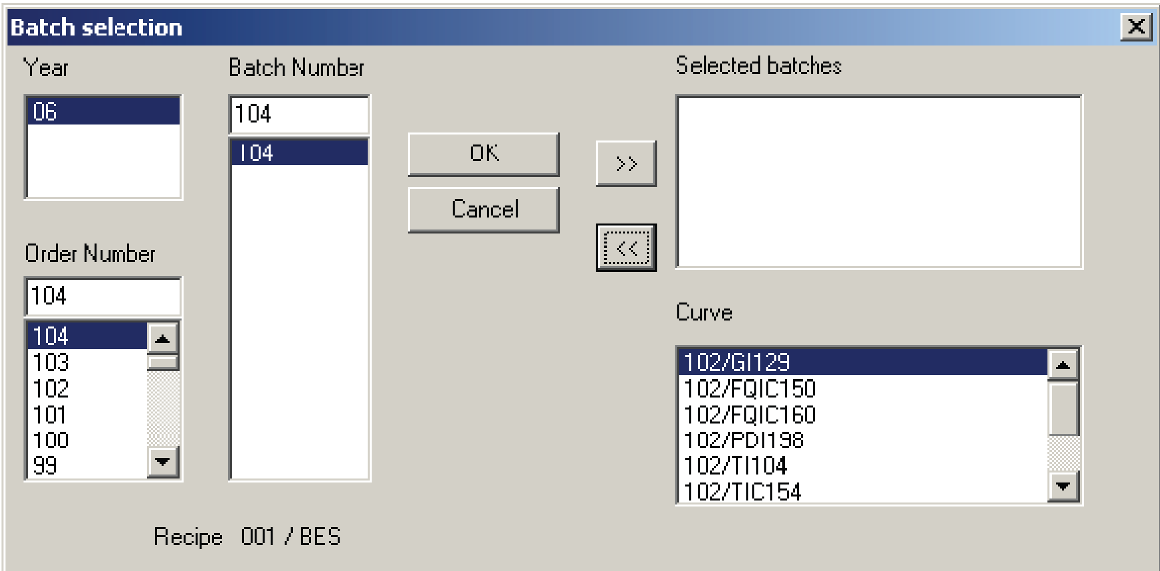
For selection of the trends for comparison, a distinction is made between two options: "Free selection" and "Compare curves".

- Use the "Free selection" option to display and compare individual curve shapes (of an individual measured value) with recorded curve shapes of the same measured value for different batches or weeks in one curve display. It is possible to compare up to 8 measured values of different batches simultaneously.
- Use the "Compare curves" option to compare defined reference curves with all curves of a curve display of a different batch (or calendar week in the case of weekly curves) using the "Free selection" option. An example is shown below.

11.5.10.1 Curve comparison with the "Free selection" option

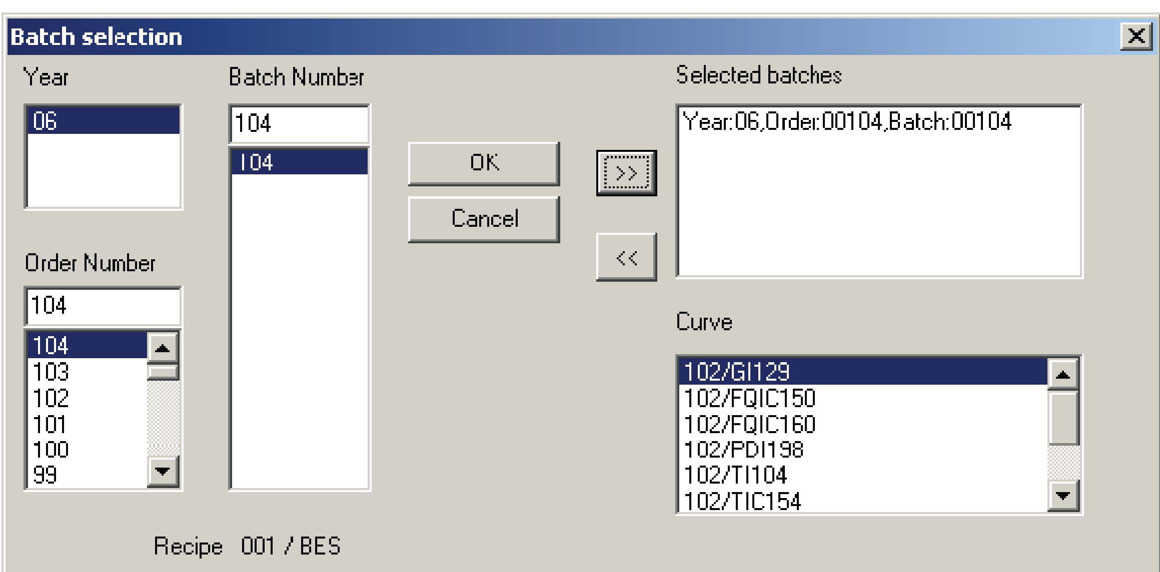


Select "batch curves" and "image name" (here LT.ltb) and the "Free selection" option to open the "Batch selection" window, which enables the normal selection of "Year/Order number/Batch number" on the left, the selection of the curves of this curve image at the bottom right, and the selection of "Selected batches" at the top right. In the following, no batches have been selected yet for the "GI129" measurement.

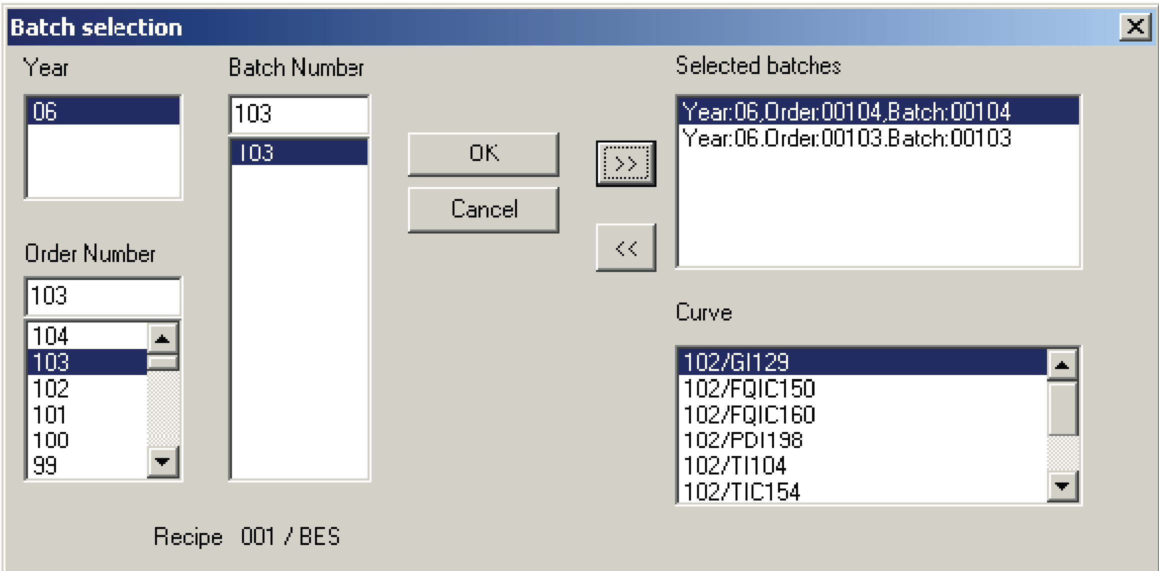


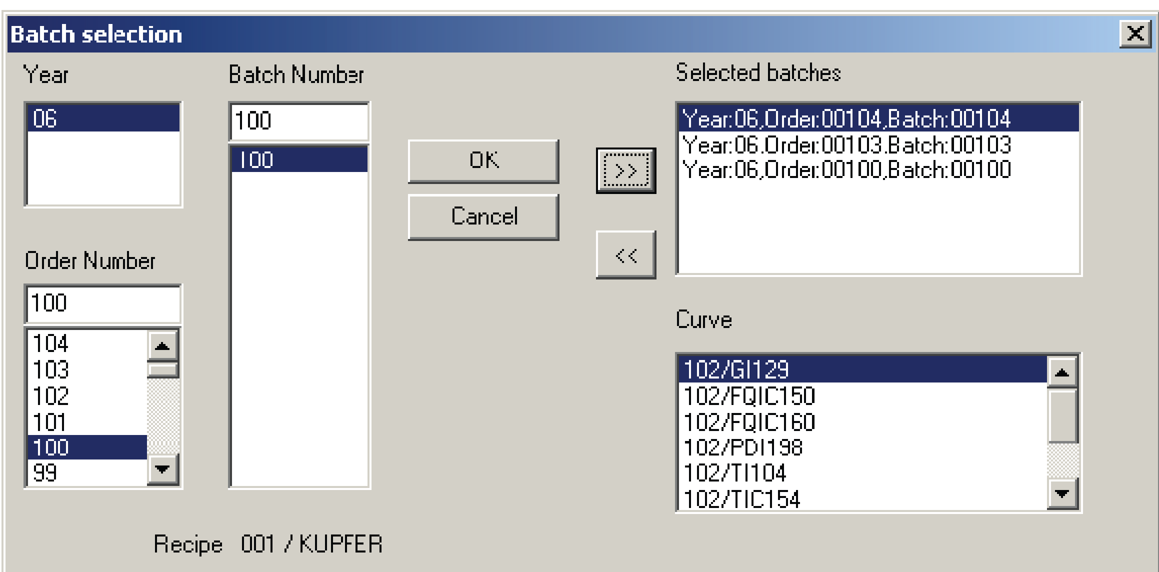
Batches are selected separately for each curve (measured value in lower right field). In the example, the measured value GI129 is selected here, along with the Year 2006, Order number 104, and Batch number 104 on the left. Clicking the ">>" button causes the selection to be displayed in the top right field (the selection also works by double-clicking on year, order number, or batch number in the left selection area).

As shown in the figure, the measured value GI129 of the server IOS102 is selected from the year 2006, order 104, and batch 104.

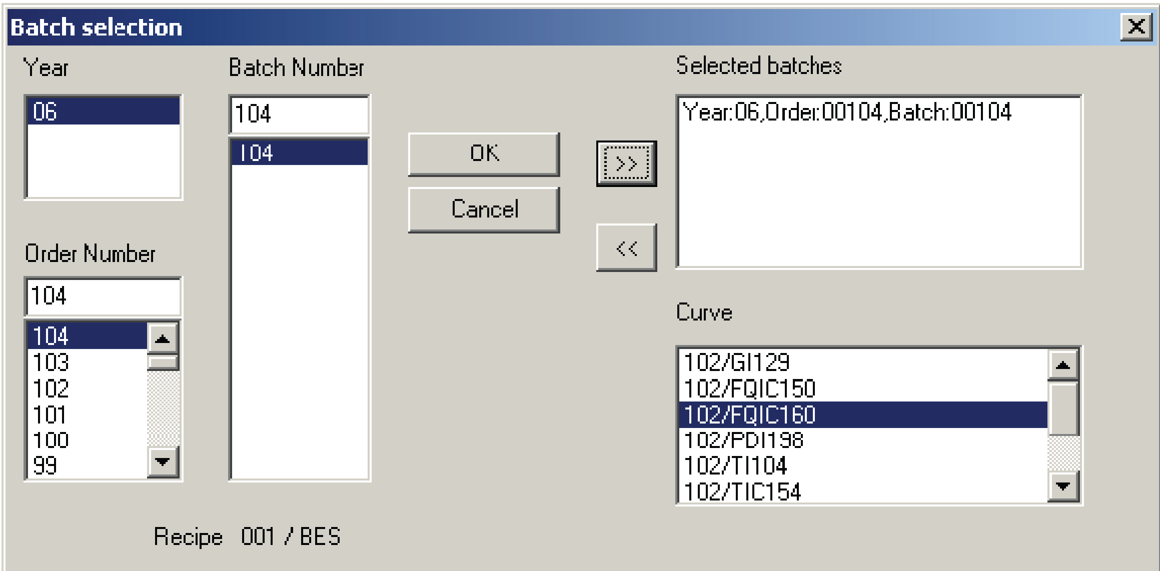


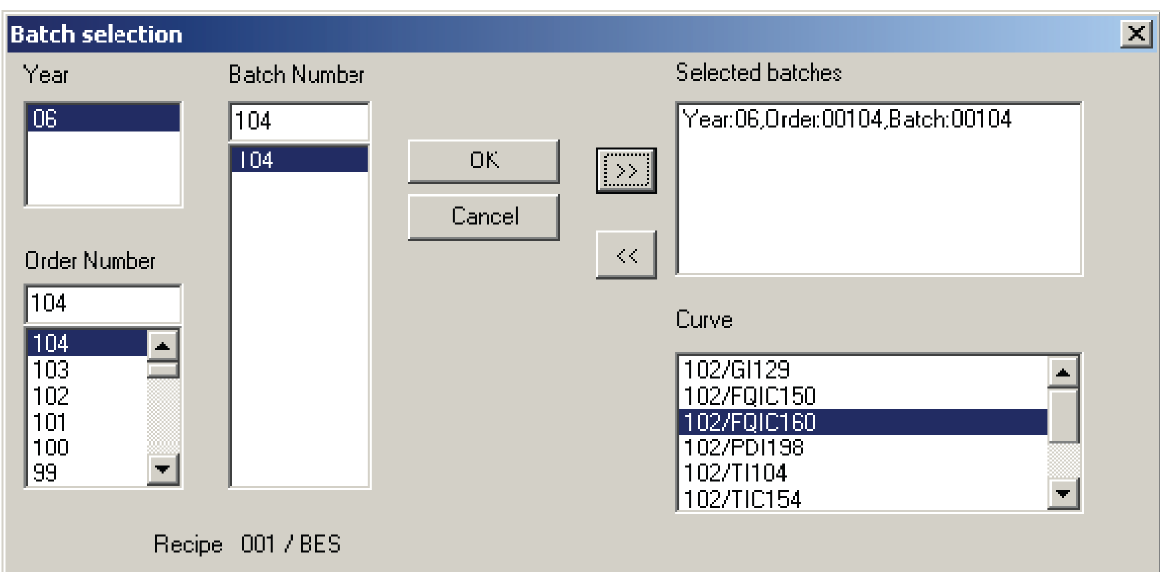
By selecting multiple batches on the left, it is possible to select up to 8 candidates for a measured value, which are then displayed together in a curve window.

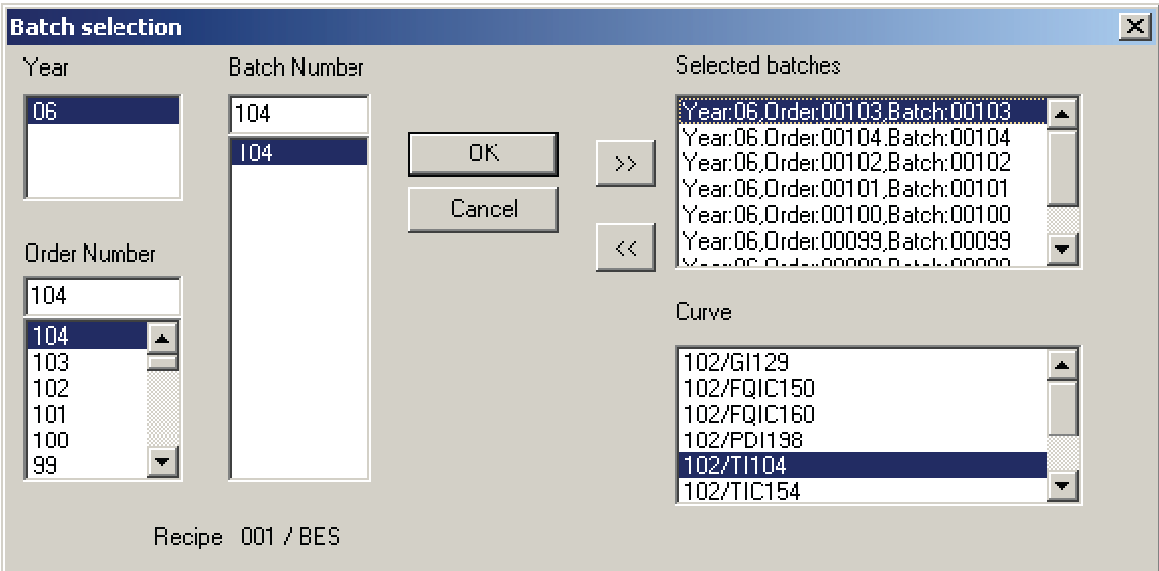




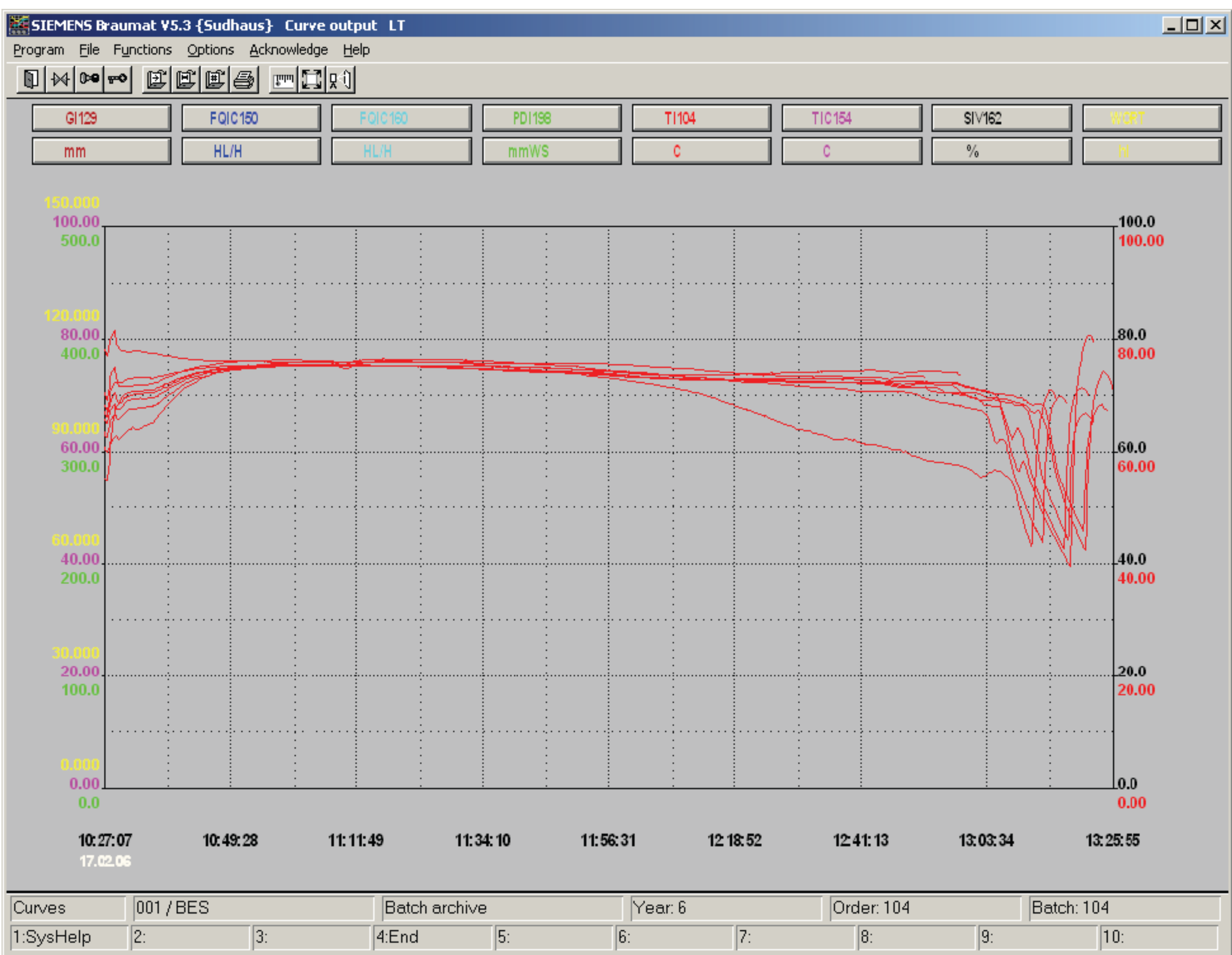
Select additional curves (measured values) in the same way if you want to display a common representation of various measured values in one curve window later. The selection of the selected batches of other curves will be retained.


Batch selection
Year
06
Batch Number
104
104
OK
Cancel
>>
<<
Selected batches
Year:06,Order:00104,Batch:00104
Order Number
104
104
103
102
101
100
99
Curve
102/GI129
102/FQIC150
102/FQIC160
102/PDI198
102/TI104
102/TIC154
Recipe 001 / BES

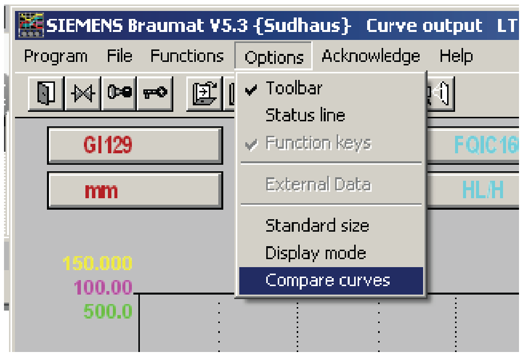

Batch selection
Year
06
Batch Number
104
104
OK
Cancel
>>
<<
Selected batches
Year:06,Order:00104,Batch:00104
Order Number
104
104
103
102
101
100
99
Curve
102/GI129
102/FQIC150
102/FQIC160
102/PDI198
102/TI104
102/TIC154
Recipe 001 / BES



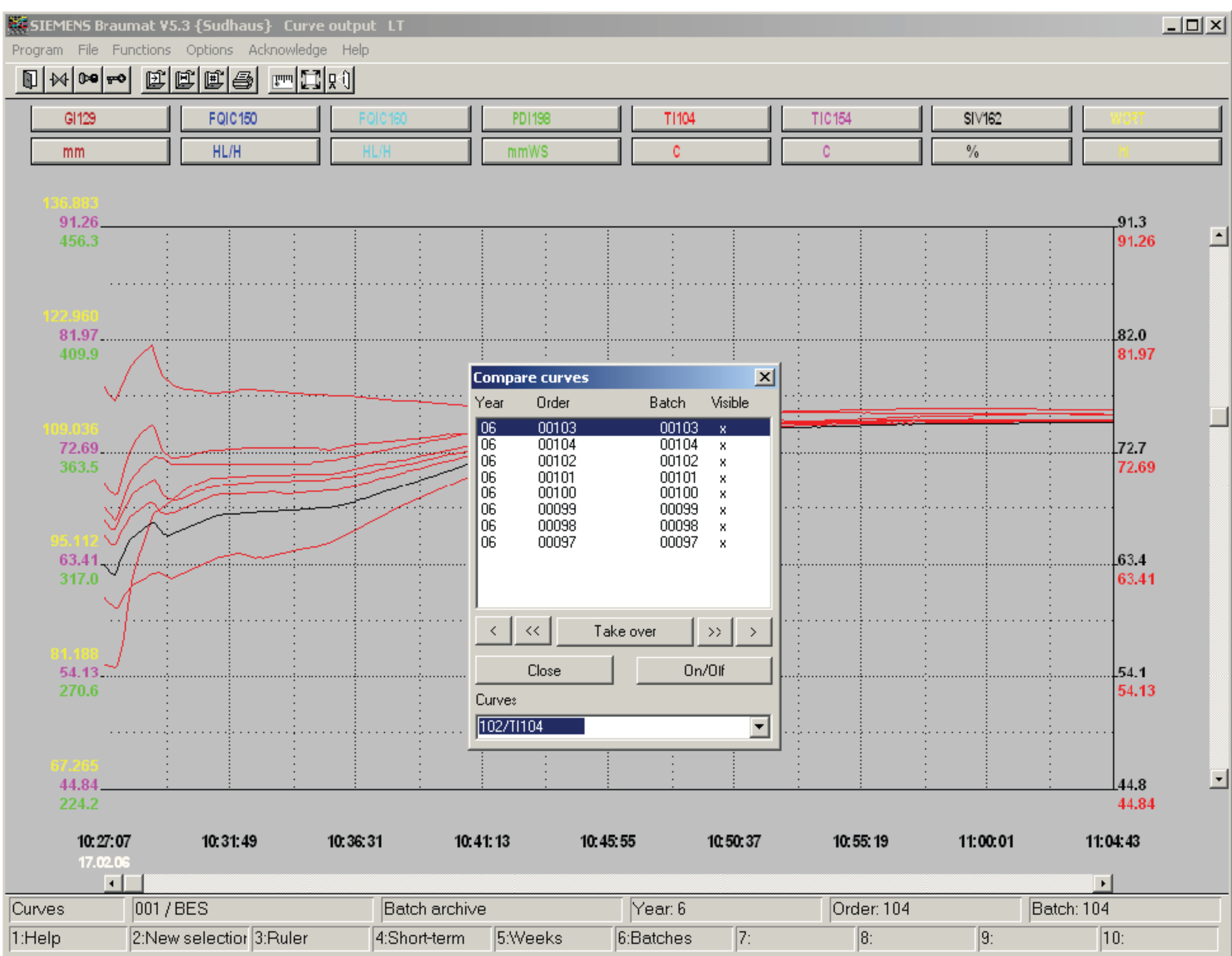
Press OK to open the following curve display:



When comparison curves are open, you access the operating window for the comparison curves with "Options - Compare curves".

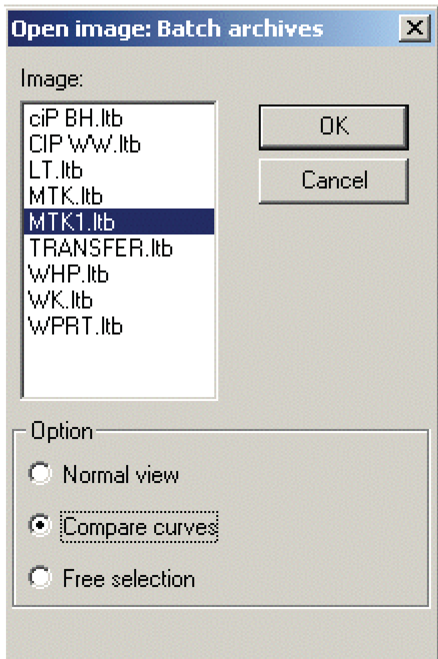


In the Compare curves selection window, the following is possible: 1) after selecting the "Curve" via the drop-down menu at the very bottom, individual curves can be selected in the list field (the blue bar shows the selected batch), which then turn black; 2) curves can be displayed and hidden ("ON/OFF" button), which is indicated with an X in the "Visible" column; 3) curves can be shifted over the time axis using the arrow or double-arrow buttons (small or large increments, respectively, to left or right).

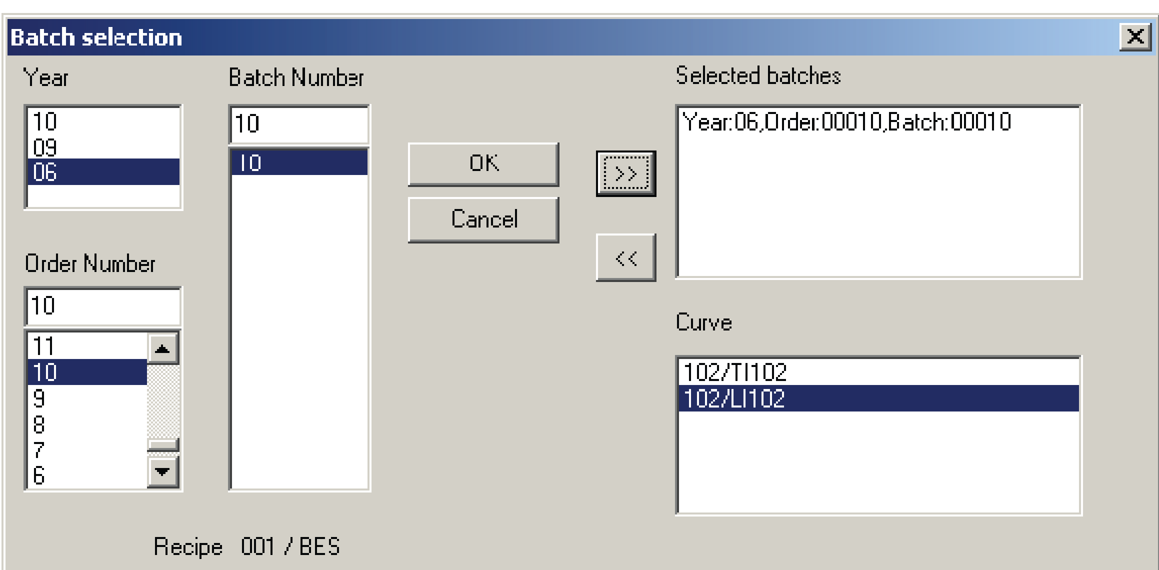


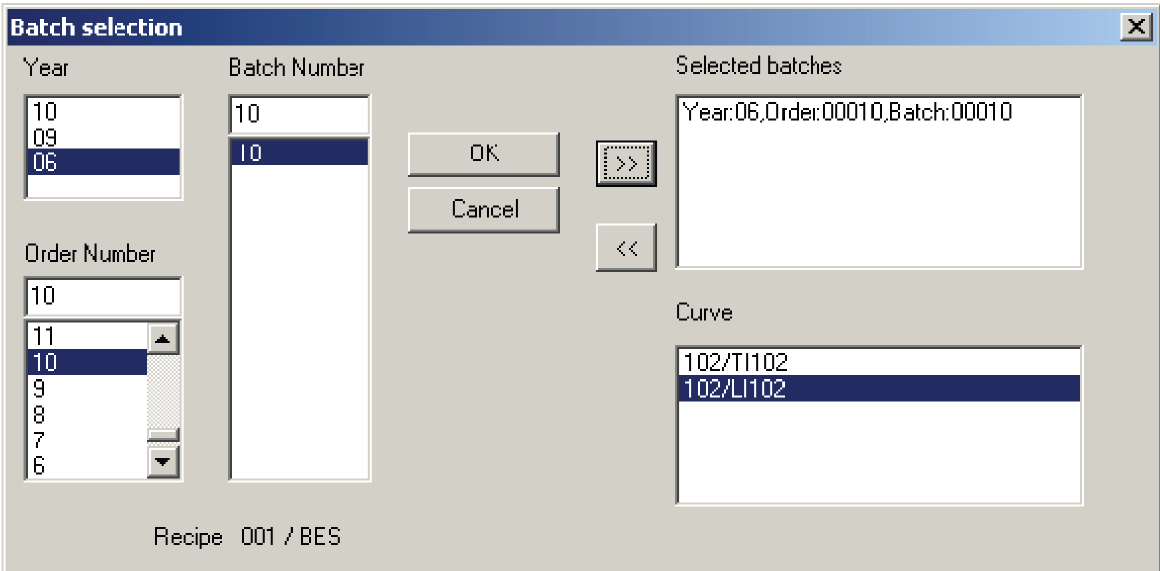
11.5.10.2 Curve comparison with the "Compare curves" option

To use the "Compare curves" option, the reference curves must first be defined for the comparison using the "Free selection" option. This takes place as described above, with curve image "MTK1" in this case.

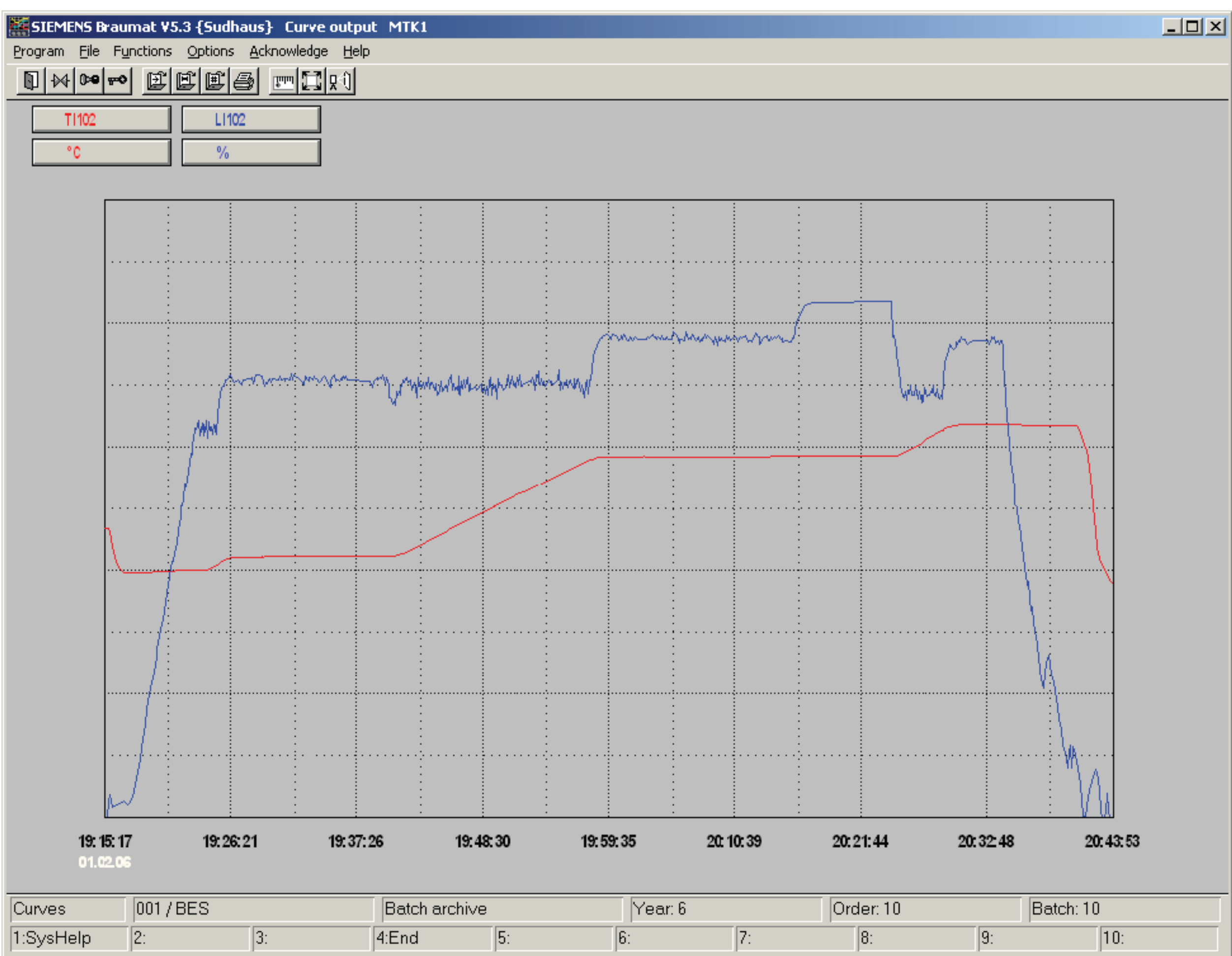


The "TI102" and "LI102" curves of year 2006 with order/batch number 10 are defined as references.

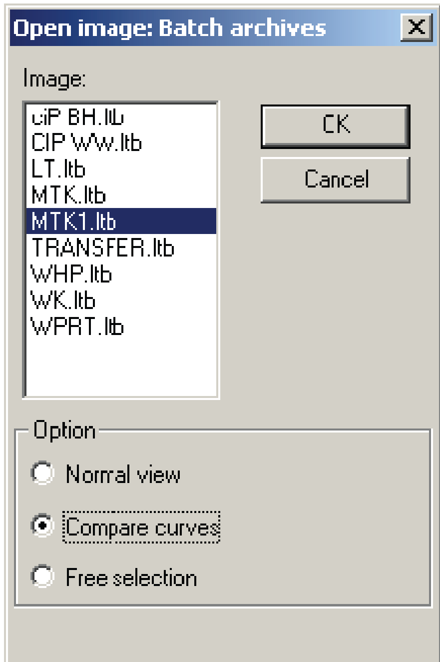




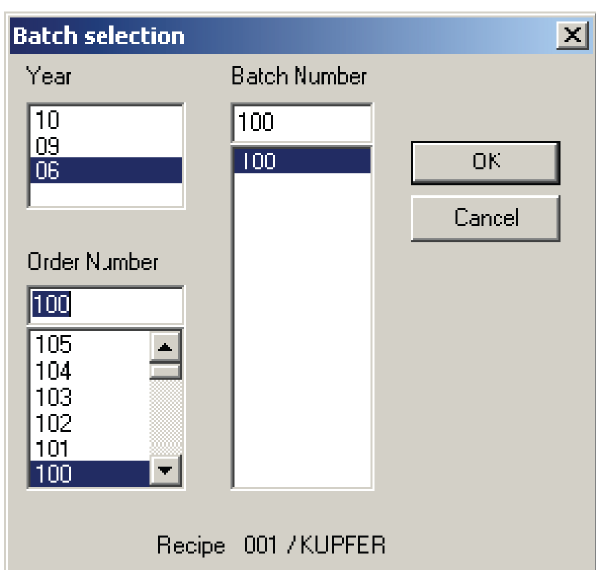
After confirmation with "OK", these two curve shapes will be displayed.



The comparison curve is selected using the "File->Open->Batch archives" function:

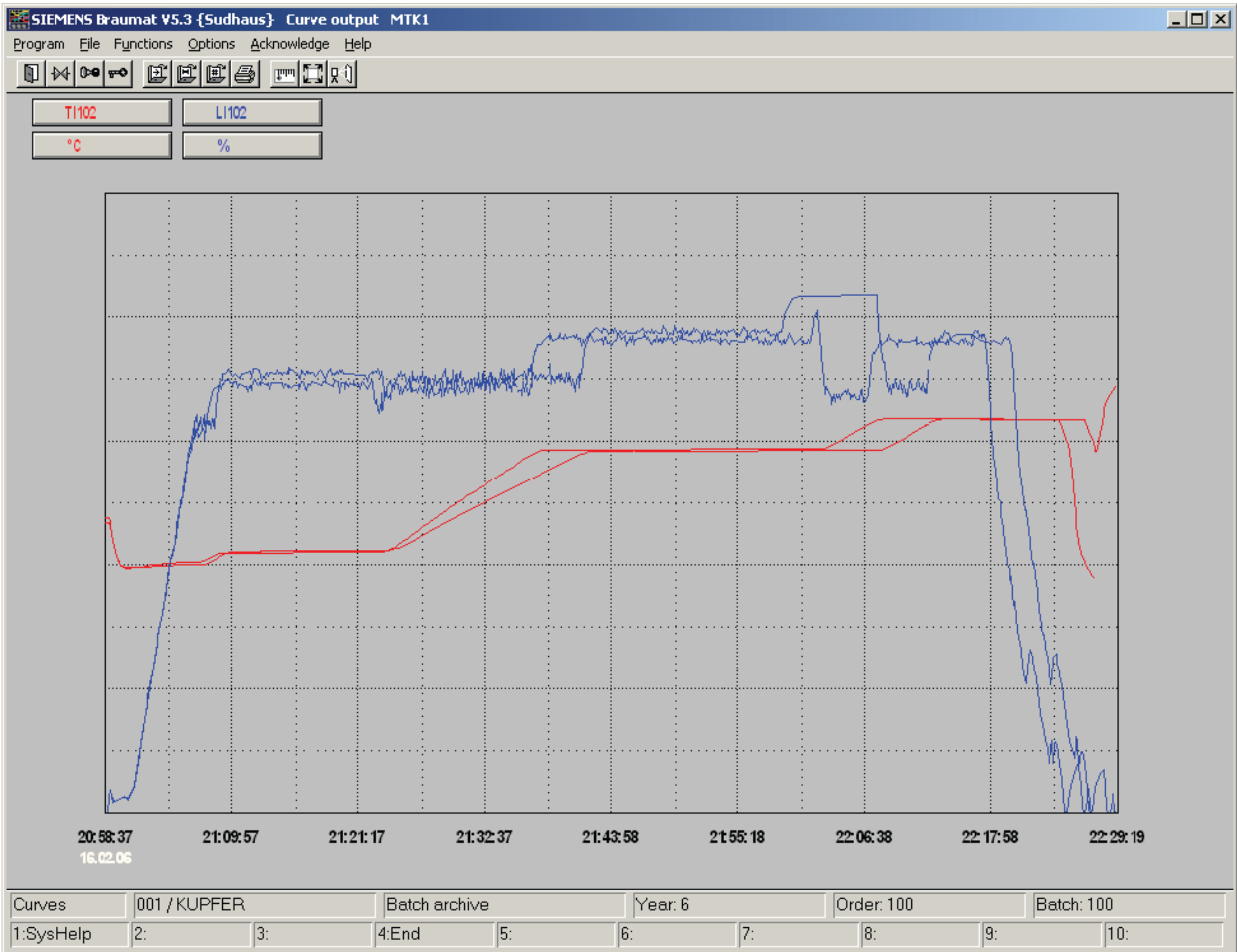


The "Compare curves" option is used to open the selection window for the batches:

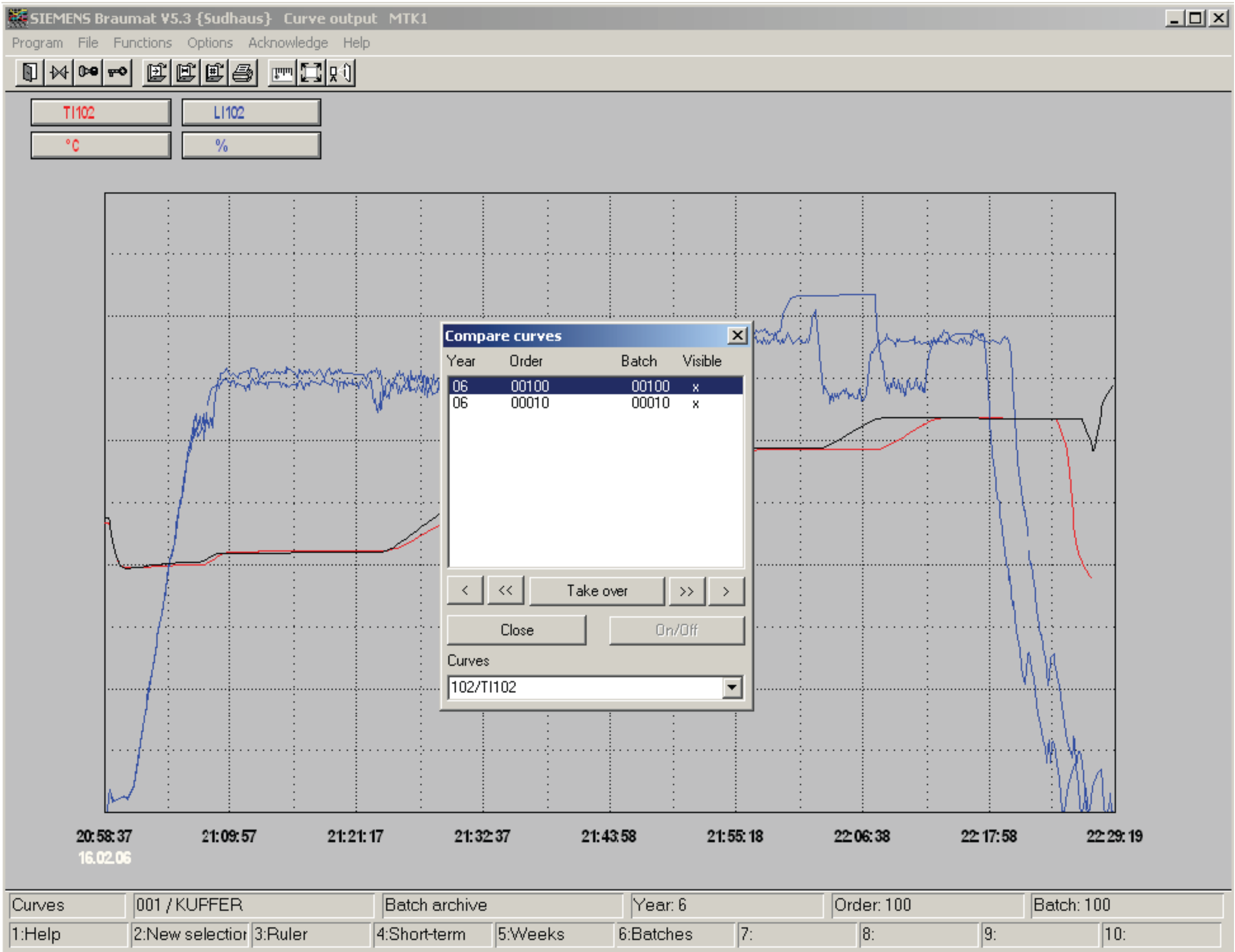


Here you select the comparison batch. You do not define the curve(s) themselves here, because all curves of the respective curve image of this batch will be displayed. It is therefore important to define a meaningful grouping of curves with this option (less is more).

After confirmation with "OK", the curve display will now appear as follows:



Use "Options - Compare curves" to display a window in which you select both the curve itself (drop-down menu at the bottom) as well as the reference and/or comparison curve. The selected curve of the batch - here batch number 100 – that is highlighted with a blue bar, is represented in the curve display in black. It can be displayed and hidden ("ON/OFF" key) and shifted with respect to time using the arrow keys, in order to achieve overlap with the reference curve.



If the curves of another batch are to be compared, this batch is selected with the "Compare curves" option as described above. The reference curves remain the same as before. Thus, it is possible to successively compare multiple batches with references.

Logging

12

12.1 System logs

12.1.1 General

The process control system supports system logs,

where the following logs are included in the system delivery:

- Recipe log
- Stepping log
- Parameterization log
- Order log
- Curve archive log
- Curve input log
- Maintenance log

12.1.2 Format files

Report printouts use format files that may contain system-specific texts.

Each footer file consists of five lines, the meaning of which is specified on the system.

Line	Meaning	Position	Default value
1	Prefix for the data item	Top right	Date:
2	Page header	Top center	(file-specific)
3	Footer 1	Bottom left	Customer:
4	Footer 2	Bottom left	System:
5	Prefix for the page number	Bottom right	Page:

Table 12-1 Table of the format files in the global project text directories (…<poj-path\texte.x)

File	Assignment
ARCH_FRM	Print curve archive
BALIFRM	Print order system
BRGRFRM	Print recipe procedure
BRLIFRM	Print RUP lists

File	Assignment
DATACONN_FRM	Print graphical source views
KURVFRM	Print curve input
MAINTFRM	Print maintenance data
PARAMFRM	Print parametrization
PTXTFRM	Print text parametrization
RECCLFRM	Print list of input materials of a master recipe
RECIOFRM	Print list of input materials of a batch
RECMAFRM	Print process parameters
RECOVFRM	Print recipe overview
SRPROTFR	Print step log

12.1.3 Recipe log

In the "Recipe editor" program, you can select the recipe elements and their layout for output to the printer.

Recipe overview window:

Outputs the master recipe list

Master recipe window

After you have selected and enabled a master recipe, you can output the following lists to the printer:

- Process parameters list
- Process input list

Recipe procedure window

After you have opened the recipe procedure, you can output the following information to the printer:

- The recipe procedure diagram with options for selecting:
  - the current size
  - the complete recipe procedure on one page
  - Poster print
- RUP lists with options for selecting
  - the current recipe unit procedure
  - all RUPs of the recipe procedure

The following topic is deleted – Stepping log -

12.1.4 Stepping log

Basics

A step report is output on the printer.

The printout of the step report can be started in two ways:

- from the equipment operation (EOP).
- by the operator

Printout from the EOP:

To initiate the automatic print-out of the step report at the end of a technological process after the last EOP), the flag M 101.2 must be set in the this last EOP.

Print-out through user interaction:

The user can start printing out a 'stepping log' (application ‘stepping log’). after selecting

- Year
- Recipe type
- Order number
- Batch number
- Unit

The selected stepping log is shown on the screen and via 'file, print' the user has two possibilities: print log for the selected unit only or print log for all units

---

Note

The stepping logs are stored on an IOS (server) only if the datagram type 5 is configured for FIFOs related to this IOS.

It is possible not to log individual EOPs in the step report. To do this flag M 102.5 is set in the EOP. This EOP the does not appear in any step report.

12.1.5 Parameterization log

The parameterization (a setting) is logged on the printer.

Application 'parameterization':

In the application 'parameterization', a record is opened via the selection of the PLC and the object class. This record set is shown on the display. With 'file, printing' the user can select what to print:

- all
- current record
- range: 'record from ... to ...'

Application ‘text parameterization’:

Within the application 'text parameterization', we have two types of text lists: global text lists (PLC-independent) and PLC-related text lists. After selecting 'global' or a PLC-related text file, it is shown on the screen. With 'file, printing', the user can select what to print:

- current file (means whole list)
- current range

12.1.6 Order log

After calling the application ‘order system’ and selecting the order list or the batch list, it can be printed out via ‘file, print’. A dialog appears where the user can select between three types of log prints:

- current view
- table on one page
- whole table

12.1.7 Curve archive log

Via the application 'PI curves', the user can select one of three archive types:

- short term archive
- batch archive
- weekly archive

The curve chosen from an archive and displayed on the screen can be printed out via 'file, printing'.

12.1.8 Curve input log

After selecting a curve via

- group number and
- curve number,

the curve is shown on the screen and can be logged on the printer via 'file, print table'.

12.1.9 Maintenance log

In the application 'maintenance data', the group is selected. Via ‘file, print’ a dialog appears where the user can select some printing options:

Page content	(Actual value, setpoint value, indication IW>SW, finished yes/no, short text, long text)
Filter	(no filter, pending jobs, completed jobs)
Records	(all, range from ... to ...) In the next step, the selected item is printed out.

12.1.10 Printer

One printer can be installed per IOS according to the operating system WINDOWS NT.

12.2 Freely definable logs

12.2.1 Overview

Brief description of the function

The function can be divided into the following two procedures:

Procedure in the PCU

- In the PCU, up to 5 data words can be arranged together into a partial telegram by calling the 'Free protocols' FB multiple times. For this purpose, every FB call by the partial telegram must be configured with the same partial telegram number.
- The data to be arranged is determined by the parameters 'iDataDest', 'iFunction', 'iAna' , etc.
- The partial telegram is concluded by calling the 'Free protocols' FB with iFunction = 0. This transfers the partial telegram to the IOS.
- Multiple partial telegrams can be sent for each recipe type and order/batch number combination. All partial telegrams for an order/batch number combination are combined on the IOS to form a data set in a dBase archive file.

Procedure in the IOS

- The order year and recipe type determine the storage location of the generated dBase file.
- There are three prescribed options for forming the directory structure and file names.
- The combination of order and batch number determines the data set within the dBase file.
- In the `'<proj-path>\FrProt\...'` directory, a definition file must be stored for each recipe type with free protocols. If recipe-independent data is to be transferred with free protocols, a 'Dummy' recipe type must be used.
- In the `'FEPR_xxx.def'` file (xxx = recipe type), both the data sent using the Free protocols FB and the data generated by copying from the step report are defined.
- The source information shown in square brackets [ ] identifies the data sent via the 'Free protocols' FB.

12.2.2 PCU blocks

'FREE PROTOCOLS' FB Collection and temporary storage of batch data

The FB <FREE DEF. LOG> is the most import part for collecting batch data in the PLC. It must be called by the user within procedures.

The FB is used to transfer batch data from the user program to the PCU for archiving on the IOS. Furthermore, the user can close active batch archives. Which actions the FB is supposed to carry out is given via a parameter list.

The FB also carries out the telegram data transfer and transfer to the IOS.

'FREE PROTOCOLS' DB Instance data block

This block is the instance DB of the 'FREE PROTOCOLS' FB. It is taken from the disk supplied without any changes being made.

'INTERMEDIATE BUFFER' DB

The maximum of 5 generated words of user data are intermediately stored in this block before they are transferred from the 'FREE PROTOCOLS' FB with call type = 0 into FIFO and from there to the IOS.

Block assignment on the PCU

The following table contains the block assignment and the basic addressing in the PCU. The descriptions that follow always relate to the parameter designations in the left-hand column.

Parameter	S7 block parameter
'Free protocols' FB	FB 580
'Free protocols' DB	DB 580
'Intermediate buffer' DB	DB 581
Call range 1	Entry of data when calling FB580 from OB1
Call range 2	Entry of data when calling FB580 from OB3x
System blocks	e.g. DB725, DB600, DB721, etc.
Start DW for indirect parameterization	DBW148
Error ID	DB 581, from DBW 12

12.2.3 Call parameters of the "FREE PROTOCOLS" FB

Important:

Using indirect parameterization (call type = -1), the parameters are fetched from the <FREE PROTOCOLS> instance DB.

Call parameters of the function block:

Call parameters	Meaning
iPartTeleNr; DB580.DBW148, INT	Partial telegram number
iBatch; DB580.DBW150, INT	Batch number
iOrder; DB580.DBW152, INT	Order number
iRecipe; DB580.DBW154, INT	Recipe number
iRecType; DB580.DBW156, INT	Recipe type
Not used; the FB580 itself detects the OB from which it was called	Identification of OB
iDataDest; DB580.DBW158, INT 1 - 5 possible	Identification for block transfer (and/or destination data word)
iFunction; DB580.DBW160, INT	Call type
iANA; DB580.DBW162, INT	ANA
rRealInput; DB580.DBD164, REAL	Floating-point number
iDBNr; DB580.DBW168, INT	Data block
iStartDW; DB580.DBW170, INT	Data word (and/or start data word for block transfer)
BoRetVal DB581 DBW12	Return value

Parameter description:

The following table indicates the available range of the parameters.

Parameter	Description	
Sub data gram number iPartTeleNr	-4	The corresponding data set with order number/batch number is reset (texts are overwritten with blanks, numbers set to 0).
	-3	The corresponding record with order number/batch number is transferred into the corresponding copy archive (same recipe type).
	-2	End of a batch, only the sub data gram number from the telegram is used.
	-1	Only the time of the telegram is adopted by the IOS
	0	All data from the telegram is adopted and written into the dBase file
	1 - n:	Sub datagram number; writing position within the dBASE file depends on the corresponding definition file (*.DEF).

Parameter	Description	
iBatch		Identification for origin of the parameters or batch number
	< 0	The indicated value corresponds to the sequence number, recipe number, recipe type, order number and batch number of this unit.
	= 0	Recipe number, recipe type, order number and batch number of the current unit are used
	> 0	Direct information of recipe number, recipe type, order number and batch number via the identifier iBatch, iOrder, iRecipe, iRecType.
iOrder		Order number Direct order number information
iRecipe		Recipe number Direct information recipe number
iRecType		Recipe type Direct recipe model information
iDataDest		Identification for block transfer and/or destination data word
	-1	If iFunction = 7, the whole block of 5 DWs is transferred into the telegram.
	No.:	In this destination data word within the telegram, the determined value is transferred.
iFunction		Function of the FB <FREE PROTOCOLS>: • Type, origin and destination of the data to be sent are given by this parameter. • The next table describes this parameter in detail.
iANA		Depending on the function, iANA is interpreted as a value or number • With iFunction = 9 , iANA is interpreted as a value and transferred to the address given by the iDataDest parameter. • With iFunction = 1 or iFunction = 2, iANA is interpreted as a DFM number.
rRealInput		Direct input of a real value in connection with iFunction = 11 → not advisable for dBASE archives, therefore do not use as standard
iDBNr		Data block with iFunction = 7: The word contains the block number from which the data is fetched.
iStartDW		Data word and/or starting offset for block transfer • iFunction = 7: Number of the data word from which the value is to be read. • iFunction = 7and iDataDest = -1: Number of the data word from which 5 data words are read
Return value	0:	no error occurred
boRetVal	1:	an error occurred, explained in detail in DB186, DW40.

Types of call (ART, iFunction):

Value	Description
= -1:	Indirect parameterization; Parameters are located at DB <FREE PROTOCOLS> at address ‘start DW’ for indirect parameterization
= 0:	Setting the end flag of a datagram. The data is moved into the FIFO and sent to one or more server.
= 1:	Reading the actual value (double-word) of the DFM module with the index (number) given by ‘ANA’: 001 ... 255 : DFM Group 0 257 ... 511 : DFM Group 1 513 ... 767 : DFM Group 2 769...1023 : DFM Group 3 1025...1279 : DFM Group 4 1281...1635 : DFM Group 5 1537...1791 : DFM Group 6 1793...2047 : DFM Group 7 2049...2303 : DFM Group 8 in the identifier 'ANA'. The parameter 'identification for block transfer' indicates the address (offset) DW (1...5) within the telegram.
= 2:	Reading the setpoint value (double-word). See iFunction = 1.
= 3:	Reading the actual value (double word) of the DFM module from the corresponding unit with the relative number (1 to 20) provided in the iANA description. The iDataDest parameter indicates the start DW (1 to 5) in the telegram.
= 4:	Reading the setpoint value (double-word). See iFunction = 3.
= 5:	Not used
= 6:	Reading the actual value of the MESS block with the number in the identifier 'iANA'. The iDataDest parameter indicates the start DW (1 to 5) in the telegram.
= 7:	Loading a word from a DB (parameters iDBNr and iStartDW). The iDataDest parameter indicates the start DW in the telegram.
= 8:	Not used.
= 9:	Loading a value from identifier iANA. With this function, iANA is not interpreted as a DFM number, but as any user-defined value. The iDataDest parameter indicates the start DW in the telegram.
= 10:	Not used.
= 11:	Loading a double word from identifier rRealInput. The iDataDest parameter indicates the start DW (1 to 5) in the telegram. → not advisable for dBASE archives, therefore do not use as standard
= 12:	Loading the date from the real-time clock of the S7-CPU; format: month, day, day of the week (1=Sunday,...), year; the data is always in the 1st and 2nd DBW in the telegram. → not advisable for dBASE archives, therefore do not use as standard
= 13:	Loading the time of day from the real-time clock of the S7-CPU; format: minute, hour, 0, second, year; the data is always in the 1st and 2nd DBW in the telegram. → not advisable for dBASE archives, therefore do not use as standard
= 14 - 19:	Not used
= 20	Loading an input byte with the number in the identifier iANA

Value	Description
= 21	Loading an input word with the number in the identifier iANA
= 22	Loading an input double word with the number in the identifier iANA.
= 23	Loading an output byte with the number in the identifier iANA
= 24	Loading an output word with the number in the identifier iANA
= 25	Loading an output double word with the number in the identifier iANA
= 26	Loading a flag byte with the number in the identifier iANA
= 27	Loading a flag word with the number in the identifier iANA
= 28	Loading a flag double word with the number in the identifier iANA
= 29 - 31:	Not used
= 32	Loading a time value (BCD, time base + value) with the number in the identifier iANA
= 33	Loading a counter value (BCD) with the number in the identifier iANA
= 34 - 98:	Not used
= 99	Initialization of the temporary entry (DB <FREE PROTOCOLS>) and the temp. buffer (DB <FREE PROTOCOLS>) with zero.

12.2.4 Telegram structure (PLC IOS) and sending

The FB<Free def. log> assembles a telegram being sent to the servers which contains the following information:

Telegram structure		DB<Fr.Prot.> Call range 1	DB<Fr.Prot.> Call range 2	DB<buffer>
Telegram length		DBW 50	DBW 100	DBW 300
Tele. type	PLC no.	DBW 52	DBW 102	DBW302
Year	Month	DBW 54	DBW 104	DBW304
Day	Hour	DBW 56	DBW 106	DBW306
Minute	Second	DBW 58	DBW 108	DBW308
Milliseconds		DBW 60	DBW 110	DBW310
Recipe year	Recipe type	DBW 62	DBW 112	DBW312
Recipe number		DBW 64	DBW 114	DBW314
Order number		DBW 66	DBW 116	DBW316
Batch number		DBW 68	DBW 118	DBW318
sub datagram number		DBW 70	DBW 120	DBW320
user data 1		DBW 72	DBW 122	DBW322
user data 2		DBW 74	DBW 124	DBW324
user data 3		DBW 76	DBW 126	DBW326
user data 4		DBW 78	DBW 128	DBW328
user data 5		DBW 80	DBW 130	DBW330

Depending on the call to the FB<FREE DEF. LOG >, the telegram is assembled and stored to the calling ranges 1 or 2 according to calling type. At the end of the block, the corresponding user data is transferred into the temp. buffer. After a maximum of 5 user data words are collected, all data is written into a FIFO through a call to the FB<FREE DEF. LOG> (where type = 0) and sent to the server (IOS). Telegram type 6 has to be configured before for this FIFO.

The parameters shown with a gray background in the table above are updated by the system when a telegram is sent and must not be modified by the user.

On the hard disk of the IOS, a dBASE file is created after arrival of the telegram. Depending on the date and the recipe type the server (PROT_006.DLL) creates the folder and file name. According to the order and batch number, one record is created.

The definition file (FEPR_<Rezepttypnr.>.DEF) provides the structure of the dBASE files and where the user data is written to (offset within the record). Data set elements with a corresponding sub telegram number are written, all others remain unconsidered.

12.2.5 Examples of calls for the <FREE PROTOCOLS> FB

Basics

The calls of the FB are made from within the procedures. The call can be made directly or symbolically. In the table, the direct call is given. The symbolic call is: Call "FREE_PROT_FB", "FREE_PROT".

Call FB 580 function 7

In this example, five data words of DB300, i.e. DBW10, DBW12, DBW14, DBW16 and DBW18 are copied to DB580 for datagram element 1.

Parameter	Transfer values	Description
	: A M x.y; : BEC;	the block can only be called directly
	: Call FB 580, DB580;	Taking over the date for batch start
iPartTeleNr	:= 1	
iBatch	:= 0	Batch data is taken from the current dataset
iOrder	:= 0	
iRecipe	:= 0	
iRecTyp	:= 0	
iDataDest	:= -1	Transfer block of 5 data words
IFunction	:= 7	iFunction = 7, i.e. load data word block with time
IANA	:= 0	all other identifiers are ignored
RRealInput	:= 0.000000e+00	
iDBNr	:= 300	Date is in DB300, Format: (Year, month, day, Hrs., Min., Sec.)

Parameter	Transfer values	Description
iStartDW	:= 10	Date is located at DBW10,12 and 14
boRetVal	:=M a.b	Error returned value; 0: No error; 1: Error

Call FB 580 function 1

In this example, the DFM process value of DFM4 is copied from DFM group 0 to user data 3 and 4 of DB580.

DFM 0/ 4 actual value	DB580
DB736 DBW366 and DBW367	DBW126 and 128 or DBW326 and 328

Parameter	Transfer values	Description
	: A M x.y; : BEC;	
	: Call FB 580, DB580;	Take over malt quantity from DFM groats flowmeter
iPartTeleNr	:= 1	
IBatch	:= 0	Batch data is taken from the current dataset
IOrder	:= 0	
IRecipe	:= 0	
IRecTyp	:= 0	
IDataDest	:= 3	the target data words in the telegram are located at address 3.DW and 4.DW
IFunction	:= 1	iFunction = 1, i.e. read the actual value (double-word!) from the DFM module addressed by the parameter 'ANA'
IANA	:= 4	Take actual value from DFM No. 4
RRealInput	:= 0.000000e+00	all other identifiers are ignored
IDBNr	:= 0	
IStartDW	:= 0	
BoRetVal	:=M a.b	Error returned value; 0: No error; 1: Error

Call FB 580 function 4

In this example, the DFM setpoint of DFM3 is copied from DFM group 0 to user data 1 and 2 of DB580.

DFM 0/ 3 setpoint	DB580
DB736 DBW342 and DBW344	DBW122 and 124 or DBW322 and 324

Parameter	Transfer values	Description
	: A M x.y; : BEC;	
	: Call FB 580, DB580;	Take malt quantity from unit groats load
IpartTeleNr	:= 1	
Ibatch	:= 0	Batch data is taken from the current dataset
Iorder	:= 0	
Irecipe	:= 0	
IrecTyp	:= 0	
IdataDest	:= 1	the target data words within the telegram are at address 1.DW and 2.DW
Ifunction	:= 4	iFunction = 4, i.e. loading the setpoint value (double-word!) from the TA (unit) module with the relative number addressed by the parameter 'ANA'
IANA	:= 3	Read the setpoint value from the DFM module, which is addressed within the unit as the 3rd DFM module
RRealInput	:= 0.000000e+00	all other identifiers are ignored
IDBNr	:= 0	
IStartDW	:= 0	
BoRetVal	:=M a.b	Error returned value; 0: No error; 1: Error

Call FB 580 function 9

In this example, the value (2) of parameter ANA is copied to user data 1 of DB580.

Parameter ANA	DB580
2	DBW122 or DBW322

Parameter	Transfer values	Description
	: A M x.y; : BEC;	
	: Call FB 580, DB580;	Taking the number of the WPF (Würzepfanne/wort settler) used for heating
iPartTeleNr	:= 1	
iBatch	:= 0	Batch data is taken from the current dataset
iOrder	:= 0	
iRecipe	:= 0	
iRecTyp	:= 0	
iDataDest	:= 1	the target data word in the telegram is the 1.DW
iFunction	:= 9	iFunction = 9, i.e. load a value from the parameter 'ANA'
iANA	:= 2	Value to be loaded

Parameter	Transfer values	Description
rRealInput	:= 0.000000e+00	all other identifiers are ignored
iDBNr	:= 0	
iStartDW	:= 0	
boRetVal	:=M a.b	Error returned value; 0: No error; 1: Error

Call FB 580 function 7

In this example, the five data words DBW10, DBW12, DBW14, DBW16 and DBW18 are copied from DB100 to datagram element 1.

Parameter	Transfer values	Description
	: A M x.y; : BEC;	
	: Call FB 580, DB580;	read data block from DB
iPartTeleNr	:= 1	
iBatch	:= 12345	
iOrder	:= 12	
iRecipe	:= 1	
iRecTyp	:= 1	
iDataDest	:= -1	Transferring data word block
iFunction	:= 7	iFunction = 7, i.e. copy a block of 5 data words into the telegram
iANA	:= 0	
rRealInput	:= 0.000000e+00	
iDBNr	:= 100	from DB100
iStartDW	:= 10	from DBW 10 to DBW 18 (Caution: S7 is byte-oriented!)
boRetVal	:=M a.b	Error returned value; 0: No error; 1: Error

Call FB 580 function 0

In this example, datagram element 1 is transmitted as datagram 6. The data to be sent must be available in DB580 prior to the call. The folder and file names are based on the year, the month and the recipe type.

Parameter	Transfer values	Description
	: A M x.y; : BEC;	
	: Call FB 580, DB580;	Sending data block to the server
IPartTeleNr	:= 1	
IBatch	:= 0	
IOrder	:= 0	
IRecipe	:= 0	

Parameter	Transfer values	Description
IRecTyp	:= 0	
IDataDest	:= 0	
IFunction	:= 0	iFunction = 0, set ‘end’ flag for the batch
IANA	:= 0	
RRealInput	:= 0.000000e+00	
IDBNr	:= 0	
IStartDW	:= 0	
BoRetVal	:=M a.b	Error returned value; 0: No error; 1: Error

12.2.6 Error messages

In order to easier locate the fault localization with the parameterization of the identifiers, the **FB <FREE DEF. LOG>** returns the last error of **DB <FREE DEF. LOG>**.

In order to easier locate the fault localization with the parameterization of the identifiers, the **error code** is returned by FB80 via DB186 DW 40.

Table 12-2 Additional information is returned via AKKU1 and AKKU2.

Error ID	Description
FE01	DB not available
FE02	Common error (please refer to error code AKKU2)
FE03	DB too short
Error code AKKU2 and/or DB186, DW40:	
1	'CHNR' exceeds the number of units available
2	called with 'CHNR' = 0, not called from a running procedure
3	DFM No. exceeds number of DFMs
4	Mess No. exceeds number of MESS
5	FIFO occupied
6	'W-NR' out of the valid range
7	'CHNR' > 0 not allowed (invalid)
8	DFM No. not allowed (invalid)
Return parameter of the FB <FREE DEF. LOG>	Description
0: 1:	no error occurred an error occurred, explained in detail in DB186, DW40.

Error code working with a SIMATIC S7

The error code consists of 2 bytes in a hexadecimal format. The calling type indicates the first byte, the second byte is a continuous number within calling type (sub type).

Table 12-3 Error code DB581.DBW12:

Error with calling type= xy (DBB12, hexadecimal)	error code. (DBB13, hexadecimal)	Description
01 or 07	01	DB not available or too short
Xy	02	
FF	03	no data exists in the data words 1 to 5
FF	04	Error writing data into FIFO
01, 03, 06, 07, 09 or 11	05	the temp. buffer is full; Transfer data with calling type= 0 in FIFO
00	10	'iBatch' exceeds number of units
03	21	'iBatch' > 0 is invalid with calling type = 3 or = 4
01	30	DFM No. exceeds number of DFMs
06	40	Mess No. exceeds the number of MESS
03	50	Error at recipe data block: too few steps
03	51	'ANA' exceeds number of DFM in the unit
03	52	'ANA' out of the valid range: 'ANA' <= 0
xy	60	
xy	70	
xy	80	
01, 03, 06, 07, 09 or 11	90	'Code for block transfer' out of the valid range: 'Identification for block transfer' > 5
01, 03, 06, 07, 09 or 11	91	'Code for block transfer' out of the valid range: 'Identification for block transfer' <= 0
01, 03 or 11	92	'Code for block transfer' out of the valid range: 'Identification for block transfer' == 5

Return parameter of the FB <FREE DEF. LOG> boRetVal

Return parameter of the FB <FREE DEF. LOG>	Description
BoRetVal	0: no error occurred 1: an error which is specified in the DB581.DBW12 in detail (see above)

12.2.7 Parameterization in the IOS

12.2.7.1 Archive file structure (Def file), receive data

With the `'...<proj-path>\FrProt\FEPR_xxx.def'` DEF files (xxx = recipe type), the structures of future archive database files are defined.

The file initially has a defined structure. An element offset must be entered for each definition line. For each new element, the offset is calculated from the previous offset + length of the previous data type. The overall data set length in line 2 must be adapted for each element.

A definition line always consists of:

- Element offset
- Element name (= column name in the dBASE file)
- Data type (CINT, CHAR)
- Length of the data type
- User data description (partial telegram source) or comment (max. 39 characters)

When opening a 'free protocol' file the first time and/or when a frame is sent the first time, the system creates a dbf - File takes the structure as described in the DEF file. The following is a sample of such a file.

Table 12-4 **Unrestricted listing/received data configuration/telegram model 6**

0	Offset:	always 0		
121	Record length:	must be modified		
1	AUFTR_NR	CINT	5	Order number
6	CHARG_NR	CINT	5	Batch number
11	SZ_JAHR	CINT	2	Start time year
13	SZ_MONAT	CINT	2	Start time month
15	SZ_TAG	CINT	2	Start time day
7	SZ_STUNDE	CINT	2	Start time hour
19	SZ_MINUTE	CINT	2	Start time minute
21	SZ_SEKUNDE	CINT	2	Start time second
23	REZ_TYP	CHAR	16	Name of the recipe type
39	RECIPE	CHAR	16	Recipe name
55	TEILTELNR	CINT	6	Partial telegram number
61	DATEN1	CINT	7.2	[1,1,1,1] PCU1,Teilt.1,DW1
68	DATEN2	CINT	12.2	[1,1,2,2] PCU1,Teilt.1,DW2/3
80	DATEN3	CINT	6	[2,2,1,1] PCU2,Teilt.2,DW1
86	DATEN4	CINT	11	[2,2,2,2] PCU2,Teilt.2,DW2/3
97	TEXTE5	CHAR	16	[1,3,3,1,"mult.txt"] PCU1,Teilt.3,DW3,Textl.
113	TEXTE6	CHAR	8	[1,3,3,1,"dfm0.txt"] PCU1,Teilt.3,DW3,Textl.

Up to the element 'TEILTELNR', the structure is fixed for all use cases, but is optional with the exception of order number and batch number. The lines below ‘TEILTELNR’ are optional, too, but can be configured in many different ways. 16 bit and 32 bit values as well as texts from text lists are allowed. The corresponding address within the telegram received from the PLC is given in the column on the far right (normally the comment column).

12.2.7.2 Description file syntax

Basics

The description file must be built up as follows:

Description file				
Offset				
Record length				
Offset of element	Element name	Data type	element size[.deci-mal points]	if applicable, de-scription of user data or comment

The configuration of the DEF file is to be interpreted as follows:

Offset:

The offset defines the number of bytes at the beginning of the (dBASE) file that do not contain any user data (header data), but are necessary for the file organization. This header data contains the dBASE file size, number of records and record structure.

In case of using `Free def. logs' this value may equal 0, as during evaluation of the telegrams this offset is computed.

Record length:

The record length is the record size in bytes. Normally, it is the sum of all elements (4th column of DEF file). As it is allowed to leave some elements in the DEF file, this record length parameter can be higher than this byte sum.

data element structure:

Every line describes the structure of one element as follows:

element offset:

This value gives a (relative) offset within the record, where the data of this element is stored. start with byte address 0.

element name:

This is the symbolic name of the element. Contains at most 10 characters whereas the first must start with a character. This name later becomes the column name within a dBASE file.

Data type:

As only ASCII characters are allowed within a dBASE file, only a subset of all possible data types for DEF files is allowed to be used:

Type	Description
CHAR:	Character string
CINT:	Number as a character string, is printed/displayed later with right justification

element item, decimal point position:

Number of the characters and/or bytes of an element.

The minimum size is:

- 6 if the number has the range of a data word and no decimal places
- 7 if the number has the range of a data word and decimal places
- 11 if the number has the range of a data double word and no decimal places
- 12 if the number has the range of a data double word and decimal places

Either it is a single number (X) or the number is followed by a decimal point and another number (X.Y) indicating the number of decimal spaces. Rule: 1 <= Y <= (element size(X) - 3).

Comment:

max. 39 characters comment or user data description

User data description:

Additional information within squared brackets [...].

Every telegram from a PLC contains the following information: PLC number and sub data gram number. Based on these two parameters and together with the additional information given in squared brackets the system takes certain data out of the data gram and assigns it to specific elements (lines in DEF file), and therefore of course to the dBASE file record. Every data gram contains up to 5 data words (10 bytes) of user data. As a record may contain more than 5 words normally, several data grams have to be sent. Using sub data gram numbers provides the option of separating a record and assigning it to several (sub) data grams.

According to the element number the system extracts the value(s) from the data gram(s). If the name of a file has been specified beyond the number of decimal spaces, the system uses it as an index within the text file to retrieve a text line.

The description has the following syntax:

- **PLC number, sub data gram number, element number, data size in words**
  or for text files
- **PLC number, sub data gram number, element number, 1, “file name”**

Table 12-5 Meaning of the different elements:

Structure elements	Description
PLC number:	Number of the PLC whose telegram contains a value that is supposed to be extracted (valid range: 1 - <number of PLCs in system>).
sub data gram number:	Number of the (sub) telegram, which a value is supposed to be taken from (valid range: 1 - <number of sub data grams>).
element number:	Number of the user data element within the data gram, whose value is supposed to be taken (valid range: 0 ... 4).
data size in words:	Number of user data elements that will form the value = 1 for data words = 2 for data double words = 1 if followed by specification of a text file = 3 with "DT' / "DT1" identifier (enter date)
additions/subtractions flag (optional) or	'+' the newly received value is added to the old value or '-' the newly received value is subtracted from the old value
Name Text file: (Optional)	The value in element number corresponds to the text number in the text file. If the "RecipeMode=GLOBAL" switch is set in SYS.INI, the text file in the `<proj-path>\TEXTE.X\` folder is read, otherwise in the `<proj-path>\PCU.xx\TEXTE\` folder Note: Umlauts in text files To ensure correct processing of umlauts, parameter Konv=1 must be set in [KonvChar] of "DB.ini"
Date/time: (Optional)	If the element size 1 is specified and in addition 'DT' or 'DT1', the first 3 words (user data) of the data gram are interpreted as a date/time in long or short format. DT: e.g. "10.01.1995 12:55:00" DT1: e.g. "10.01.1995 12:55:00 PM"

12.2.8 Examples of a telegram entry

12.2.8.1 Engineering in the PLC

In the PLC up to 5 words can be assembled by calling the FB580 up to five times, whereas the same sub data gram number has to be specified.

The values handed over are specified by the parameters 'DataDest', 'Function', and 'Ana' etc.

The last call to FB580 uses 'function = 0',

this triggers sending the data gram to the IOS.

Per recipe type/order/batch number combination more than one sub data gram can be sent. All sub data grams with the same order/batch number combination are stored into the same dBASE file record,

where the recipe type defines the location (folder name) of the dBASE file

and the order and batch number indicates the record within this dBASE file.

12.2.8.2 Engineering in the IOS

Placing of numbers & texts

Sub data gram from PLC

type=6	PCU=1
year=94	month=9
day=3	hours=12
minute=1	seconds=0
year = 03	recipe type=01:
order no.	= 13000
batch no.	= 124
sub data gram no.	= 3
DW 0	= 1278
DW 1	= 0x0001
DW 2	= 0x12BA
DW 3	= 3
DW 4	= 10

Datasets in archive file: \FRPROT\FRJA_94\REZ_001\FE_09_94.DBF

order_no.	batch_no	...	Weight	Flow	Text	Amount
.....		...				
.....		...				
.....		...				
13000	124	...	12.78	70.330	cell 12 / tank 3	10

sub set of the DEF file FEPR_001.DEF

Def. Offset	Name	Format.	Reference to telegram
...	...	...	....
55	Weight	CINT 7.2	[1,3,0,1]
62	Flow	CINT12.3	[1,3,1,2]
74	Text	CHAR 16	[1,3,3,1,"Anlag.txt"]
90	Amount	CINT 6	[1,3,4,1,"+"]

sub set of the text file "Anlag.txt"

Line	Content
2	cell 12 / tank 2
3	cell 12 / tank 3

Explanations for implementation

- The value 1278 sent via sub data gram 3 (1st user data) is logged at byte offset 55 into the dBASE record FEPR_001.DEF in FE_09_94.DBF.
- 55 the beginning of the data type within the definition. This offset starts at 0 and is incremented.
- The next definition begins at 62 as at 55 a data type of size 7 (bytes) is specified.
- ‘Weight’ is the column name of the record in the dBASE file. Remark: Don’t use special characters, e. g. "ß" as in ‘Durchfluß’.
- The CINT 7.2 value is supposed to be stored as a number with a maximum of 7 characters (comma included) and 2 decimal places.
- [1,3,0,1] describes where the data is located within the data gram. [PLC number, sub data gram number, offset of the first user data word, number of data words]. Additional information is possible, as some other examples show (‘Text list’, 'Add', 'Subtract' etc.).

Annotation: The assignment of a datagram to the DEF file is related to the recipe type.

Configuration of ‘date & time’ logging

Sub data gram from PLC

type=6	PCU=1
year=94	month=9
day=3	hours=12
minute=1	seconds=0
year = 03	recipe type=01:
order no.	= 13000
batch no.	= 124
sub data gram no.	= 4
DW 0	= 95.01
DW 1	= 10.12
DW 2	= 55.04
DW 3	=
DW 4	=

Data sets in archive file: \FRPROT\FRJA_94\REZ_001\FE_09_94.DBF

order_no.	batch_no	...	Date1		Date2	
.....		...				
.....		...				
.....		...				
13000	124	...	10.01.1995	12:55	10.01.1995 12:55:04	

sub set of the DEF file FEPR_001.DEF

		.	.	
96	Date1	CHAR 16	[1,4,0,3,"DT1"]	
112	Date2	CHAR 19	[1,4,0,3,"DT"]	

12.2.9 File/directory structure receive data

Core statement

The ‘Free def. log’ records are recorded into the corresponding files automatically by the system after receiving a data gram. According to the data and the recipe type the file name is created, if the file doesn’t exist, it will be created ‘on the fly’.

Yearly/monthly archive structure

The path/file structure here is as follows:

..\FRPROT\FRJA_<Year>\REZ_<Recipe type>\FE_<Year>_<Month>.DBF

According to the order and batch number, a single record is created in the dBASE file (both are used as a key). Thus these two pieces of information are unambiguous and are evaluated as search keys. Whenever a telegram with exactly these numbers arrives from the PCU, the matching data set from the file is first found and the corresponding user data is entered or overwritten. All other elements are retained, and if not yet evaluated by a predecessor partial telegram, are not taken into account.

With partial telegram number = -1, a data set can initially be created for which only header data and the date are explicitly entered. Otherwise, this data is created when the first partial telegram is received.

Yearly archive structure <..... tbd!">

The path/file structure here is as follows:

..\FRPROT\xxxxxxxxx\yyyyyyyyyy\FE_<year>_<month>.DBF

The "PROT_006.INI" file must contain an entry for the month and year for each recipe type:

```
[REZ_xxx]
...
;Only for old path-systematic. If month=AC ( or no entry), the
program enters the telegram-month into the filename. Otherwise, it
enters this special name.
;month=00
;Only for old path-systematic. If year=AC ( or no entry), the program
enters the telegram-year into the filename. Otherwise, it enters this
```

```
special name.
;year=00
```

Example:

```
[REZ_001]
month=00
year=00
```

This yields the file name: FE_00_00.DBF

Order/batch archive structure

With this version, the path/file structure is as follows:

..\FRPROT\FRJA_<Year>\REZ_<Recipe type>\ANR <A No. 5-digit>\FE_<Batch No. 5-digit>.DBF.

A separate subdirectory is created here for each order number block and a separate dbf file is created for each batch number block. The block size can be set individually and is 50 by default (see below).

The following switches in "PROT_006.INI" are used for activation and configuration:

```
[GLOBAL]

;Global for all recipetypes. New path-systematic on/off (order-batch-sorting/month-year-sorting).

New_Path=1

;Global for all recipetypes. It means, how much files get in one path, dependent on orderno.. (Only for new path-systematic.)

Block_Order=50

;Global for all recipetypes. It means, how much datasets get in one file, dependent on batchno.. (Only for new path-systematic.)

Block_Batch=50
```

These settings can also be defined individually for a recipe type by setting the same switch in the respective `[Rez_xxx]` section.

12.2.10 Copy archive

The `'...<proj-path>\FrProt\FKPR_xxx.def'` DEF file (xxx = recipe type) of the copy data is described here.

With partial telegram number '-3' , it is possible to create a copy data archive from the PCU. Depending on the recipe type, the order number and the batch number, parts of the receive data archive are copied into a new archive.

	Free def. log / copy data structure / telegram type 6		
0	Offset:	always 0	

44	Record length:	must be modified		
1	AUFTR_NR	CINT	5	["AUFTR_NR"]
6	CHARG_NR	CINT	5	["CHARG_NR"]
11	D1	CINT	7.2	["DATEN1"]
18	D2	CINT	12.2	["DATEN2"]
30	D3	CINT	6	["DATEN3"]
36	T6	CHAR	8	["TEXTE6"]

The User data description column contains the element name of the corresponding element in the receive archive. When copying over, the system searches for this name in the original file and then the corresponding entry is made in the destination file.

Note

This function can only be used if the following switch is set in the `'...\sys\prot_006.ini'` file and the function is therefore enabled:

```
[prot_cpy]
;Activate the PROT_CPY.DLL: Copy some datasets from the free
protocolls into a special archive
Akt=1
```

12.2.10.1 Example of copying data

Partial telegram from PLC			**Data sets in archive file: \FRPROT\FRJA_94\REZ_001\FE_09_94.DBF**						
type=6	PCU=1		**Order_nr.**	**batch_no**	**...**	**Weight**	**Flow**	**Text**	**No.**
year=94	month=9		.....		...				
day=3	hours=12		.....		...				
minute=1	seconds=0		.....		...				
year = 03	recipe type=01		13000	124	...	12.78	70.330	cell 12 / tank 3	10
order no.	= 13000								
batch no.	= 124		**Extract from the DEF file: FKPR_001.DEF**						
sub data gram no.	= -3		5	ORDER		CINT 5	["Order_nr"]		
DW 0	irrelev.		6	CHA		CINT 5	["Char_nr"]		
DW 1	irrelev.		10	T_1		CHAR 8	[“Text“]		
DW 2	irrelev.		18	D_1		CINT 12.3	[“Flow“]		
DW 3	irrelev.		30	G_7		CINT 7.2	[“Weight“]		
DW 4	irrelev.								
			Data sets in archive file: \FRPROT\FRJA_94\REZ_001\FE_09_94.DBF						
			ORDER	CHA		T_1	D_1	G_7	
			....	....		....	....	....	
			13000	124		cell 12	70.330	12.78	

12.2.10.2 Copy archive folder

The ‘Free def. log’ copy data sets are copied automatically to the destination by the system after receiving a data gram type = -3. Dependent on the recipe type, the order and the batch number, the record is first searched within the source files. If the files do not exist, they will be created. The syntax is as follows:

..\FRPROT\FRJA_<Year>\REZ_<Recipe type>\FE_<Year>_<Month>.DBF

Every record within this (destination or copied) archive is a subset of the receive archive. The keys are again the order and batch number.

12.2.10.3 Programming the copy archive

We suggest the following sequence for calling FB580 for datagram 6:

Call of FB580 with iFunction=99 i.e. deletes data in the buffer

Call of FB580 with iFunction=7 i.e. enters the datagram subset

Call of FB580 with iFunction=0 i.e. sends datagram 6

Note

If an error occurs when data are written to the archive, you can perform a diagnosis by means of the "Tracesvr.exe" program. The cause of error is in most cases found in a faulty *.def file.

12.2.11 Summary

Note the following points for working with Free Protocols:

- Telegram 6 must be enabled in a FIFO.
- In directory `'<proj-path>\frprot\...'`, definition files must be created (see section "Parameterization in the IOS (Page 908)")
  Fepr_xxx.def xxx = Recipe type (range 1 to 255)
  As a template, the fepr_def.def file can be copied and edited.
- Telegrams 06 must be assembled in the controller by calling FB <Free protocols> in DB <Free protocols>.
  Refer to the section Call parameters of the "FREE PROTOCOLS" FB (Page 897)
  FB 580/DB580 with iFunction > 0 e.g. 7
  If a maximum of 5 values were entered, this data must be sent as a partial telegram (see item 5). After this send call, the data for the next partial telegram can be collected.

- Telegrams 06 must be initiated in the controller for each partial telegram by calling the FB <Free Protocols>.
  Refer to the section Call parameters of the "FREE PROTOCOLS" FB (Page 897)
  FB 580 with iFunction = 0
- The telegrams trigger creation of DBase files in directory `<proj-path>\frprot\frja_YY\rez_XXX\fe_YY_MM.dbf`. These can then be reused with standard tools such as Microsoft Access or Excel.
- YY = year from the frame
  XXX = recipe type from the frame
  MM = month created from the frame.

12.3 Generating free protocols from step protocols

12.3.1 General

Description

The option exists to transfer a subset of the received step protocols to other archives.

There following two methods available for this:

1. Extracting individual setpoint/actual values from specific EOPs of the step logs and saving in the table "<proj-path>\SRPROT\SR_FRPR.DBF" ("Free step logs")
2. Selecting general step log data by criteria that are stored in a so-called filter talbe and saving to the archives of the 'Free logs' <proj-path>\FRPROT\FRJA_<year>\REC_<Recipe type> \FE_<year>_<month>.DBF

Method 1: Free step logs

The selection criteria are defined by the configuration engineer in the file <proj-path>\SYS \SR_FRPR.INI fest. The step logs from all or only one special PCU can be selected. For the EOPs too all or only very specific ones can be entered. The DFM setpoint/actual value column positions must also be specified.

Installation and configuration

Copy the following file from the template project in the installation path to the current project:

<proj-path>\SYS\SR_FRPR.INI

Edit the configuration file SR_FRPR.INI according to the requirements of the example below. In this example, the setpoints 2 (component or material) and 3 (quantity) are to be extracted from the transfer EOPs 250, 251, 252, 253 of PCU 1.

```
;*******************************************************************
**
;
; \WINDCS\SYS\SR_FRPR.INI
;
; This file is used by the following modules:
; - SR_FRPR.DLL
;*******************************************************************
**
[GLOBAL]
;Max. number of datasets in the archive
Max_Ds=10000
[GOP]
;GOP-Selection: All GOP's from the step-report are allowed
All=0
;GOP-Selection: PCU-No., global for all GOP's, which are in the list
;PcuNrGlob=1
```

```
;GOP-Selection: Component-position in the step report dataset,
global for all GOPs, which are in the list
;KompPosGlob=1
;GOP-Selection: Weight-position in the step report dataset, global
for all GOPs, which are in the list
;GewiPosGlob=2
;GOP-Selection: String, which decides the selected GOPs from the step
report
Select=250,251,252,253
;GOP-Selection: String, which decides the PCU-No. for each GOP
PcuNr=1,1,1,1
;GOP-Selection: String, which decides the component-position for
each GOP
KompPos=2,2,2,2
;GOP-Selection: String, which decides the weight-position for each
GOP
GewiPos=3,3,3,3
```

You can configure either the same 2 setpoints for all EOPs of a PCU (**All=1**) or multiple EOP/PCU/setpoint combinations (**All=0**):

1. **All=1**
   Parameterize one value each in: PcuNrGlob, KompPosGlob, GewiPosGlob
2. **All=0**
   Parameterize in order multiple values separated by "," in: Select, PcuNr, KompPos, GewiPos

Activation

This function is activated in the INI file **"<proj-path>\SYS\SRPROT5.INI"** with the switch:

[sr_frpr]

Akt = 1

After restart of the PCU servers, a dBASE file "<proj-path>\SRPROT\SR_FRPR.DBF" is created. The table structure corresponds to the definition file "<sys-path>\SRPROT\SR_FRPR.DEF".

An additional evaluation can be executed with this table, for example, by means of ACCESS. The file is a ring buffer with the length "Max_Ds". The operator must ensure that the file is deleted after processing.

Method 2: Free logs from step logs via filter table

This generalized and comprehensive data extraction from the step logs is discussed in the sections below.

12.3.2 Filter criteria

Following parameters of a step telegram can be used as a filter criterion:

- PCU number
- unit number
- EOP number
- recipe number

For EOPs (steps) that are used in several recipes (units) or more than once in the same unit, it is also possible to specify whether or not values of this EOP are stored. One of the DFMs of an EOP is reserved for this purpose. The DFM number can be easily selected. The user can apply a filter ID which is provided as a setpoint value (DFM) of a GOP (step). This filter ID is also configured within the filter table.

12.3.3 Installation

To make use of the "Free protocols via step protocols" functionality, you must do the following:

- Copying required files from the project template to the current project
  ...<sys-path>\!BM_Proj\FRPROT\GENFRPR.DBF → ...<proj-path>\frprot\
  ...<sys-path>\!BM_Proj\FRPROT\GENFRPR.DEF → ...<proj-path>\frprot\
  ...<sys-path>\!BM_Proj\FRPROT\GENFRPR.MDB → ...<proj-path>\frprot\
  ...<sys-path>\!BM_Proj\FRPROT\GENFRPR.LDB → ...<proj-path>\frprot\
  Files GENFRPR.MDB and GENFRPR.LDB are not required for proper functioning; they merely provide a relatively convenient MS-Access screen form for the filter and assignment tables.
- To activate the function, the following switches must be set in the configuration file for step reports '...<proj-path>\sys\SRPROT5.INI':
  ```
  [sr_frpr]
  ;Activate the SR_FRPR.DLL: Copy some datasets from the step-
  reports into a special archive
  Akt=1
  ;0 = forwarding disabled; 1=forwarding of step datagrams to module
  free protocols enabled
  ForwardDGrams=1
  ```
- The filter definitions and the source/destination assignments must be created in the GENFRPR.DBF table.

12.3.4 Structure of the filter table GENFRPR.DBF

Basics

Type	Field name		Field type	Meaning	Min. value	Max. value	DEF file
F	PCUNR	*	Numeric	PCU number	1	255	CINT 5
F	TEILANLNR	*	Numeric	Unit number in the step telegram	1	64	CINT 3
F	BOPNR	*	Numeric	EOP number	1	32.767	CINT 5
F	REZEPTNR	*	Numeric	Master recipe number (A recipe number exactly matches recipe number in the step frame)	1	32.767	CINT 5
F	SCHRITTNR	*	Numeric	Step number to distinguish of EOPs with identical EOP numbers	1	32767	CINT 5
F	FILTER_ID	*	Numeric	ID, if EOP uses the DFM x.y	1	"__int32"	CINT 10
F	FILTERDFM	*	Numeric	DFM x.y, (= x*256 + y) e.g DFM3.255 = 1.023 Only relevant if FilterID was defined and doesn't equal 0	0	1.023	CINT 5
Q	PARAM_TYP		Numeric	Parameter type: 0 = actual value 1 = setpoint value 2 = EOP start time 3 = EOP end time	0	3	CINT 5
Q	PARAM_IDX		Numeric	Only relevant when PARAMTYPE=actual/setpoint value 0 = step run time 1..20=Parameter number (setpoint 1..20)	0	20	CINT 3
Z	DATEN_TYP		Numeric	0 = Int16 (5 characters) 1 = Int32 (10 characters) 2 = text (Value is interpreted as an index, see below: parameter "TEXT FILE") 3 = time,format "dd.mm.yyyy hh:mm:ss", 4 = time, format "dd.mm.yyyy hh:mm" 3 and 4, when PARAMTYPE = start or end time EOP 5 = time, format "hhhh:mm:ss" 6 = time, format "hhhh:mm" 5 and 6 when PARAM_IDX = 0 (Step operating time) and PARAMTYPE = 0/1 (setpoint/actual value)	0	3	CINT 6
Z	SPALTENAM		Character string	Field name (name of the column) of the parameter in the db file, uniquely describes a column in the above table	---	---	CHAR 9
Z	TEXTDATEI		Character string	Text file name without TXT extension Only available when DATA_TYPE = Text, is otherwise ignored	---	---	CHAR 9

Legend:

- F Filtering criteria (conditions for the storage of a value from a frame)
  * If criteria = 0 then not relevant, if <> 0 then relevant and will be used as filter criterion
- Q Specifications of the source (What is supposed to be stored?)
- Z Target information (Where should the information be stored to?)

Note:

- The step number is not evaluated currently, as it is not included in a step data gram in the current version of BRAUMAT/SISTAR Classic V6.0. therefore it cannot be used as filter criteria at the moment!
- A line within the filter table always describes only one data item from a step telegram. If for example several DFM setpoints or actual values of the same EOP need to be stored, several lines must be created.
- The Filter DFM means the number of the DFM is only necessary if a Filter ID was indicated (Filter ID > 0), otherwise both values are ignored.
- PCU number, EOP number, recipe type and recipe number (= master recipe number) are ignored if they are 0. If they are larger than 0, they are used as a filter criterion.
- The type of parameter (PARAMTYPE) and parameter index (PARAM_IDX) indicate which data element is supposed to be stored from the step telegram.
- The parameter index is only relevant if the type of the parameter indicates an actual or setpoint value to be stored.
- The parameter table type (TABLETYPE) and column name (COLNAME) of the filter table indicate where the value from the step telegram is supposed to be stored to. The data type (DATATYPE) indicates in which format the value is stored. If the data type "text" is specified, the parameter "text file name" (TEXTFILE) also has to be configured. The value from the step telegram is taken as an index within the given text file. The character string from the text file is stored in the dBASE files of the Free def. log and not the value itself.
- The table type (TABLETYP) corresponds to the file FEPR_<tabletype number>.DEF, which describes the structure of the files of the Free def. log, for example FE_<order number>.DBF.
- The column name (COLNAME) is the name of the column in one of the files of the Free def. logs; this name also appears in the description file (FEPR_<table type>.DEF). If a column name is specified, which does not exist in the description file, then a warning message (WARNING) is stored into the trace file (see chapter "Diagnosis")
- If a file name is entered for TEXTFILE that does not exist in the text directory (`...<proj-path>\texte.y\` or `...<proj-path>\pcu.xxx\texte.y\`), then a warning message (WARNING) is stored into the trace file (refer to the section "Diagnostics")
- If a configured value exceeds the valid range for one of the parameters, then an error trace (ERROR) is stored into the trace file (see chapter "Diagnosis")
- There are also still some "specific data types":

Action	INT16	INT32	Meaning
---	DT_INT16 (0)	DT_INT32 (1)	New value = value from telegram
Addition	DT_INT16_ADD (65.536)	DT_INT32_ADD (65.537)	New value = old value + value from telegram
Subtraction	DT_INT16_SUB (131.072)	DT_INT32_SUB (131.073)	New value = old value - value from telegram

Tab title: Table : Data types - Addition and subtraction with old values

- Old value: Value from dBASE file (Value which is read from record)
- New value: Value for dBASE file (Value which is written in record)
- Value from telegram: Value from the step telegram

Topic deleted: Assignment table PCU/Unit to recipe type number

12.3.5 Examples

Limits

There are four cases:

Case	Procedure	Control recipe	Description
1	EOP 1 EOP 2 EOP 3	EOP 1 EOP 2 EOP 3	Only different EOPs are used in the procedure none of which are multiplied (master recipe without dosing EOPs, old recipe system 2.13)
2	EOP 1 EOP 2 EOP 1	EOP 1 EOP 2 EOP 1	EOPs are used several times in the RP, therefore also in the control recipe. Distinction through Filter DFM and Filter ID.
3	EOP 1 EOP 2 EOP 3	EOP 1 EOP 2 EOP 2 EOP 2 EOP 3	Only different EOPs are used in the RP, one or more of which is however multiplied in the control recipe, for example dosing EOPs, as is possible in the new recipe system (version 3).
4	EOP 1 EOP 2 EOP 3 EOP 2	EOP 1 EOP 2 EOP 2 EOP 2 EOP 3 EOP 2	EOPs are used several times in the RP, therefore also in the control recipe. In addition, EOPs are multiplied for the control recipe (most complex case)

The cases 1 and 2 are currently covered by the functionality described here. In case 3 and 4, the **multiplied EOPs cannot be distinguished**, since only one EOP exists in the recipe procedure.

These EOPs cannot be distinguished by the step number since the step number is **not** part of the step frame.

These EOPs cannot even be distinguished based on the mechanism of the filter DFM or the filter ID.

Saving the start time of a specific EOP that appears only once in all sequences of a PCU

Type	Field name	Entry	Example	Remarks
F	PCU_NO	Yes	3	
F	UNIT_NO	Yes	30	
F	EOP_NO	Yes	199	
F	GR_NO	Yes	1.000	
F	FILTER_ID	No	0	Because EOP appears only once in all sequences
F	FILTER_DFM	No	0	Because Filter ID is not used
Q	PARAMTYPE	Yes	2	Start time of the EOP
Q	PARAM_IDX	No	0	Because PARAMTYPE is neither 0 nor 1 (actual or setpoint value)
Z	DATA_TYPE	Yes	3	Time
Z	COLNAME	Yes	"start time"	Always necessary
Z	TEXT FILE	No	""	because start time is not reasonable as an index in a text file

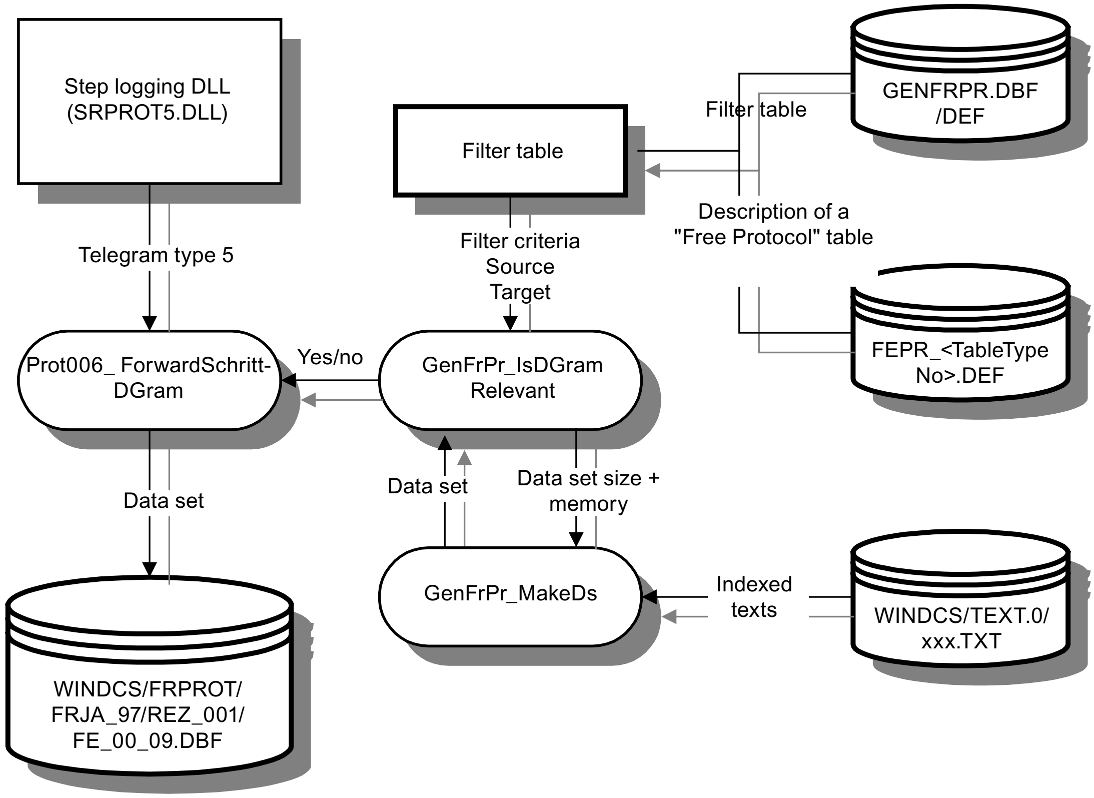
Saving the end time of a specific EOP that appears in several sequences (of the same PCU!) (filter DFM is DFM 3.255)

Type	Field name	Entry	Example	Remarks
F	PCU_NO	Yes	3	
F	UNIT_NO	Yes	30	
F	EOP_NO	Yes	199	
F	GR_NO	Yes	1000	
F	FILTER_ID	Yes	10.000	Because EOP appears several times in all sequences.
F	FILTER_DFM	Yes	1.023	The EOP is selected that uses this FILTER_DFM. All other EOPs with the same EOP number must not use this FILTER_DFM -> 3.255 = 3*256 + 255 = 1.023
Q	PARAMTYPE	Yes	3	End time of the EOP
Q	PARAM_IDX	No	0	Because PARAMTYPE is neither 0 nor 1 (actual or setpoint value)
Z	DATA_TYPE	Yes	3	Time
Z	COLNAME	Yes	"Finish at"	Always necessary
Z	TEXT FILE	No	""	Not relevant, because end time is not logical as an index in a text file

Saving the actual value of the fourth DFM (DFM number = 2.44) of an EOP that appears only once in the sequences of the PCU

Type	Field name	Entry	Example	Remarks
F	PCU_NO	Yes	3	
F	UNIT_NO	Yes	30	
F	EOP_NO	Yes	199	
F	GR_NO	Yes	1000	
F	FILTER_ID	No	0	Because EOP appears only once in all sequences
F	FILTER_DFM	No	0	Because Filter ID is not used
Q	PARAMTYPE	Yes	0	Actual value of a DFM of an EOP
Q	PARAM_IDX	Yes	4	4th DFM
Z	DATA_TYPE	Yes	1	32-bit value
Z	COLNAME	Yes	"DFM4"	Always necessary
Z	TEXT FILE	No	""	Not relevant, because actual value is not used as an index in a text file

12.3.6 Picture showing the interaction



12.4 Stepping log

12.4.1 Overview

Record

The setpoints and actual values occurring during processing of the units such as times, quantities, temperatures etc. and information about the executed equipment operations are stored in files chronologically.

- The storing of data is done after a step change has finished.
- The data are stored into dBASE files.
- The order and batch number are unique keys to the stored batch data.
- The batch data can be printed at any time or deleted.

Data

The step log is recorded automatically by the system. The following data is logged on each step:

- Start time (year, month, day, hour, minute, second)
- Batch ID (year, recipe category, job and batch number)
- Recipe name
- PCU and unit name
- Target and actual time of the step
- For each setpoint
  - Name
  - Dimension
  - Setpoint and actual value at step end

Visualization and printout

The application 'Step logs' has to be used for the visualization and printout.

- The batch data can be printed at any time or deleted.
- A complete batch print-out can also be activated via the PCU. The flag M101.2 (PrintStepProt) must be set for this purpose via the flag M102.3 (EopStop) in the last EOP of a batch to be reported.

12.4.2 'Step logging' program

12.4.2.1 Functionality

Basics

The Step logging program is used to view and print Step log files.

Client area

The client area can be split into these sections:

- Toolbar
- Display area for step log entries
- Status bar
- Function key line

Table 12-6 Display area

Column 1: Step	Continuous numbering of the individual positions for equipment operations
Column 2: EOP	Number of the equipment operation
Column 3: User ID *)	ID / Identification for user log entry
Column 4 : Name	EOP text that describes the equipment operation
Column 5 : Date	Date of the step log entry
Column 6 : Time	Time in EOP name line: EOP start time Time in actual value line EOP end time / for user telegrams: Telegram trigger time
Column 7: Time	EOP runtime setpoint EOP runtime actual value / for user telegrams: Telegram trigger time
As of column 8: Setpoint 1 ... Setpoint 24	Name, setpoint, actual value

*) Only with enabled option "User logs" :

- The "User ID" is displayed in column 3. This column is empty for normal system step log entries.
- User logs are shown with a gray shadow to distinguish them from system logs.
- User logs are additional EOP-specific step log entries. These can be triggered in the user program code of the EOP FC (see section User step log (Page 671)).

12.4.2.2 Working with the program

Your work with the **step logging** program comprises the following tasks:

Selecting the step log file

Select **File** -> **Open**, then press F2 or click the **OPEN FILE** button to open the **Select** step log file dialog box.

Options available:

- Year
- Recipe category
- Order number
- Batch number
- Unit / PCU

Confirm your selection with **OK**.

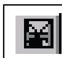
Printing step log files

Select **File** -> **Print**, then press **F4** or click the **PRINT FILE** toolbar button to open a dialog box for setting up the following functions:

Print:

- Log file of the selected unit
- All unit log files.

Deleting step log files

	Select **File** -> **Delete**, or double-click the **DELETE FILE** toolbar button to delete logs from the hard disk drive. Select the log files to be deleted in the dialog box.

Order in which you select log files you want to delete:

- Year
- Recipe type
- Order number
- Batch number

Access to the Delete log files menu can be password protected.

Input of comments

A comment can be called for the current order number. Select **Options** -> **Comment**, or press **F10** to open the comment file for the current batch number.

12.5 Logging of user operations

12.5.1 Overview

The system logs operator activities in the process as well as changes in parameterization and the recipes. The changes are structured like messages and stored in dBASE files.

This function covers:

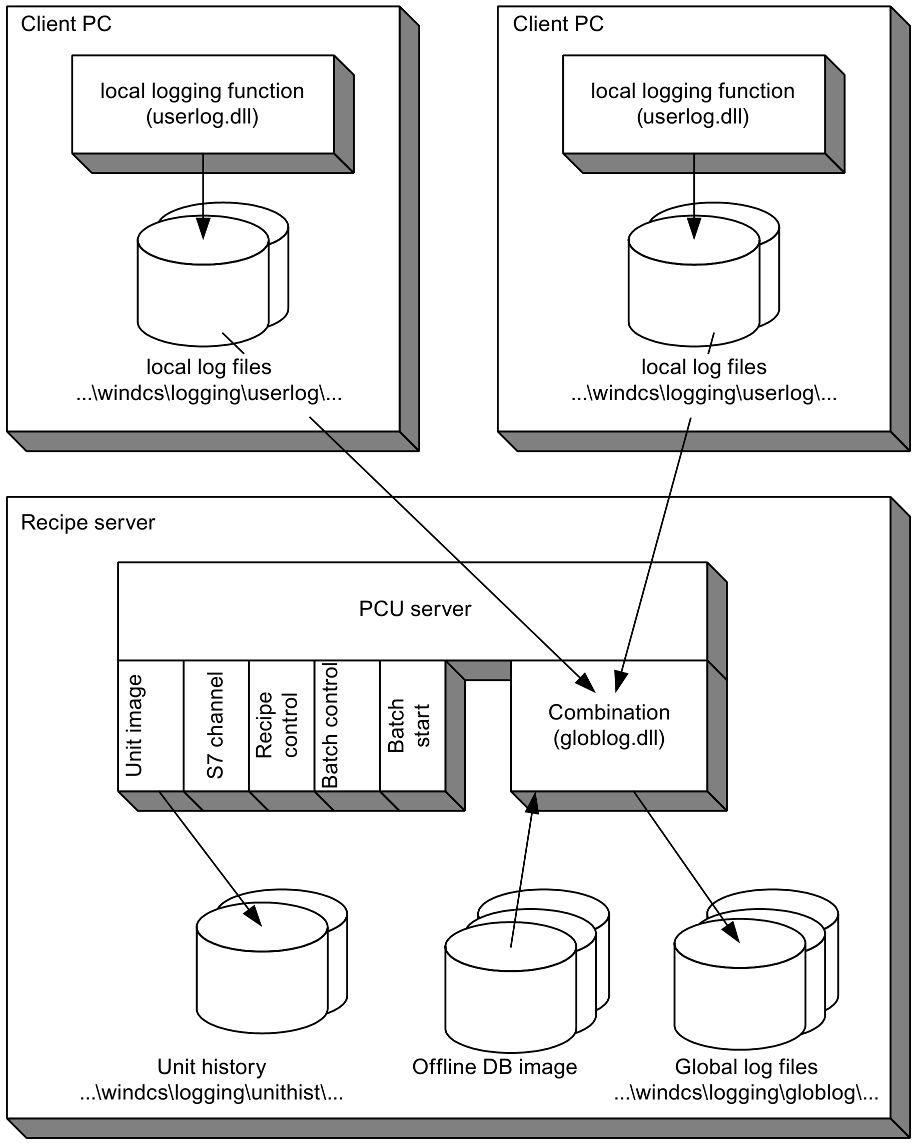
- local logging of changes on a client and/or on a server
- Recording of a unit history
- Joining several local files into one ‘global’ file
- Apply batch relation to changes

Unit history

The history is recorded from the unit data image. The following changes are logged:

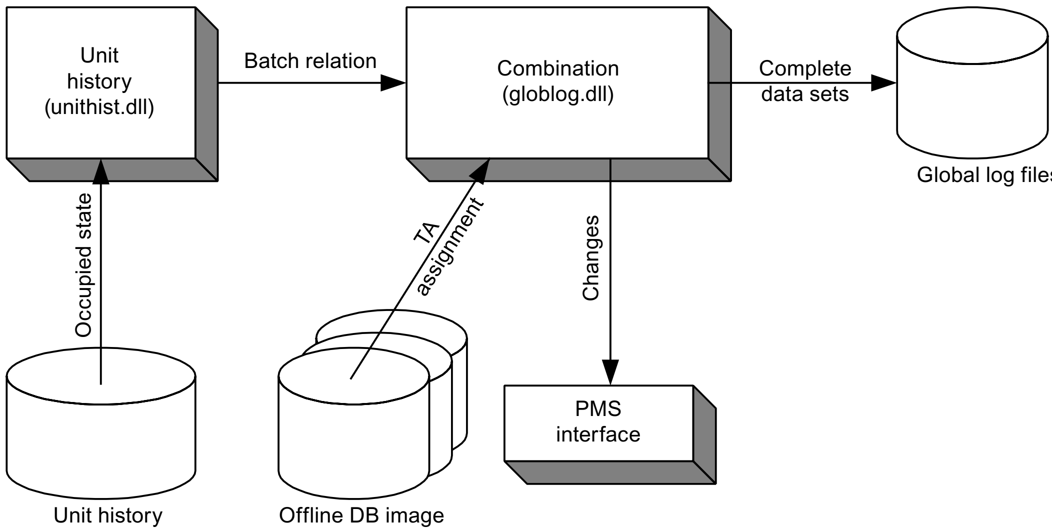
- Unit starts (leaving step 0 and going to a step unequal to 0; start and allocation)
- Unit terminates (goes from step not equal to 0 into the step 0; end and de-allocation)
- Unit changes its operation mode (automatic or manual)

- The fault indicator of the unit changes its status
- The running condition of the unit changes its status



The changes are stored on every client temporarily. On every AREA server a service is running, which combines the clients' changes referred to this AREA and applies the batch relation (if possible). The offline data blocks are used to get the batch information.

12.5.2 Server functionality



On the server a service is running which combines the changes made for that AREA on the clients. The unit history dll also runs on the server. For all changes related to objects with a unit reference (ICMs, regulators...), the batch information is added. For that, the unit assignments are read from the offline data blocks. If objects are assigned to units in a dynamic way, it is not possible yet to get the correct batch reference.

Note!

It is the user's responsibility to keep the offline data blocks up-to-date.

12.5.2.1 Configuration of the unit history

Basic principles:

To get the reference from the changes to a batch, a history of the unit allocation must be recorded. The batch data is recorded for every start and every termination of a unit.

Enable logging of the unit history

File: "<proj-path>\sys\Logging.ini"

```
[Global]

EnableUnitHistory=1
```

Location of the files

The files are stored in folder '<proj-path>\logging\unithist' in accordance with the structure described below.

- One subfolder per year: <...\UH_yy>
- A separate file is created for each day inside the "year" folder: <UHyymmdd.dbf>.

Example: '<proj-path>\logging\unithist\UH_02\UH021221.dbf'

12.5.2.2 Configuration of the server function

Basic principle

For the server function, the AREA memberships and computer names have to be fully configured in the configuration tool "SiteCfg.exe".

Cycle time for the combination of the changes

File: '<proj-path>\sys\logging.ini'

```
[GlobLog]
```

```
CycleTime=10
```

Deactivation of the globlog.dll

The server function can be disabled via a switch, which is read also at runtime - a restart of the PCU server is not necessary.

File: '<proj-path>\sys\logging.ini'

```
[Global]
```

```
EnableGlobalLogging=0
```

Note

The change log service is running only when the "Recipe Server" mode is activated on the Server IOS. Consequently only the active server on a redundant server pair runs that service.

Data storage:

The UserLog and UnitHistory files are combined to create a 'globlog.dbf' file.

The files are stored in folder '<proj-path>\logging\globlog\...' with the following structure.

- One subfolder per year: <...\GL_yy>
- A separate file is created for each day inside the "year" folder: <GLyymmdd.dbf>.

Example: '<proj-path>\logging\globlog\GL_02\GL021221.dbf'

Redundancy

On a redundant system the data is copied to the standby machines by the active server. The necessary path specifications for this are read from the area.ini.

12.5.3 Client functionality

Basic principle

All the system applications log the changes made to them in files that are stored locally in the client's project folder. The user currently logged in is also saved after each change entry. For changes, the time of the client computer is entered.

Enable user logging on a client

File: '<proj-path>\sys\Logging.ini'

```
[Global]

EnableUserLogging=1
```

12.5.4 Which changes are recorded?

Changes to parameterization

Changes to data records of the block classes (old and new values each) via the parameterization application (Param.exe) are recorded. However, this concerns the ‘Online mode’ of parameterization only.

Note on text parameterization

The changes to the text parameterization are not logged.

Runtime process images/faceplates

In the faceplates of the runtime process images "LzSys.exe", all parameter or value changes of the linked variable tags or block classes are logged with old and new values each. Particularly the following controls are concerned:

- Analog.ocx/Analog2.ocx
- Controller.ocx
- Digital.ocx
- ICM.ocx
- UnitCtrl.ocx
- SmartUnit.ocx

Recipes

The following changes to a recipe are logged:

(*) are not available in this release

- Status change
  Recipe procedure
- Changes to ROPs
  TUE
- Changes to the setpoint values
  Value
  EditNr. (Replacement, Scaling etc.)
  Status
- Changes to the recipe structure
  Deletion of an ROP
  Insertion of an ROP
  Deletion of an SW
  Insertion of an SW
  RP Type (Simple,(*)Free candidates, (*)XStreaming, Tracls)
  Deletion of a TRP
  Insertion of a TRP
- Unit recipe procedure (TRP) (RUP)
  Name of the TRP
  Change of position
  TRP deleted
  TRP inserted
  (*)Unit class of the TRP
  (*)Type of the candidates (Reference, automatic, manual)
  Deletion of a candidate
  Insertion of a candidate
- Track - Recipe procedure (RP)
  Change of the reference lines
  Reference line virtual/not virtual
  Number of the parallel line
  Deletion of a parallel line
  Insertion of a parallel line
- Master recipe (GR)(MR)
  Note: Changes are logged only in MR states not equal to ‘In work’!
  Name of the MR
  Full name
  Nominal batch size
  RP assignment
  PIL configuration: Insert an input material
  PIL configuration: Delete an input material
  PIL configuration: set quantity of input material
  PIL configuration: Dosing group
  PPL configuration: Number of process parameters
  PPL configuration: Value of the PP
  PPL configuration: PP Scaling

- Recipe procedure (RP)
  Name
  Recipe category
  minimum, maximum, nominal batch size
  Cycle time
  Start unit
  Start PCU
  Creation date and "creator"
  Date of change and user name
- DFM – Definition
  Change of the definition
- Technical operation (TOP)(EOP) – Definition
  Type (weighing, standard, etc.)
  Unit of assignment
  Number of setpoints
  Setpoint values
  Editability of the setpoints
  TOP - Name

Route Control System

The following conditions are logged in RCS-Online and in Route Control OCX:

- Switch route to manual/auto
- Manual mode requirements
- Manual destination requirement
- Manual request and start of routes

12.5.5 Structure of the files

12.5.5.1 Columns of the log files

The columns of the local and global change files are identical. At the local definition the last column 'AddCounter' is left.

Column	Type	Remark
Time_GMT	INT 11	Date and time according to GMT
Date_Time	TEXT 20	Date and time according to the current IOS configuration
Log_Type	INT 3	Type of entry
Usr	TEXT 20	User who made the change
IOS	TEXT 16	Name of the IOS
IOS_No	INT 5	IOS number on which the changes were made
PCU	TEXT 16	Name of the PLC
PCU_No	INT 5	Number of the PLC
Block_Typ	INT 3	Block type

Column	Type	Remark
Block_No	INT 5	Block number
DW_No	INT 5	Data word
Bit_No	INT 5	Bit number
Unit	TEXT 16	Name of the unit assigned to the object
UnitNo	INT 5	Number of the unit assigned to the object
Modules	TEXT 12	Name of the module which was modified
Object	TEXT 16	Name of the object which was modified
ObjectNo	INT 5	Number of the object
Attributes	TEXT 16	Attribute that was modified
RecType	TEXT 16	Name of the recipe type
RecTypeNo	INT 5	Number of the recipe type
BatchYear	INT 5	Year when the batch was created
OrderNo	INT 5	Order number of the batch
BatchNo	INT 5	Batch number
OldValue	TEXT32	old value
NewValue	TEXT32	new value
Text	TEXT64	Additional information about the changes
AddCounter	INT 5	number to make identical entries unique

12.5.5.2 Columns of the units history

…

Column	Type	Remark
Time_GMT	INT 11	Date and time according to GMT
Date_Time	TEXT 20	Date and time according to the current IOS configuration
PCU_No	INT 5	Number of the PLC
Unit_No	INT 5	Number of the unit assigned to the object
UnitState	TEXT 8	Status of the unit as a hexadecimal number
Step_No	INT 5	Step number
Rtype_No	INT 5	Number of the recipe type
Rec_No	INT 5	Number of the recipe
BatchYear	INT 5	Year when the batch was created
Order_No	INT 5	Order number of the batch
Batch_No	INT 5	Batch number

12.5.6 Logging the last most significant recipe modifications

When the logging function is activated and a recipe procedure is saved, not only are entries made in the Userlog file and the Globlog file, but the change with the highest priority is also written to the master recipe header. This entry consists of the modified text and the ID.

In the Manufacturing Execution System (MES), an entry is made in the MES coupling table when the data are saved, and the entry is downloaded to the master system.

Of several changes to the recipe system made in a session between two archiving operations, only the changes with highest priority are saved. The priorities of the changes categories are found in the table. Existing entries are overwritten by the new data when saved again (LMSM Logging: Last Most Significant Modification).

Change	Priority	ID
RP production enables status is set or reset	Maximum	32767
RP type modified		70
TRP deleted		60
New TRP added		50
TRP deleted		40
New RUP added		30
Recipe updated		20
Setpoint modified	Minimum	10

12.6 Archive display

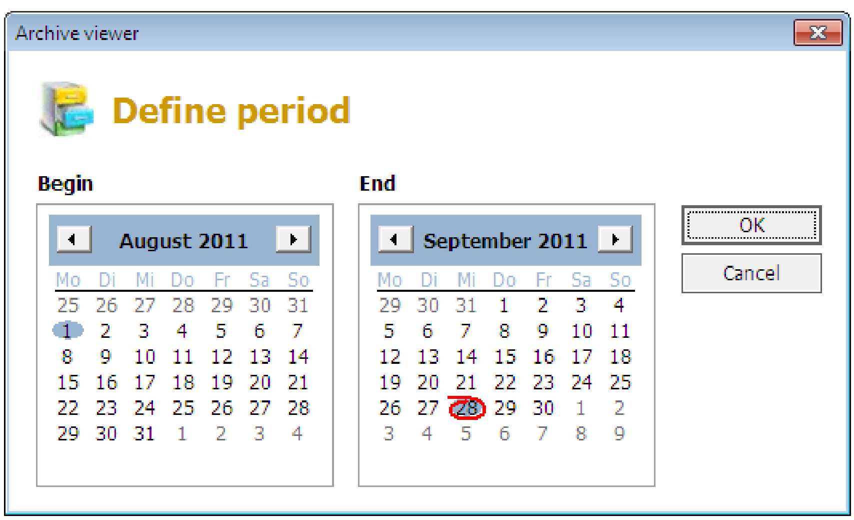
Basics

Open the dialog via General Menu / Archive / Changes and Messages.

The application allows you to display and print messages as well as any recorded process operations and configuration changes from the same time period.

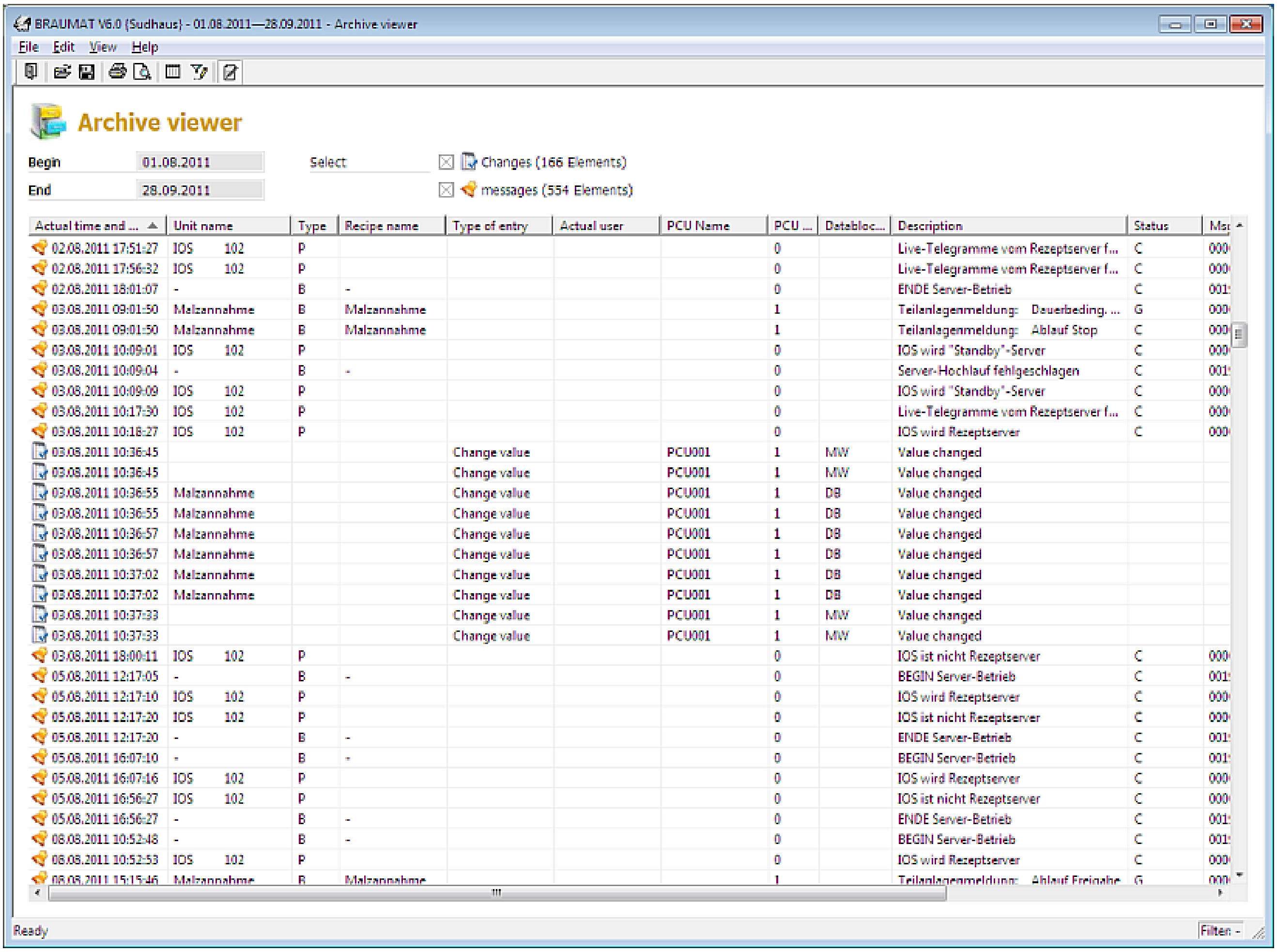
- You can apply your own filter criteria and column settings for searching and displaying the associated archive files for the selected type and period (based on days). It is also possible to sort the archive files by column within this context.
- The currently selected settings for archive selection, column selection and sorting are saved for the next call of the application.
- When starting this application as message archive from the "PCU server" application, only the messages are displayed and the sorting takes place by date/time in descending order, regardless of the saved setting.
  → The new command line switch for "MeldArch" operation is "/meldarch".

Specifying the period



- You can open the dialog in the following ways:
  – Menu command: "File/Open"
  – Toolbar: "Open" icon
  – Double-click the begin/end date fields in the main view
- To specify the archive selection period, select the month, year, and day in the "from" and "to" fields in the relevant calendar.

Viewing an archive



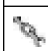
The main window contains the following elements:

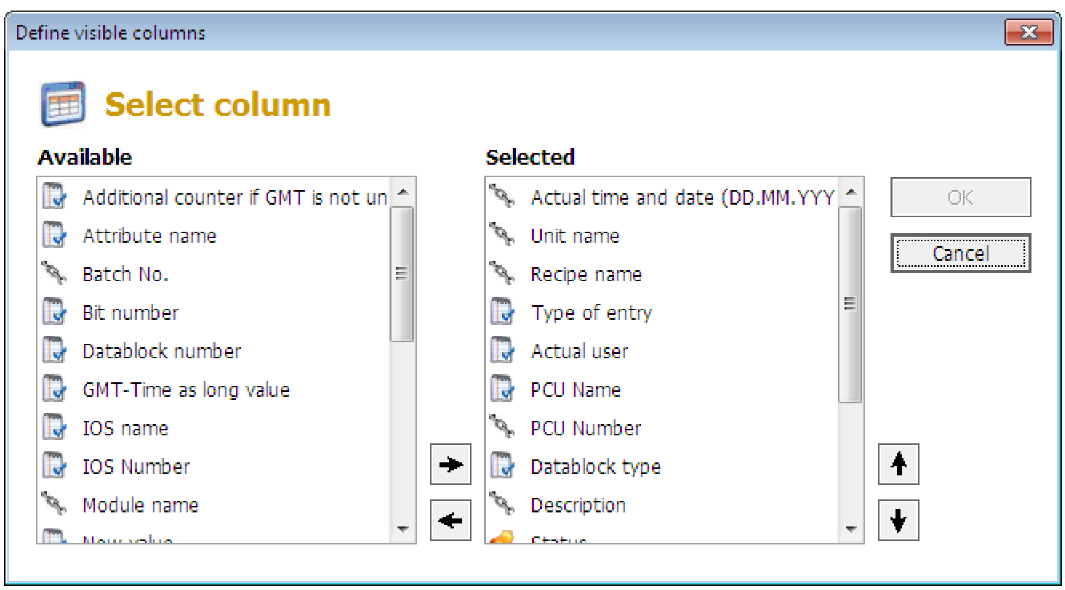
- The start and end dates for the selected period are displayed
- Selection:
  Allows you to select messages, process operations, and configuration changes
  - Changes → Process operations and configuration changes from the '...<proj-path>\logging\globlog\GL_yy\GLyymmdd.dbf' archive files are displayed
  - Messages → Messages from the '...<proj-path>\meld\ME_yy_mm\MAyymmdd.txt' message archives are displayed
  - If you activate both archive types, the relevant data sets are displayed together in tabular format and in chronological order. The symbol at the start of the line identifies the type of archive concerned.
- You can select the columns you want and the order in which they should appear (toolbar → "Select columns")

- The lines can be sorted by column in ascending or descending order (click the relevant column header)
- Only the lines that meet the adjustable filter criteria are displayed (toolbar → Filter output)

12.6.1 Selection of columns

The "Define visible columns" dialog allows you to select which columns should be displayed in the main window ("Selected" list) as well as the order in which they appear. The columns that are available for the selected archive type are all listed together and are marked as follows:

Symbol	Archive type
	Column available in changes archive
	Column available in message archive
	Column available in changes archive and message archive

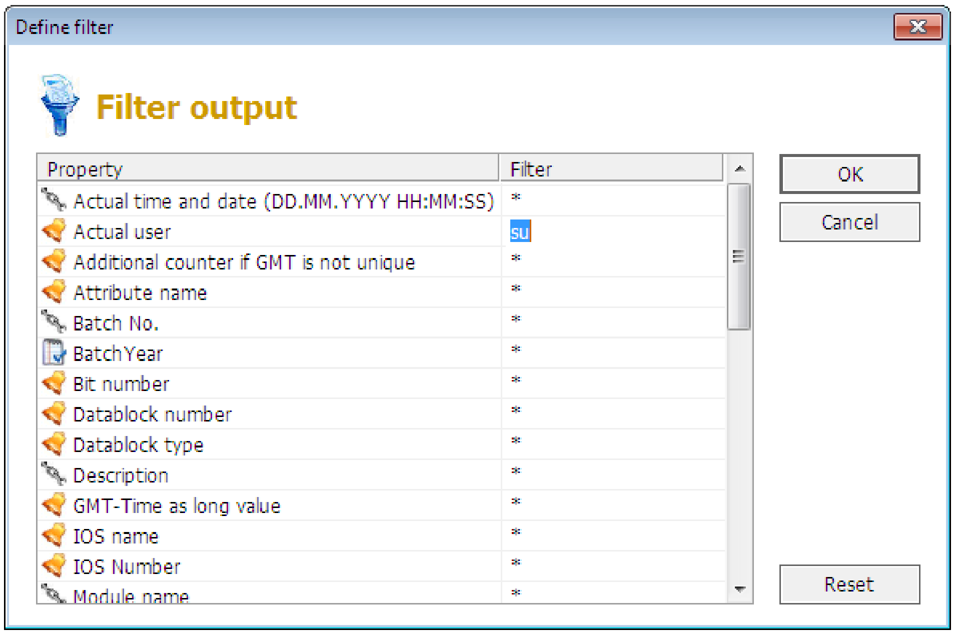


- Select the elements you require and then use the → / ← arrow key to move them into the relevant list box. In the "Selected" list, the elements are inserted **below** the item that is currently highlighted.
- Select the elements in the "Selected" list box and use the ↑ / ↓ arrow key to move them to the required column position.
- Click "OK" to save the drop-down list.

12.6.2 Filter output

Filter output

You can use the "Define filter" dialog to apply filter criteria for displaying and printing specific lines.



Procedure

- Only those elements defined using the "Select column" function are available
- A filter string can be defined for each element (default ='*')
- Filter syntax
  - All the characters leading up to the "*" are included in the comparison
  - The "*" must always be placed at the far right-hand end
  - The function is case-sensitive
  - The filter strings are ANDed
- Use RESET to set all the filter strings back to "*" (= Show all)
- The filter is applied as soon as you quit the dialog by pressing OK

Example filter criteria

- Filter set to "MT*" unit → The search function finds all the texts that start with "MT"
- Empty filter string '' → The search function finds all blank texts
- Filter set to the specific IOS name "IOS1" → The search function finds all the texts that match "IOS1" exactly

12.6.3 Printing

The selected columns can be printed on the standard printer.

Using the menu item 'printer setup', a different printer can be selected.

Applications for diagnostics and maintenance 13

13.1 Diagnosis recipe control 'RecContr'

For the diagnosis of the recipe control in the recipe server there is the recipe control application 'Diagnosis recipe control' (reccontr.exe).

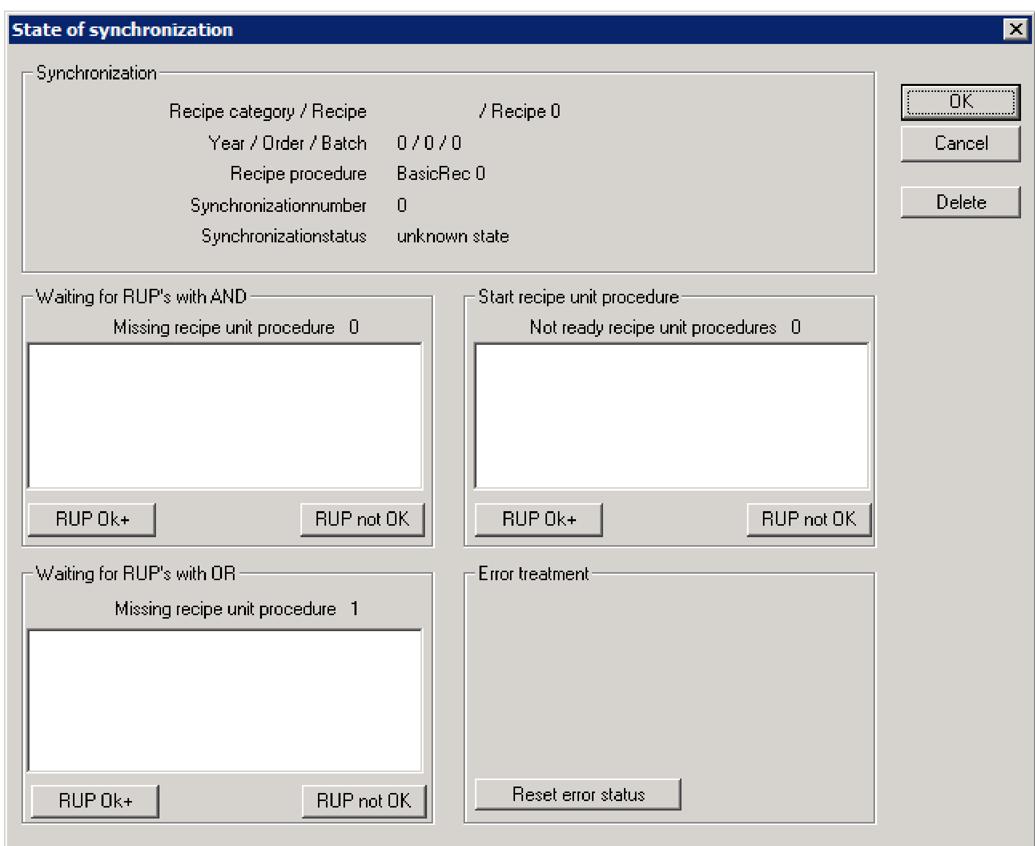
In this application the status of synchronizations and alternatives can be queried.

Under the menu 'file', there is the selection of synchronizations and alternatives.

Every synchronization or alternative is marked by the year, the recipe category, the order and batch number and the synchronization number.

13.1.1 Synchronizations

Dialog selection: File\Synchronization\OK Button



Synchronization

Here, the data of the synchronization is displayed.

Waiting for units AND

Here the units are listed on which AND is waited for. 'OK ' or 'WAIT' can be found behind the unit

- OK: Unit is already at the synchronization
- WAIT: Unit hasn't already reached synchronization

Activating sequences

Here, the units are listed which are supposed to be started with the synchronization

- OK: Unit is ready to start
- WAIT: Unit is occupied, is set to hand, hold or the permanent condition is missing

Waiting for units OR

Here, the units on which OR is waited for are listed. 'OK ' or 'WAIT' can be found behind the unit

- OK: Unit is already at the synchronization
- WAIT: Unit hasn't already reached synchronization

Button 'RUP OK +'

With this button the status of a unit can be set for the synchronization to OK . A following unit telegram overwrites the status.

Button 'RUP Not OK'

With this button the status of a unit can be set for the synchronization on Wait. A following unit telegram overwrites the status.

Delete

With the button the synchronization can be deleted. Following unit telegrams create the synchronization again.

Reset error state

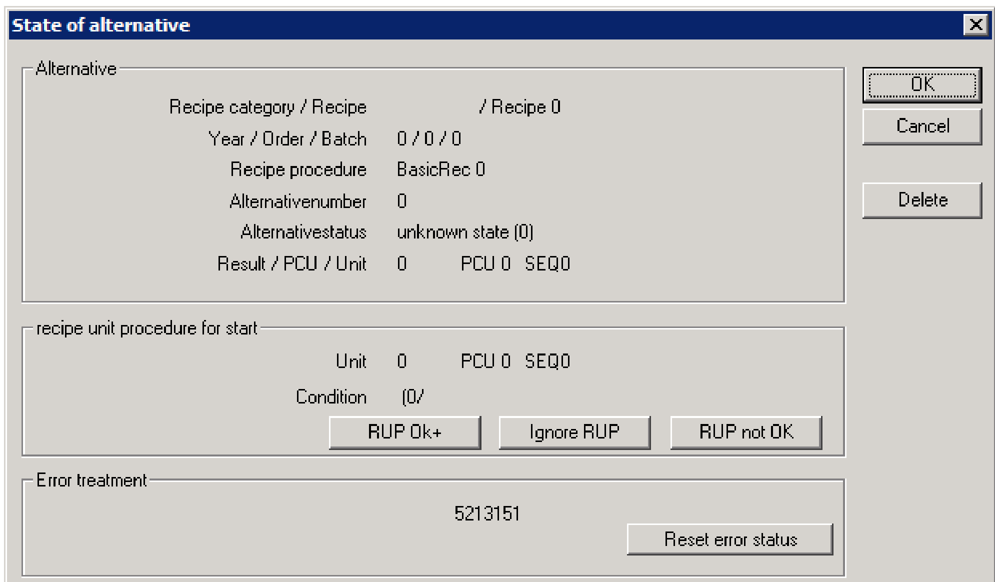
With the button, the current error status of the synchronization is deleted.

13.1.2 Alternatives

Alternative

Here the data of the alternative are displayed.

The meaning of the buttons is the same as the synchronization.



13.2 PCU Server dialog 'Recipe control'

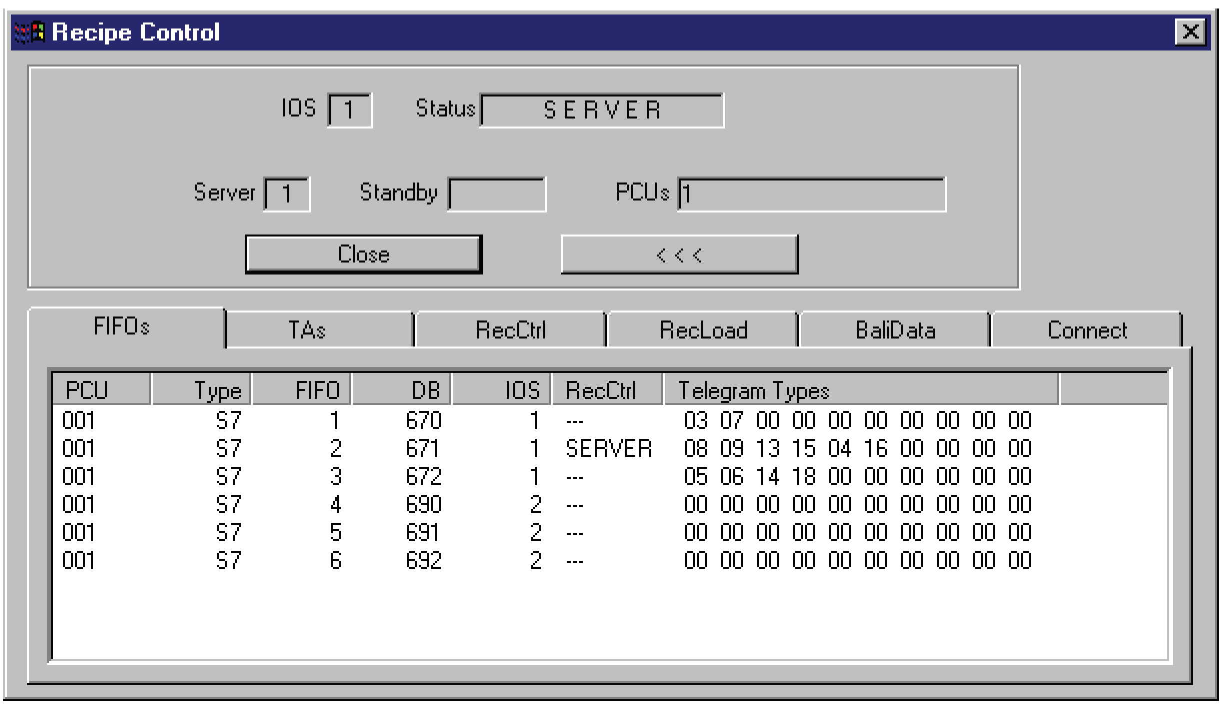
13.2.1 PCU Server

On the active recipe server, a diagnosis window can be opened via the menu item 'Options'->'Recipe control' in the application PCU server.

Via the button '>>>', an expanded representation can be opened.

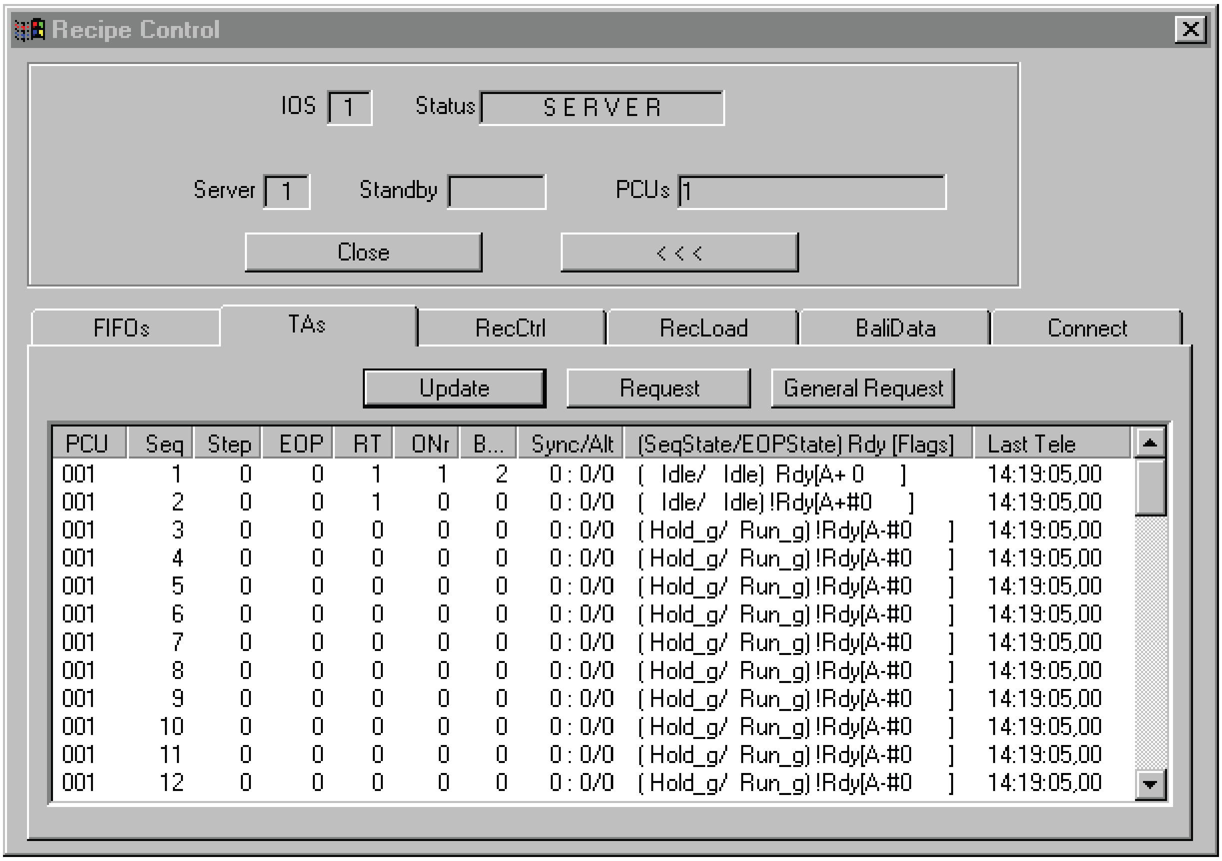
13.2.2 FIFO allocation

In the view, the FIFO-allocation is displayed to all PCUs.



13.2.3 Sequence image

In this window (TAs), all sequences of all PCUs are displayed.



The views correspond to a large extent to the signaling in the plant overview. The data are transmitted via the telegram type 13 of the SIMATIC.

Update

The view is built up from the sequencer image again.

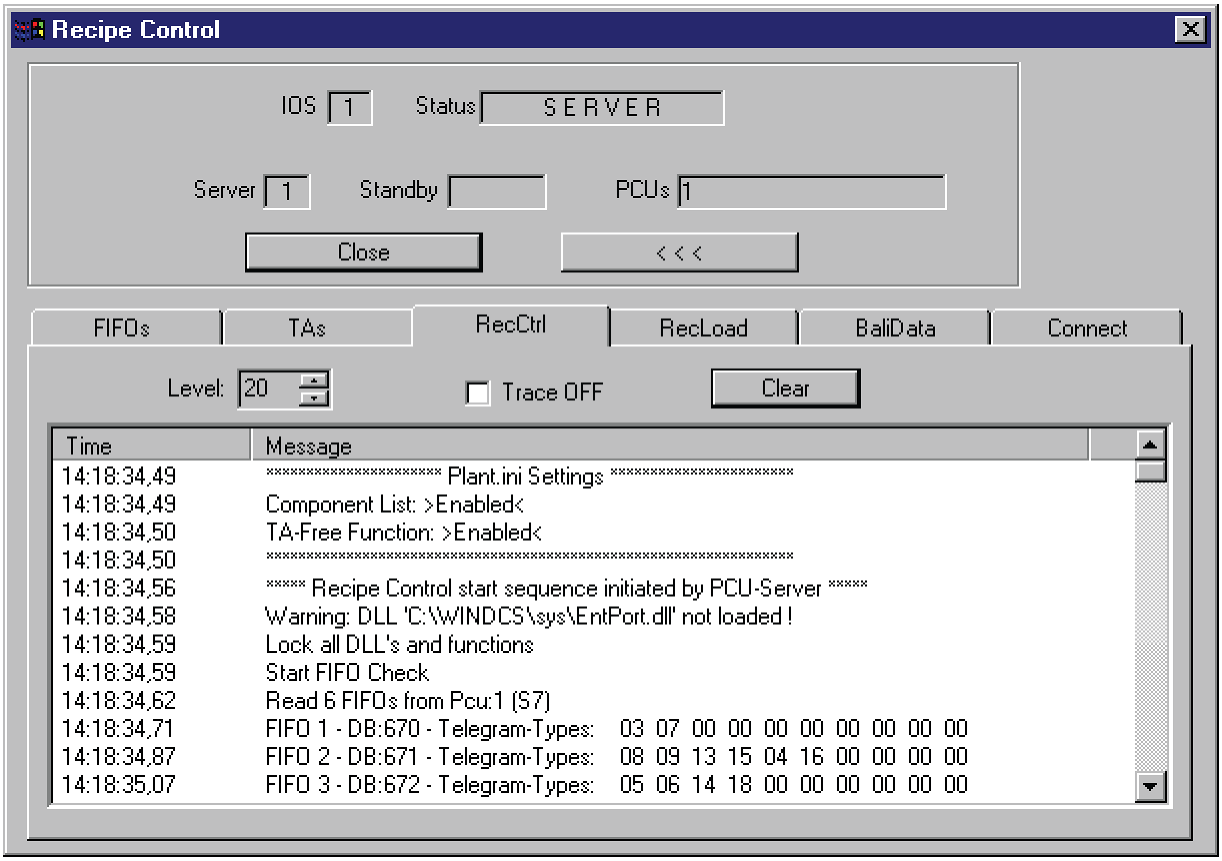
Request

Data for the selected sequence are newly requested by the SIMATIC.

General request

Data of all sequences are newly requested by the SIMATIC controls.

13.2.4 Recipe control diagnosis



In the window (RecCtrl), diagnosis messages relating to the server startup and the recipe control are output.

Level

The messages are grouped with a different level. It is determined, which messages are displayed by the indication.

- With level 100, all messages are displayed.
- With level 0, no messages are displayed.
- Start messages have the level 5
- Error reports have the level 10
- Repairable errors have the level 20
- above this limit, there are information and other messages

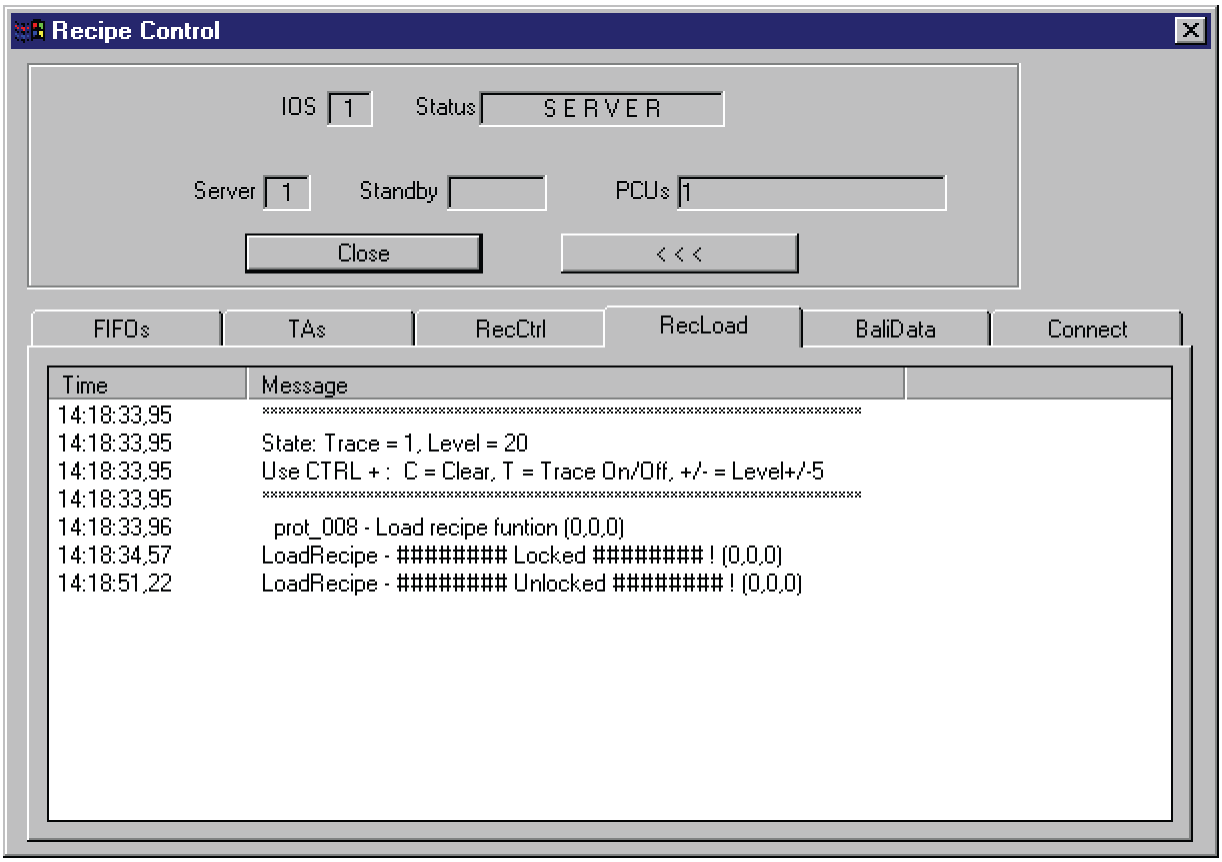
Trace Off

With this the trace output can be disabled.

Clear

With this, the window can be deleted.

13.2.5 Recipe load function



In this window (RecLoad), messages relating to the loading of control recipes into the SIMATIC are displayed.

This window provides diagnostic information with recipe load errors of the sequencers.

The following commands can only be set after one line is selected with the mouse.

Level

With Ctrl + and Ctrl – the level of the signaling can be changed.

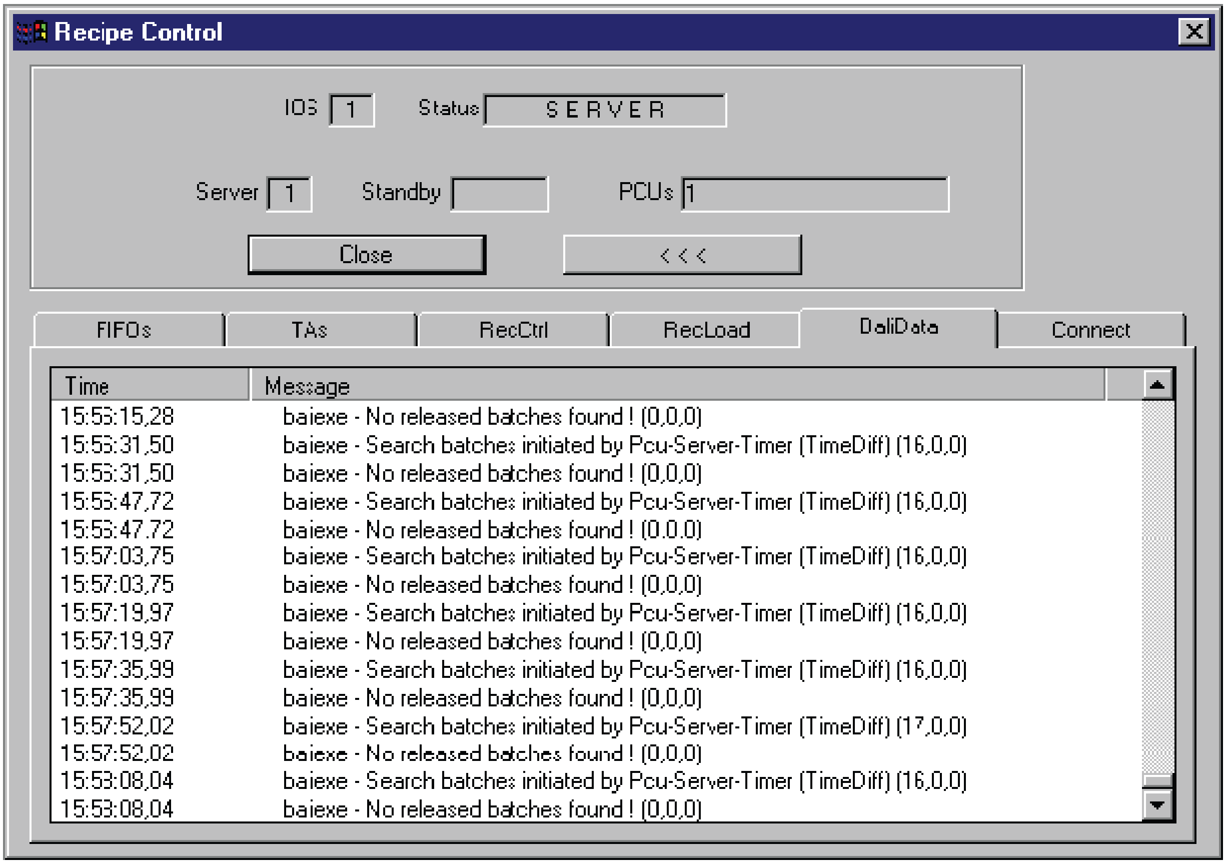
Clear

With Ctrl + C the signaling can be deleted.

On/Off trace

With Ctrl + T the trace function can be switched on/off.

13.2.6 Order system



In this window (BaliData), messages relating to the Runtime part of the order system are displayed.

Level

With Ctrl + and Ctrl – the level of the signaling can be changed.

Clear

With Ctrl + C the signaling can be deleted.

On/Off trace

With Ctrl + T the trace function can be switched on/off.

Writing on file

With Ctrl +F messages can be written into a file.

13.3 Controller operation

13.3.1 Functionality

The **Controller operation** application is used for the operation and the control of controllers.

13.3.2 Client area

Basics

The Client area contains up to 4 controller schematics. Depending on the type of controller each type includes the following displays and operating elements:

Name of the controller

Every controller gets an explicit meaning.

Scaling for X & W

For the process value and the setpoint, a scaling with maximum value, minimum value and unit is output.

X & W. as bar graph and/or numerical value

For the process value and the setpoint one bar graph and a numerical value is output.

Scaling for Y

For the manipulated variable a scaling with maximum value, minimum value and unit is output.

Y as bar graph and/or numerical value

For the manipulated variable a bar graph and a numerical value in percent is output.

Auto/Manual

Automatic system/Manual transfer; in the **manual** operating mode the input is possible for the manipulated variable Y.

External/Internal

External/internal transfer; with **External** the input of the setpoint W is possible.

KP part

P amplification

TV part *

Rate time

TN part *

Readjusting factor

On/Off **

On/Off operating for manual operation

STWE **

Manipulated variable in manual operation

M **

M = 1 for xd > TOB M = 0 for xd = 0

On/Off **

Controller On/Off, switchable

***** only for PID controllers

****** only for Three-point controllers

13.3.3 Working with the application

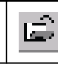
Working with the **Controller operation** application involves the following activities:

Selection of the operating elements

The user can select the controller by clicking the controller with the mouse and/or via the function keys of the operating elements of the controllers. In turn with the mouse and/or with the TAB key the single elements are jumped to and made operable. Some edit windows are protected by password.

Selection of the group

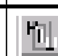
You can select the following options here:

Menu functions	Function key	Toolbar
Group	F2	

After the selection the configured group names are output in a list box. With a double-click on the names or with the OK key the program builds up a new overview with the corresponding controllers.

Selection of the controllers:

You can select the following options here:

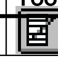
Menu functions	Function key	Toolbar
PID control module	F3	
Three-position controller	F4	
Controller 1	F5	
Controller 2	F6	
Controller 3	F7	
Controller 4	F8	

By selecting the menu item **Settings** or by pressing the function keys the following can be adjusted.

Auto	**F9**	Automatic operation
Manual	**SHIFT+F9**	Manual operation
External	**F10**	External
Internal	**SHIFT+F10**	Internal
Setpoint W	**SHIFT+F3**	Input of the setpoint W
Setpoint Y	**SHIFT+F4**	Input of the setpoint Y
KP part	**SHIFT+F5**	P amplification
TV part *	**SHIFT+F7**	Rate time
TN part *	**SHIFT+F8**	Readjusting factor
STWE **	**SHIFT+F1**	Manipulated variable in manual operation
Controller on **	**ALT+F5**	Controller on
Controller off **	**ALT+F6**	Controller off
Setting impulse open**	**ALT+F7**	Setting impulse open
Setting impulse close**	**ALT+F8**	Setting impulse close
* only for PID controllers		
** only for three-point controllers		

Configuration of the group

You can select the following options here:

Options menu	Function key	Toolbar
Parameterization	-/-	

After the selection, the standard Windows editor with the corresponding group file is called. The user can change the corresponding parameters here and save the file afterwards.

The files have e.g. the following structure:

Meaning	Description
[Group001]	Identification for group start
Name="Lautering"	Group name (Default: e.g. GROUP001)
Controller=5,7,13,23	Data record numbers in the DB of the controllers involved (a maximum of 4) (Default: 1,2,3,4)
Dim="mbar,mm,l,%"	Dimensions for the controllers involved (a maximum of 4) (Default: "Dim1,Dim2,Dim3,Dim4")
Dep=2,1,0,3	Decimal point position for the controllers involved (a maximum of 4) (Default: 0,0,0,0)
[Group002]	Any further groups
Name=" "	
Controller=	

The following limiting values are defined in the program:

Group number ID:	min. 1	Max. 64
Name string length:		max.16
Record number:	min. 1	Max. 64
Dimension string length :		max.6
Decimal point position:	min.0	max.4

If the limiting value is exceeded, the limited value is assumed automatically.

If the files are not available, the default values are used (see table).

13.4 Final operating elements

13.4.1 Functionality

The application **Actuators** gives an overview of all actuator statuses, with the possibility to switch on and off actuators or acknowledge errors.

13.4.2 Client area

Overview line:

- Input field for ICM number
- Input field for ICM name

If an ICM name is entered and finished with the return key, this ICM will be searched for. If it is found, the area/group of this IMC is displayed in the overview picture and at the same time the detailed picture (actuator status) of this IMC is shown. If no IMC with the entered name is found, the old IMC number and the corresponding name are retained in the overview row.

Column 1: No.

Continuous numbering for the single ICMs from 1 to 255.

Column 2: Name

Specifies the symbolic ICM name. The ICM name is allocated by the text configuration.

Column 3: Status

Displaying the current operating state of the ICMs

1. Displaying green on or black off
   On = ICM is switched on
   Off = ICM is switched off
2. Displaying red **error** or blank
   Error = ICM error is in line

Horizontal scrollbar for block-by-block paging.

13.4.3 Working with the application

Working with the application **Actuators** involves the following activities:

Displaying ICM data

The display of the ICM data occurs PCU-related. In an area-crossed operation the requested PCU can be selected via the menu **Area**. Only all ICM data of a PCU are indicated. A mixed display of ICM data from several PCUs is not possible. The ICMs of the selected PCUs can be operated.
The display of the ICM data occurs continuously in two blocks. However, it is possible to scroll forth and back page by page using the cursor keys or the horizontal scrollbars. The corresponding selected ICM is marked by color. For this currently selected ICM the corresponding detailed image can be selected via Return. The selection of the ICMs can also occur by clicking the corresponding line with the mouse.

Selection of the area

For an area-crossed operation the area which should be operated can be selected via the menu **Area** (=PCU). The display and operation of ICMs refers to the selected section exclusively. A mixed operation and control of several ICMs of different areas (i.e. different PCUs) is not possible.

Selection of the ICMs

The colored background of number and ICM name shows the currently selected ICM.

The selection of the ICMs occurs in three different ways:

- Cursor keys
- Click the ICM names with the mouse
- Input of the ICM number in the input field ICM.

Selection of the ICM detailed image

The selection of the ICM detailed image can occur in three different types:

- Selection of the ICMs and activating Return and/or clicking with the mouse
- Double-click on the ICM names with the mouse
- Input of the ICM name in the overview line and activating Return

Displaying the ICM detailed image

The ICM detailed image includes the following information:

- Symbolic name of the selected area (= PCU)
- Current ICM group

ICM number and symbolic name of the current ICMs

Table 13-1 Representation of the interface signals

1	=	logical signal "1" is connected to the component
0	=	logical signal "0" is connected to the component

BA	=	Command Automatic	
		Control from procedure or user program	
		0 = automatic system command On	
		1 = automatic system command Off	
BV	=	Operation locking	
		1 = enable:	ICM can be switched on
		0 = barrier:	ICM is switched off automatically, a restart is not possible
RE	=	Response On	
		1 = response of ICM "On"	
RA	=	Response Off	
		1 = response of ICM "Off"	
HD	=	Manual operation	
		1 = manual operation is active	
		0 = automatic switching is active	
QL	=	Load output	
		1 = load output is controlled	
Status	=	as in overview (On/Off, error)	
TÜS	=	Monitoring time setpoint [s]	
TÜI	=	Monitoring time process value [s]	
TYPE	=	ICM type	
Zuo	=	ICM manual group assignment	
BSP	=	ICM command file room	
		1 = ICM is controlled, specified circuit state Automatic operation	
QSP	=	Error acknowledgment memory	
		1 = error is in line	

Operation of the ICM detailed image

Table 13-2 The operation of the ICM occurs with the following buttons/function keys:

Help	**F1**	Calling a help function
Close	**F3**	Closing the ICM detailed image
Status	**F4**	Calling the application **Status** with the display of the corresponding interlock program
Quit	**F5**	Acknowledgement of the ICM error in line
Off	**F6**	Switching off the final controlling element; the monitoring time is restarted if the ICM was on before and not disturbed.
On	**F7**	Switching on the final control element; the monitoring time is retriggered if the ICM was on before and not disturbed. Switching on is only possible, if BV = 1
Memo	**F10**	Opening a text editor for entering a memo by the user. A default text is offered which can be changed, added and filed by the user in any way. The text is also operable with the application **Message archive**.

Special properties:

ICMs of the type "Motor" turn off automatically in case of error; the error should be acknowledged and removed before the switch-on is possible again.

ICMs of the type "Valve" remain switched on also in the case of fault.

13.5 Setpoint trends

13.5.1 Functionality

Basics

The **Curve entry** application is used for the graphic or tabular input and display of curves

which enables the user to enter IOS setpoints as a graphic curve shape. The interpolation points of the curve are stored in a data block in the PCU. After the setpoint curve is triggered, a standard FB always calculates the current setpoint value (interpolation between the interpolation points) according to the set time base.

The current setpoint value can be output to a PID controller for further processing, for example.

Application example:

Preset of cooling curves for fermenting silos.

You can define several trigger points for the same curve. A flag is set in the PCU when the trigger condition is met.

You can trigger the curve recording in two ways:

- Internal trigger:
  Curve recording is triggered manually. Specify the curve group ID and the relative curve number in the curve dataset, and then trigger the recording by means of the start bit ‘Start from internal = 1’ of the dataset.
- External trigger:
  Curve recording is triggered externally by a sequence. The value of the curve number is defined in comma-separated decimal format, with two decimals.
  - External trigger:
    Curve group number
  - Decimal:
    Relative curve number in the group

  A setpoint SW = 0 must be preset in the recipe list after the curve is triggered, because otherwise the curve would be re-triggered after its expiration.

See also

CURVSCAN - Curve target values (Page 291)

13.5.2 Client area

Basics

- After the application starts, new curves can be configured and existing curves or pictures (=collection of curves) can be created.
- All open curves (max. 8) are always displayed simultaneously on the graphic field.
- The currently open curves can be saved as a picture (=collection of curves).
- Only the curve selected in the toolbar is active. For the active curve, the curve shape can be edited with the mouse.
- When the mouse pointer is in the graphic field, it is displayed as a cross-hair cursor. The coordinates of the pointer are also output below the vertical axis.
- The curve data is converted to the time unit of the display. If the represented time window is smaller, the curves are cut off at the edge of the graphic field. If the window is bigger, the curves end at the last interpolation point. The graphic field can be shifted horizontally (chronologically), rotated or squeezed.

Setup of the Client area for graphics representation:

- Vertical scaling of the active curve (on the left next to the graphics)
- Vertical scaling of the inactive curves (on the right next to the graphics)
- Scaling of the time axis below the graphics
- Scrollbar for displacing the time window
- Scrollbar for vertical displacement
- Graphics window for the graph
- Displaying the trigger points

Structure of the Client area for table representation:

- Scrollbar for vertical displacement
- Scrollbar for horizontal displacement
- Table with the interpolation points for the open curves

Column 1

Continuous numbering of the interpolation points

Column 2

- Time field:
  If "Tr" appears in front of the time information, the time is a trigger point

Column 3-10

Interpolation points of the curves 1-8

13.5.3 Working with the application

Requirement

In order to specify setpoint curves on the IOS, the PCU data blocks in which the interpolation points of the curves are to be stored must first be specified in the `...<proj-path>\sys\kurvein.ini` file in the [DB_List] section. The following rules apply:

- A separate data block must be used for each group of curves - curve group and DB number form a 1:1 relationship.
- The first DB number of each PCU must be configured in the global data of the "CURVSCAN" class.
- The group and DB numbers must be unique within the AREA server, i.e. they must not be repeated in the PCUs of this AREA.
- The following relationship applies to the curve scanning in the PCU: DB No. = <Start DB> + Groupx - 1

KURVEIN.INI	Description
[DB_List]	List of data blocks
Groupx=PCU,DB,Type	Group No.= PCU No., DB No., data type *)
Groupx=PCU,DB,Type	Group No.= PCU No., DB No., data type *)
etc.	
	*) Only the "short" data type is supported in the standard system

Example: 1 PCU, 2 curve groups

The curves of the PCU 001 are to be stored as 16-bit numbers in DB 50 for Group 1 and in DB 51 for Group 2.

KURVEIN.INI	Description
[DB_List]	
Group1=1,DB50,short	Group No. 1= PCU No. 1, DB No. 50, data type short=16 BIT number
Group2=1,DB51,short	Group No. 2= PCU No. 1, DB No. 51, data type short=16 BIT number

Parameter assignment:

PCU 1:

- Class 'CURVSCAN' / Global data / FirstGroupDbNo = 50
- Class 'CURVSCAN' / Instance xy / CrvGrp = 1
- Class 'CURVSCAN' / Instance xy / CrvGrp = 2

Example: 2 PCU, 2 curve groups

The curves of Group 1 are based on PCU 013, those of Group 2 on PCU 014.

KURVEIN.INI	Description
[DB_List]	
Group1=13,DB50,short	Group No. 1 : PCU No. 13, DB No. 50, data type short =16 BIT number
Group2=14,DB51,short	Group No. 2 : PCU No. 14, DB No. 51, data type short =16 BIT number

Parameter assignment:

PCU 13:

- Class 'CURVSCAN' / Global data / FirstGroupDbNo = 50
- Class 'CURVSCAN' / Instance xy / CrvGrp = 1

PCU 14:

- Class 'CURVSCAN' / Global data / FirstGroupDbNo = 50
- Class 'CURVSCAN' / Instance xy / CrvGrp = 2

Note

Once the data blocks have been specified, the curves can be created with the "Curve entry" application.

Opening images:

Previously defined pictures with a collection of curves can be opened using menu item "Images / Open".

A new picture can be generated by entering an as yet unused name.

Up to 8 different curves can be displayed in a picture. They do not necessarily have to come from the same curve group.

The currently open picture is indicated in the title bar.

Saving images:

Images can be saved under a new name using menu item "Images / Save as..."

All currently open curves are saved within the image.

The information for all configured curves and images (=collection of curves) is saved in the `...<proj-path>\sys\kurvein.ini` file.

Deleting images:

Images can be deleted using menu item "Images / Delete". After the function is selected, a dialog box for selection of the image appears followed by a confirmation dialog.

Information about a curve image

Information about the curves that are loaded in the work memory (buffer = 1...8) is displayed using menu item "Images / Information".

Column 1:	Buffer
Column 2:	Number
Column 3:	Name
Column 4:	Vertical unit
Column 5:	Dec. pt. Decimal point
Column 6:	Time base, e.g. (seconds)
Column 7:	No. int.pts. Number of interpolation points
Column 8:	Max.no. int.pts. maximum number of interpolation points

Opening curves:

A previously defined curve can be opened using menu item "Curves / Open". A new curve can be generated by entering an as yet unused curve number.

Note

Assignment of curve numbers

The application supports only curve numbers in the form **"a.mn"** where a=curve group (interpolation point DB) and mn=curve number (rel. curve within the group)!

- Using **Save as...** the curve data can be saved as a different target file.
- All selected curves are saved using **Save all**.
- The configured curve interpolation points are initially saved in PCU-related offline DBs.

Deleting curves:

Curves can be deleted using menu item "Curves / Delete". After selecting the function, a dialog box for selecting a curve will appear.

Removing curves:

A curve can be removed from the work memory using menu item "Curve / Remove".

Loading curves

The offline DB of the currently open curve group is transferred to the PCU using menu item "Curves / Load to PCU". For this purpose, the "Block transfer" application is opened with preassigned PCU / block information.

Note

PCU type S7-1500

All the data blocks used for setpoint curves must be created in the TIA Portal project and loaded in the AS before the "Load curve data to PCU" function is called. Otherwise, an error message appears when loading.

The S7-1500 delivered project contains the "...!System\Runtime Interfaces\03_Miscellaneous \BmCurveSpTableGroup1 [DB50]" block, which can be expanded as needed by copying and pasting to other examples (DB 51, DB 52, etc.).

Reason:

For this PCU type, the IOS loading function cannot create new data blocks in the AS. Rather it can only update existing DBs!

Insert line

Lines are inserted in the table via the menu **Curves** via the function **Insert line** or by pressing **Shift+Insert**. After pressing the keys one line is inserted in the table. The cursor is positioned in the time field.

Attach line

Lines are attached to the table via the menu **Curves**, via the function **Attach line**. After the selection one line is attached in the table.

Delete line

This function deletes all locations with the time value. After activating the function the interrogation appears:

"Do you really want to delete?"

Assignment

Here you can view the assignment of the current curve to the PCU DB. The DB number is shown in relation to DB50. The data are not selected or modified here.

Graphic input and display of curves

In the graphics field all opened curves (a maximum of 8) are represented. Only one curve is always active. The active curve is determined by clicking the corresponding icon in the button bar. For the active curve, the curve shape can be edited with the mouse. The curve data is converted to the time unit of the display. If the represented time window is smaller, the curves are cut off at the edge of the graphic field. If the window is bigger, the curves end at the last interpolation point. The graphics field can be moved, stretched or compressed horizontally (chronologically) and vertically.

Stretch/compress time view

The following options are available for stretching / compressing the time range displayed:

View menu	Function key	Toolbar
Stretch view	F8	
Last view/compress	-/-	

- When making a selection via the menu/function key the desired time range is provided in a dialog box (format for "from" / "to": hhh:mm:ss). If the format is not kept, the message "Unauthorized time" appears.
- All curves are initially deactivated when selecting with the mouse. In the graphics field a time interval can be selected with the mouse (hold right mouse button pressed, hold pressed, displace, enable, end of the stretching function)
- You can return to the last view via the Last view/compress function. If no last view is filed, the area will be compressed by the factor 1.5 (resolution becomes smaller, area becomes larger).

View Displace time

The represented time period can be displaced horizontally. After selecting the function **Displace time** in the menu **View**, a dialog box in which the "New start time value" (for example 12:30) can be preset.

The displacement can also be selected with the scrollbar below the graphics field.

Stretch/compress vertical view

The following options are available for stretching / compressing the time range displayed:

View menu	Function key	Toolbar
Stretch vertical	F7	
Vertical Last view/compress	-/-	

- When making a selection via the menu/function key the desired time range is provided in a dialog box ("from" / "to"). An error message appears if the measurement range is exceeded in the input.
- All curves are initially deactivated when selecting with the mouse. In the graphics field a time interval can be selected with the mouse (hold right mouse button pressed, hold pressed, displace, enable, end of the stretching function)
- You can return to the last view via the Last view/compress function. If no last view is filed, the area will be compressed by the factor 1.5 (resolution becomes smaller, area becomes larger).

Vertical view displace

The represented area can be displaced. After the selection of the **function Displace vertically** in the menu **View** a dialog box in which the "New minimum value" can be preset appears. The displacement can also be selected with the scrollbar to the right of the graphics field.

Time grid

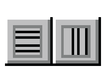
With this function a time grid can be adjusted i.e. a grid enables only mouse inputs at equidistant times.

After selecting the function **Time grid catch** in the menu **View** a dialog box appears. The requested grid can be entered behind **Distance** in the form hh:mm:ss. The function becomes active when **Grid catch on** is crossed.

Grid

The graphics field can be coated with a grid network.

After selecting the function **Grid** in the menu **View** a dialog box appears. The requested distance can be entered for vertical and horizontal lines separately. In addition both lines can be hidden as well as shown separately. Or by selecting the icon:



Curve parameters

Curve parameters can be determined for every opened curve.

The selection of the dialog box occurs after the operation of the function **Curve parameter** in the menu **Curves**.

With new curves the parameters are preset (values in brackets).

The curve parameters are:

- Name (cuxxx.yy)
- Unit (unitxxx.yy)
- Decimal point (0)
- Minimum input value (0)

- Maximum input value (100)
- Time base (Seconds)

Note

The time base can be selected only at new, still empty curves. After setpoint values have been defined, the time base can no longer be changed.

Edit trigger points

With the function **Trigger points** in the menu **Curves** trigger points can be entered. A dialog box appears for the input.

In the graphic representation, trigger points can also be edited by clicking the following icon in the toolbar.

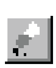
The trigger values are represented in the graphic representation as vertical lines at whose top end the setpoint is indicated.

In the tabular representation the trigger time is marked with a prefixed "Tr".

Trigger points are operating points. In this way operating activities can be executed e.g. by reaching the "trigger time", for example enabling one step of the step chain. A synchronization with the recipe system can be achieved.

Delete curves

In the graphic representation curve points can be removed by selecting the following icon.



Set-up auto

By selecting the function **Autosetup** in the menu **View** the original representation after compressing, stretching etc. is signed again.

Sign new

By selecting the function **Sign new** in the menu **View** the curve is signed new – after the curve shape has been changed (delete).

Table/graphics

You have the following options for showing the tabular view:

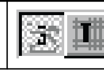
View menu	Function key	Toolbar
Table/graphics	F5	

Table type

By selecting the function **Table type** in the menu **View** you can branch out into different representation types.

Tabular input and display of curves

For the table representation there are two options:

Representation 1 (all interpolation points):

All interpolation points (all curves) are represented in ascending order within a given time. If there are no interpolation points for a point of time, there will be a linear interpolation between the interpolation points which are before or behind them. The interpolated values are labeled with *). It won't be interpolated over the last interpolation point (represented in the table with a line).

Representation 2 (interpolation points of a curve):

Interpolation points of a curve (Master) are used as "Trigger pulses". All interpolation points of the "master curve" are output. If there are no interpolation points for these time values, the values will be interpolated in a linear fashion. It won't be interpolated over the last interpolation point.

There is a field cursor in the table. This can be positioned with the mouse or the cursor navigation keys. The field in which the cursor is located can be edited. If a time value is changed, the table is put in the right order and is indicated.

Printing

With the **Print table** function in the **Print** menu the curves can be output in tabular format on a printer. After the selection, a dialog box appears in which the table type - master form and/or all interpolation points of all curves - can be preselected.

13.6 Special values

13.6.1 Functionality

The application **Special values** is used for the representation of up to 511 special values. These special values can be represented in the formats channel lock-in facility (Fixed-point number) KH (Hexadecimal number), KT (Time value) or KZ (Count). The special values represent the content of the data block DB 709 of the selected area. The selected area corresponds exactly to a PCU. A mixed operation of special values via several PCUs is not possible.

13.6.2 Client area

Column 1: No.

Continuous numbering of the individual special values from 1 to 255

The insertion of blanks is not possible.

Column 2: Type

Specification of the special value type:

- CHANNEL LOCK-IN FACILITY = fixed-point number (Integer)
- KH = Hexadecimal representation
- KT = Representation as a time value
- KZ = Representation as count

Column 2 remains empty with not-configured special values.

Column 3: Name

Name of the special value the can be freely defined by the user. This name is edited with the **application Text configuration.**

Column 4: Comment

User definable comment for the special value with a maximum of
32 ASCII characters.

Column 5: Value

Value of the special value in the respective preset format. With a non-available configuration the value is indicated in the format KT.

Column 6: Dimension

User-friendly and definable name for the dimension of the special value with a maximum of 8 ASCII characters.

13.6.3 Working with the application

Basics

Working with the **Special value** application involves the following activities:

Displaying the values with data source "Offline IOS"

The special values currently filed on the hard disk are indicated. The reading and writing of special values refers exclusively to the special values which are filed on the hard disk

On the hard disk of the IOS, the special values of the individual PCUs are stored in the paths `<proj-path>\PCU.nnn\DB`.

The special values are stored in the db.709 file.

According to the valid PCU selection the special values are fetched from the assigned registers.

Display of the values with data source "Online PCU"

The current special values of the selected PCUs are indicated. The reading and writing of special values refer exclusively to the special values which are filed to the selected PCU in block 709.

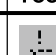
Selection of the PCUs

For an area-crossed operation the PCU which has to be operated can be selected via the menu **Area**. The reading and writing of special values refer exclusively to the selected PCU. A mixed operation and control of several special values of different PCUs is not possible.

The PCU selection refers to the corresponding IOS subfolders or directly to the corresponding PCU.

Selection of a special value

You have the following options for selecting this function:

Menu functions	Function key	Toolbar
Special value	F2	

- After selecting this function a dialog box appears.
- The entered special value is selected and represented in the first line of the Client area.
- With the input of values outside the entered value area a message box with the note "Value too large!" or "Value too small!" will appear.

Parameterize a special value

The parameters of the special values (type, comment and dimension) are always saved on the hard disk and always relate to the special value selected. You have the following options for selecting this function:

Menu functions	Function key	Toolbar
Parameterization	F3	

- double-click on the corresponding line in the Client area

The call of this function is protected by the password. After successful password request a dialog box which contains the following data appears:

- **Number** of the selected special value
- **Name** of the selected special value from the file:
  ..\ PCU.xxx \ TEXTE \ SONDWERT.TXT
- **Type** of the selected special value from the file:
  ..\ PCU.xxx \ TEXTE \ SONDWERT.TYP
  If the special value has no existing configuration yet, the **KT** format is offered as a default setting.
- **Comment**
  from the file... \ PCU.xxx \ TEXTE \ SONDWERT.NAM
- **Dimension**
  from the file... \ PCU.xxx \ TEXTE \ SONDWERT.DIM
- **Current value**
  with Online PCU from the data block DB 709, DW data word (Special value no.) of the selected PCU
  when offline - IOS from the file ... \ PCU.xxx \ DB \ DB.709 The current values are displayed in the formats KF, KH, KT and KZ.

The type can be selected via the option button and the comment and the dimension can be entered.

OK command button

The inputs are filed in the corresponding files:

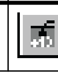
The dialog box is closed.

Delete command button

The warning "Do you want to delete the special value?" appears. If the **Yes** button is pressed the value is set to 8000 H. The note box and the dialog box are closed.

Value input for special value

You have the following options for selecting this function:

Menu functions	Function key	Toolbar
Value entry	F4	

Or alternatively:

- Return key on the corresponding line in the Client area

The call of this function is protected by the password. After successful password request a type-dependent dialog box appears. The following data are indicated:

- **Number** of the selected special value
- **Type** of the selected special value from the file:
  ... \ PCU.xxx \ TEXTE \ SONDWERT.TYP
- **Name** of the selected special value from the file:
  ...\ PCU.xxx \ TEXTE \ SONDWERT.TXT
- **Comment** from the file ...
  ... \ PCU.xxx \ TEXTE \ SONDWERT.NAM
- **Dimension** from the file:
  ... \ PCU.xxx \ TEXTE \ SONDWERT.DIM
- **current value**
  with Online PCU from the data block DB 709, DW data word (Special value no.) of the selected PCU
  when offline - IOS from the file ... \ PCU.xxx \ DB \ DB.709 The current value is only displayed in the configured format.

The representation of the input area with an example according to the set format. According to the configured type, different dialog boxes appear for the value input:

Type KF:

- Input of integer numbers (whole numbers)
- Input area of KF -32768 to +32767

The input is limited to six characters (minus sign + 5 digits). Entering the sign "+" is not necessary.
With inputs outside the input area the note box "Wrong entry" appears. After their confirmation the note box and the dialog box are closed without a value transfer.

Type KH:

- Input of hexadecimal numbers
- Input area of KH 0000 H to FFFFH
- Only digits and letters A, B, C, D, E and F can be entered.
- The input is limited to four characters.

KT type:

- Input of time values
- Input area of KT 0.000 to 3.999

The input occurs in two areas:

- Time grid:
  - Only the numbers 0 for 10 msec, 1 for 100 msec, 2 for 1 sec
  - and 3 for 10 sec can be entered.
  - The input is limited to one character.
- Time value:
  - Only the digits 0 to 9 can be entered.
  - The input is limited to three characters.

KZ type:

- Input of count values
- Input area of KZ 000 to 999
- Only the digits 0 to 9 can be entered.
- The input is limited to three characters.

OK command button

- The entered value is saved.
- After storing, this special value is filed in the data block DB 209, DW data word (Special value no. - 1).
- With Online PCU the value is written into the data block in DB 709 of the PCU.
- With Offline IOS the value is written to the file ... \ PCU.xxx \ DB \ DB.709.
- Thus it should be considered that the contents of the saved DB 209 differ between IOS and PCU.

Selection of the memo

The selection of this function occurs in two different ways:

- Menu item **Memo** in the menu **Options**
- Function key **F10**
- A text editor is called.
- Texts can be set aside now for every special value.
- For each single data record there is the corresponding file:
  - ... \ PCU.xxx \ NOTETXT.y \ SONDWERT \ SONDWERT.nnn
    If there is no file for a special value, a default text will be offered from the file
  - ... \ PCU.xxx \ NOTETXT.y \ SONDWERT \ SONDWERT.TXT
    and copied in the corresponding file
  - ... \ SONDWERT.nnn .
    <y> symbolizes the adjusted language (0 = German, 1 = English, etc.).
    <nnn> represents the special value number.
    Fundamentally all memo texts of the special values can be called.

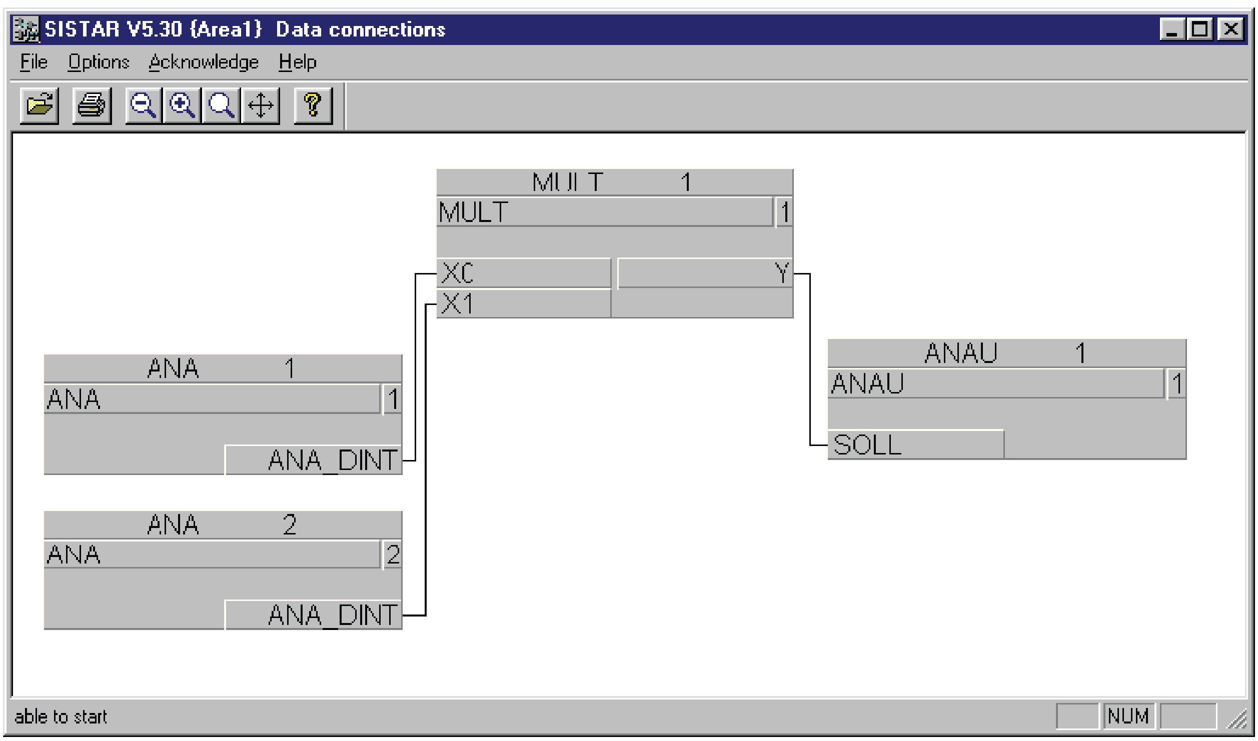
The editing is protected by a password. After a successful password query the text editor can be used. After pressing the **OK** button the file is saved and the dialog box is closed.

13.7 Graphical source view

13.7.1 Functionality

The function Data connection gives the user an overview of the interconnection of the blocks. The interconnections are represented via the application dataconn.exe. The application shows all interconnections starting from a block (e.g. MULT). The interconnections are continued recursively for other blocks which are referenced.

13.7.2 Client area



13.7.3 Working with the application

Open file

You have the following options for opening an existing data connection:

File menu	Function key	Toolbar
Opening	CTRL+O	

In the list fields you must select the PCU and the connection and click on OK. The graphic view of this instance is loaded into the working area.

Free selection

By calling the function **Free selection** in the **File** menu it can be searched for existing connection views.

- In the ‘Search’ tab an arbitrary block instance name can be searched for. In this case the options ‘Match case’, ‘Whole word only’ and ‘Search in ICMs only’ are selectable.
- In the ‘Browser’ tab any block instance may be selected

This block is the starting object. All connections starting from this block are displayed by pressing the ‘Goto Process image’ Button.

Organize

Via the menu item **File\organize** the connection views can be managed. Proceed as follows in order to create a connection view:

- You select the PCU via combo box PCU.
- With the button SEEK you arrive at the browser dialog.
- After having selected and confirmed the block, the name of the selected block appears in the edit section ‘title’
- Comment, creator and date can be entered for the selected connection view.
- The new connection view is accepted by the button Add.

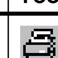
The data is stored in the file dataconn.txt of the selected PCU `(...<proj-path>\pcu.xxx \texte)`. There is one line per connection view in this file. Data are separated by a comma in this line.

In order to change the attributes of the available connection views it must be selected and the ‘Change’ box must be activated. After the change and pressing the 'Take over' button the new attributes are stored in dataconn.txt.

The selected connection view is deleted in the list by pressing the delete button.

Printing

You have the following options for printing an open data connection:

File menu	Function key	Toolbar
Printing	CTRL+P	

First, the print function opens a dialog, via which the legend can either be entered or changed.

Page preview

The function **Page preview** in the menu **File** opens a dialog via which the legend can either be entered or changed. Then the preview window with the page/pages is shown.

Printer setup

The function **Printer setup** in the menu **File** opens the Standard printer setup.

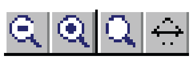
Documentation

Via the function **Documentation** in the menu **File** a print function which prints all connection views can be activated.

Before printing you can determine via the dialog if all interconnections of a PCU or all interconnections of all PCUs should be printed.

Toolbar

With the following icons of the toolbar the represented connection view can be reduced, extended, zoomed value-wise or aligned in the image.



Additional attributes at the blocks

In addition, the attributes of the block can be displayed for each block. File paramX.ini contains the settings to determine which attributes should be displayed. The file is located in the `<proj-path>\pcu.x\texte.x\` folder and should be created with an editor first if it is not available.

Example: paramX.ini

```
 [MULT]
AID_MULT_ART=ENUM_MULT_ART
AID_MULT_K=VALUE
AID_MULT_M=SYS_MULT

[PID]
AID_PID_YNF=SYS_PID

[ENUM_MULT_ART]
Lenght=9
1=0;ASL
2=1;MIN
3=2;MAX
4=3;ADD
5=4;SUB
6=5;MUL
7=6;DIV
8=7;LI+
9=8;LI-
```

This file is created based on the file `<proj-path>\param.pcu\paramS7.000`, which contains the available blocks and corresponding attributes.

For a ‘Mult’ block, for example, you can find the following attributes which you can use in "paramx.ini"

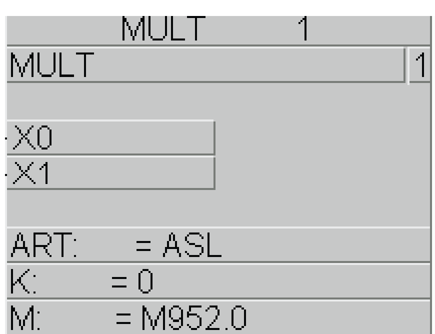
```
AID_MULT_Y
AID_MULT_X0
AID_MULT_X1
AID_MULT_ART
AID_MULT_K
AID_MULT_M
```

The file "paramX.ini" may contain one entry for each block. A block entry consists of the block name, e.g. [MULT], and the attribute, e.g. "AID_MULT_M" which is also to be displayed. This attribute value defines the displayed value.

The displayed value can be defined in three ways:

- Enumeration
  This means that a numeric value will be replaced by text. It is used for such attributes that can accept several values (e.g. DFM TYPE, MULT TYPE). For the enumeration the number (LENGTH=) of the parameters is defined. A value of the attributes and (separated by comma) the text is assigned to each parameter of the enumeration. For the definition there must be a definition block for the attribute.
  [ENUM_xxxxxxx]   xxx=Attribute name without AID
  Length=
  n=value;Textn=continuous number
- Value (=VALUE)
  The current numeric value of the attribute is shown
- Address of the user interface (=SYS_MULT, =SYS_PID etc.),
  i.e. the attribute source is shown.
  e.g.: AID_MULT_M=SYS_MULT
  This is a control input of the Mult block.
  A flag is assigned to this control input. Hence, M952.0. is shown for Mult1.

Graphic view of the MULT block in the above example:



13.8 Status

13.8.1 General

After a new installation (delivery state), a "Status" icon is displayed on the main menu of the "Diagnostics" tab.

- This icon is linked to the Status_S7.exe application by default.
- Alternatively, the KOP/FUP/AWL Editor of the STEP 7 program pack can be opened, provided that it is installed on the PC station.

Note

PCU type S7-1500 is not supported

- Due to technical circumstances, the status view with the assembled code sequences from the respective online FBs/FCs is not possible for PCU type S7-1500.
- For this reason, PCUs of this type do not appear in the selection list.
- The appropriate STEP 7 TIA Portal programming package must be used for the diagnostics.

13.8.2 Status for Simatic S7 via ‘Status_S7.exe’ application

Functionality

The application "Status_S7.exe" enables status displays of linked program in online operation to be displayed (directly and without using the SIMATIC Manager), but with the following limitations through the system:

- AWL-CALL calls are not currently decompiled.
- Global flags (e.g. HornGrpError, EopMonTmError, OperatorRequest, etc.) are not always updated correctly at the present time

Note

For viewing symbolic information (allocation list), this internal Status application uses the PCU-specific symbol file "s7_sym.seq". When the user wishes to see his/her project-specific changes (engineered with Step 7 Symbolic editor) in the status view, this file has to be exported again with the Step 7 Symbol Editor (format type "assignment list (*.SEQ)), where the address column should be sorted **in ascending order first**. Otherwise symbol information may not be viewed correctly in that application.

The Symbol table should be located in the following system folders:

- In case of 'Single Engineering Language':
  "…<proj-path>\pcu.nnn\s7_sym.seq" or "…<proj-path>\pcu.nnn\texte\s7_sym.seq" – the system uses the more recent file
- In case of 'Multi-Engineering-Language'
  "…<proj-path>\pcu.nnn**texte.x**\s7_sym.seq"

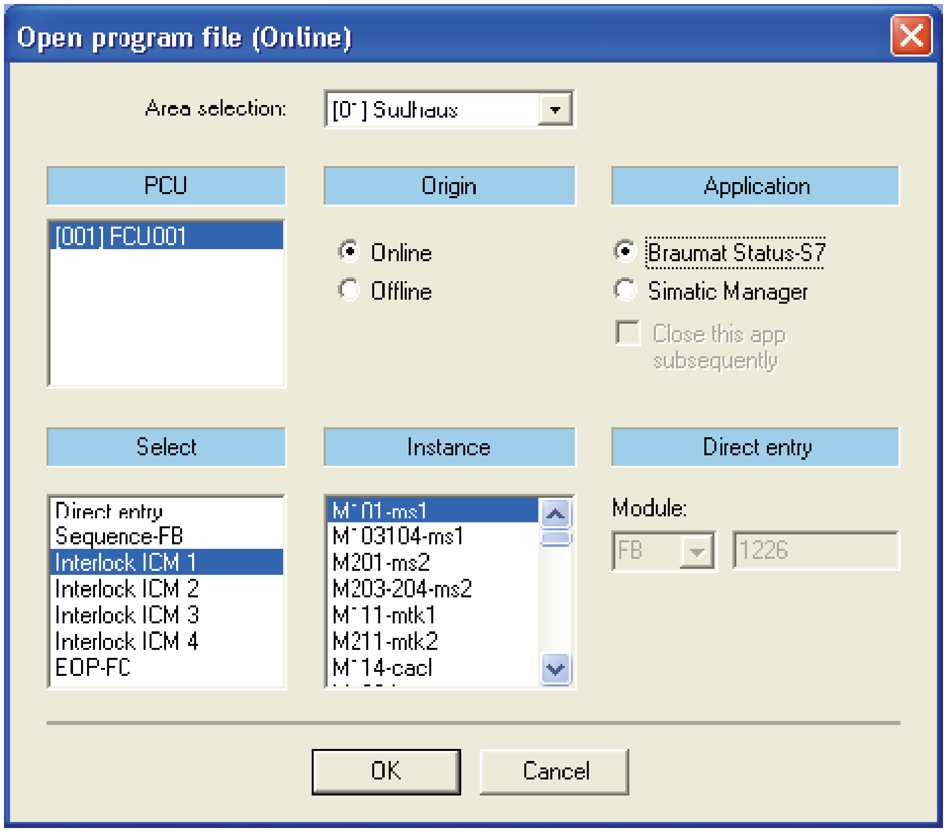
Client area

Display area

- Column 1: Address
  address rel. to the block start in hexadecimal code
- Column 2: Code
  Number of the data word
- Column 3: Status
  Status of the bit commands
- Column 4: Symbol
  Symbolic name from the assignment list
- Column 5: Comment
  Comment from the assignment list

Open file

After selecting the menu item **Open** in the menu **File** or after clicking the **FILE** icon in the button bar a dialog box "Open program file" for opening a file appears:



- AREA selection
- PCU selection; (only PCUs of type Simatic S7 are listed)
- In the box 'Select', first select the method; direct entry or the object classes sequence FB, BV-ESG1, ESG 2 (operator interlock ESG group). After that the corresponding block or object instance is selected.

With the OK button the requested block is loaded.

On the hard disk the blocks of the individual PCUs are filed in their own paths. The registers are built up as follows:

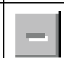
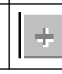
- <proj-path>\PCU.001\ für symbolisches PCU1 (e.g. boiler house)
- <proj-path>\PCU.nnn\ für symbolisches PCU n (e.g. silo store) etc.

with the respective subfolders:

- <proj-path>\PCU.nnn\ FB\
- <proj-path>\PCU.nnn \FC\

Network selection

You have the following options for the network selection:

Menu functions	Function key	Toolbar
Network	F4	
Network - 1	F7	
Network + 1	F8	

- After selecting the network a dialog box appears.
- Via the check box **All Networks** the display mode can be selected, whether all networks or only one network is supposed to be indicated.
- The following functions are only selectable for a non-selected check box **'All networks'**.

Selection of address

After selecting the menu item **Address** in the menu **Functions** a dialog box appears in which a valid address can be entered. The entered address appears in the uppermost line of the Client area.

Representation of the Step 7 Mnemonics

Representation of the Step 7 Mnemonics can be switched between DE and EN via the corresponding buttons in the toolbar. This is independent of the Simatic Manager language set on exporting the '*.seq' file.

Configuring colors

In the file `<proj-path>\sys\status.ini` in the section `[Settings]`, the text and background colors of the "Status" column can be configured. These are specified as RGB values (red, green, blue, 0 ... 255 in each case). Per operand type, the configured coloring can enabled or disabled separately.

The table below shows the possible setting attributes and the default values:

OP. type	Background color	Text color	Activation
Flag	`ColorBk_M=255,255,255`	`ColorTx_M=100,0,100`	`SetColorText_M=0/1`
Input	`ColorBk_E=255,255,255`	`ColorTx_E=100,100,0`	`SetColorText_E=0/1`
Output	`ColorBk_A=255,255,255`	`ColorTx_A=0,100,100`	`SetColorText_A=0/1`
Timer	`ColorBk_T=255,255,255`	`ColorTx_T=0,0,255`	`SetColorText_T=0/1`
Counter	`ColorBk_Z=255,255,255`	`ColorTx_Z=0,255,255`	`SetColorText_Z=0/1`
Data block	`ColorBk_D=255,255,255`	`ColorTx_D=255,0,255`	`SetColorText_D=0/1`
Negative *)	`ColorBk_Neg=255,0,0`	`ColorTx_Neg=255,255,0`	`SetColorText_Neg=0/1`

*) "Negative" is used to represent logical opposites. Only binary queries are affected. Depending on whether they are positively or negatively formulated the expectation for the signal value is different and it will be recolored or not accordingly.

Command	Signal value 0	Signal value 1
A `M 12.0`	It will be colored according to `ColorXX_Neg` because a positive query meets a negative signal value.	The operand-specific coloring is used.
AN `M 12.0`	The operand-specific coloring is used.	It will be colored according to `ColorXX_Neg` because a negative query meets a positive signal value.

The table below shows examples of the RGB values for some of the primary colors.

	Lighter tones	Darker tones
White	255,255,255	
Gray	200,200,200	
Red	255,0,0	125,0,0
Blue	0,0,255	0,0,125
Green	0,255.0	0,125,0
Yellow		255,255,0
Black	0,0,0.	

13.8.3 Status for Simatic S7 via "Status_S7.exe" and "STEP 7 KOP/FUP/AWL"

Functionality

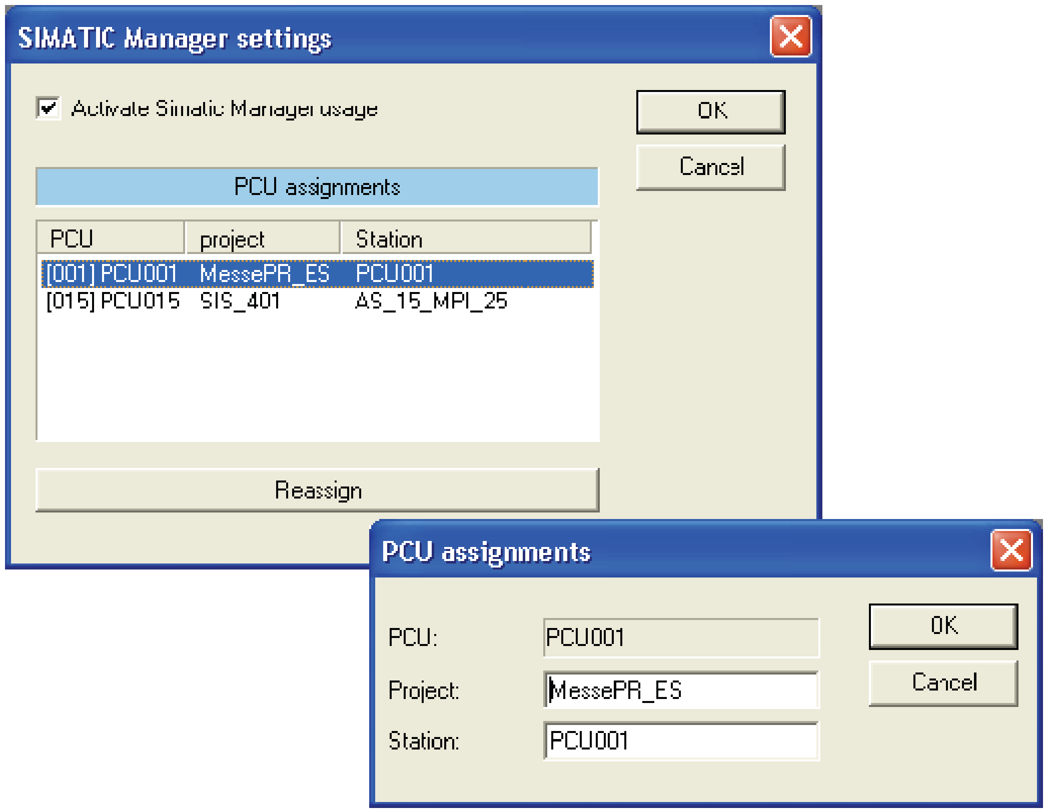
The status function for this variant is made available via the programming package STEP-7 for LAD/STL/FBD editor with the function "Control in the on-line mode". The operation of the editor is described in the corresponding manual instruction of the programming package STEP-7 and is not explained in more detail here.

Note!

This function is only available if the STEP-7 programming package was installed and authorized on the PC.

Activation

Activation is performed via the following configuration dialog, which is called in the Status_S7 – menu item ‘Options → SIMATIC Manager settings’:



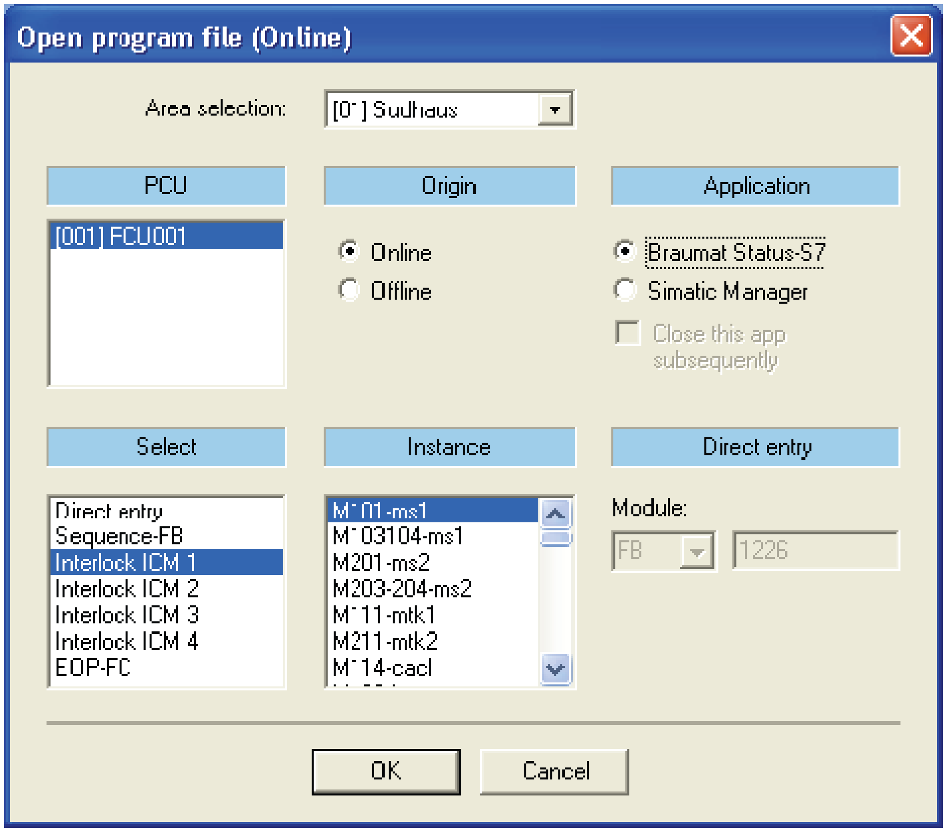
For each PCU the corresponding STEP 7 project and station name entries are assigned with the ‘Reassign’ button. The entries must exactly match the original SIMATIC Manager object names, otherwise the LAD/CSF/STL editor cannot be initialized correctly.

These settings are stored in the configuration file "`.....<proj-path>\sys\status.ini`"in the section "S7" as follows:

```
[S7]
PCU1=MessePR_ES\PCU001

...

PCU15=SIS_401\AS_15_MPI_25

UseStep7App=1
```

If you work with the delivered Simatic projects, only the ‘Simatic Manager’ option is to be activated; the PCUs are already established correctly.

In the Open file dialog, the Status S7 application or SIMATIC Manager is selected under "Application". The last selection remains stored.



All other selection fields correspond to the Status S7 application only (see previous section).

Note

After the LAD/STL/FBD editor has started, you should change to ‘Monitoring mode’ for displaying status information.

Other INI switches in "…<proj-path>\sys\status.ini":

```
[S7-Window]

;Set position automatic
```

`SetPosition=0;` Set position and size of AWL-Editor. If not, AWL-Editor starts with last settings`;Left border`

```
Left=0

;Upper border

Top=0

;Window width

Width=800

;Window height

Height=600
```

```
;Set AWL-Editor 'topmost'
```

```
TopMost=0
```

Show AWL-Editor 'always on top'

13.9 Forcing

13.9.1 Functionality

The application '**Steuern.exe'** is used for changing and viewing the values in the PCU and saving the definition file and data on disk.

13.9.2 Client area

The Client area is subdivided into two areas:

- Table heading

Display and editing area:

- **Column 1: No.**
  Line number
- **Column 2: Variable**
  Description of the parameter which has to be indicated. The input of the parameter must read as follows:
  – <PCU no>, <Operand1>, <Operand2>, whereby separation can take place through comma or <space>.

Addresses	Range of the S7 CPU 416-2	Range of the S7 CPU 417-4
DB	1...4095	1…8095
DX	-	
DW	-	
DBB	0...32767	0...32767
DBW	0...32766	0...32766
DL, DR		
MW	0…16382	0…16382
AW (QW), EW (IW)	0…510	0…510 *)
PAW (PQW), PEW (PIW)	0...16382	0...16382
MB	0…16383	0…16383
AB (QB), EB (IB)	0…511	0…511 *)
PAB (PQB), PEB (PIB)	0...16383	0...16383
SW	-	
SY	-	
T, Z	0…511	0…511
*) Note: The CPU 417-4 does not utilize all the I/O resources if EW/AW and EB/AB are used!		

- Input example.:
  - 1,DB124,DW5or1 DB124 DW5
  - 1,MW166or1 MW166
  - 1,QB10 or 1 QB10
- Column 3: KH
  - Display of the data word content in the format KH
  - KH = four-digit, hexadecimal representation
  - Input area of KH 0000 H to FFFFH
    Example: "1CAB"
- Column 4: KF
  - Display of the data word content in the format KF
  - KF = up to five-digit fixed point representation (integer value)
  - Input area of KF -32768 to +32767
    Example: "7339"
- Column 5: KY
  - Display of the data word content in the KY format
  - KY = two-digit, byte-by-byte representation, separated by comma
    Example: "28,171"
- Column 6: KC
  - Display of the data word content in the format KC
  - KC = two-digit ASCII character representation
    Example: "1A"
- Column 7: KM
  - Display of the data word content in the format KM
  - KM = 16-digit, bit-by-bit representation
    Example: "00000101 01101010"

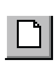
- **Column 8: KT**
  - Display of the data word content in the format KT = input of time values
  - Input area
  - h from KT 0.000 to 9993
    The input occurs in two areas:
  - **Time grid**: Only the numbers 0 for 100 msec, 1 for 10 msec, 2 for 1 s and 3 for 10 s can be entered. The input is limited to one character.
  - **Time value**: Only the digits 0 to 9 can be entered. The input is limited to three characters.
- **Column 9: KZ**
  - Display of the data word content in the KZ format
  - KZ = input of count values
  - Input area of KZ 000 to 999
  - Only the digits 0 to 9 can be entered.
  - The input is limited to three characters.

13.9.3 Working with the application

Working with the **Forcing** application involves the following activities:

Create new file

After selecting the **New** menu item in the **File** menu or clicking on the following icon in the toolbar a new display mask is created.



Open a file

After selecting the **Open** menu item in the **File** menu or clicking the following icon in the toolbar a modal dialog box appears.

A program is selected by clicking in the list box **File**; after pressing **OK** it is loaded.

Save

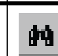
- After selecting the **Save** menu item in the Menu file or clicking the following icon in the toolbar the current control file is saved under the name which is displayed in the title bar.

- After activating the menu item **Save under** in the menu file a dialog box appears. The current control can be saved under a new name.

Data refresh

You have the following options for updating and for navigating in the current Client area:

Menu functions	Function key	Toolbar
Data refresh	F4	
Scroll forward	F6	
Scroll backward	F5	

13.10 DB editor

13.10.1 Functionality

The **DB editor** application is used for editing data blocks DB, DX in the PCU (Online) or on the hard disk of the IOS (Offline).

The length of the data blocks can be changed and data words are preset with editable values.

13.10.2 Client area

The Client area is subdivided into two areas:

- Table heading

Display and editing area:

- **Column 1: DW**
  Number of the data word
- **Column 2: KH**
  - Display of the data word content in the format KH
  - KH = four-digit, hexadecimal representation
  - Input area of KH 0000 H to FFFFH
    Example: "1CAB"
  - **Column 3: KF**
    Display of the data word content in the format KF
  - KF = up to five-digit, fixed point representation
  - (integer value)
  - Input area of KF -32768 to +32767
    Example: "7339"
- **Column 4: KY**
  - Display of the data word content in the KY format
  - KY = two-digit, byte-by-byte representation,
  - separated by a comma
    Example: "28.171"

- **Column 5: KC**
  - Display of the data word content in the format KC
  - KC = two-digit ASCII character representation
    Example: "1A"
- **Column 6: KM**
  - Display of the data word content in the format KM
  - KM = 16-digit, bit-by-bit representation
    Example: "00000101 01101010"

13.10.3 Working with the application

Basics

Working with the **DB editor** application involves the following activities:

Display for data source IOS

The data word contents currently filed on the hard disk of the IOS are indicated. The selection is made via the Offline IOS menu item in the data source menu or via the following icon.

The reading and writing of file contents refer exclusively to the data blocks which are set aside on the hard disk.

On the hard disk of the IOS the data blocks of the individual PCUs are stored in the paths `<proj-path>\PCU.nnn\DB`.

The data blocks are stored in files with the names db.001 for DB 1 to db.4095 for DB4095.

According to the valid PCU selection the data blocks are fetched from the assigned folders.

Display for data source PCU

The current data word contents of the selected PCU are indicated. The selection is made via the Online PCU menu item in the data source menu or via the following icon

The reading and writing of file contents refer exclusively to the data blocks which are filed to the selected PCU in the CPU.

The file contents are updated continuously.

Selection of the PCU

For an area-crossed operation the PCU which has to be operated can be selected via the menu **Area**. The selection occurs via the menu **Area**. The reading and writing of data word contents refer exclusively to the selected PCU. A mixed operation and control of several data words of different PCUs is not possible.

The PCU selection refers to the corresponding IOS subfolders or directly to the corresponding PCU.

The reading and writing of data word contents always refer only to a data block.

Data block selection

You have the following options for selecting the data module:

Menu functions	Function key	Toolbar
Data module	F2	

Following selection the modal dialog box **Module** appears.

- The input of **Data element type** and **Number** must occur (for example DB701). After closing the input with Return the data word contents of the new selected data block of the corresponding PCUs are fetched either from the hard disk or from the PCU and indicated.
- If a bad entry is made, values outside the limits or data block does not exist, message boxes to this effect are displayed.

Data word selection

After selection of the **Data word** menu item in the menu **Functions** or activating the key **F3** a modal dialog box appears **Selection data word**.

- The input of the data word number must occur (for example 123). After closing the input with Return the uppermost line is formed by the selected data word.
- With inputs of values exceeding the range of values the note boxes appear "Value too small", "Value too large".

DB length default

After selecting the menu item **DB length** in the menu **Functions** or activating the function key **F4** a modal dialog box **Change DB length** appears.

- The default of the DB length is only possible if as a data target the IOS of the hard disk is selected **(in an offline way).** In the **online operation** this function is blocked.
- The length of the selected data block can be changed in any way. The maximum length in this case is 4091 (at S5) and 32000 (at S7) data words. For an increase of a data block, the new data words are preset with KH 0000. For a shortening of a data block, the contents of the data words which are cut off disappear.

Page forth and back

You have the following options for navigating in the selected module:

Menu functions	Function key	Toolbar
Scroll forward	F6	
Scroll backward	F5	

13.11 Maintenance data

13.11.1 Functionality

The application maintenance data is used for default, checking and acknowledging maintenance intervals.

The maintenance data are divided into two parts:

- Maintenance data for ICMs
- Maintenance data for user aggregates

Per ICM or user aggregate the switching cycle and the hours of operation are registered.

For the switching cycle and the hours of operation a maximum of 5 setpoints each can be registered. The exceeding of these setpoints is output as a message and should be acknowledged after the maintenance has occurred.

Note

PCU type S7-1500 supports only maintenance data for ICMs.

The MAINT_USR parameter assignment and the user groups in the maintenance data application are not available for PCU type S7-1500.

13.11.2 Client area

In the Client area the aggregates of the selected area are indicated with the process value and a maximum of 5 setpoints.

The Client area is divided as follows:

- **Column 1**: Number of the aggregate
- **Column 2**: Aggregate
  Name of the aggregate
- **Column 3**: Hour/switching clearance
  Process value in hours or switching cycles
- **Columns 4 - 8**: Setpoints 1- 5
  Setpoints in hours or switching cycles

13.11.3 Working with the application

Basics

Working with the application maintenance data involves the following activities:

Selection of the area

Menu:	**File → Open...**
Keyboard:	**F2**
Button bar:	

One group can be assigned to every PCU. The standard name for the Group 0 is "ICM1/2/3/4". For every group it can be decided whether hour or switching clearance meters are supposed to be indicated.

Display

After the selection the corresponding data are read in from the file **maintesg.dbf for the ICMs (Group00), or from maintusr.dbf for the user groups (Group01...)** and this is indicated in the work window.

The number of lines is specified by the number of datasets in DB 682/684.

If the **required DBase** file is smaller or not there at all, a database with the required length is manufactured (with message). The database is not shortened.

During the generation the setpoints and texts are preset. The Aggr.No, Aggr.Name, process value and the setpoint are indicated.

Process value with power-on time meter in the format hh:mm.

If the process value exceeds a setpoint, the representation color of the corresponding setpoint is changed. The colors can be determined in the file **wardat.ini.**

Determine parameters

Menu:	**File → Parameter**

For every one of the maximum 5 setpoints of an aggregate the following parameters can be determined:

- **Value:** The setpoints can be selected by clicking on them and they can be overwritten by entering a new value. This is also possible in the Client area.
- **Short text:** For fast information; the selected setpoint is displayed in the status bar.
- **Long text:** For longer remarks a text file can be specified that can be viewed or edited using the Edit button.

The input data are stored in **maintesg.dbf or maintusr.dbf**.

Group definition

Menu:	**File → Group definition**

Can be used to change or delete the currently selected group definition (except group 0).

New groups can also be created for ICMs and user data in this dialog. A newly created group must first be selected to collect data.

For user data, the data block MAINT_USR / DB 684 must be generated and supplied with data by the user.

Printing

Menu:	**File -> Print**
Button bar:	

For the printing of maintenance data the following settings can be filed:

Sheet content: Selection of the required information.

Filter:

no filter	= all setpoints
queuing jobs	= setpoints which were exceeded and not acknowledged.
completed jobs	= setpoints which were exceeded and acknowledged.

Data sets: All, or a restricted number.

Reset process value

Menu:	**Acknowledgment →Reset counter/hours**
Keyboard:	**F9**

The resetting of the process value can be documented by a dialog box correspondingly and is registered in the message archive.

Acknowledgment

Menu:	**Acknowledgment → Acknowledge maintenance**
Keyboard:	**F10**
Toolbar	

The acknowledgement of maintenance can also be documented correspondingly and is registered in the message archive.

If the process value exceeds the setpoint either for the switching cycles or for the hours of operation, this value is marked in color (default value red). The color settings can be made out in the Wartdat.ini file (refer to the section "INI files (Page 1003)", Wartdat.ini).

For these marked aggregates a maintenance has to be performed.

After the execution of the jobs every setpoint of the aggregate which is exceeded can be acknowledged. During this acknowledgement a dialog is opened in which the operator, the date and a comment can be entered. This acknowledgement is saved in a database quitt.dbf.

The acknowledgement is filed in the data block of the PCU. An exceeded but acknowledged setpoint is characterized in color (default value green).

Message at acknowledgment

During acknowledgement a message is output, which appears at the message-enabled IOSs in the PCU server and is saved in the message archive.

Table 13-3 The columns of the message are allocated as follows:

Column	Function
Time	
Type	M
Recipe category	Type ("switching cycles" or "hours") 0 = hour; 1 = switching clearance
JobNo	GroupID
BatchNo	"Reset" or setpoint number
PCU	PCU number
Block	"MAINT"
BlockNo	Aggregate number
Name	Aggregate name
Text	"Group: " Group name

The meters for the switching clearance or the operating hours can be reset. During the resetting all jobs are also reset automatically for the setpoints.

During the resetting of the meters a dialog is also opened, in which the operator, the date and a comment can be entered. This resetting is also saved in a database quitt.dbf.

13.11.4 Engineering

Core statement

The function of the maintenance data is already integrated in the supplied AS project for the ICMs and for the user data. The user should allocate the states in the data block DB 618. The allocation of the data block is the same as in data block DB 605 for the ICMs.

The Standard DBs are:

- MAINT_DB (DB 682)
- MAINT_USR_DB (DB 684)

Structure of the maintenance database

A database exists for the maintenance data of the System ICMs (maintesg.dbf) and is a database which is user-defined for the user aggregates (maintusr.dbf). The structure is identical.

In the maintenance database, five setpoints are saved per ICM for the switching cycles and the hours of operation. Next to the setpoint a short text can be filed, as well as a file name for long texts.

- Name of the files: maintesg.dbffor the ICM maintenance data
  maintusr.dbffor the user maintenance data
- Path: `...<proj-path>\pcu.xxx\maint`

The database is manufactured according to the definition maint.def.

Table 13-4 Structure of the file 'maintICM.dbf' and/or 'maintusr.dbf':

	Offset:	0	
	Record length:	861	
001	SSP_SOLL1	CINT	10
011	SSP_FILE1	CHAR	12
023	SSP_KTEXT1	CHAR	64
087	SSP_SOLL2	CINT	10
097	SSP_FILE2	CHAR	12
109	SSP_KTEXT2	CHAR	64
173	SSP_SOLL3	CINT	10
183	SSP_FILE3	CHAR	12
195	SSP_KTEXT3	CHAR	64
259	SSP_SOLL4	CINT	10
269	SSP_FILE4	CHAR	12
281	SSP_KTEXT4	CHAR	64
345	SSP_SOLL5	CINT	10
355	SSP_FILE5	CHAR	12
367	SSP_KTEXT5	CHAR	64
431	STD_SOLL1	CINT	10
441	STD_FILE1	CHAR	12
453	STD_KTEXT1	CHAR	64
517	STD_SOLL2	CINT	10
527	STD_FILE2	CHAR	12
539	STD_KTEXT2	CHAR	64
603	STD_SOLL3	CINT	10
613	STD_FILE3	CHAR	12
625	STD_KTEXT3	CHAR	64
689	STD_SOLL4	CINT	10
699	STD_FILE4	CHAR	12
711	STD_KTEXT4	CHAR	64
775	STD_SOLL5	CINT	10
785	STD_FILE5	CHAR	12
797	STD_KTEXT5	CHAR	64

Structure of the quitt.dbf file

The acknowledgements are written to the database quitt.dbf if this is enabled in the Wartdat.ini file (refer to the section "INI files (Page 1003)", Wartdat.ini).

Thus the data in the database can later be evaluated with dBase or for example MS Access.

The database is saved in the catalog of the PCU in which the ICMs or user aggregates are defined.

- File name: quitt.dbf
- Path: <proj-path>\pcu.xxx\maint

The database is manufactured according to the definition Quitt.def.

Structure of the file 'quitt.dbf' :

Column	Type	Function
PCU	CINT 7	PCU number
GROUP	CINT 7	GroupID
TYPE	CHAR 4	Type "Time" or "Cnt"
AGGR_NO	CINT 7	Aggregate number
AGGR_NAME	CHAR 16	Aggregate name
SETP	CHAR 1	Setpoint
NAME	CHAR 20	Operator (asked in dialog)
DATE	CHAR 10	Date (asked in dialog)
COMMENT	CHAR 64	Comment (asked in dialog)

The writing of the acknowledgements into the database can be enabled or locked via a switch in the files Wartdat.ini.

13.11.5 INI files

Basic principle

Two INI files are available in order to adjust settings for the maintenance data.

These are the files:

- `...<proj-path>\sys\Wartdat.ini`
  In these files settings for the application can be adjusted.
- `...<proj-path>\pcu.nnn\maint\Maint.ini`
  In this file the maintenance group is described.

WARTDAT.INI:

Wartdat.ini	Description
[Settings]	
Actualize=60	Update time in seconds

Wartdat.ini	Description
ActualizeShow=0	Time until the next update occurs; display in the status indicator line
ShowAggrNumber=1	Fading in column with aggregate number
ColorTextSp=255,0,0	Exceeded and not acknowledged color for setpoint. Colors red, green, blue (every value between 0 and 255)
ColorBkSp=-1	Background exceeded for setpoint and not acknowledged. Background red, green, blue or -1 for transparent
ColorTextSpOk=0,255,0	Exceeded and acknowledged color for setpoint. Colors red, green, blue (every value between 0 and 255)
ColorBkSpOk=-1	Background exceeded for setpoint and not acknowledged. Background red, green, blue or -1 for transparent
AutoStart=0	opening to last setting after start of the application
StoreQuitt=1	Setting up database for acknowledgements
QuittMessage=1	Input into the message archive
PCU=3	last selected PCU number
Group=0	last selected group
Type=1	Type of last selection (1=operations counter, 2=operating hours meter)
MenuAppl=bed	Maintenance data belong to the operation or to the system (bed/sys)

MAINT.INI:

The group 0 is reserved for the ICMs and already defined in the file maint.ini. For this no further configuration is necessary.

For every further maintenance group which is supposed to be defined a section [Groupyy] is to be drawn up in this file (yy=Group number). This should have the structure described below.

maint.ini	Description
[Groupyy]	
Name=ICM1_ICM1023	Description of the group yy Group name
Type=0	Group type: 0=ICM, 1=User
FromDS=1	first data record
ToDS=1024	last data record
MaxDS=1024	maximum number of data records
TextFileName=	Name of the text file for the aggregate meanings (only with user groups, with ICM groups ICM1.txt is used always to ICM4.txt!) **Note:** **The path name and extension (*.txt) may not be specified in the file name.** Example: TextFileName=MaintUsr ; file used: ...<proj-path>\Pcu.xxx \texte\MaintUsr.txt

13.11.6 Miniport for generating maintenance messages

13.11.6.1 Miniport for generating maintenance messages

The setpoints for switching cycle and/or hour of operation can also be monitored by the system itself over the Miniport DLL "MAINTSUPV.DLL". If the respective actual value exceeds a configured setpoint, a message is entered in the message archive. Additionally an entry into a report (text) file can occur. Different report files are generated for each day (see below). You can find information on activating the miniport DLL in manual "*Installation and Configuration*" in "Section 6 "*PCU server miniports*".

13.11.6.2 Configuring the function

The following settings are necessary in `...<proj-path>\SYS\MainSupv.ini`:

```
[Settings]
;Startup delay in seconds
StartupDelay=20

;Cycle time for checks in minutes
CycleTime=15

;Enable (1) or disable (0) trace messages (PCU-Server trace window
only)
Trace=0

[TextSources]
;Language of message texts is fixed; specify them here:
SwitchingCycle=Schaltspiele
Hours=Stunden
SetpointReached=Grenzwert erreicht
Unit=Aggregat
ICMGroupName=ESG1_ESG1024

;Text files containing ICM names
ICM1File=ESG1.txt
ICM2File=ESG2.txt
ICM3File=ESG3.txt
ICM4File=ESG4.txt
```

13.11.6.3 Report files

Basics

Maintenance messages are generated and stored in the directory and file structure:

...<proj-path>\pcu.xxx\MAINT\maint_YYYY_MM\maint_YYYY_MM_DD.TXT

- xxx → PCU no.
- YYYY → year
- MM → month
- DD → day

Example of the structure:

Time	Type	PCU	No.	Aggregate	Group	Message	ActVal	Setp	Shorttext
13:18:56	Switching cycles	16	1	ICM1 1	ICM1_ICM1024	Limit value reached	2	5	Unit 0001 / Setpoint 2
13:18:56	Hours	16	1	ICM1 1	ICM1_ICM1024	Limit value reached	1:00	2:46	Unit 0001 / Setpoint 1

These reports are stored in the message archive, too.

13.12 Maintenance groups

13.12.1 Overview

An "Entity" is an object that enables secure closure of line sections in case of maintenance work. In this case the ICM functionality ABM/M (affected by maintenance/maintenance) is used.

If an ICM needs to be set to "Maintenance", all ICMs that belong to this partial route must be in the safe status (valves must not open, motors or pumps not run, etc.). This switch-over should be guaranteed by the PCU Server.

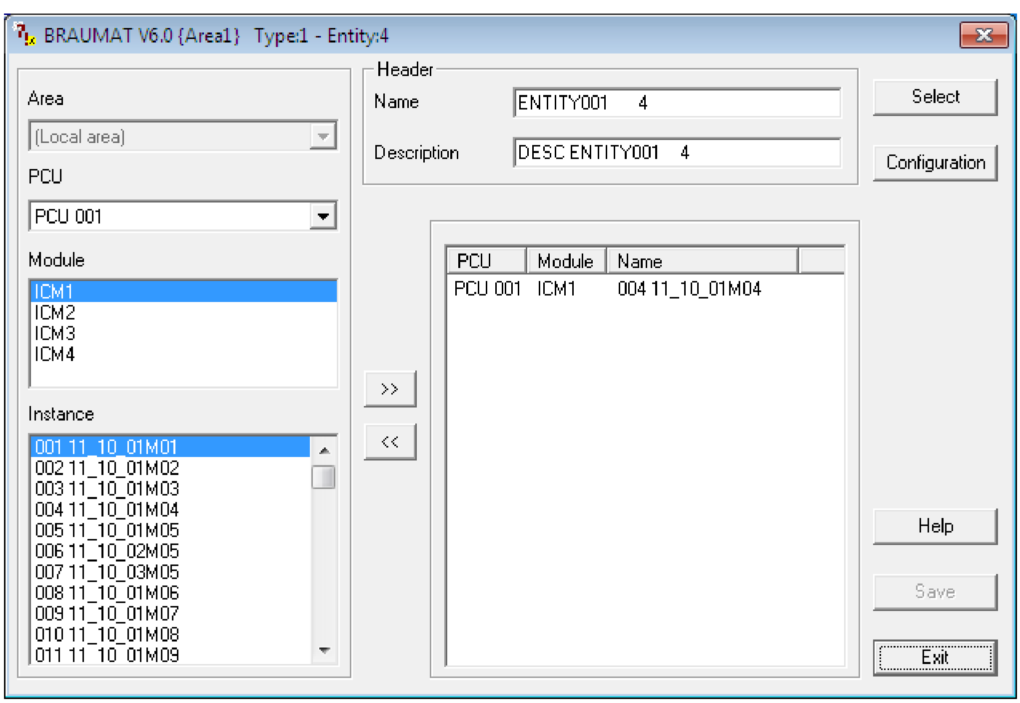
The assignment of ICMs to individual Entities is achieved with the "Configuration / Maintenance groups" application which is described below. The ICMs are stored in the Entity PCU-related.

13.12.2 Working with the application

Note:

This tool can only be called up if the text file "" ...<proj-path>\entities \pcunnn.txt"" exists (where nnn is the number of the required PCU).

This can be created by the configuration engineer (with empty content) or can be copied from the supplied version.



- Via the command button **Select** you reach a selection window via which the entity type and the entity can be selected.
- Via the command button **Configuration** new entities can be created.
- After an entity has been selected, the individual ICMs (area → PCU → module → instance) can be assigned or removed again with the "<< / >>" buttons.
- Via the command button **Save** the configuration is stored.

14 Message system

14.1 Overview

14.1.1 General

Basics

Messages are generated for certain operational events and errors in the automation system and sent to the IOSs. This is not required for messages of technological blocks, of the recipe system and of the route control.

Only the user messages have to be configured during the course of system engineering.

Overview

The messages are sent to the IOSs in whose FIFO the telegram type 3 is configured (see also: FIFO1 to FIFO6 - PCU system data general (Page 286)). Therefore it is possible to send the messages only to the IOSs which require them. By removing the message type 3 in the FIFO, all messages for this IOS are blocked.

The messages are filed in the message archive of the IOS which receives the messages. The message archives can be processed in the application 'Message archive'. There is the possibility to select and print the messages according to specific criteria.

If the PCU server window is opened on the IOS and the view 'Messages' is selected, the last current message will be output at the bottom of the window.

There are three types of messages:

- PCU system messages
- IOS messages
- User messages

PCU system messages are initiated by the system in the PCU, the message text is generated in the IOS.

The PCU creates system status messages and system fault messages. A system status message is e.g. a step selection for a unit or the switch-over of a controller. A system fault message is e.g. the exceeding of the monitoring time of a unit or an individual control element.

IOS messages are created by the system on the IOS, the text is pre-configured.

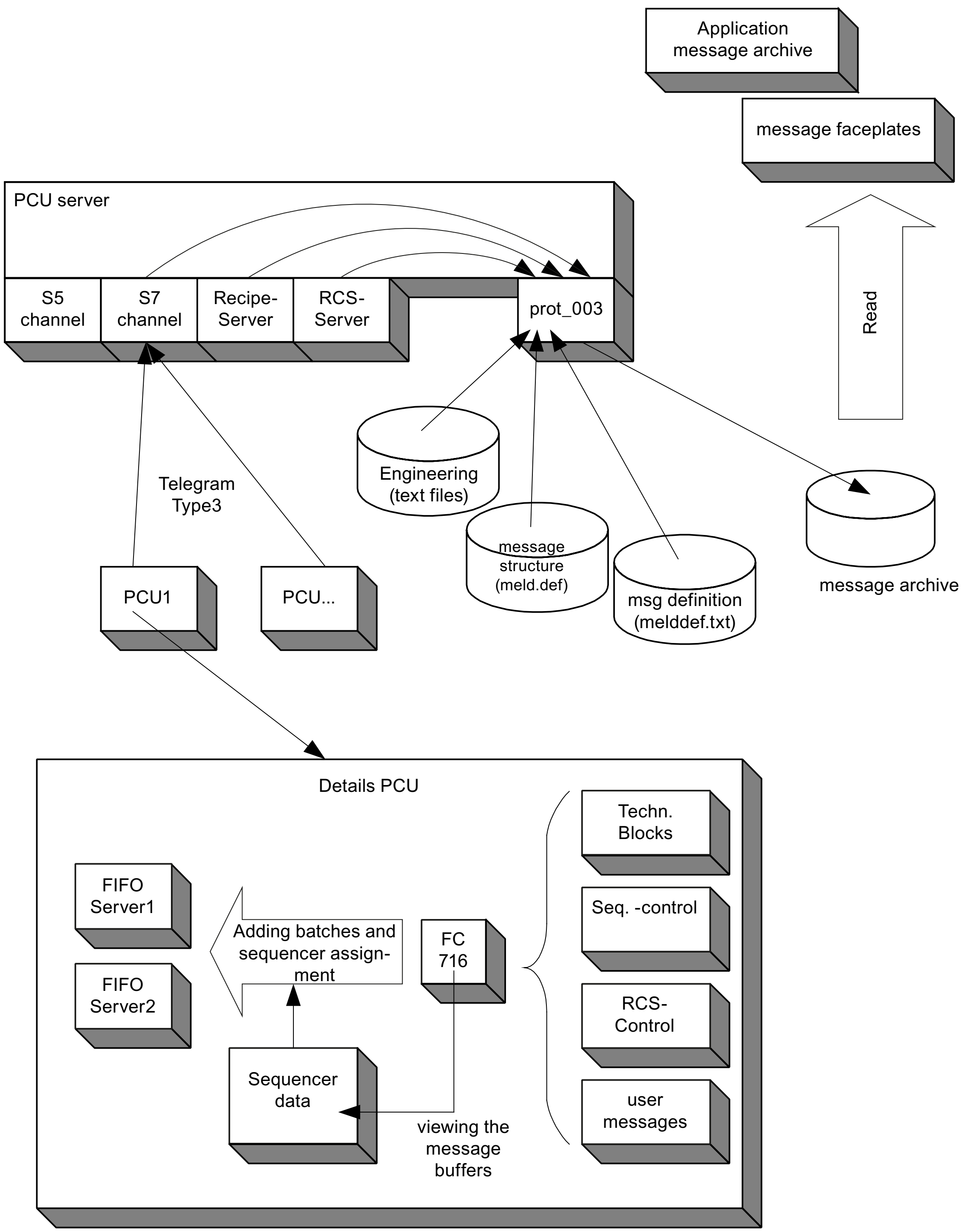
User messages are created on demand (user triggers initial flag). The texts of the user messages can be parameterized. There are status messages, fault messages, errors (faults) and control messages.

The texts for the incoming and outgoing message are configured in the application 'text parameterization' and for the PCU in which the messages are deposited, in files ‘Meldkom.txt’ and ‘Meldgeh.txt’ These files are located in the `‘<proj-path>\PCU.xxx\Texte’` or `‘<proj-path>\PCU.xxx\Texte.x\'` folder.

The indication as to whether the message is supposed to be output as a status message, fault message, error or control message is configured on the function block MSG (see also: MSG - Message block (Page 307)).

14.1.2 Message preparation

Basics



The raw data of messages are created by the technological blocks, by the sequencer control or by the route control in the PCU. The raw data are sent via a message of type 3 from the PCU to the IOS. The message type 3 must be configured in the FIFO head data.

Besides the PCU messages, raw data messages are also created from the recipe and route control in the PCU server. These messages also create messages; they are also of the message type 3.

These messages are handled by an IOS function within the prot_003.dll, which creates the real, readable message based on this raw data. For this, the system uses the specified files ""meld.def"" and ""meldsys.txt"". In the case of user messages, instead of ""meldsys.txt"" the project-specific file ""<proj-path>\texte.x\meldusr.txt"" is used to generate messages.

Visualization

The visualization is possible via:

- Message view of the PCU server window
- Application 'Message archive'
- Message faceplate in process images
- Message faceplate in the unit overview

Reference/assignment of messages

In this type all messages are related to a batch as well as to a unit.

The batch reference consists of:

- Batch year
- Recipe category of the batch
- Order number of the batch
- Batch number

Standard unit reference

All reporting blocks can be assigned to a unit. Based on this assignment, the batch reference is retrieved at the time a message is created. The unit reference is part of the message.

Unit reference with RCS

A route can be assigned to a unit whereby all messages of the RCS are assigned to this unit. The PCU number of the reporting object and the PCU number of the unit which controls a route can be different.

14.1.3 Message classes and types

14.1.3.1 Message classes and types

Basics

The messages are structured hierarchically according to the following features:

Message classes, message types

A message class can consist of several message types. Available message classes are: E - error, W - warning, M - process message, O - operator request.

Table 14-1 Names, assignment and colors with system delivery:

Message class	Description	Message type	Description	Color
E	Error/failure	F	User fault message	Violet
		S	System fault message	Red
W	Warnings	W	Warning	Orange
M	Process Messages	M	User status message	Green
		B	Operator message	Blue
O	Operation Request	O	Operation Request	Dark green
	- no class assigned -	P	IOS Message	Light brown

These assignments and the following settings are defined in the system configuration file '<proj-path>\sys\MsgClass.ini':

- Displayed shortcut character
- Assigned message types
- Flashing behavior
- Reset behavior
- Deletion behavior

Each message definition is assigned to a message type. The message type is displayed as the first character next to the time field in the message row. The whole number of message definitions and their configuration is distinguished as follows:

Message definitions / user messages

Additionally, there are 64 message definitions for the user. These messages need to be triggered via the block FC713. The message structure can be configured in the following definition file for every user message definition:

- Message definitions 64 – 127 → File '<proj-path>\texte.x\MELDUSR.TXT

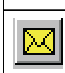
14.1.3.2 File MsgClass.ini

For every message class a section exists in the file MsgClass.ini where the following settings are defined:

`SignalChar=`	describes the sign that will be displayed in the plant overview with the defined message type.
`SignalChar10=`	describes the sign that will be displayed in the plant overview with the defined message type. Insert this line only if the output is language-dependent. If this line is entered this will be taken instead of the character defined in the SignalChar line. The number after the SignalChar describes the language.
`Displayed=`	describes the allocation of a message type to this message class. Several message types are entered, separated by commas.
`Flushing=`	defines which of the assigned message types is to be shown flashing.
`Reset=`	defines which of the assigned message types is to be reset by activating the corresponding button in the plant overview.
`Deleted=`	defines which message text of the assigned message type is deleted from the plant overview by resetting.

14.1.3.3 Message acknowledgment

The acknowledgment of the individual unit-related messages can be performed in the sequence control application or via the SmartUnit-OCX within process pictures.

	Delete the buffer for message class "Fault/Error"
	Delete the buffer for message class "Warning"
	Delete the buffer for message class "Process message"

14.1.4 ICM allocation

14.1.4.1 ICM allocation

- Every ICM has been assigned to a specific unit.
- With an error of this ICM, one message is created in which the recipe type, the job number and the batch number of the assigned unit will be entered.
- This allows you to retrace which unit was affected by the ICM error in a later evaluation of the message archive. This is an important condition in order to guarantee the safety of the product.

14.1.4.2 Allocation to the unit

If ICMs are used by different units at different times, the configured and assigned unit number should be changed according to the unit that uses the ICM.

Only then will the recipe type, the job number and the batch number be entered in the message with an ICM error that also shows an error.

The user can change the configured and assigned unit number in the user program.

1. Two functions are provided:
   All ICMs of one manual group are assigned to a specific unit:
   This is e.g. the case if two CIP units are available and both use the same ICMs. In this way it will be guaranteed that the correct recipe type, the right job and batch number will be entered and for the ICMs of the manual group depending on which unit is currently active it is considered that the assigned unit number should be rewritten.
2. An ICM is assigned to a specific unit:
   If ICMs are used by different units at different times, the configured and assigned unit number should be changed according to the unit that uses the ICM.

For the implementation of function 1, the block FC723 is provided for the user for the SIMATIC S7.

Function 2 works without a function block for the SIMATIC S7, since it can access all parameters of the data record by opening the data blocks as well as the UDTs easily and symbolically.

Function FC723:

By calling the function FC723 two parameters are to be transferred.

Formal operands of the FC723:

FC 723		
ITA	:=	Unit number
IHGrp	:=	Manual group

Example: Call of the FC723 for function 1:

All ICMs of the manual group 4 are supposed to be assigned to unit 7 in the EOP 31.

Call of the FC723 in FC1031:

FC 1031			
	UN	M 102.2	EOP Start
	SPB	NEXT	Jump
	CALL	FC 723	Call of the FC723 'GRP_TA_FC'
	ITA	:=7	Unit number 7
	IHGrp	:=4	Manual group 4
NEXT:			

Example: Implementation of function 2 by a symbolic access to the parameter 'byTA' of the ICM

The ICM no. 125 of the ICM group 1 should be assigned to unit 13 in the EOP 15.

Writing parameters 'byTA' of the ICM 5 in the FC 1015 on TA=13

FC1015			
	UN	M 102.2	EOP Start
	SPB	NEXT	Jump
	L	13	Unit number 13
	T	"ICM1".au[125].byTA	Access to DB "ICM1" (DB 726) on the field (array) of the 125th ICM on the parameter byTA (from ICM UDT)
NEXT:			

Example: Implementation of function 2, but for 100 ICMs in a loop with the use of the FC500 'ATTRIB_PTR_FC'

The ICM Nos. 100 to 200 of ICM group 1 should be assigned to unit 44 in the EOP15.

The FC500 'ATTRIB_PTR_FC' calculates the pointer on a data record attribute and transfers it to the AR1 register.

Parameters 'byTA' of the ICMs 100 to 200 in the FC1015 on TA=44 write

FC 1015			
	UN	M 102.2	EOP Start
	SPB	NTZ	Jump
	ONTO	"ICM1"	Looking up "ICM1" (DB726)
	L	100	From ICM 100
	T	#iESGNR	Presetting variables
NEXT:	CALL	FC500	Call FC500
	IDB	:=726	DB726
	Irecord	:=#iESGNR	Current ICM no.
	IrelByteOffset	:=5	byTA is the 5.Byte in the UDT
	IretVal	:=iRETVAL	Value = 0 => no error
	L	44	Unit number 44
	T	DBB[AR1,P#0.0]	Writing value on byTA
	L	#iESGNR	ICM no.
	+	1	Increment
	T	#iESGNR	
	L	200	Processing 200 ICMs
	<=I		as long as not all processed
	SPB	NEXT	Process next ICM no.
NTZ:			

14.1.4.3 Allocation to a group

Basic principles

The unit allocation in the blocks can also be used for the allocation to a group. All allocations are possible for PCU-crossed units. The current recipe type, the current recipe number, the job number and the batch number of the assigned unit are transferred via the cross-coupling.

In order to switch the unit allocation to a group, the allocation number should be more than 100. The group number results from the allocation number minus 100. The user should consider that the current data of recipe type, recipe number, job number and batch number can be found in the block GRUP_TA' .

Only then is it guaranteed that a fault signal can be assigned correctly later (see chapter GRUP_TA - Group block (Page 288)).

Group block 'GRUP_TA' DB723

Setup of the data block DB723	
DBW4	Extent 8 bytes per data record
DBW8	Number of data records
From DBW20	1st data record
DBB20	Year for Rtyp, JobNo, BatchNo
DBB21	Recipe type
DBW22	Recipe number
DBW24	Job number
DBW26	Batch number
From DBW28	2. Data record

14.2 System messages

14.2.1 Overview

The system messages described below are issued by the individual technological blocks.

For more detailed information on the message structure and message types, refer to the following sections. For the meaning of the messages, refer to the description of the individual system blocks.

14.2.2 Messages of the block 'SEQU'

SEQU Abort:

- if the unit is set to step 0.

```
12:05:23 B Production 00122 03421 002 SEQU 007 LB1 unit message abort
```

'SEQU step selection':

- if a step is selected manually.

```
12:07:15 B Production 00122 03421 002 SEQU 007 LB1 unit message step
selection
```

'Sequence error EopMonTmError':

- if the monitoring time has expired and the control bit EnableMonTm (M101.7) is set.

```
12:09:28 S Production 00122 03421 002 SEQU 007 LB1 error monitoring
time beginning
```

SEQU sequence Stop/sequence enabled:

- by changing Enabled/Stop by the user

```
12:10:03 B Production 00122 03421 002 SEQU 007 LB1 unit message
sequence stop
```

```
12:10:47 B Production 00122 03421 002 SEQU 007 LB1 unit message
sequence enabled
```

SEQU Manual operation ON/OFF:

- Setting/resetting the flag bit for the assigned manual group

```
12:11:10 B Production 00122 03421 002 SEQU 007 LB1 unit message
Manual mode on
```

```
12:11:15 B Production 00122 03421 002 SEQU 007 LB1 unit message Auto
mode on
```

SEQU permanent conditions OFF:

- if no unit conditions are queued (change ON → OFF).

```
12:05:05 B Production 00122 03421 002 SEQU 007 unit message LB1:
continuous condition On
```

Configuration error SEQU:

- if a false DFM type is configured.

```
12:05:23 B Production 00122 03421 002 SEQU 007 LB1 unit message
Parameterization error
```

14.2.3 Messages of the block 'ICM'

ICM error:

if the error memory QSP is set. This message is a fault message

```
12:15:23 S Production 00122 03420 002 ICM1 003 ICM1 monitoring time
V312 error. Beginning
```

```
12:18:27 S Production 00122 03423 002 ICM2 014 ICM2 monitoring time
W157 error. Beginning
```

14.2.4 Message of the block 'AIN'

AIN error:

if from the analog module the variable error bit is set, and the inhibit bit is not set in the AIN data record (Fault message)

```
12:18:43 S Production 00122 03421 002 AIN 012 measured value
monitoring TempWK1 error. Beginning
```

AIN Limit violation:

if the determined values for upper or lower limits are exceeded or fallen below and the enabling bit for error upper or lower limit is set in the AIN data record respectively.

```
12:18:43 S Production 00122 03421 002 AIN 012 TempWK1 upper limit
measured value monitoring exceeded
```

```
12:18:43 S Production 00122 03421 002 AIN 012 TempWK1 measured value
monitoring below lower limit
```

14.2.5 Messages of the block 'PID'

Manual operation PID ON/OFF:

by switching over from manual to automatic and reverse (status message).

```
12:21:18 B Production 00122 03421 002 PID 017 controller operating
mode WK:Temp manual on
```

```
12:21:55 B Production 00122 03421 002 PID 017 WK:Temp automatic
controller operating mode on
```

Internal operation PID ON/OFF:

by switching the controller on/off from external to internal operation and reverse (status message)

```
12:22:15 B Production 00122 03421 002 PID 017 WK:Temp controller
operating mode internal
```

```
12:23:01 B Production 00122 03421 002 PID 017 WK:Temp controller
operating mode external
```

14.2.6 Messages of the block 'ThreeStep'

ThreeStep manual operation ON/OFF:

by switching over from manual to automatic and reverse (status message).

```
12:34:22 B Production 00122 03421 002 ThreeStep 005 controller
operating mode HWT:Niv manual on
```

```
12:37:46 B Production 00122 03421 002 ThreeStep 005 HWT.Niv
controller operating mode automatic on
```

ThreeStep internal operation ON/OFF:

by switching the controller on/off from external to internal operation and reverse (status message)

```
12:38:29 B Production 00122 03421 002 ThreeStep 005 HWT:Niv
controller operating mode internal
```

```
12:38:43 B Production 00122 03421 002 ThreeStep 005 HWT.Niv
controller operating mode external
```

ThreeStep controller ON/OFF:

by switching the controller on/off (status message)

```
12:44:22 B Production 00122 03421 002 ThreeStep 005 HWT:Niv
controller controller operating mode off
```

```
12:45:46 B Production 00122 03421 002 ThreeStep 005 HWT.Niv
controller controller operating mode on
```

14.2.7 Message of the block 'SEQS'

Start error:

if the unit cannot be started (manual, no continuous condition, not step 0)(fault message)

```
12:46:35 S Production 00122 03421 002 SEQS 013 WK sequence start
fault start
```

Configuration error:

wrong transfer of the target/source parameters (fault message)

```
12:48:13 S Production 00122 03421 002 SEQS 013 WK sequence start
configuration error
```

14.2.8 Messages of the block 'DFMx'

Configuration error:

- General configuration errors

```
12:48:13 S Production 00122 03421 002 DFM0 013 IDM-027
```

14.2.9 Further system messages

Additional system messages

The table below includes system messages that are generated due to different error states.

The "leading" fields of the message line are specified here.

RecCatName	UnitName	PCU	Modules	InstName	MsgText
<Recipe type>		<PCU no>	PCU	DB no.: <DB no>	DB missing or too short
Lock path Release path	<Source IOS>		Path monitoring	<Target IOS>	Path=PROJ, Index=.. Path=SREP, Index=..

RecCatName	Uni-tName	PCU	Modules	InstName	MsgText
Recipe control	<Source IOS>		Redundancy		IOS becomes recipe server IOS becomes 'Standby Server' Live telegrams from recipe server are missing
PCU status	<Source IOS>	<PCU no>	PCU		PCU is in STOP PCU is in RUN PCU is offline PCU is online
TCP/IP status	<Source IOS>		TCP/IP	<Target IOS>	Connection established Connection disconnected
SQL-ADP sta-tus	<Source IOS>		SQL Adapter	<SistarProxy>	Connection established Connection disconnected
Login Logout	<Source IOS>		IOS		<username>
			RCS_Svr1 RCS_Svr2	<Target IOS>	Server power-up failed

14.3 User messages

14.3.1 Overview

There are two possibilities for creating user messages:

- Initiating via block MSG (standard interface)
- Initiating via FC713 (SIS_BAS library)

14.3.2 Initiation via the MSG block

Basic principles

User messages are transmitted at a specific time by the user. The text field is configurable by the user. There are operating messages and fault messages.

The texts for the incoming and outgoing messages are configured in the application 'text configuration' for the PCU, in which the messages are deposited in the files ‘Meldkom.txt’ and ‘Meldgeh.txt’. These files are located in the ‘<proj-path>\PCU.xxx\Texte’ or ‘<proj-path>\PCU.xxx\Texte.x\’ directory with engineering language switch over.

The individual message type is configured with the parameterization of the block MSG. The following types are supported:

EnableW/Op	S/BM	Type	Description
0	0	M	Process message
0	1	F	Error
1	0	B	Operator message
1	1	W	Warning

- For detailed information on the block, see chapter MSG - Message block (Page 307).
- For detailed information on message construction, refer to chapter Structure of the user messages (Page 1028).

Messages of the block 'MSG'

Take saccharification test

- Initiation of the message by setting the corresponding flags in the equipment operations or in the user programs
- Process message

```
12:21:55 M Production 00122 03421 002 MSG 053 message in take
saccharification sample
```

```
12:24:53 M Production 00122 03421 002 MSG 053 message out take
saccharification sample
```

Motor protection case of a drive

- Initiation of the message by setting the corresponding flags in the equipment operations or in the user programs
- Fault message

```
14:44:56 F Production 00122 03421 002 MSG 054 message in motor
protection drive 47M1
```

```
14:56:13 F Production 00122 03421 002 MSG 054 message out motor
protection drive 47M1
```

14.3.3 Initiation via FC713

Basics

User-configurable messages can be initiated by the user with the standard block **FC713** via the message system.

Parameterization

In the file '<proj-path>\texte.x\Meldusr.txt' the user configures the definitions for the messages to be output (see example below). The structure rules from chapter "Structure of the user messages (Page 1028)" apply here.

Call interface of Block FC713:

Call parameters:

With the first parameter 'iTaNr' a decision is made as to whether the recipe, order and batch data are taken from the handover parameters or from the relevant unit data set (see the following table).

Name	Type	Description
iTaNr	INT;	//=unit no. (0, 1 to 128) • **'iTaNr'** = 0: – Parameter 'iMsgType' irrelevant. – The 'byValDL4', 'byValDR4', 'iValDW5', 'iValDW6', 'iValDW7' parameters are assessed • **'iTaNr' =** 1 to 128: – The 'iMsgType' parameter is assessed – Parameter 'byValDL4', 'byValDR4', 'iValDW5', 'iValDW6', 'iValDW7' irrelevant → comes from the unit data set
iMsgType	INT;	//= Message types (1, 2, 5, 6) • Sets the corresponding bit of the configured message type in the unit data record. • If the relevant bit is set, the corresponding button for the message class to acknowledge the message will be enabled based on the '...<proj-path>\sys\MsgClass.ini' file. 1 - Error 2 - Process message 5 - Warning 6 - Operation
iMsgDef	INT;	//=iMsgType, message definition, DW8 (64...127) Index for user message in the 'Meldusr.txt' file
byValDL4	BYTE;	//=byTa_Year, year relevant for iTaNr=0 only
byValDR4	BYTE;	//=byRecType, recipe type relevant for iTaNr=0 only
iValDW5	INT;	//=iRecipe, recipe relevant for iTaNr=0 only
iValDW6	INT;	//=iOrder, order relevant for iTaNr=0 only
iValDW7	INT;	//=iBatch, batch relevant for iTaNr=0 only
iValDW9	INT;	//=iRecord, data record, instance
iValDW10	INT;	//=iSysTxt, error text
iValDW11	INT;	//=iTA, unit
wValDW12	WORD;	//=w24, Res.1
wValDW13	WORD;	//=w26, Res.2
wValDW14	WORD;	//=w28, Res.3

Note

Calling up the FC 713

- The FC713 should be called, edge-triggered, as one message is created with each call.
- If an output value is no longer needed, the corresponding transfer value of the column with the default value is allocated (default value).

Example user message in 'Meldusr.txt':

The following image shows a section of the 'Meldusr.txt' file:

```
 1 F·@················@······@······@···@···@················@···SEQU·····@····fault·start······@¶
 2 /+QS/DR4T16RTYP·DW6I6·DW7I6·DR0I3·DW11I3·DW11t16TEILANL·DR0I3·DW9I4·DW12T48MSGF¶
 3 B·@················@······@······@···@···@················@···SEQU·····@····fault·end········@¶
 4 /-S/DR4T16RTYP·DW6I6·DW7I6·DR0I3·DW11I3·DW11t16TEILANL·DR0I3·DW9I4·DW12T48MSGF¶
 5 W·@················@······@······@···@···@················@···SEQU·····@····alarm·start······@¶
 6 /+QS/DR4T16RTYP·DW6I6·DW7I6·DR0I3·DW11I3·DW11t16TEILANL·DR0I3·DW9I4·DW12T48MSGW¶
 7 B·@················@······@······@···@···@················@···SEQU·····@····alarm·end········@¶
 8 /-S/DR4T16RTYP·DW6I6·DW7I6·DR0I3·DW11I3·DW11t16TEILANL·DR0I3·DW9I4·DW12T48MSGW¶
 9 M·@················@······@······@···@···@················@···SEQU·····@····message·start····@¶
10 /+QS/DR4T16RTYP·DW6I6·DW7I6·DR0I3·DW11I3·DW11t16TEILANL·DR0I3·DW9I4·DW12T48MSGM¶
11 B·@················@······@······@···@···@················@···SEQU·····@····message·end······@¶
12 /-S/DR4T16RTYP·DW6I6·DW7I6·DR0I3·DW11I3·DW11t16TEILANL·DR0I3·DW9I4·DW12T48MSGM¶
13 B·@················@······@······@···@···@················@···SEQU·····@····log·start········@¶
14 /+QS/DR4T16RTYP·DW6I6·DW7I6·DR0I3·DW11I3·DW11t16TEILANL·DR0I3·DW9I4·DW12T48MSGB¶
15 B·@················@······@······@···@···@················@···SEQU·····@····log·end··········@¶
16 /-S/DR4T16RTYP·DW6I6·DW7I6·DR0I3·DW11I3·DW11t16TEILANL·DR0I3·DW9I4·DW12T48MSGB¶
17 F·@················@······@······@···@···@················@···SEQU·····@····fault·start······@¶
18 /+QS/DR4T16RTYP·DW6I6·DW7I6·DR0I3·DW11I3·DW11t16TEILANL·DR0I3·DW9I4·DW12t48MSGF¶
19 B·@················@······@······@···@···@················@···SEQU·····@····fault·end········@¶
20 /-S/DR4T16RTYP·DW6I6·DW7I6·DR0I3·DW11I3·DW11t16TEILANL·DR0I3·DW9I4·DW12t48MSGF¶
21 W·@················@······@······@···@···@················@···SEQU·····@····alarm·start······@¶
22 /+QS/DR4T16RTYP·DW6I6·DW7I6·DR0I3·DW11I3·DW11t16TEILANL·DR0I3·DW9I4·DW12t48MSGW¶
```

The parameters of a message definition are explained in the following table using the message number as an example: '68' (= 5th user message = lines 9/10).

Line column	Line 9 = Text / Place-holder	Line 10 = Associated tele-gram parameter	Message display/meaning
1	`M`	`/+QS/`	'Type' column User operator message (text color green) Incoming error message / Error message requiring acknowledg-ment
2	`@`	`DR4T16RTYP`	'Recipe type' column Text 16 char-acters from '...\texte.x\RTYP.txt'
19	`@`	`DW6I6`	'Order no.' column INT 6 characters
26	`@`	`DW7I6`	'Batch INT 6 characters' column
33	`@`	`DR0I3`	'TaPCU' column INT 3 characters

Line column	Line 9 = Text / Place-holder	Line 10 = Associated tele-gram parameter	Message display/meaning
37	`@`	`DW11I3`	'TA' column
41	`@`	`DW11t16TEILANL`	'TaName' column Text 16 charac-ters from '...\pcu.nnn\texte\Teilanl.txt'
58	`@`	`DR0I3`	'PCU' column INT 3 characters
62	`SEQU`	`-/-`	'Module' column fixed text
71	`@`	`DW9I4`	'No.' column INT 4 characters
76	`Message start`	- / -	'Name' column fixed text
93	`@`	`DW12T48MSGM`	'Text' column Message text 48 char-acters from '...\texte.x\MsgM.txt'

Example for the FC 713 call:

```
U M 3078.2
 FP M 3079.2
 SPBNB M005
 CALL "FC713"
 iTaNr :=0 // byValDL4, byValDR4, iValDW5, iValDW6, iValDW7 are
relevant
 iMsgType:=2 // 2=Process message
 iMsgDef :=68 // message number relating 'MELDUSR.TXT'
 byValDL4:=B#16#b // Year 2011;
 byValDR4:=B#16#3 // recipe type
 iValDW5 :=1234 // recipe number
 iValDW6 :=1234 // job number
 iValDW7 :=12345 // batch number
 iValDW9 :=1 // message instance
 iValDW10:=0 // not used
 iValDW11:=1 // unit identifier
 wValDW12:=W#16#3 // message text
 wValDW13:=W#16#0 // Res.2
 wValDW14:=W#16#0 // Res.3
M005: NOP 0
```

Associated user message in the message window (PCU server)

Time	Type	Recipetype		OrderNo	Batch	UnPCU	Unit	UnName	PCU	Module	No	Name	Text
18:12:52	M	RTYP	3	001234	012345	001	001	Malzannahme	001	SEQU	0001	message start	Standby tank selection required

14.3.4 Structure of the user messages

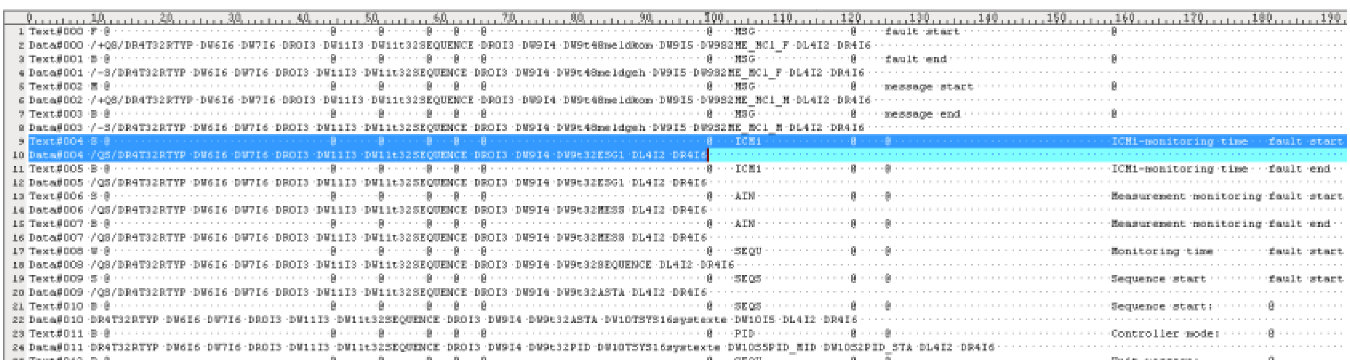
Definition file for user messages

The definition file '<proj-path>\texte.x\Meldusr.txt' provides the format template for user messages (64 - 127).

Creating a definition file

Each message definition occupies 2 lines in the file. The message definition is numbered consecutively. For example, the first and second lines (see the section in the following image) determine the message definition 64 (= first user message), the next two lines determine message definition 65, etc. The character positions or the start of the different text fields within a definition line define the message structure for the message archive files, but without the time field, as this is created by the system itself.

Example:



Structure of the odd lines

The odd lines may consist of fixed text or placeholders. The column position within the line determines the text or placeholder position. The placeholders are numbered continuously, and each one requires a definition in the next line.

- The message type shortcut is entered at column position 1: for example 'S', with the text 'ICM1' beginning at column position 103.
- The text of the first placeholder is entered starting at column position 3, the text of the second placeholder at column position 19. Placeholder and following space characters determine and limit the actual string length of the substituted parameters. Column-separating spaces have to be considered here, too.

Structure of the even lines

The even lines contain the definitions for the placeholders of the previous odd line.

Table 14-2 Structure:

Column	Characters	Meaning
1	/	Initial character
2	+ or -	Incoming or outgoing message
3	Q or space	Message with compulsory acknowledgment
4	S or space	Fault message
5	/	End character
6 … and following	...	Placeholder definitions for substitution, in the same sequence as in the odd lines.

Definition of placeholders

The definition describes the conversion of the raw data (see below, structure of telegram type 3) in a text.

Table 14-3 Syntax: <Address type><Address><Data type><Length><Text file>

<address type>	DW = data word DR = right data word DL = left data word
<address>	represents the DW number in the datagram.
<data type>	I = Integer number S, T = global text substitution (following text file, see below) t = PCU-specific text substitution. The PCU-specific text substitution always refers to the object from which the message originates (datagram origin). If this is not the case, which is possible with RCS messages, the PCU name appended to the file name can be defined in more detail, i.e. the address must be configured in the datagram that contains the PCU number. e.g. DW11t16SEQUENCE(PCU=DL12)
<length>	length of the value or text written to the message line.
<file name>	The file name must be defined without *.txt extension. Depending on the data type the following folders are valid: 't' → '<proj-path>\PCU.nnn\texte' 'T' → '<proj-path>\texte.x' 'S' → '<proj-path>\sys\msg'

Example:

Definition of a system message in lines 9 and 10 of the 'Melddef.txt' file (see image above)

Explanation of the placeholder:

- What is defined here is a "fault" message which must be acknowledged with the text: ‘ICM1 monitoring time fault start’
- DW4T16RTYP
  The value is read from DW4 of the datagram and is then interpreted as index. The indexed text line of the shared text file "rtyp.txt" substitutes the first @ placeholder. The text has a length of 16 characters with one separating blank to the next @ placeholder.
- DR0I3
  From DW 0 (LSB), the value is entered as an integer number with a length of three characters.
- DW11t16SEQUENCE
  The line of text of the PCU-dependent text file "teilanl.txt" indicated by the value of DW11 is read in and replaces the placeholder with a length of 16 characters.

Structure of type 3 message telegrams

DW	DL	DR
0	3	PCU number
1	Year	Month
2	Day	Hour
3	Minute	Second
4	Year for batch	Recipe type
5	Recipe number	
6	Order number	
7	Batch number	
8	Message type	
9	Message definition number	
10	Depends on the message definition (usually a system message text number)	
11	Depends on the message definition (usually an assigned unit)	
12	Depends on the message definition	
13	Depends on the message definition	
14	Depends on the message definition	
15	Depends on the message definition	

See also

Initiation via FC713 (Page 1024)

14.4 Message display

14.4.1 Overview

You have three options for on-screen message output:

- Use PCU server as client-global message (→ see chapter PCU server as message window (Page 1031))
- Using the application Archive display (Page 940)
- Implementation of message faceplates in the process images
  - Configuration, see chapter Message View Control (Page 457)
  - Runtime functions, see chapter Message View Control (Page 496))

14.4.2 PCU server as message window

Basic principles

In the PCU server, select the display in the menu 'Message→Message output'. The last active message is displayed in the bottom line, starting from the left.

- When sending messages, it should be ensured that the FIFOs are programmed so that the IOS also receives type 3 message frames (see chapter Modifying PCU settings (Page 121)).
- Message types to be received may not be locked in the 'LOCK' block of the transmitting PCU.
- The messages appear in a certain color that is determined by the message type.

Assignment of colors to the message types:

Message class \ source	PCU system messages	User messages	IOS messages
System messages	B Blue on white	M Green on white	P Light brown on white
Error messages	S Red on white	F Violet on white	P Light brown on white
Warning	W Orange on white	W Orange on white	---
Operation Request	O Dark green on white	---	---

15 Route control configuration

15.1 General

15.1.1 Introduction

The Route Control is used for controlling and monitoring production routes in industrial process plants. Simple route connections up to a variety of complex route combinations are possible, which depend on the plant complexity. Route Control is one possibility to simplify and standardize configuration, processing and diagnosis of route connections. The assigned route can be determined, tested, controlled, monitored and watched with the help of a Route Control in case of a Route request (from Source via Partial Route to Destination).

All possible route connections of a production area can be defined by the offline configuration tool, which is part of this manual. This configuration is used for the possible selection, which may be started and ended by the Automation System.

Note

Note on CPU type SIMATIC S7-1500:

With the current system version, the system also supports the SIMATIC S7-1500 controller family.

The release is initially subject to the following limitations in this stage.

- The "Route Control System" option is not supported (modern solution for future version planned)
- For this reason, the following description applies only to use with the **SIMATIC S7-400** controller family.

15.1.2 Quantified Project Specification

The Route Control can transfer and monitor up to 300 material transports across the Automation System in parallel.

It is possible to configure and monitor in parallel up to a maximum of 16 Automation Systems (AS) at present.

In this connection you can create per AS,

- 1024 Control Elements
- 1024 Sensor Elements
- 1024 Parameter Elements
- 1024 Link Elements

.

There aren't any configuration limits in contrast to create:

- Locations
- Partial Routes
- Plants

Note

- The determined route must not contain more than 450 elements per AS by a Route Request. This limitation depends on the capacity of the route list data blocks in the Automation System, which may only be up to a maximum of 16 kilobytes.
- However, you can influence the search algorithm by indication of the locations.

15.1.3 Explanation of definitions

Topic

The individual components of the Route Control engage with each other strongly. The most important definitions are explained in order to classify detailed descriptions in the following section.

- Control Element (CE)
  Control elements represent the active final controlling elements in a process via which you can control a route: Motors, valves.
- Sensor Element (SE)
  They are passive elements. They enable you to consider process conditions with controlling of the current route: Flow velocity, conductivities, limit switch.
- Parameter Element (PE)
  Parameter Elements are used for setpoint allocation. Values (32Bit Integer) of this storage location can be continued with processing in the AS by the user.
- Link Element (LE)
  It is possible to consider information relating to materials in the route search with the help of the link elements. This is filed as material, which was transported last in the link elements. With a route search this information is evaluated and e.g. removed, so that a material transport sequence, which isn't allowed, participates.
- Partial Route
  A partial Route corresponds to a route section. The configured Partial Routes form the route net, where the search algorithm searches special Partial Routes and combines them into a route.
- Route
  Only Partial Routes are configured. Suitable Partial Routes are assembled into a Route and elements are loaded in the AS. A Route consists of 1-n Partial Routes, consequently.

- Locations
  Locations are only "nodes" of the Route net. Partial Routes are created by a suitable selection.
- Mode tables
  The Mode Table corresponds to a function catalog, which could include up to a maximum of 32 functions. Individual elements are assigned to this function catalog, which are started and closed by defined function steps on reverse.

15.2 Engineering

15.2.1 Preparation

15.2.1.1 Overtaking a current project

The conversion tool is installed by the installation of the configuration tool (Offline tool) too. It is used for adapting available route control databases to the current database structure of the configuration tool. Therefore it is recommended to update the configuration database after each installation of the configuration tool.

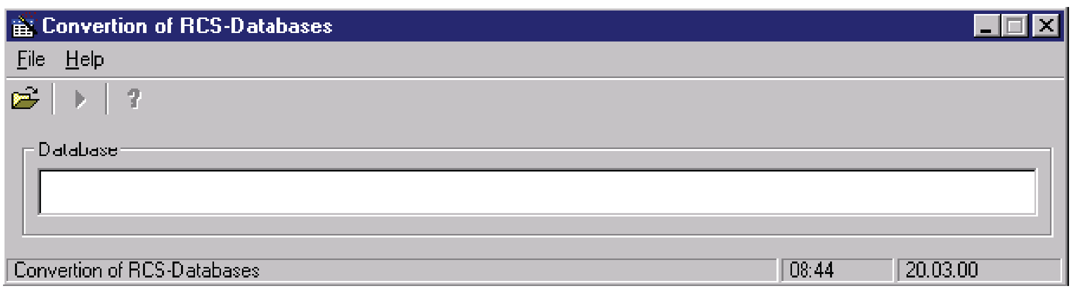


Image: Menu of the conversion tool

The database is selected via the menu under "File” and "Select database”; alternatively the symbol can be chosen. A dialog appears in which the database should be selected that will be loaded with "Open” in the configuration tool.

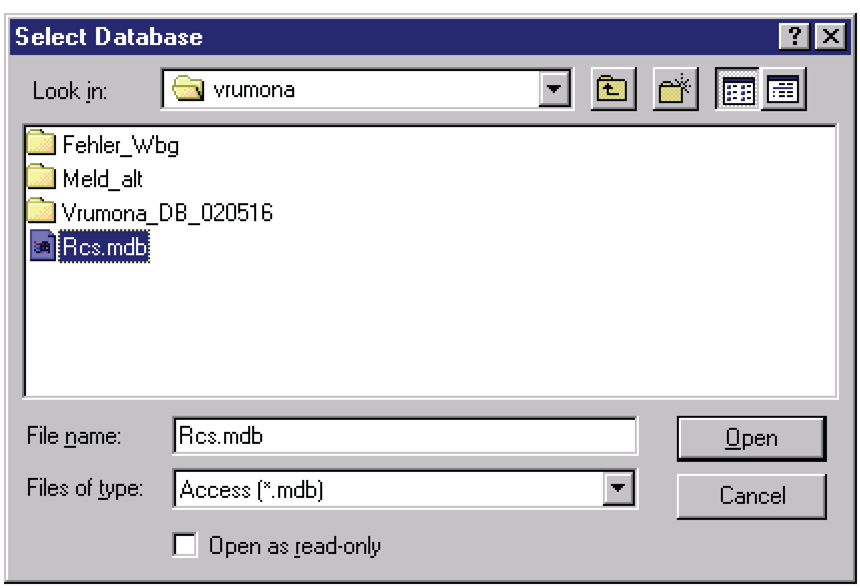


Image: Select database to convert

In order to start the conversion, please press "▸". The original database is filed in a file with the extension ".sic" in order to be safe. In case of problems with the new version later it is possible to create the original configuration again by renaming the safeguard file. If the conversion was successful the user will be informed by a dialog that should be confirmed.

15.2.1.2 Create a new project

The configuration is started with the "RCS configuration" icon ("Engineering" tab).

If the configuration tool is used for the first time, an empty editor window opens.

In the later process the name and the directory structure of the last active project are filed in the registration of the operating system. With a restart the last processed project will be opened automatically.

Creating a new RCS project

You can create a new project with the menu *"File|New"*. Here, the Windows "Open File" dialog opens with the preselected project directory "`<proj-path>\RCS\Database`".

The view of the engineering database is divided into two areas:

- The upper area shows the elements (individual blocks). This means that the process cell to be configured is configured in its technological structure in the upper area.
- In the lower area the routes and Partial Routes are combined. Here the plants will be configured in their structure.

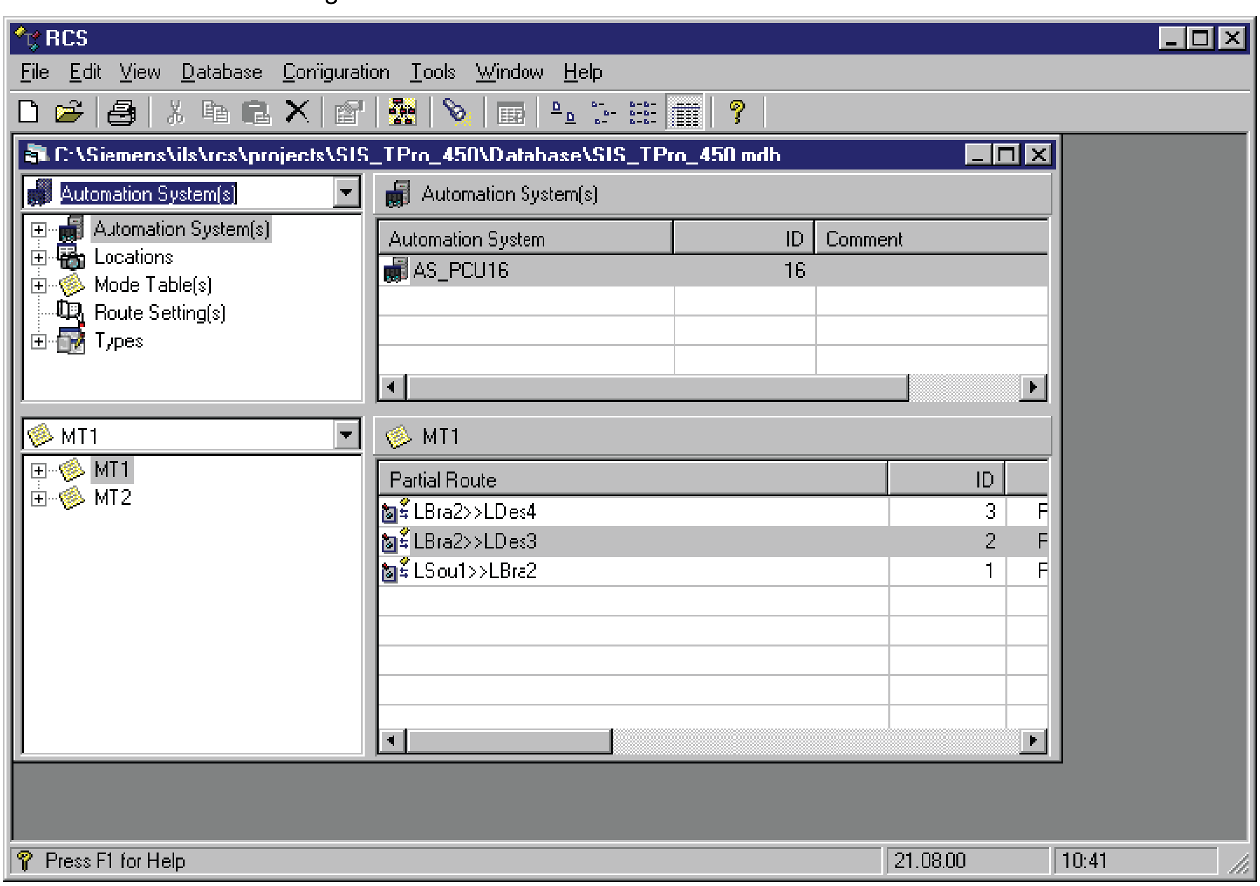


Image: Example Project

15.2.2 Configuring the objects

15.2.2.1 New Automation System (AS)

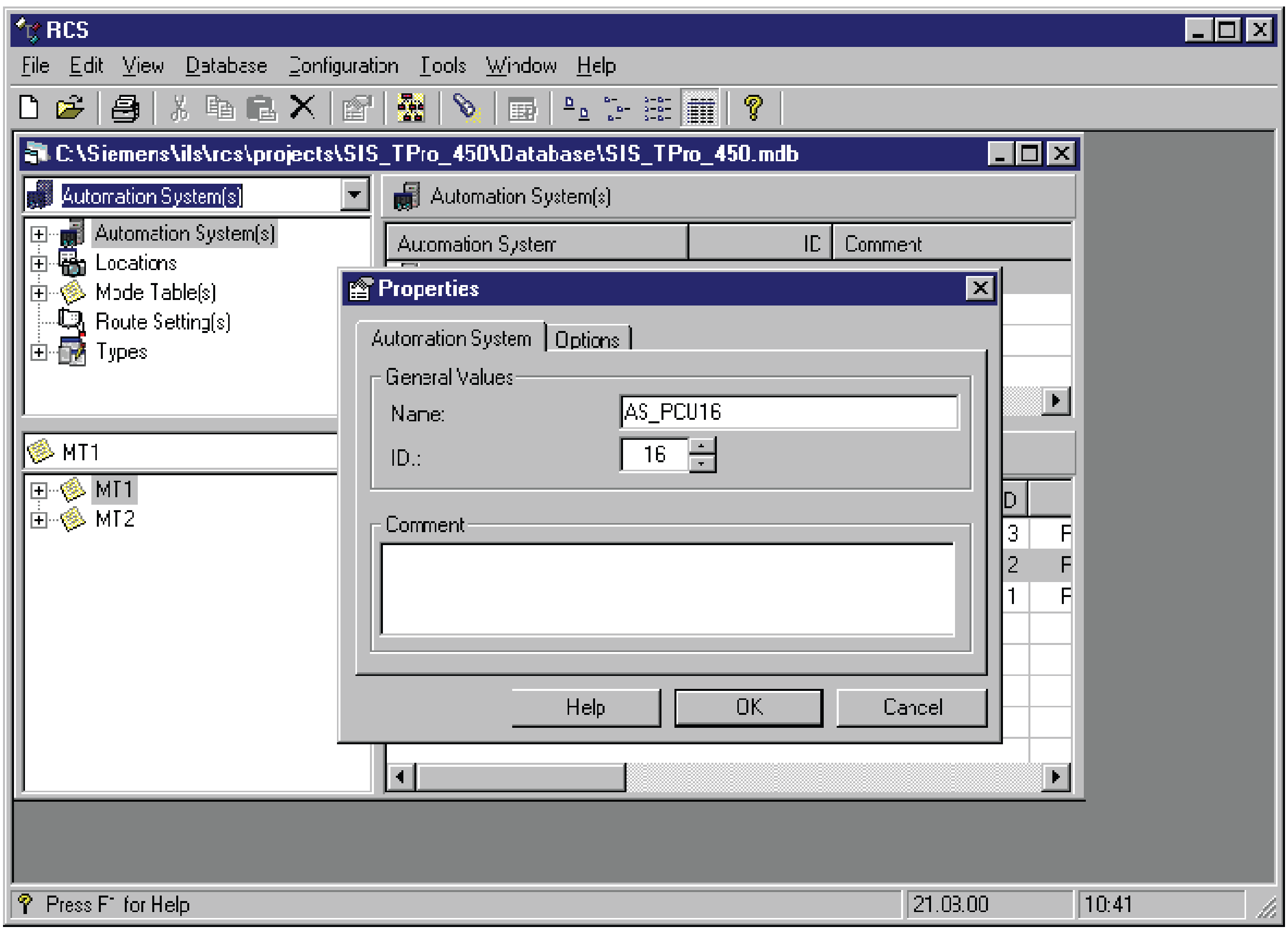


Image: Automation System

A pulldown menu opens upon pressing the right mouse key on "Automation Systems" in order to select and create a new Automation System.

The opened dialog window expects a name, an "ID" and a comment. Selecting the OK button creates the Automation System.

Notice:

The AS ID always corresponds to the configured PCU number in the system configuration.

15.2.2.2 New Elements

Basics

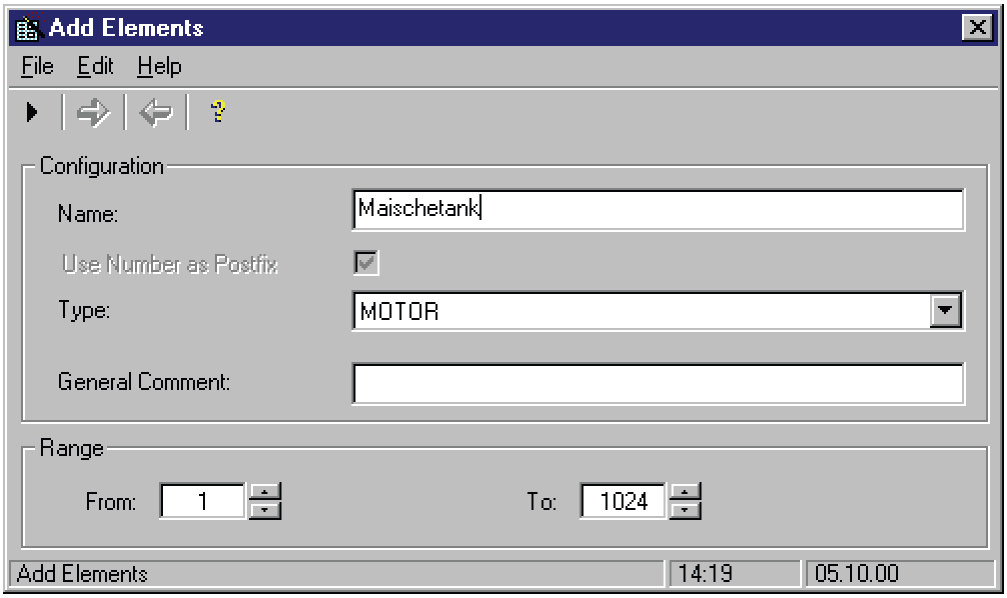


Image: Automation item

You have the possibility to allocate Control, Sensor and Parameter elements to the Automation System which was configured new. You use the node of the tree structure of the new Automation System and open tree structure to the corresponding type nodal point. In the above-demonstrated view the nodal point "MOTOR" was selected. By pressing the right mouse key you open the context menu. In the above-demonstrated view the nodal point "MOTOR" was selected.

In this way it is possible to create Control, Sensor and Parameter elements.

The following indications as well as specifications can be made in the dialog.

In the input line ”Name” it is possible to indicate the name (user-specific). Under 'type' you allocate the requested element type. You can indicate the ID number of the elements in the dialog below which has to be in a range between 1 and 1024. In our example 1024 Control Elements of the type "Motor" are created. These Control Elements receive the ID numbers 1 to 1024 as well as the name "Mash Container + consecutive number".

In order to avoid elements with duplicate names, the ID number of any element is appended to its name. This can be disabled by selecting an element individually and deactivating the "Use Number as Postfix” check box. However, this is only possible if just 1 element has been selected as area.

In the event that not all 1024 elements (CE, SE or PE) have been entered, the left and right arrow keys can be used to create new elements in the free range.

With the menu item "edit", by selecting 'next area'/'previous range' you can jump from one gap to another in the configuration database.

Elements that are no longer used can be deleted from the database. The operator selects the element, then either presses the delete key or uses the context menu (right-click) to perform the deletion.

A maximum of 1024 elements can be specified for each element group (CD, SE and PE).

Another new Automation system can be created more rapidly and easily using the copy and paste functions. The number, designation and type of each item are copied and created automatically.

Assigned items can also be copied and deleted using the "Copy/Delete AS" function.

Important

The CE number corresponds to the ICM number in the PCU.

Here the assignment for **PCU < V7** is:

- ICM1.1-255 corresponds to CE 001-255
- ICM2.1-255 corresponds to CE 257-511
- ICM3.1-255 corresponds to CE 513-767
- ICM4.1-255 corresponds to CE 765-1023

Here the assignment for **PCU V7** is:

- ICM1.1-256 corresponds to CE 001-256
- ICM2.1-256 corresponds to CE 257-512
- ICM3.1-256 corresponds to CE 513-768
- ICM4.1-256 corresponds to CE 765-1024

Note

- Only the CE types "Motor" and "Valve" are relevant in this system version.
- The ICMs 256, 512, 768 and 1024 can only be used when the respective instance names exist in the text parameter assignment in the PCU directory "texte\ESG1.txt ...ESG4.txt" or "texte.x\\ESG1.txt ...ESG4.txt". This is of particular importance for the migration of PCU version < V7 to PCU V7.

15.2.2.3 Control Element

Basics

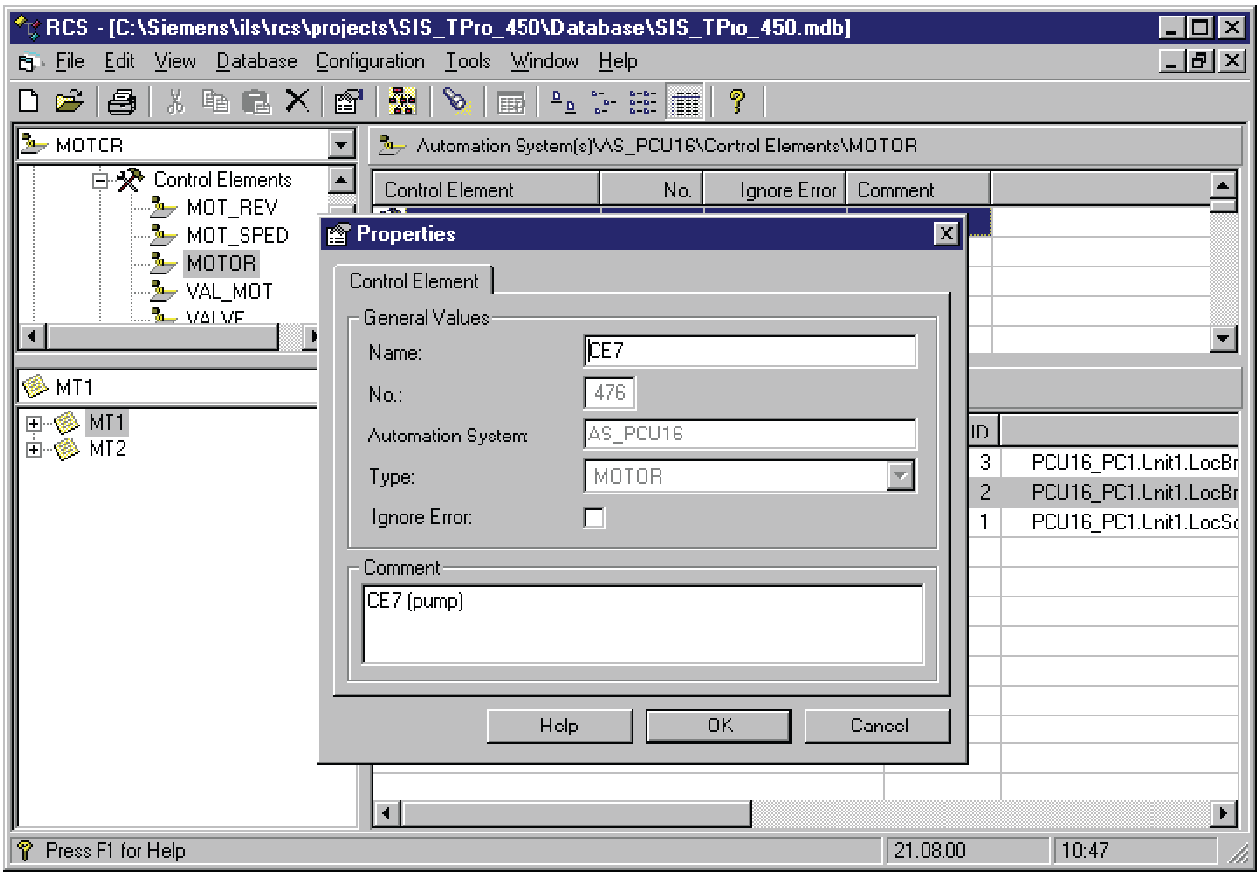


Image: Control Element

The Control Element (CE) represents the active element in an Automation System. Each Control Element is assigned a type with the corresponding reaction filters. 1024 active elements are available per Automation System.

- Only the "Motors and Valves" types are available by default. Only these types are interconnected via a standard wiring (FC819) to the individual ICMs, to an assignment which has already been named.
- By selecting the element and pressing the right mouse key, you receive a context menu in which you call the menu "Properties".
- The standard dialog window appears for all Control Elements. It is possible to assign a new name, new Control Element Type and a comment to each Control Element.
- If you make a change, all inputs will be overtaken and updated in the lists by closing the dialog window.
- If you assign a different type to a CE element, then this one will be removed from the corresponding type list and assigned to the new one.

"Ignore Error" selection box

If this option is set, the route is not switched off in the case of an error (on/off monitoring/ operation monitoring). Its activation on an individual CE always takes priority, meaning that it is active even if "Ignore Error" has not been set in RCS online for the entire route.

15.2.2.4 Sensor Element (SE)

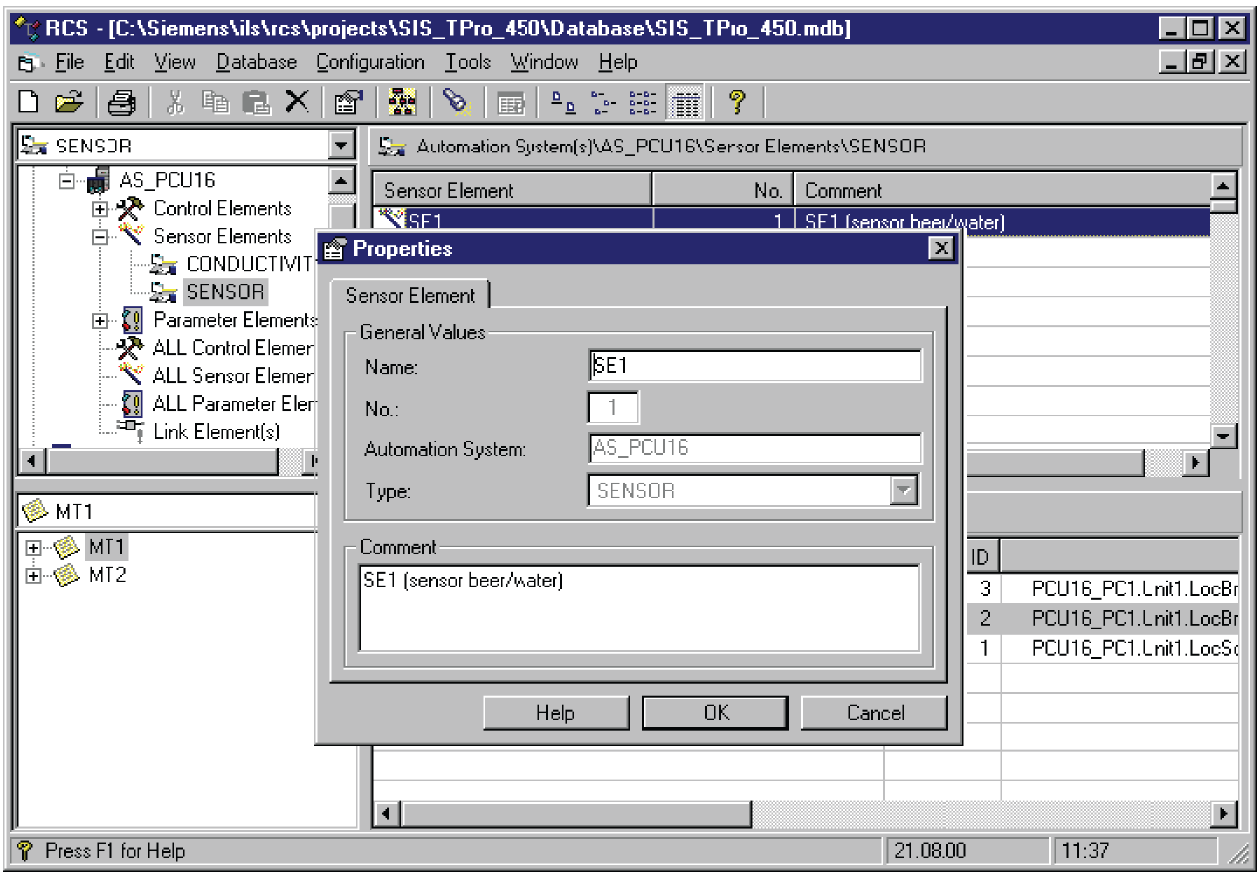


Image: Sensor Element

The Sensor Elements (SE) are used to provide status information which play an important role for a route. These are passive elements containing only signals of messages of 16 Bit values. Standard types for sensors are, for example limit switches. A fixed number of 1024 sensor elements whose type can be freely selected is preset for each automation system.

The following standard types have already been included and are described by the following interface blocks:

Table 15-1 Standard types of a Sensor Element

Type	Function	AS Block
SENSOR	Sensor	RC_IF_SENSOR FB808

15.2.2.5 Sensor Element Type (SE Type)

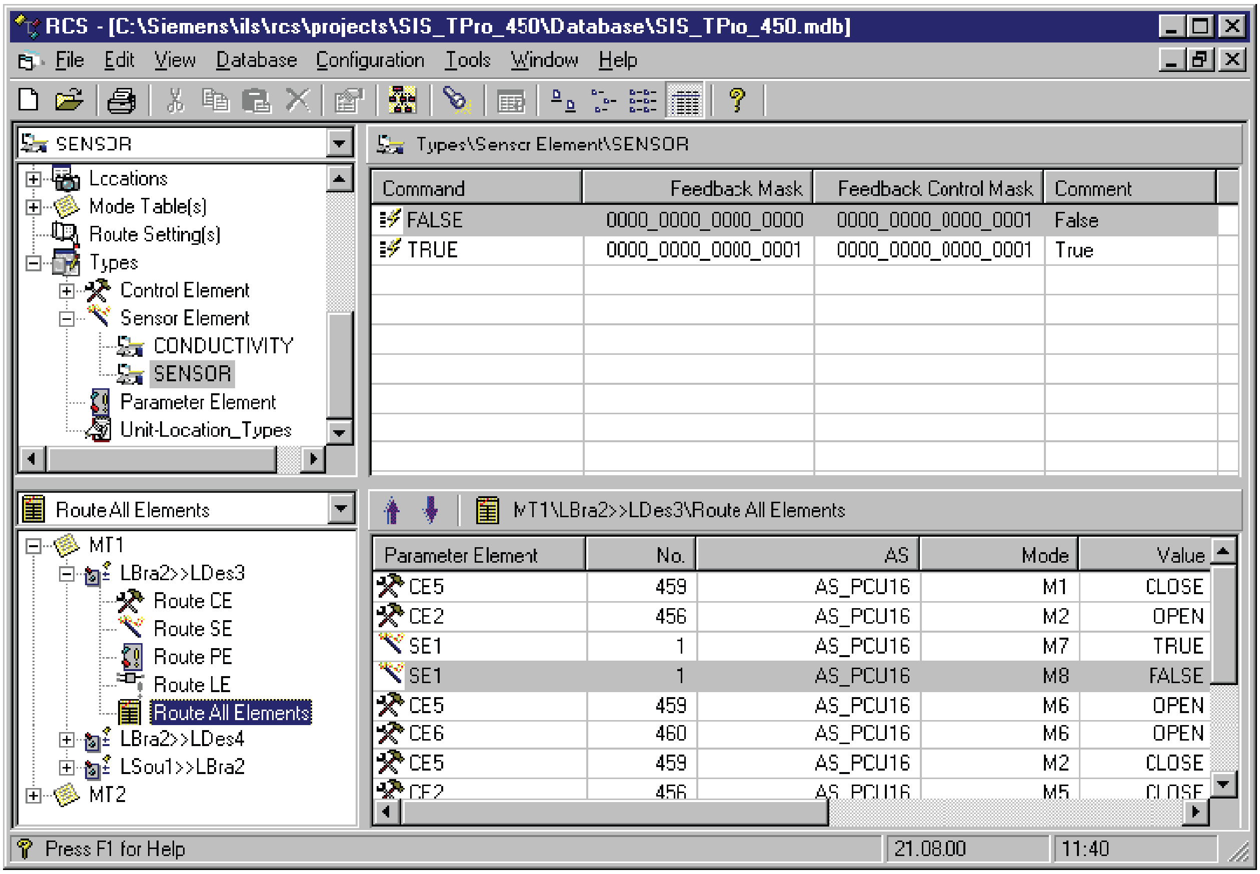


Image: Sensor Element Type (SE Type)

Here, all different types of passive Route Elements (e.g. Sensor, Conductivity) are defined. Only feedback value is generated according to the state of the element as these are only passive elements.

The list of Sensor Element types is also expandable here.

By clicking with the right mouse key and via the item "Add New Type", you can define a new Sensor Element type.

15.2.2.6 Sensor Element Feedback (SE Command)

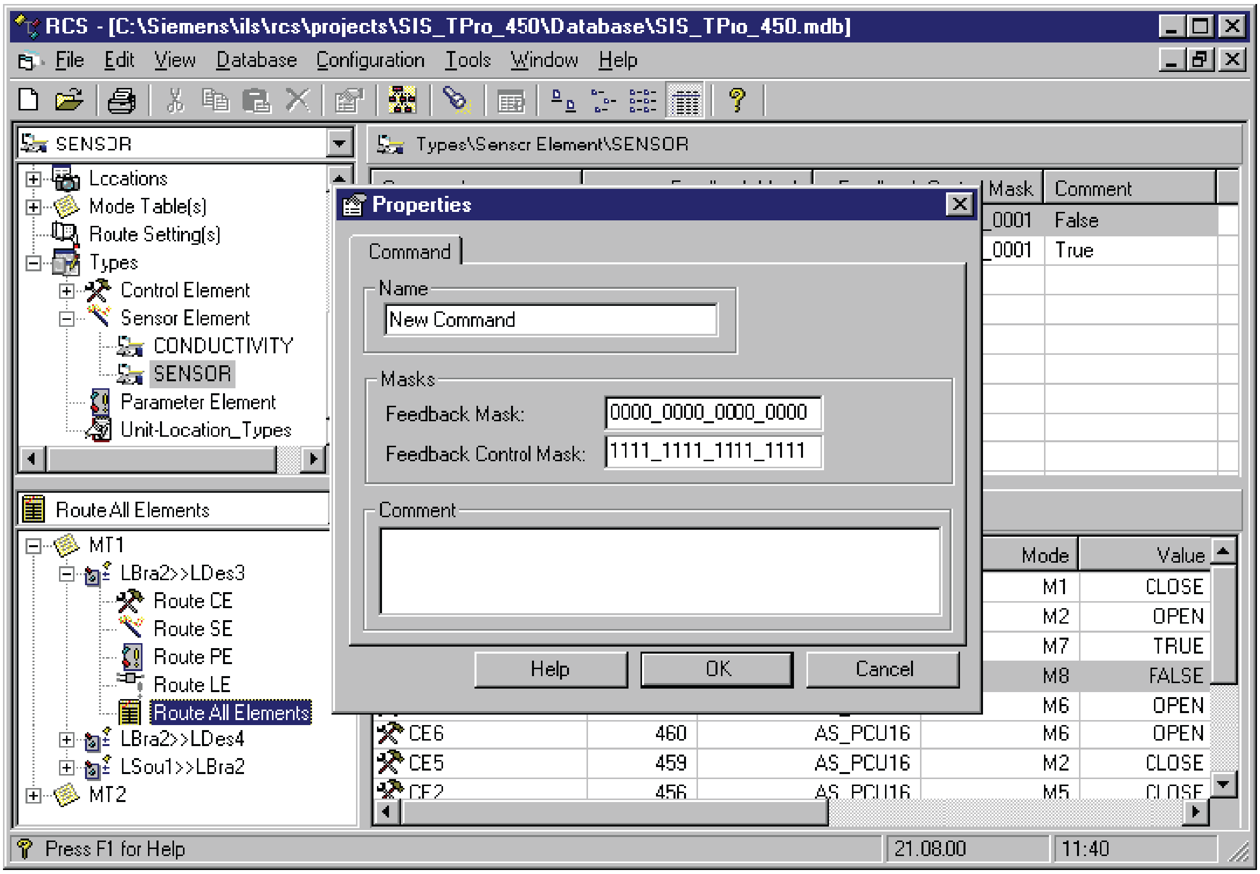


Image: Sensor Element Command

In the chapter 'Sensor Element Type' it has already been mentioned that Sensor Elements only have a Feedback value since they serve only to query information. There are 16 bits available for the Feedback and the Feedback Mask similar to Control Element. Here, a Sensor Element for a plug bow is described which occupies five positions.

Table 15-2 Example of an SE type with Commands of a quadruple plug bow as Decoder.

Plain Text	Feedback value
Not plugged	0000_0000_0000_0000
Connection to 1	0000_0000_0000_0001
Connection to 2	0000_0000_0000_0010
Connection to 3	0000_0000_0000_0100
Connection to 4	0000_0000_0000_1000

15.2.2.7 Parameter Element (PE)

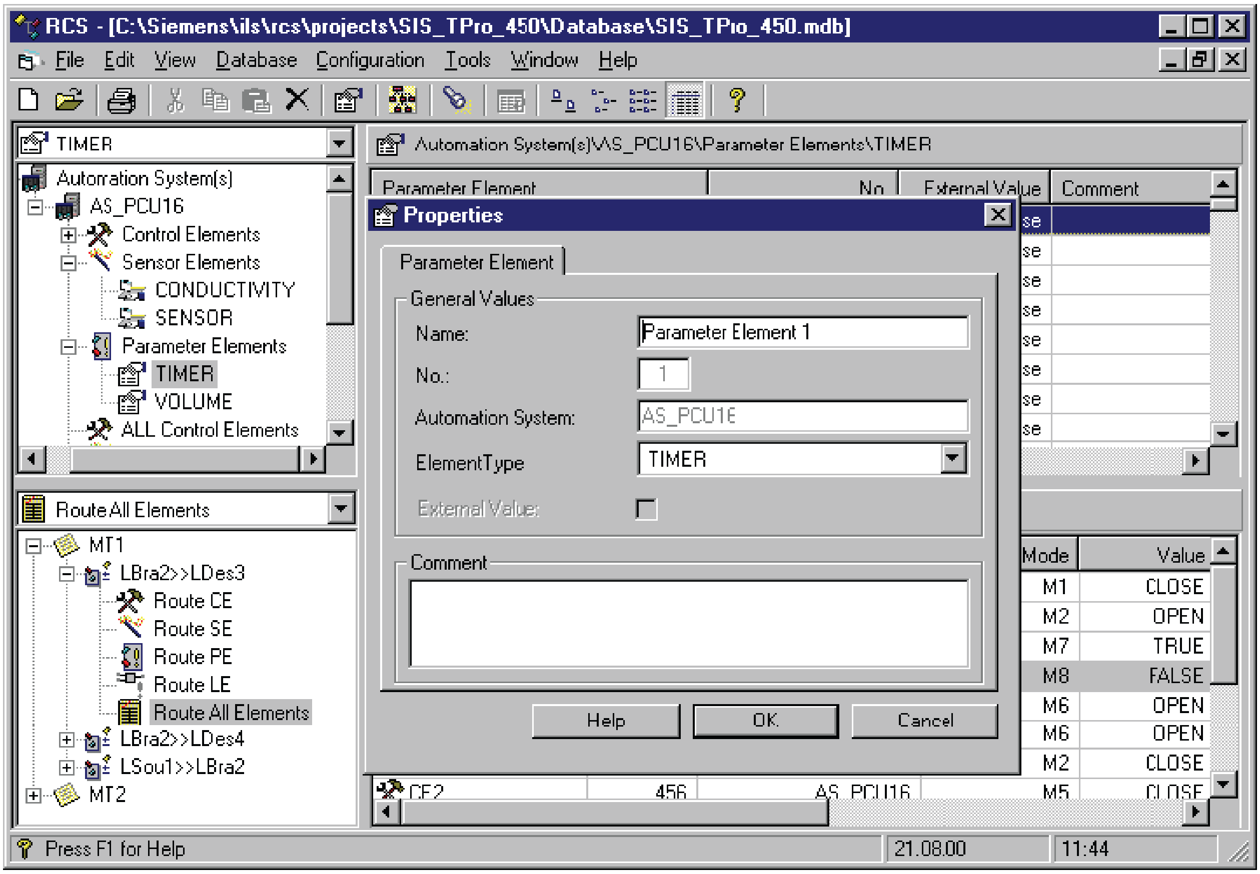


Image: Parameter Element

Parameter Elements (PE) represent additional memory storage locations which are attached to certain Routes and can contain any information. You may file e.g. emptying times, overtravel times or current flow rate measurement. These values are held in the Automation System where 1024 storage locations are configurable for each Automation System.

The value of the storage location is a 32-bit double Integer in S7 notation.

Parameter Elements also have the type attribute in the same way as the Control or Sensor Element. In this way it is possible to assign Parameter Elements to different types.

If the bit "External Value" is set, the corresponding Parameter Route Element uses the interface block and not the configure parameter value. You will find more detailed information in the section Interconnected Parameter Elements (RPE) (Page 1058).

After having closed the property dialog, changes are included in the configuration database. Upon changing the type, the Parameter Element of the list is taken and assigned to the new type.

15.2.2.8 Parameter Element Type (PE Type)

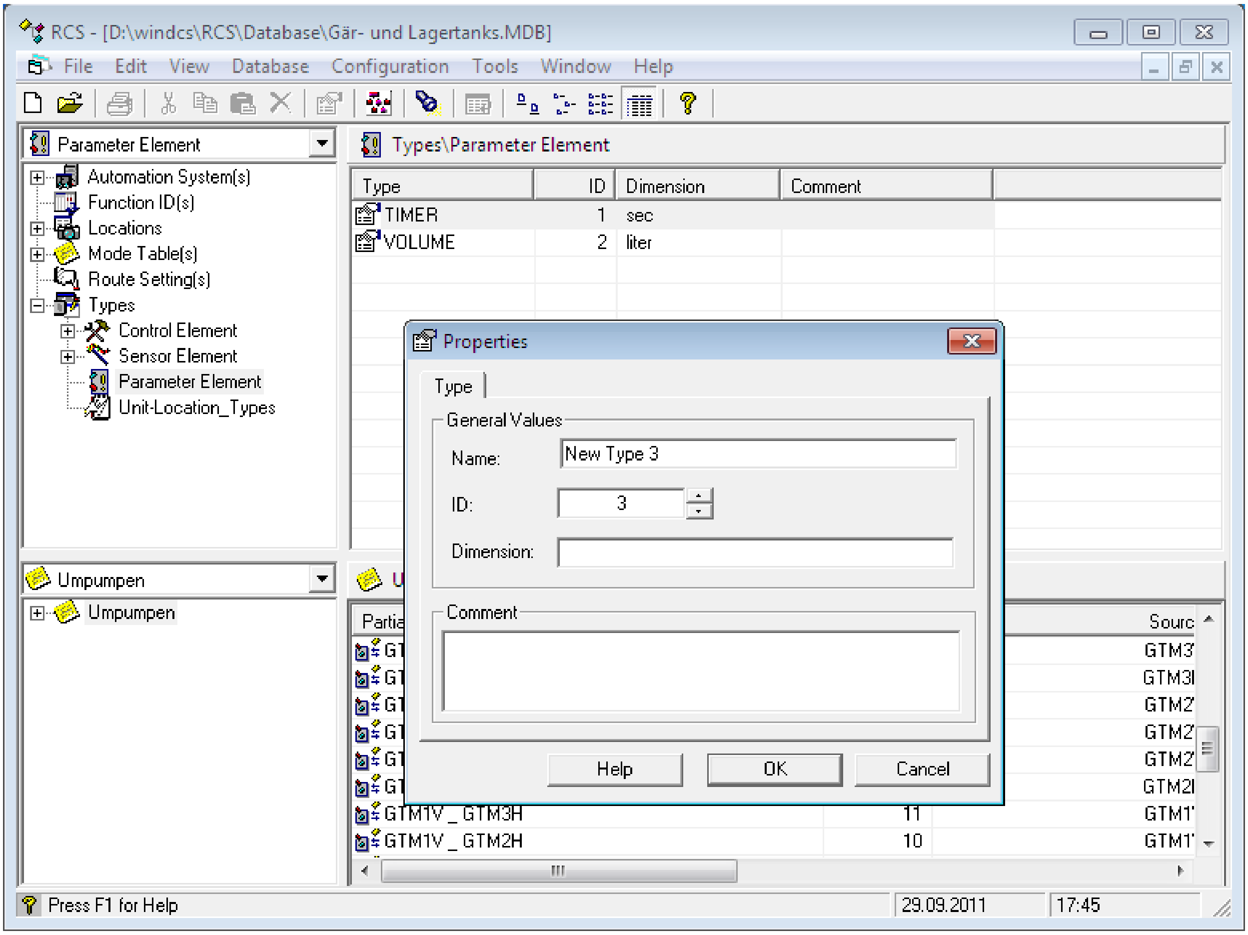


Image: Parameter Element Type

It is also possible to indicate further Parameter Element types. By clicking with the right mouse key as well as administering the call "New type", you can define a new Parameter Element type. The input of the dimension is used by the Online view. The parameter type "Timer" is displayed with the dimension "Seconds" or the type "Liquid level" with the dimension "Liter".

15.2.2.9 Link Element (LE)

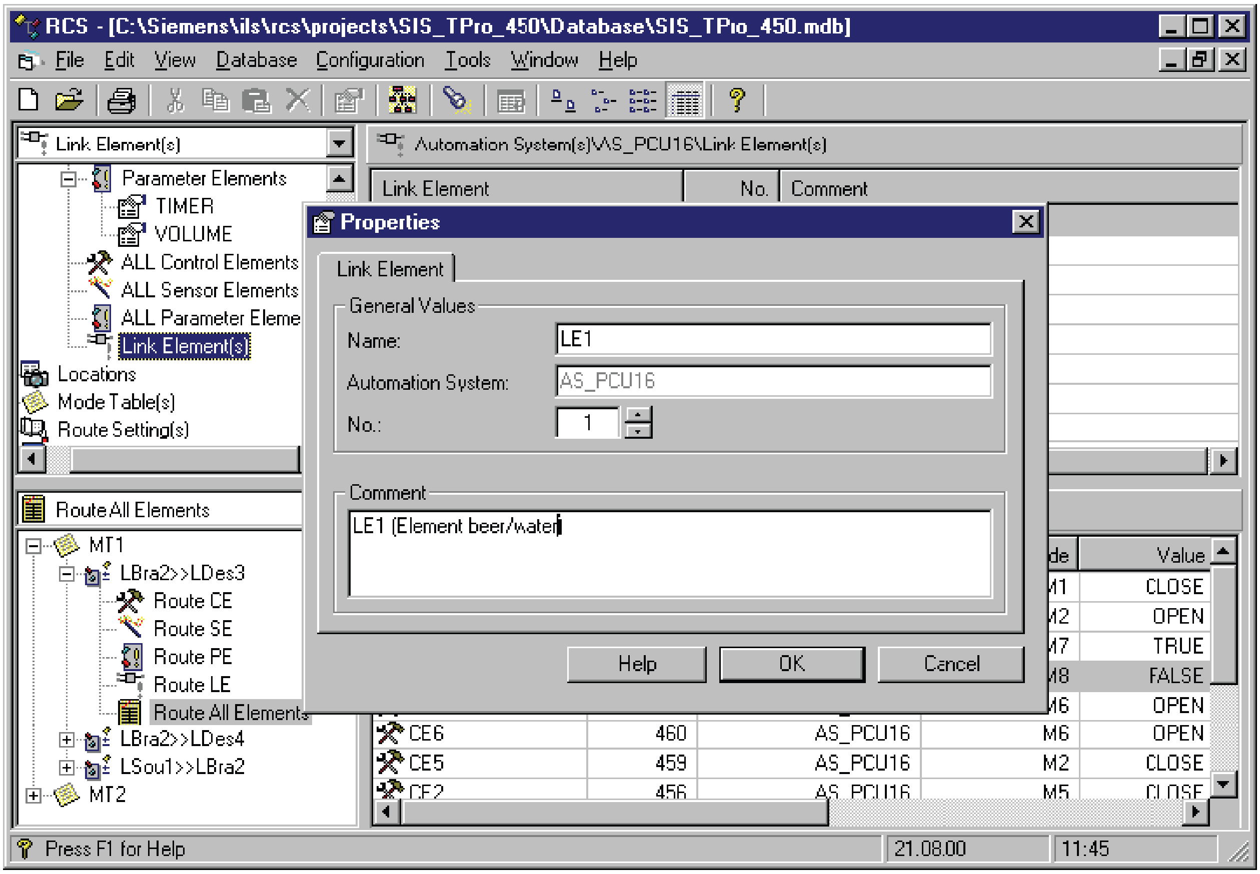


Image: Link Element

A Link Element (connector) corresponds to any definable point within the process cell, similar to the location. However, information is bound to the LE with regard to the actual led material, it serves thus for the storage of the current material allocation.

With the help of the Link Element a compatibility check of successive materials can be executed.

You will find further information on the topic of successive materials is in the section "Material and RCS / Overview (Page 1080)".

15.2.3 Technological hierarchy

15.2.3.1 Process Cells

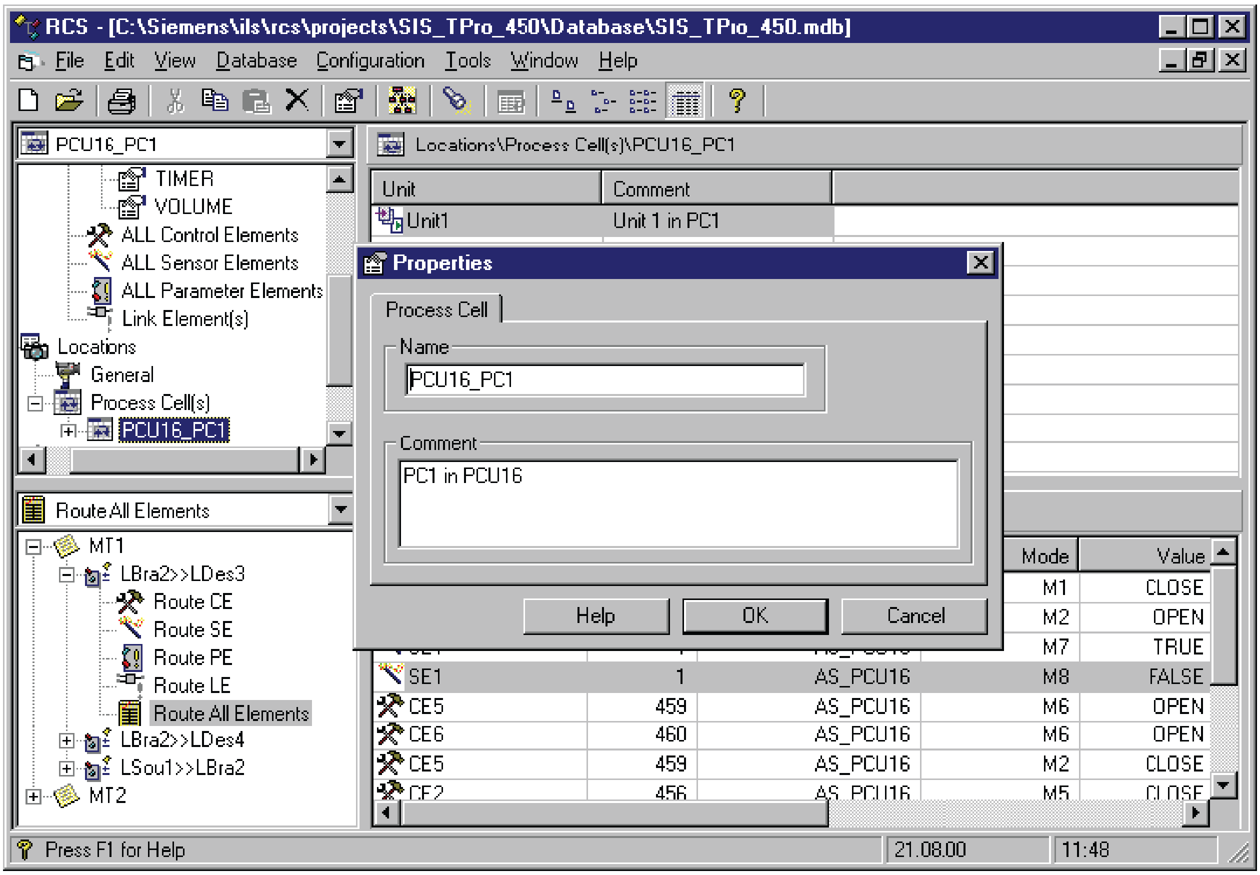


Image: Process Cell Section

'Process cells' are used to define individual areas of a plant. In this way, a plant can be structured in a more detailed manner. For example, the area of the fermenting cellar, the racking plant or stocks could be mentioned here.

15.2.3.2 Unit

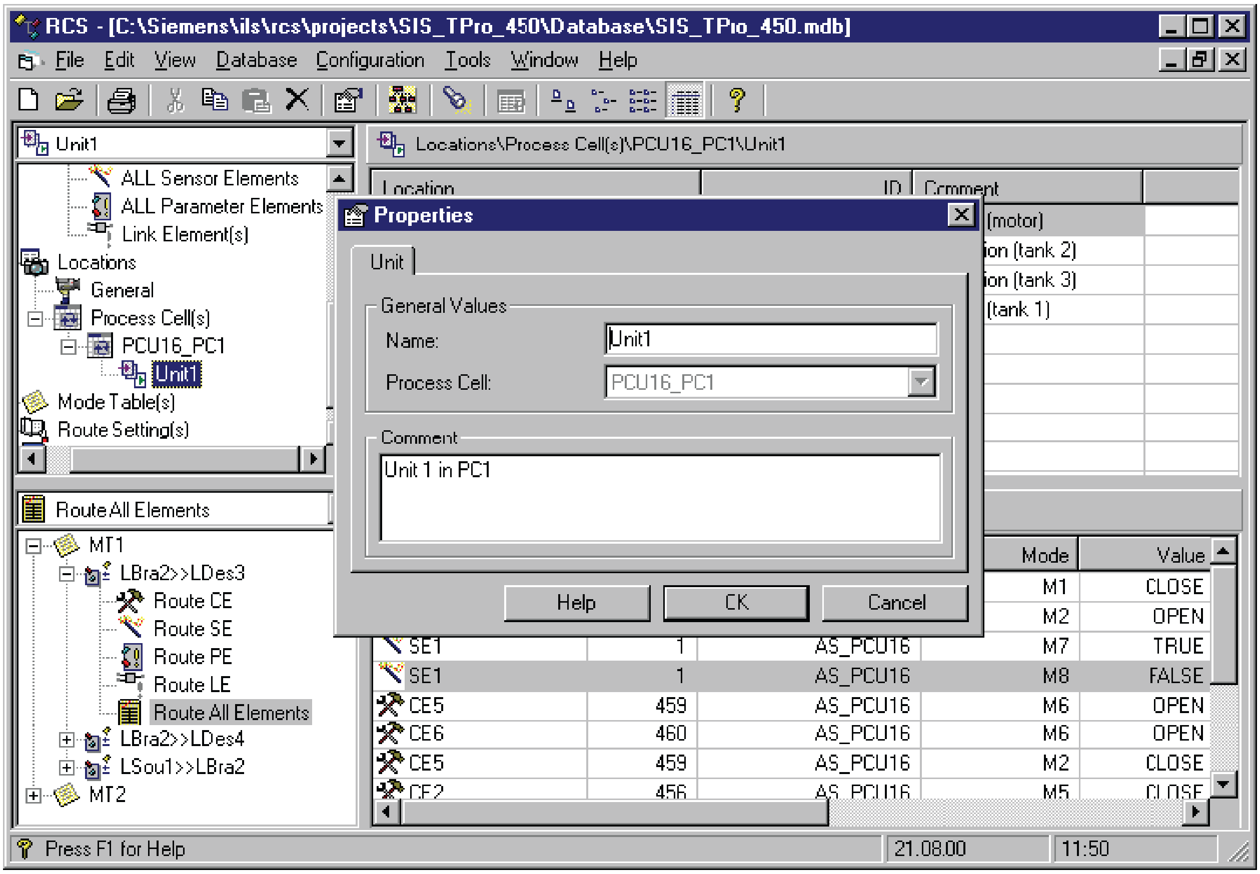


Image: Units

Units correspond to units which you can assemble as a process chain in a plant. Particular examples for the brewery industry include grinding, mash tun, lauter tun.

15.2.3.3 Locations

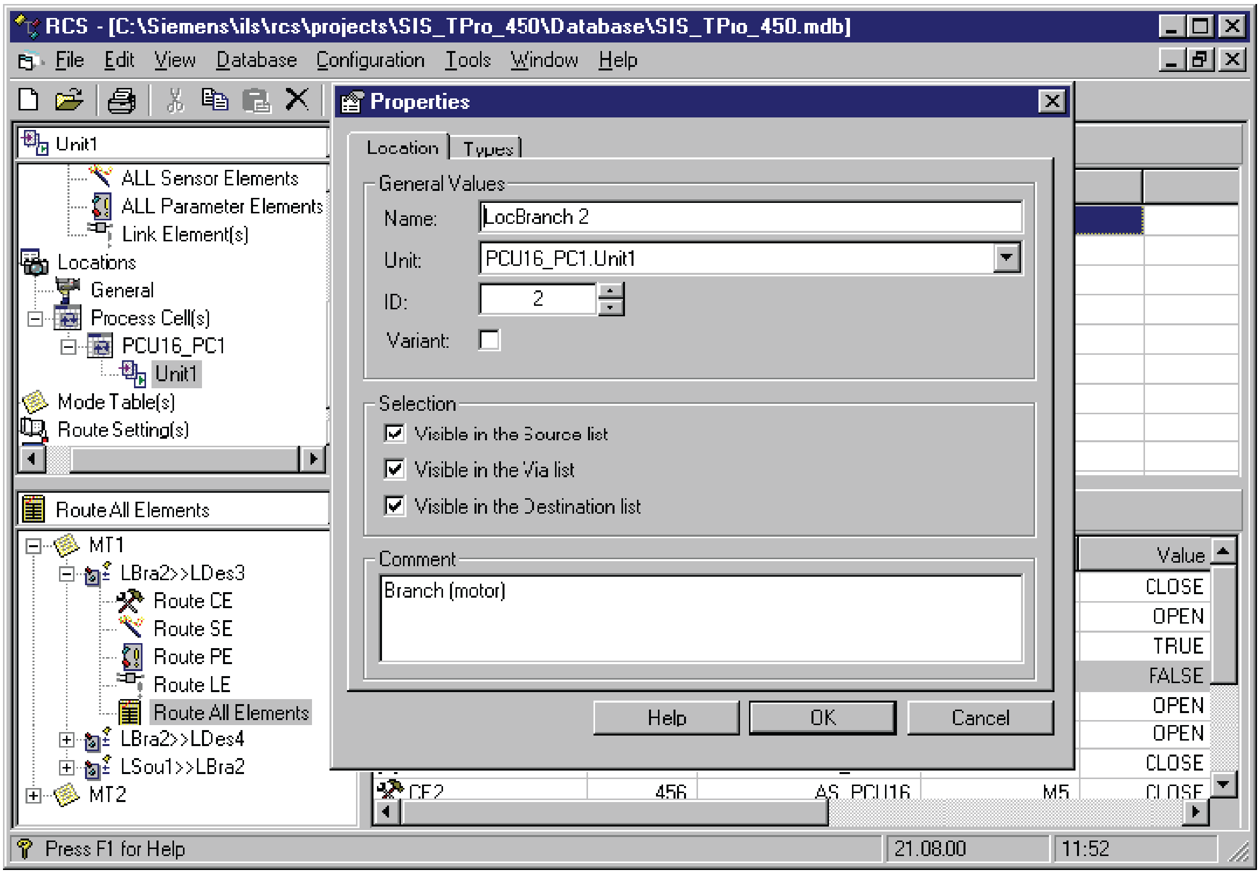


Image: Locations

Here, you specify points or locations (e.g. silo, tank, weigher, tube branching) of a plant which are assigned to a unit. They are referenced as starting or final points by the Routes and Partial Routes.

In order to prevent problems relating to Variants of a Partial Route beforehand, there is an extra attribute for the locations (Variant) describing the location's use.

Locations which are marked as Variant shouldn't and can't be used as source or destination for a Partial Route.

The selection fields are filed as locations for 'Source/Variant/Destination'. By deactivating the attribute variant during runtime the location cannot be selected as 'Source, Variant or Destination'.

In the property box you can select where to list the location. The possible selections are source, via and destination.

Drain valves at tanks would be one example for locations that should not be selectable as destinations.

15.2.3.4 Unit type

A unit, e.g. a mash tun, has several locations such as Inlet, Sprayball and Outlets. A type standardization should be explained by an example. By creating a brewing recipe there is no direct reference to the unit or a location. Therefore the recipe should be generally valid. Only on starting or at the output time of the recipe is it stated to which tank the batch will be produced and only then can you define a node. Each location can receive one or several type assignments which are generally valid because of this reason. In the example below the type "Inlet" and "Outlet" are assigned to the new locations.

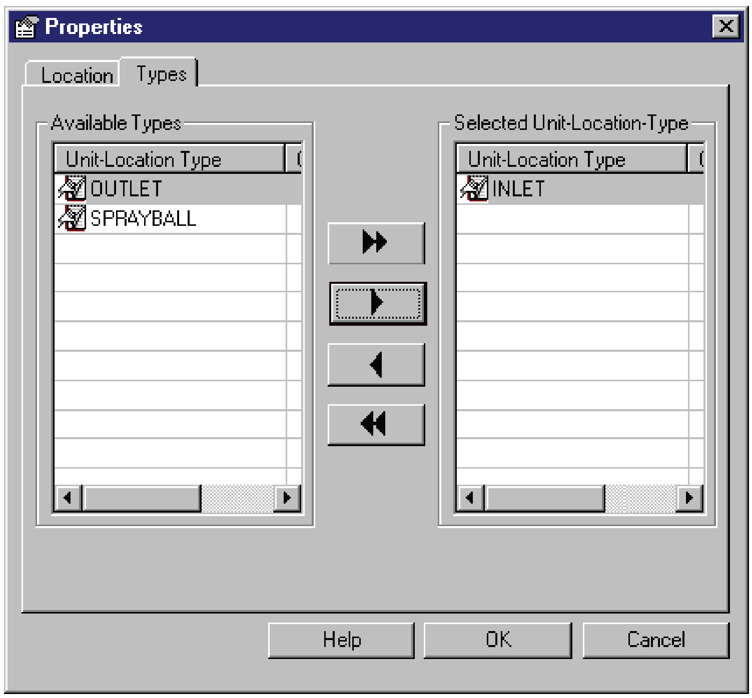


Image: Unit Type

New Unit types can be created and defined under the node "Unit point types".

15.2.3.5 Mode tables

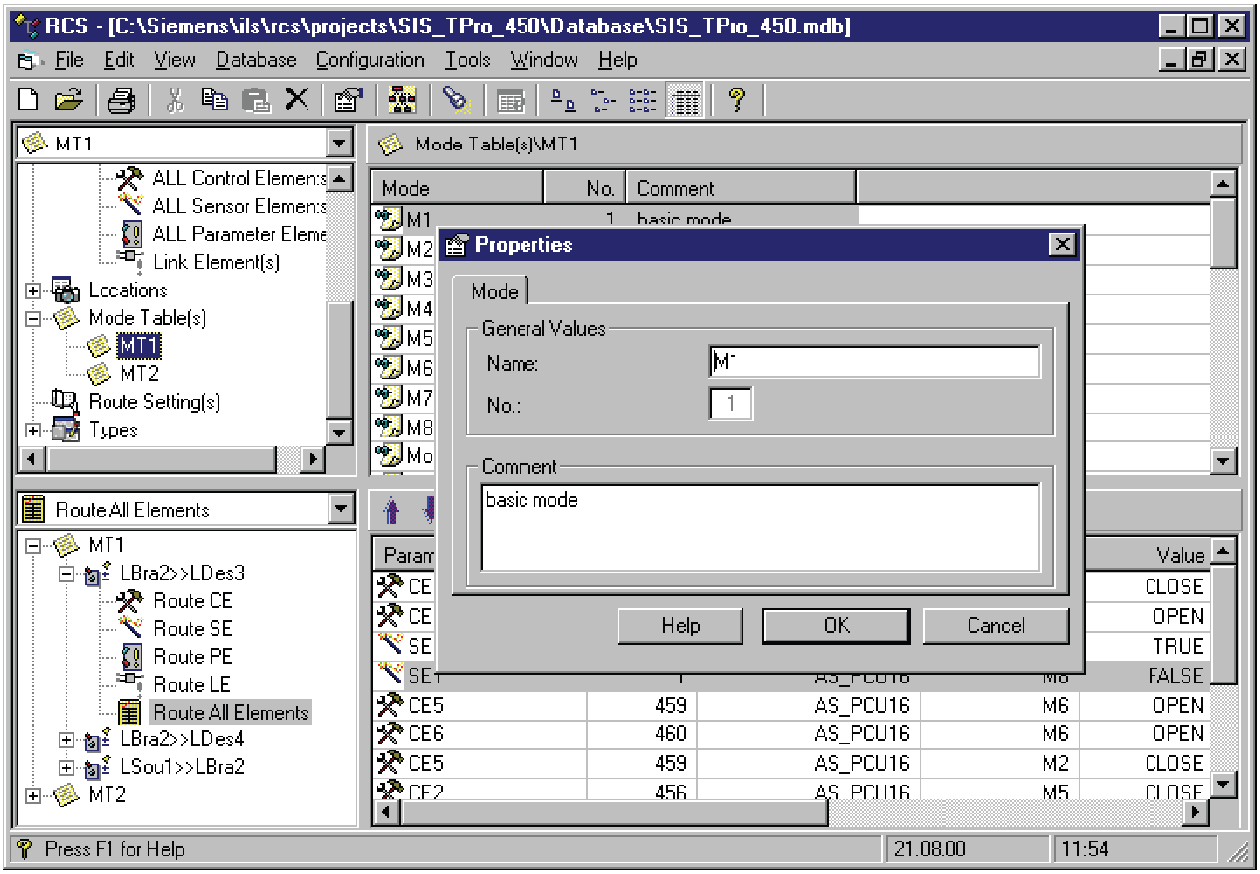


Image: Mode table

A Mode table describes the function steps. The started Route should run in this sequence. Each Mode table has a maximum of 32 function steps which are described when creating a new Mode table as function 1 to function 32. All Partial Routes which are to be configured are assigned to a Mode table.

The corresponding function steps are independent of each other. How the individual function steps are switched on or off is controlled via the corresponding user program in the Automation System and is also detected there. You have to consider that all Partial Routes of a Route refer to the same Mode table.

Which Mode table and which Partial Route are used for assembling the Route in the Online Operation will be detected in the Automation System.

15.2.3.6 Partial Route (PR)

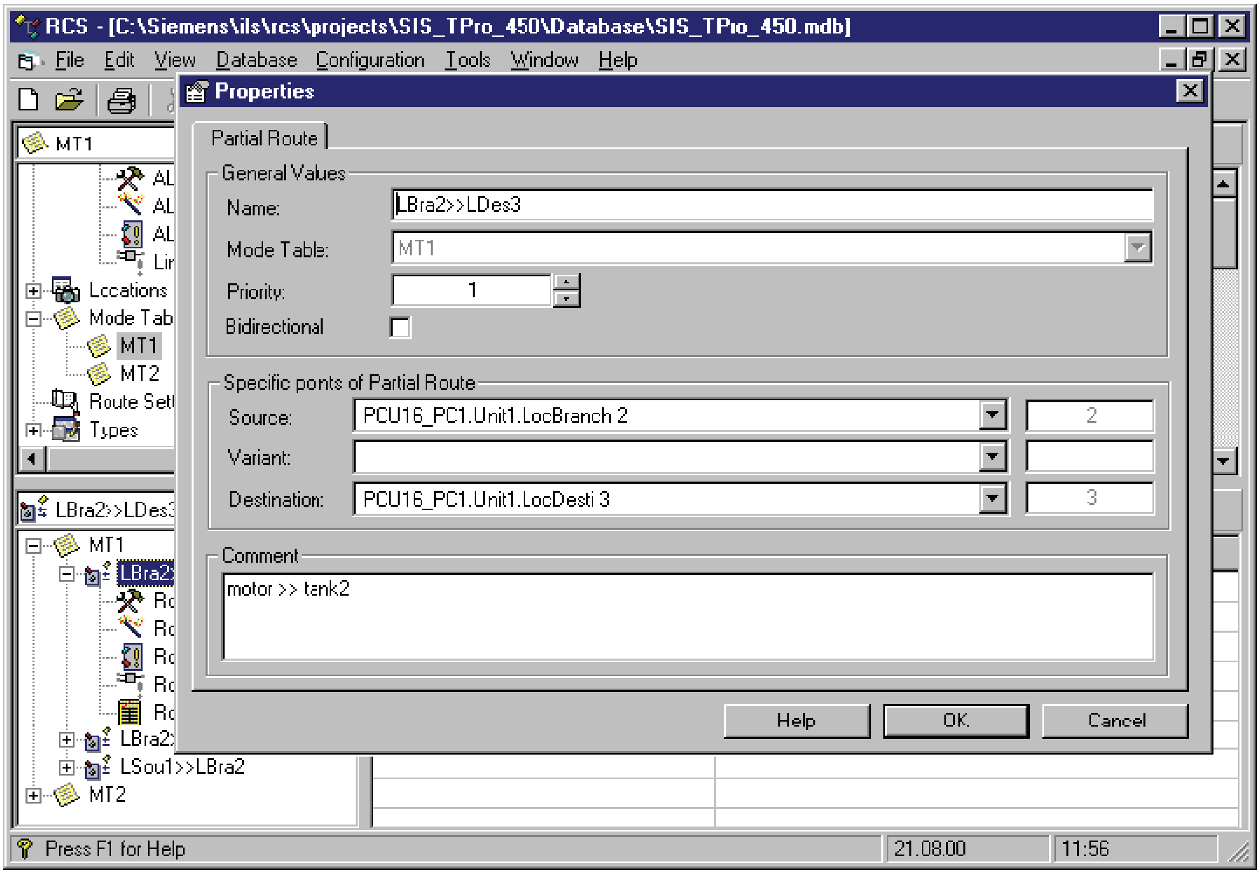


Image: Partial Route

You define a Partial Route with the specification of a starting and a final point from the amount of locations. If several Partial Routes process the same Source and Target specifications, they cannot be differentiated from each other with a search procedure based upon those parameters. For this reason additional information included - the variants. On this basis, variants specify different alternative partial routes.

The "Bidirectional" parameter, the usability in terms of the route direction can be changed. As default, the partial routes are unidirectional.

If a location is selected as Source, Variant or Destination, the identification number will be displayed in the same line.

With the parameter *Prio* (1 = the highest priority) a general criterion is available, which permits a Route section to have different priorities. This attribute is considered by performing a Route search, i.e. if equal routes are available the one with the higher priority is selected.

The length of a Partial Route is influenced in this way because it is bound closely to the number of elements associated with the Route.

Priorities for Partial Routes are allocated e.g. according to:

- Length of the pipelines
- Number of items in this Route
- Preferred elements, e.g. a pump which hasn't yet been in operation for a long time.

You could configure any Partial Routes.

The route search algorithm takes into account:

- Material
- Element allocation
- Priority
- Via From To

15.2.3.7 Pre-defined Routes

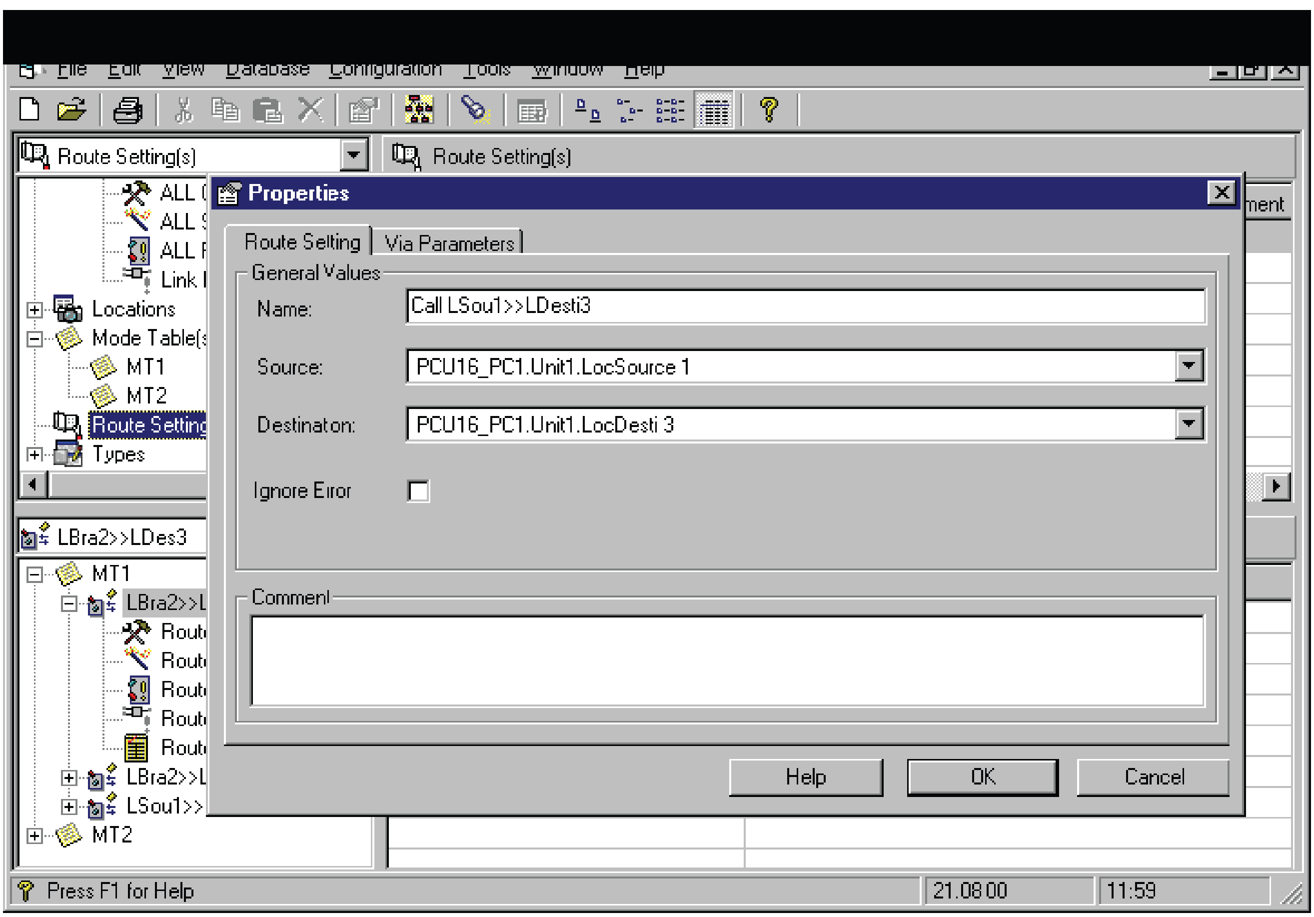


Image: Pre-defined Routes

If some Partial Routes have already been configured, it is possible to pre-define whole Routes of plants under the point "Route Setting". Therefore, it is possible to adjust the Routes in the

Online Operation more quickly. The dialog window for a Route definition expects the following inputs:

- A name which should be identified as Route
- Mandatory indications are the Source and Destination point of a Route.
- In addition to the Source and Destination, 'Via Parameter' can be defined across which the Route should be lead. Up to 10 locations can be indicated.
- If you set the "Ignore Error" Bit, the Route check of RCE and RSE involved will be deactivated in the Online Operation. Deactivation of this test function makes sense only under certain circumstances e.g. the filling and simultaneous emptying of a tank that should be made by the same valve.

15.2.3.8 Route blocks

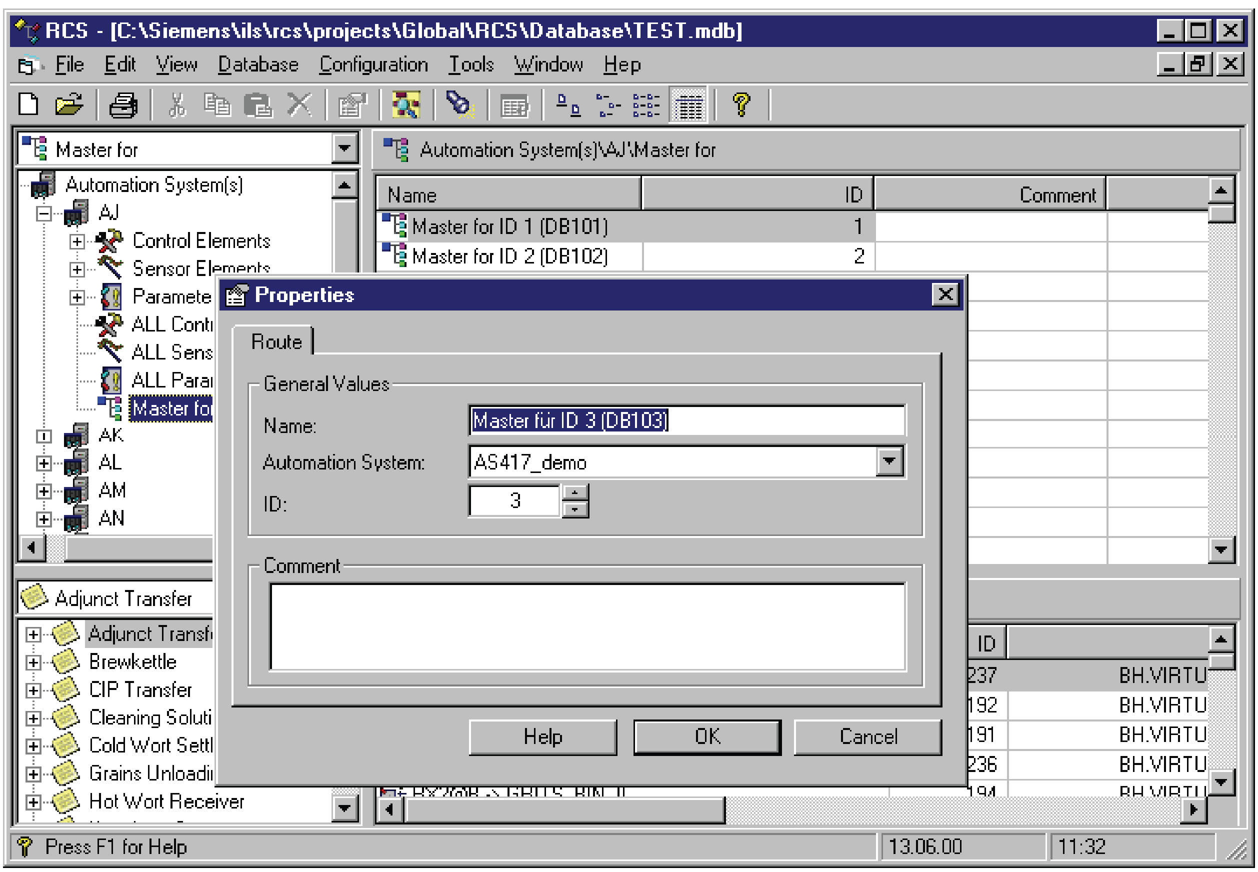


Image: Route blocks

If the "Dynamic Route ID Assignment" is not applied, then it should be stated via the dialog "Route" which Route blocks (Route_CM "FB800") are assigned to the corresponding Automation System.

The Route Control supports a maximum of 300 Routes. This number represents the maximum number of Routes that can be configured. The ID number for a Route can have any number between 1 and 300.

The diagnosis tool (Faceplate) will only display those Route CMs that were assigned to an Automation System during the setup phase.

Per Route block one complete material transport path may be controlled and monitored, i.e. the number of Routes that can run in parallel at any given time depend on the number of configured route blocks.

Route blocks can be assigned to one Automation System via the combo box "Automation System" separately. Once this configuration is performed, the Route can only be started in its assigned Automation System.

15.2.3.9 Interconnected Control Elements (RCE)

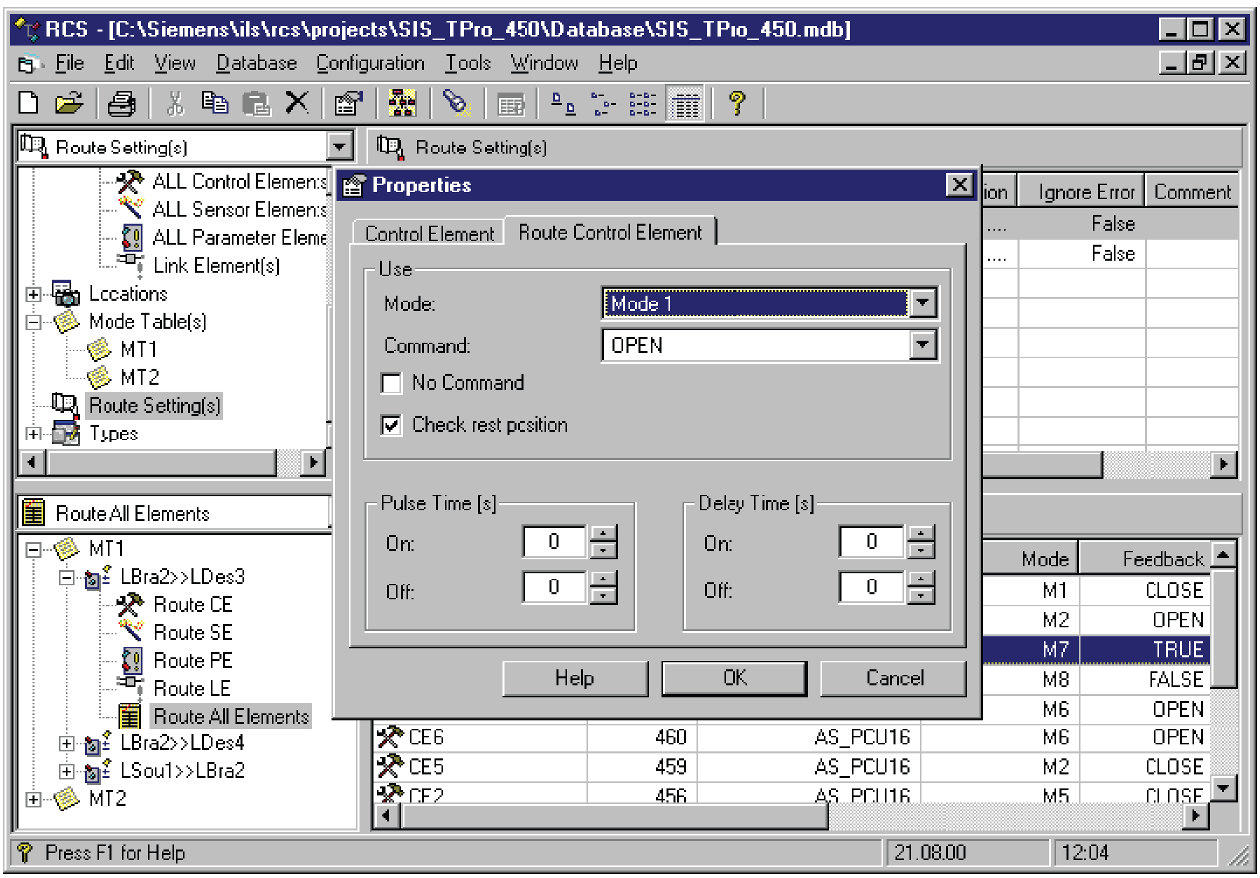


Image: Interconnected Control Element (RCE)

Installing Control Elements in a Partial Route (PR) results in a Route Control Element (RCE). A control element can be used as often as required. This means that a control element can be included any number of times in a partial route (PR). The only limitation is that a given CD cannot occur more than one time in the same mode.

With the installation of the element in a partial route (PR), the control element receives further properties as shown in the figure.

This dialog allows the user to determine the requested command mask from the list selection. Pulsing drives can be implemented as required by using the Pulse Time (s) area. An example of this would be shaking sieves or valve knockers.

Time delays for the switch-on reaction of the drives can be configured using the Delay Time(s) area.

Values from 0 can be indicated to a maximum of 255 seconds.

If the "No Command" bit is set, pulse times and delay times will be deactivated at **runtime** and automatically set to the value 0. The values of the engineering however are retained.
Important: This element is not occupied by the Route Control, only the feedback is scanned passively. That means it may be occupied and controlled by other routes.

The rest position monitoring bit decides whether or not the route element will be monitored when a route is terminated. The default value for the rest position monitoring is activated.

Note

Only the position of the rest monitoring bit is actually relevant for the Standard types of the Control Element as only the Automation System in the user program knows the position of the rest relating to the element at the moment. That means that the position of the rest monitoring bit in the Interface block FB827 won't be supported for new Control Element types.

15.2.3.10 Interconnected Sensor Elements (RSE)

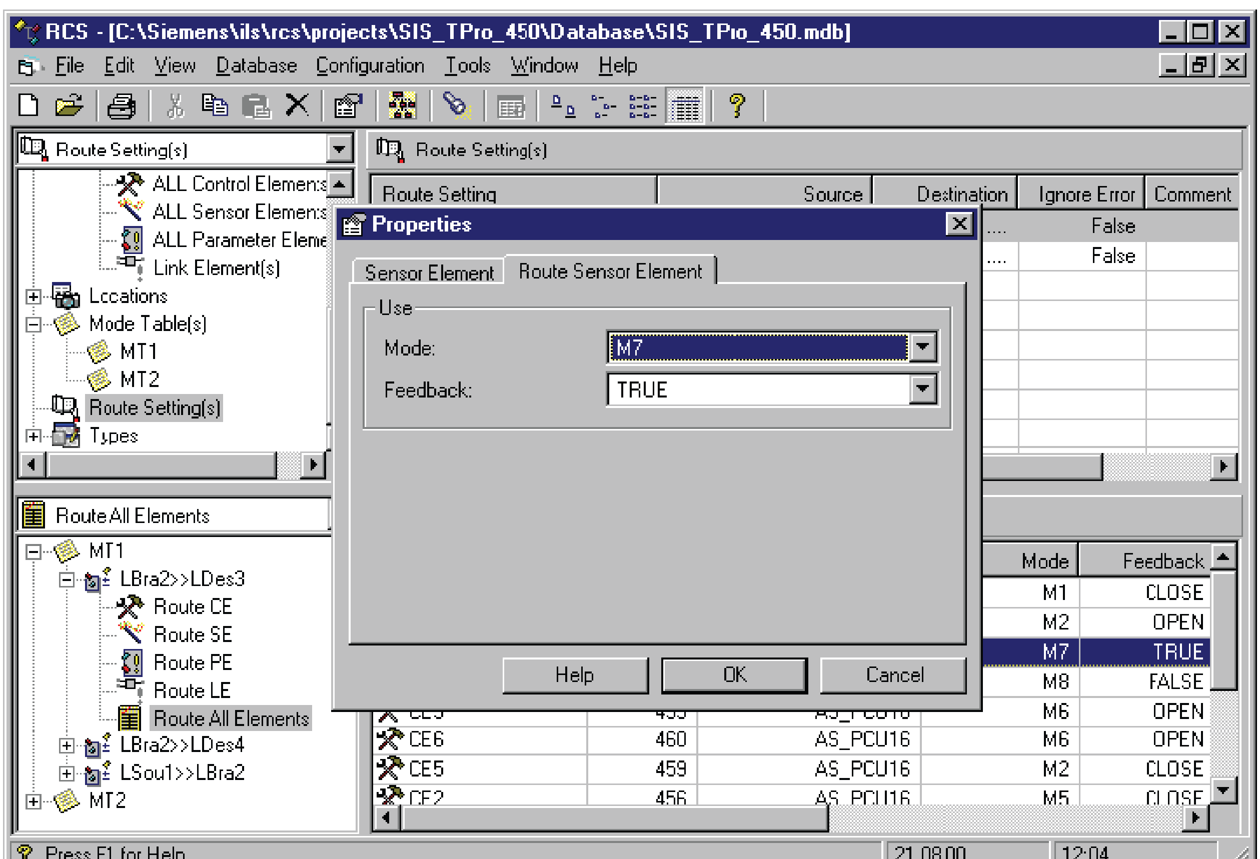


Image: Interconnected Sensor Element (RSE)

The Route Sensor Element occurs by using a Sensor Element similar to the Route Control Element.

As queries are also bound to special functions, there is also an assignment to a mode with the corresponding feedback.

The corresponding feedback is selectable via the combo box.

15.2.3.11 Interconnected Parameter Elements (RPE)

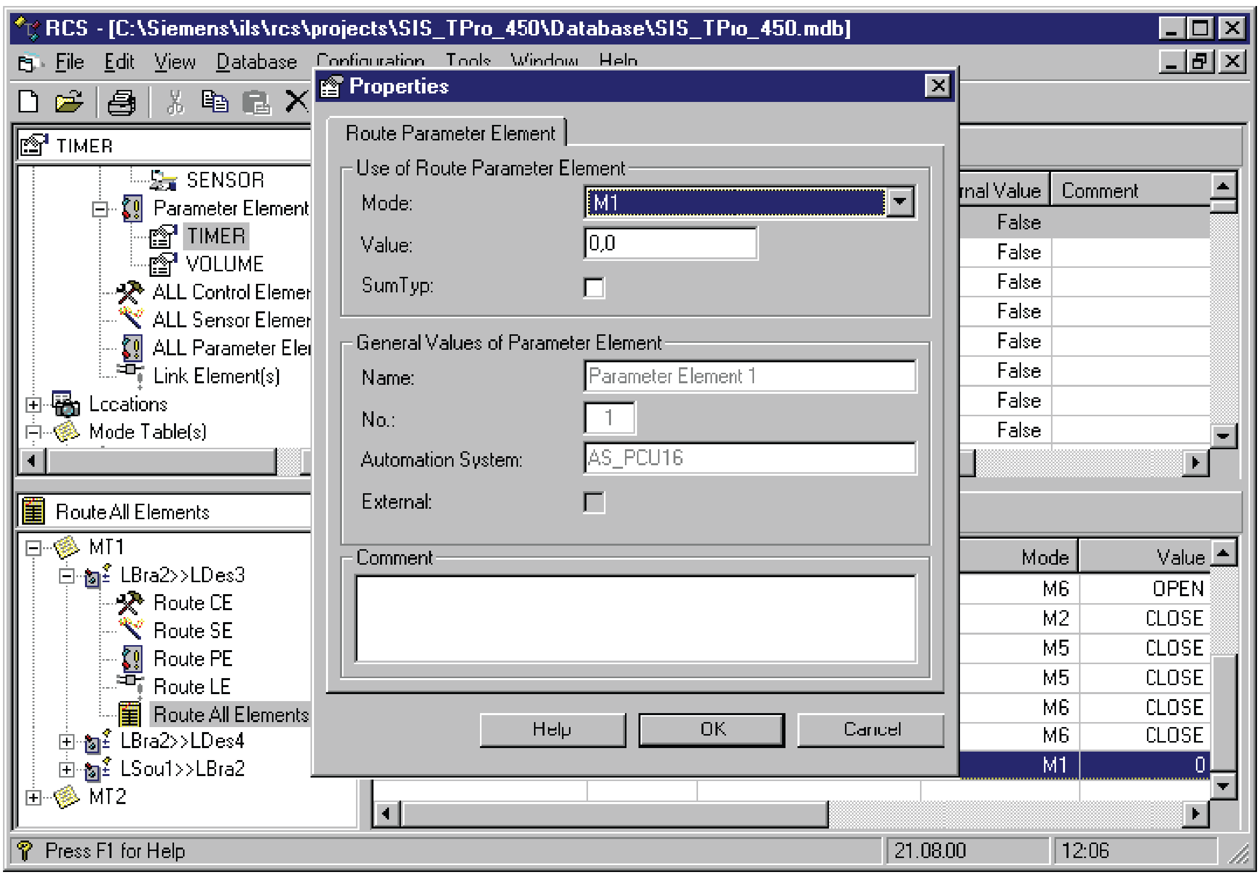


Image: Interconnected Parameter Element (RPE)

A new instance of the type Parameter Element arises with each installation of a Parameter Element in a Partial Route.

If you set the external bit at the Parameter Element, you won't be able to assign any value to the interconnected element. However, you indicate the index of the requested setpoint which should be supplied from the Route block to the Parameter Element. As the Route data block can process up to a maximum of 24 setpoints, the index can only be indicated from 1 to 24.

Otherwise you can allocate an integer or flextime value which should be processed as setpoint in the user program. Parameter Elements can be built in summation, so that the values of all Parameter Elements participating in the Route of a type can be added (time generator). This can be used for dynamic calculation (depending on the combination of the used Partial Route) of flow rates or break times. Here you have to set the bit 'sum type'.

15.2.3.12 Interconnected Link Elements (RLE)

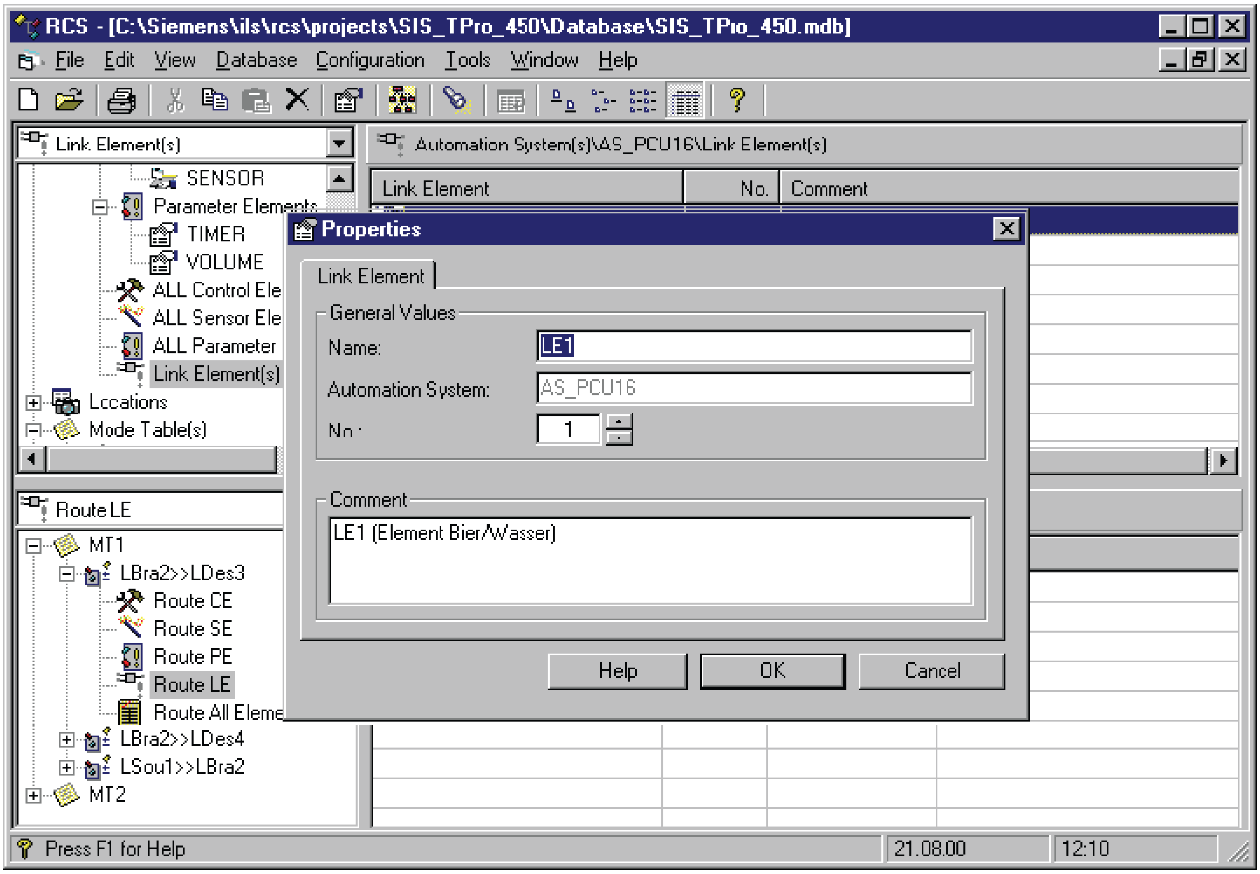


Image: Interconnected Link Element (RLE)

You won't add any further properties with the installation of a Link Element in a Partial Route. Interconnected Link elements are used as information carriers for the transported materials. This information is used for checking the material compatibility which defines the material sequence. You will find further information about material compatibility in the chapter "Material configuration".

15.2.4 Extended configuration

15.2.4.1 Function identifiers

Function identifiers describe the type of material transport and are of a more static type. Different routes also include different function identifiers. Exception: Crossing routes. Exception Crossing routes

The interface blocks of the Route control (e.g. of the FB800) include the input parameter "FUNC_ID". In the Offline tool a readable name can be configured for each ID (Number) (ID in Offline tool corresponds to FUNC_ID in the AS) in order to ease the diagnosis (Displays in the diagnosis functions of the Route Control for RCS Online and RCS Faceplate).

Note

The configuration of texts for function identifiers isn't necessary. It only influences the visualization.

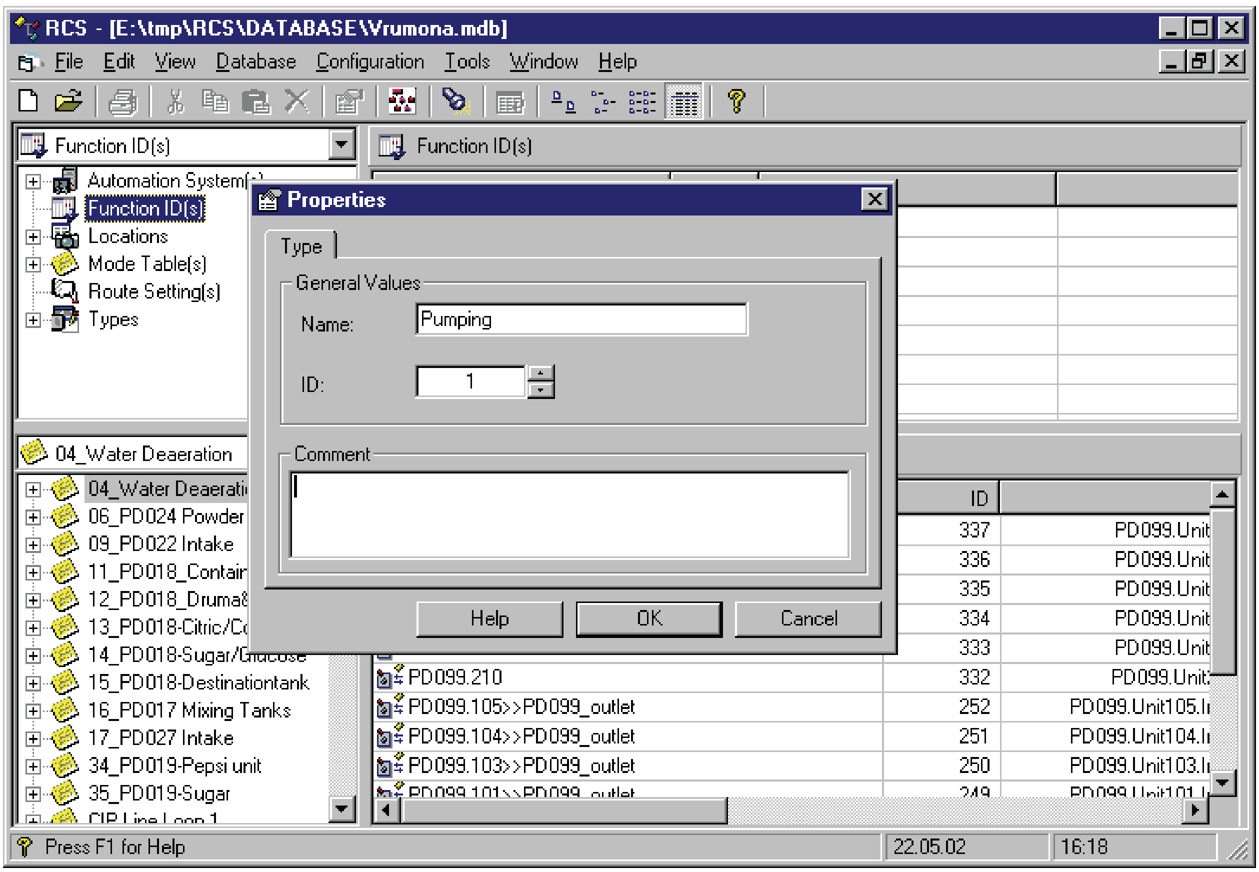


Image: Dialog for the function identification

15.2.5 Tools and adjustments

15.2.5.1 Cross Reference Lists

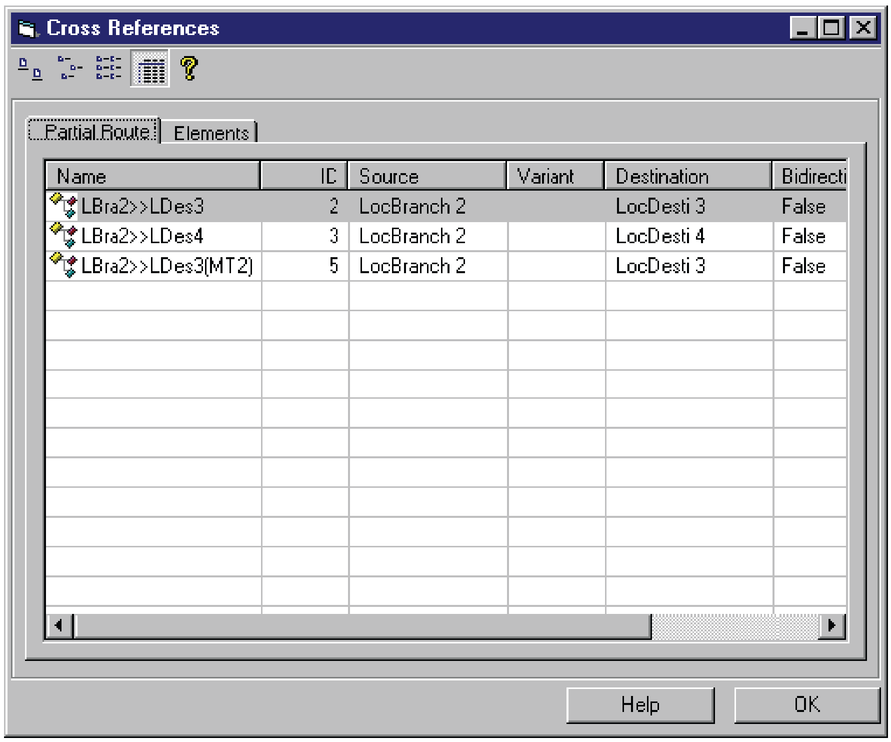
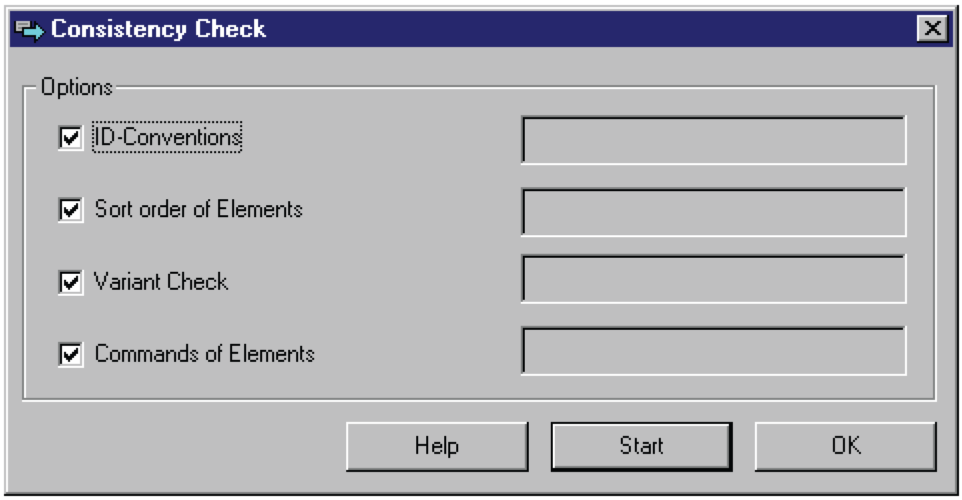


Image: Cross-references (use elements in partial routes)

The context menu (right mouse key) enables the user to view the cross references of any element. This menu lists all Partial Routes corresponding to an individual element. By double-clicking a Partial Route, you receive a list of all elements from this Partial Route on the second register card, where all elements of the selected Partial Routes are listed.

15.2.5.2 Consistency check

Inputs in the database according to logical sequence are used for checking the database with regard to special assignments. At the moment there are four different areas in the test.



- Check ID allocation:
- The "IDs" of the elements CE, SE and PE are allocated according to the AS number and element number (AS number * 10000 + element number), checked and adapted. The same is valid for the column "Name_ID" that is formed from the AS number and name of the element so that the elements are called differently within an Automation System. If the names or IDs can't be adapted the message will be displayed that the database is inconsistent and it should be changed manually.
- Check the sorting sequence:
- The mode-specific sorting number (SortNo) of the CEs is transferred in a gapless area. It is numbered according to the mounting order of the CEs per function step. The same is valid for SEs and PEs. The sorting number is checked via the total configured Partial Route and adapted in case of errors. The elements are numbered according to the table order of all elements (CE, SE, PE) without considering the function steps (SortNoAll)

- Alternative test:
- The inputs in the database according to the logical sequence filters all locations that do not correspond with the assignments and displays these in the list window. Via the cross reference list in the list you receive additional information as to in which Partial Routes the error location has already been configured. The cross reference list is called by a double-click on the corresponding location.

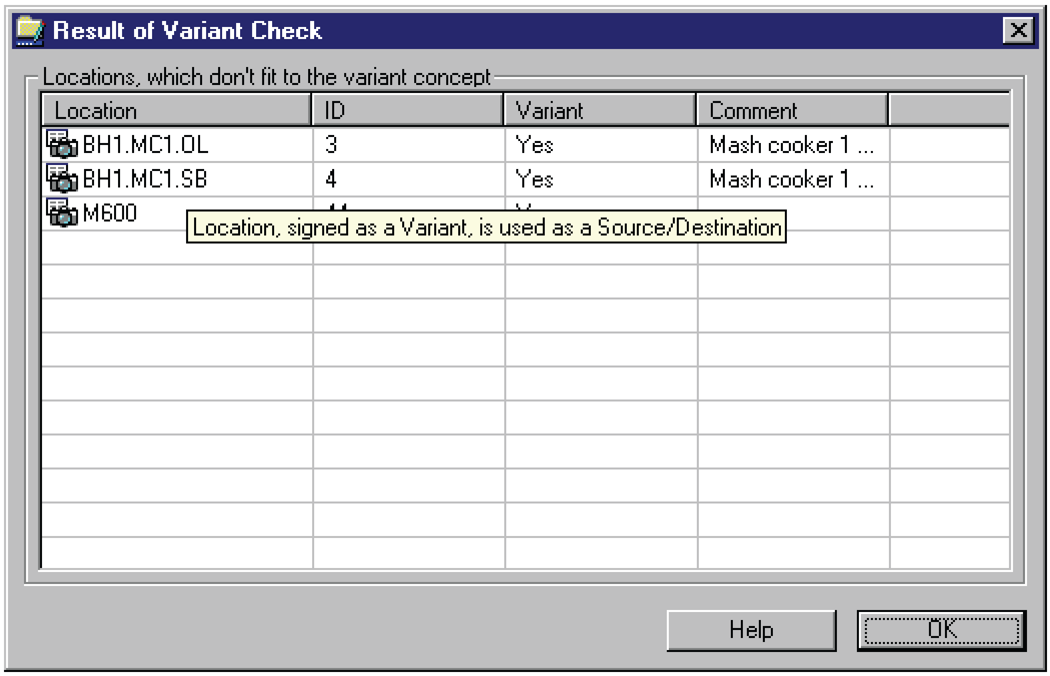


Image: Result of a variant check

- Control of the elements:
- The control of the elements will be checked with regard to the type assignment. If there are controls that don't match the configured type, a dialog with all non-reliable interconnections will be displayed. The fault configurations can be adapted.

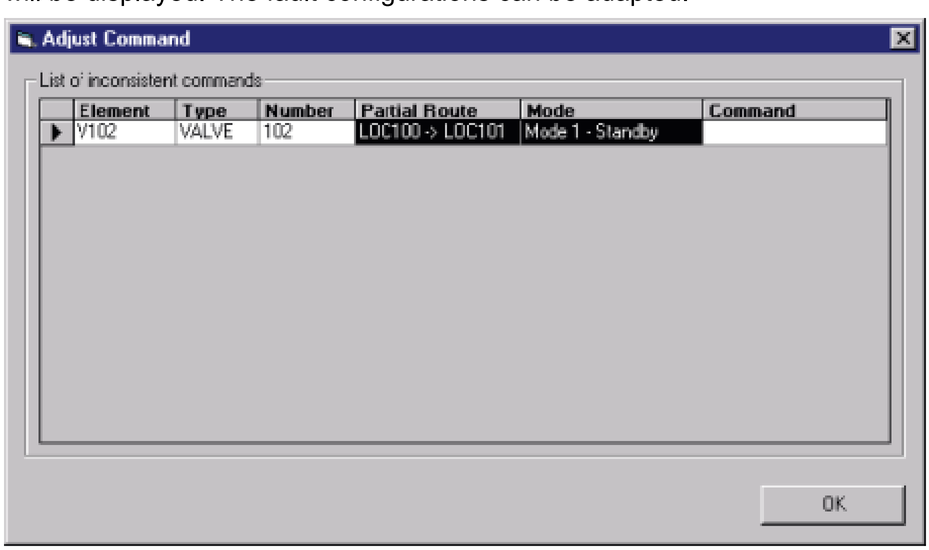


Image: Adjust command

15.2.5.3 Loader

Basics

Under accessories -> Loader the following dialog should be found:

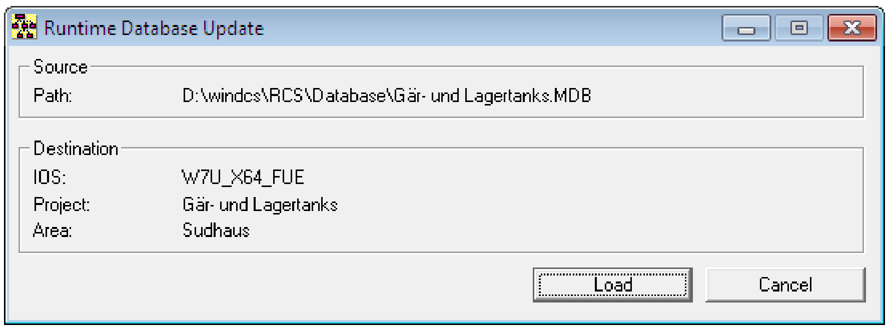


Image: Loader

If the configuration phase has been finished the database should be filed on the RCS Server. The loader function exists for this reason. It calls the RCS Server that copies the configured database onto the IOS and renames the names of the database as "RCS.mdb”. At the same time the RCS Server is in hold position. After the copy procedure the RCS Server will be run up with the new database. The Standby Server is aligned automatically.

The following information also appears in the dialog window.

Source indications:

- The route describes the current database route and includes the database that should be loaded on the IOS.

Destination indication:

- The IOS name is under 'IOS'.
- 'Project' describes the project name of the database that has currently been filed on the IOS Server.
- 'Area' is the plant area for which the configuration database was created.

Generating the text files 'rcs_xxx.txt'

The following text files "rcs_xxx.txt" with the instance names of the configured RCS elements are created by the RCS server when loading the database.

These are stored in the language-dependent "Texte.x" directory depending on the engineering language selected. A distinction is made between global RCS elements across all PCUs and PCU-specific RCS elements. With the PCU-specific elements, only those instances that are assigned into partial routes are taken into account. The remainder are generated as empty rows.

The following table provides an overview of the text files generated in this way:

Text file in '...<proj-path>\texte.x'	Text file in '...<proj-path>\pcu.nnn\texte.x'	Description
rcs_loc.txt		Locations of a plant that are assigned to a unit.
rcs_mat.txt		Material name
rcs_pr.txt		configured partial routes of the plant
rcs_rt.txt		Instance names of the max. 300 routes
rcs_svr.txt		Server name
	rcs_ce.txt	Control elements of the PCU (assigned to partial routes)
	rcs_le.txt	Connection elements of the PCU (assigned to partial routes)
	rcs_pe.txt	Parameter elements of the PCU (assigned to partial routes)
	rcs_se.txt	Sensor elements of the PCU (assigned to partial routes)

Note

Use in process images

In addition to system use in messages, these text files can also be referenced as information for the "Text variable" visualization object in process images.

15.2.5.4 Reports

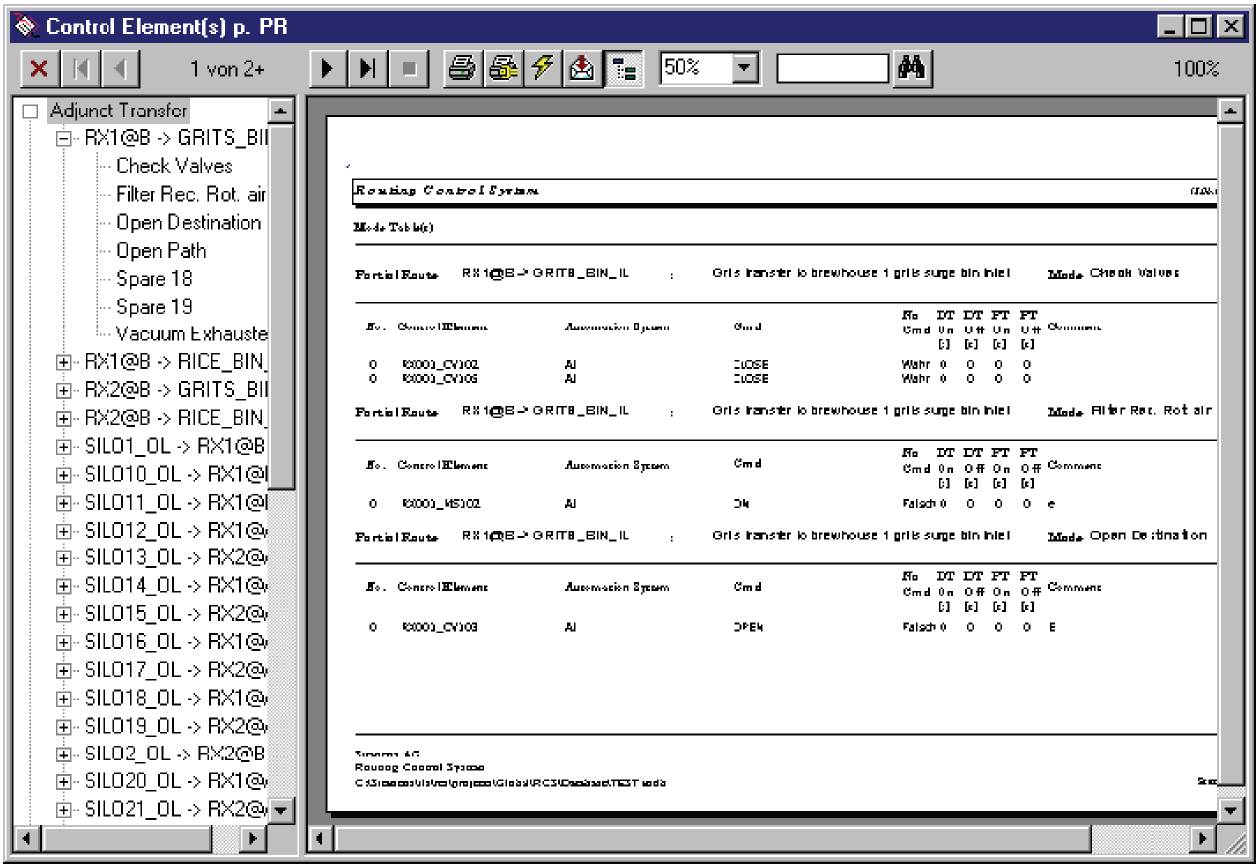


Image: Reports

Reports which provide important information can be printed out from the configuration tool. If printing is performed with the option "Preview", the user-specific changes can be made in the print format. To do this, you select the corresponding area by clicking on it and either start the preview or the printout with the appropriate menu command.

There are two possibilities to print a report:

1. Menu bar 'File'/'Report'
2. Shortcut menu (right mouse button) "Report"

15.2.5.5 Visibility of columns

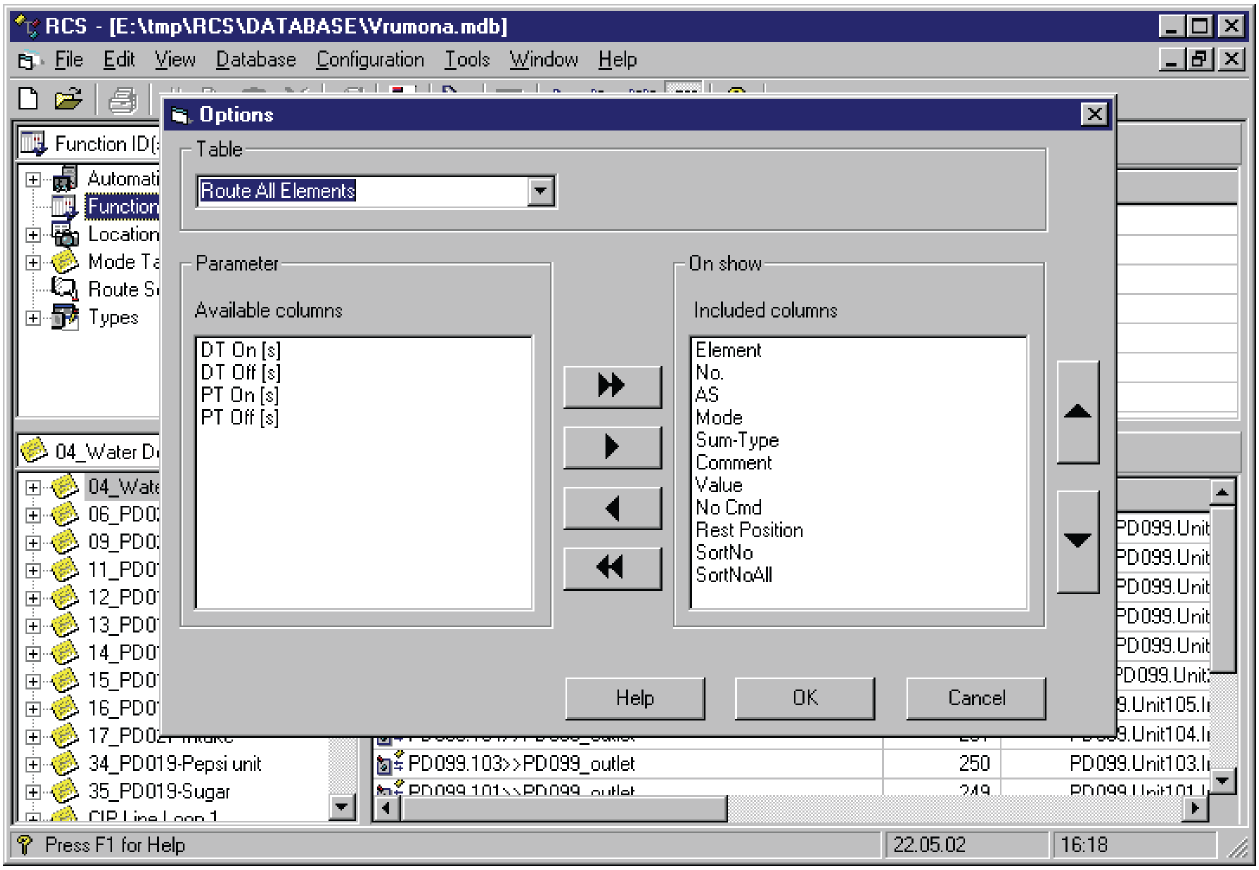


Image: Options

Via the menu 'View' 'Options' the columns can be hidden or unhidden from the displayed table. The requested table will be selected and will displace the corresponding column names. The column names that are not displayed in the list are shown on the left side. The column names that are displayed in the list are shown on the right side. By selecting the column names and pressing the corresponding button the name will be displaced from one side to the other. Besides the right list there are two buttons with which the order of the column display can be influenced.

15.2.5.6 System configuration

With the menu command "Configuration / Settings", you specify the project name and the project author.

The project name is used to verify the online database and appears when you load a new database or when you open the RCS server.

15.3 Interfaces

15.3.1 CSV files

15.3.1.1 CSV export function

Via the Menu 'Accessories' 'Transfer' 'CSV Files' you start the following dialog "Spreadsheet".

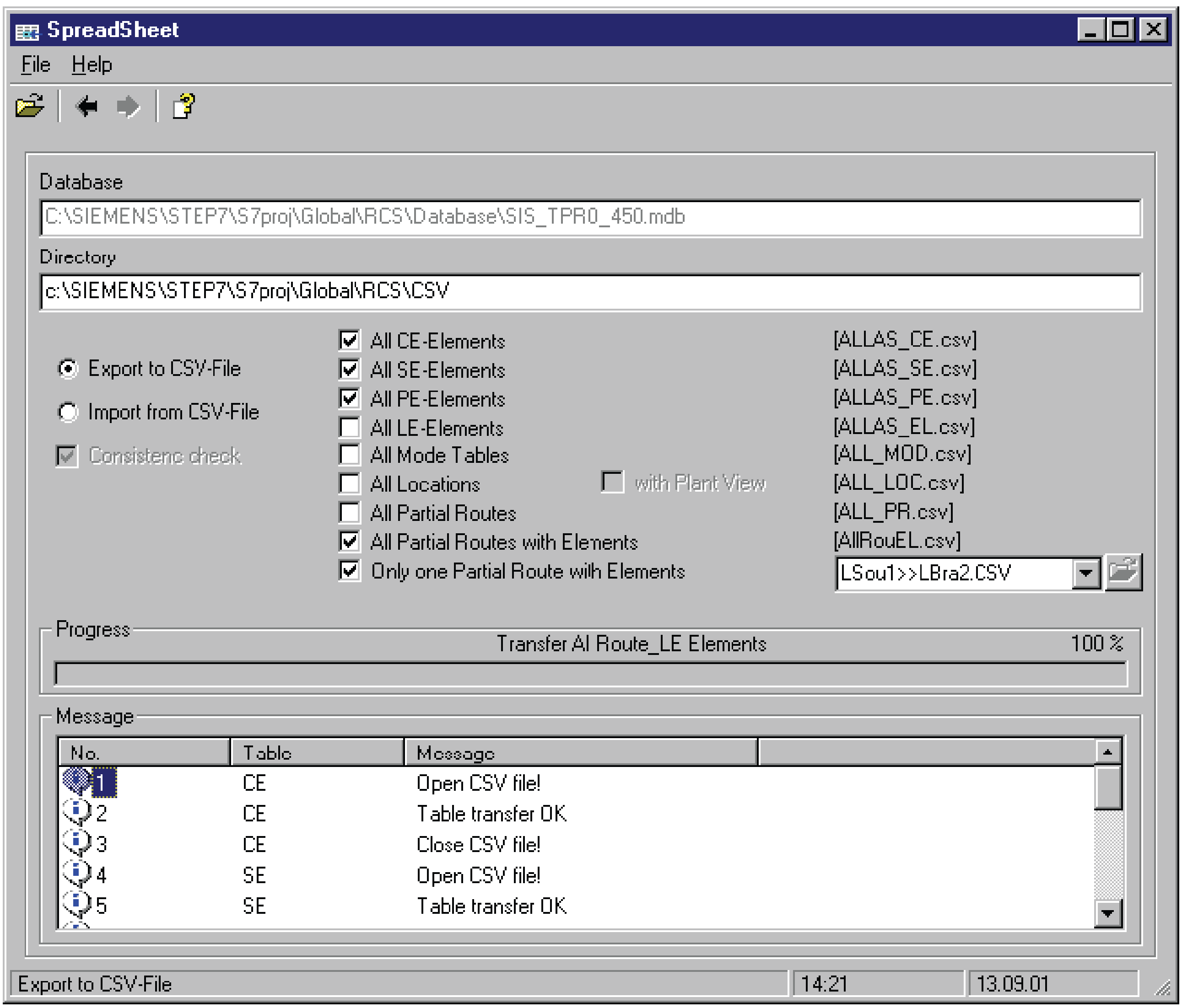


Image: Export function

By opening the dialog, the current RCS Project database is used. You can see the path and the file name as well in the first line "database".

In the second text line you can enter any directory in which you want to record CSV files. This directory input is entered in the registry when starting the export routine, so that the selected directory is preset. At the next invocation the Route can be changed any time.

Then you select functions which you want to execute. Here, you distinguish between export and import. By clicking the selection fields, you select database tables, which you transfer into a CSV file. Only by selecting the selection field will either the export or the import key be started.

The following names are assigned to the tables:

Table 15-3 Fixed tables

Menu item	Database table	CSV file
All CEs	CE	ALLAS_CE.csv
All SEs	SE	ALLAS_SE.csv
All PEs	PE	ALLAS_PE.csv
All LE elements	PE	ALLAS_LE.csv
All mode tables	ModeTable	ALL_MOD.csv
All locations	Location	ALL_LOC.csv
All partial routes	PartialRoute	ALL_PR.csv
All Partial Routes with Elements	PartialRoute RCE /RSE /RPE	ALLRouEL.csv

In contrast to this, CSV file names of the individual Partial Routes are selectable in a user-defined way:

Table 15-4 User-defined tables

Menu item	Database table	CSV file
Partial Routes with elements	PartialRoute RCE /RSE /RPE	Selectable in a user-defined way

These files are created by starting the export function in the selected directory.

It is possible to select CSV file names for each Partial Route in a user-defined way. All Partial Routes which have already been configured appear in the combo box with their name. In this way, you select the desired Partial Route which should be exported or imported relatively quickly. Here, you have to consider that the name depends on the standard for the allocation of file names. Furthermore, this means that there shouldn't be the following special properties.

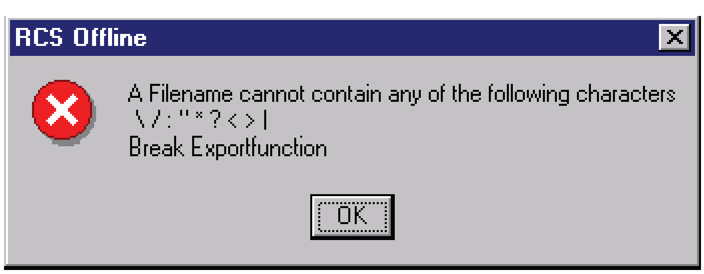


Image: File convention

However, you can adapt the file name of the exported Partial Routes which should be adapted and exported.

Note

Note that the export function overwrites existing CSV files. Any information and modifications made by EXCEL to existing CSV files may be lost.

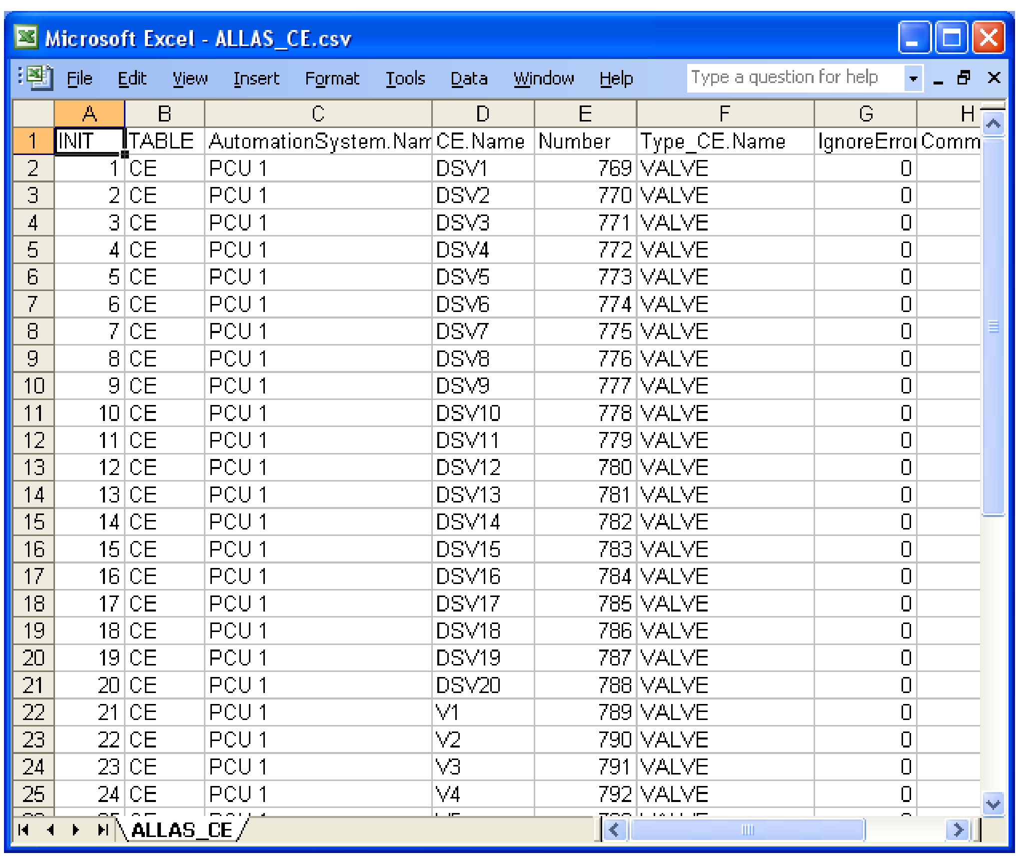
The progress bar shows the current transfer status.

Any further information about the transfer into the CSV files is provided in the message box. This bar is displayed whether the transfer was successful or not and also if the table has been selected for transfer.

The topic "Warnings and Fault Messages (Page 1075)" describes the error messages.

If the transfer procedure is completed, you can open and edit the CSV files using EXCEL.

15.3.1.2 Processing of CSV files



	A	B	C	D	E	F	G	H
1	INIT	TABLE	AutomationSystem.Nam	CE.Name	Number	Type_CE.Name	IgnoreErroi	Comm
2	1	CE	PCU 1	DSV1	769	VALVE	0	
3	2	CE	PCU 1	DSV2	770	VALVE	0	
4	3	CE	PCU 1	DSV3	771	VALVE	0	
5	4	CE	PCU 1	DSV4	772	VALVE	0	
6	5	CE	PCU 1	DSV5	773	VALVE	0	
7	6	CE	PCU 1	DSV6	774	VALVE	0	
8	7	CE	PCU 1	DSV7	775	VALVE	0	
9	8	CE	PCU 1	DSV8	776	VALVE	0	
10	9	CE	PCU 1	DSV9	777	VALVE	0	
11	10	CE	PCU 1	DSV10	778	VALVE	0	
12	11	CE	PCU 1	DSV11	779	VALVE	0	
13	12	CE	PCU 1	DSV12	780	VALVE	0	
14	13	CE	PCU 1	DSV13	781	VALVE	0	
15	14	CE	PCU 1	DSV14	782	VALVE	0	
16	15	CE	PCU 1	DSV15	783	VALVE	0	
17	16	CE	PCU 1	DSV16	784	VALVE	0	
18	17	CE	PCU 1	DSV17	785	VALVE	0	
19	18	CE	PCU 1	DSV18	786	VALVE	0	
20	19	CE	PCU 1	DSV19	787	VALVE	0	
21	20	CE	PCU 1	DSV20	788	VALVE	0	
22	21	CE	PCU 1	V1	789	VALVE	0	
23	22	CE	PCU 1	V2	790	VALVE	0	
24	23	CE	PCU 1	V3	791	VALVE	0	
25	24	CE	PCU 1	V4	792	VALVE	0	

Image: CSV File

If the CSV File is opened in Excel, the extracted database table will appear. In the above-displayed example, the CSV file for CEs will be opened.

The individual CSV files and their processing are described:

General

The tables display the amount of columns and their heading.

The last column "Changeable" describes whether the columns may be processed or not.

The first line with the name or heading of the individual database columns shouldn't be changed. Otherwise you cannot import the CSV file.

Table 15-5 All Control Elements CSV File ["ALLAS_CE.CSV]

Column	Heading	Concerns the following element	Changeable
1	INIT	CE	NO
2	TABLE	CE	NO
3	AUTOMATIONSYSTEM.NAME	CE	NO
4	CE.NAME	CE	YES
5	NUMBER	CE	NO
6	TYPE_CE.NAME	CE	YES
7	IGNOREERROR	CE	YES
8	COMMENT	CE	YES

Table 15-6 All Sensor Elements CSV File ["ALLAS_SE.CSV]

Column	Heading	Concerns the following element	Changeable
1	INIT	SE	NO
2	TABLE	SE	NO
3	AUTOMATIONSYSTEM.NAME	SE	NO
4	SE.NAME	SE	YES
5	NUMBER	SE	NO
6	TYPE_SE.NAME	SE	YES
7	COMMENT	SE	YES

Table 15-7 All Parameter Elements CSV File ["ALLAS_PE.CSV]

Column:	Heading:	Concerns the following element	Changeable:
1	INIT	PE	NO
2	TABLE	PE	NO
3	AUTOMATIONSYSTEM.NAME	PE	NO
4	PE.NAME	PE	YES
5	NUMBER	PE	NO
6	TYPE_PE.NAME	PE	YES

Column:	Heading:	Concerns the following element	Changeable:
7	EXTERNAL	PE	YES
8	COMMENT	PE	YES

Table 15-8 All Link Elements CSV File ["ALLAS_LE.CSV]

Column	Heading	Concerns the following element	Changeable
1	INIT	LE	NO
2	TABLE	LE	NO
3	AUTOMATIONSYSTEM.NAME	LE	NO
4	LE.NAME	LE	YES
5	NUMBER	LE	NO
6	COMMENT	LE	YES

Table 15-9 All Mode tables CSV file ["ALL_MOD.CSV]

Column	Heading	Concerns the following element	Changeable
1	INIT	MODE	NO
2	TABLE	MODE	NO
3	MODETABLE.NAME	MODE	YES
4	MODETABLE.COMMENT	MODE	YES
5	NUMBER	MODE	YES
6	MODE.NAME	MODE	YES
7	MODE.COMMENT	MODE	YES

Table 15-10 All Locations CSV File ["ALL_LOC.CSV]

Column	Heading	Concerns the following element	Changeable
1	INIT	LOCATION	NO
2	TABLE	LOCATION	NO
3	LOCATION.NAME	LOCATION	YES
4	PROCESSCELL.NAME	LOCATION	YES
5	UNIT.NAME	LOCATION	YES
6	VISFORSOURCE	LOCATION	YES
7	VISFORVIA	LOCATION	YES
8	VISFORDEST	LOCATION	YES
9	VARIANT	LOCATION	YES
10	COMMENT	LOCATION	YES

Table 15-11 All Partial Routes CSV File ["ALL_PR.CSV]

Column	Heading	Concerns the following element	Changeable
1	INIT	PARTIALROUTE	NO
2	TABLE	PARTIALROUTE	NO
3	LOCATION.NAME	PARTIALROUTE	YES
4	PARTIALROUTE.NAME	PARTIALROUTE	YES
5	MODETABLE.NAME	PARTIALROUTE	YES
6	SOURCE.NAME	PARTIALROUTE	YES
7	DESTINATION.NAME	PARTIALROUTE	YES
8	VARIANT.NAME	PARTIALROUTE	YES
9	PRIO	PARTIALROUTE	YES
10	COMMENT	PARTIALROUTE	YES

Table 15-12 All Partial Routes with elements CSV File ["ALLRouEL.CSV]

Column	Heading	Concerns the following element	Changeable
1	INIT	RCE /RSE /RPE	NO
2	TABLE	RCE /RSE /RPE	NO
3	MODETABLE.NAME	RCE /RSE /RPE	YES
4	PARTIALROUTE.NAME	RCE /RSE /RPE	YES
5	AUTOMATIONSYSTEM.NAME	RCE /RSE /RPE	YES
6	CE.NAME	RCE /RSE /RPE	YES
7	MODE.NAME	RCE /RSE /RPE	YES
8	COMMAND_CE.NAME	RCE / RSE	YES
9	NOCOMMAND	RCE	YES
10	VALUE	RPE	YES
11	SUM	RPE	YES
12	PULSE TIME ON	RCE	YES
13	PULSE TIME OFF	RCE	YES
14	DELAY TIME ON	RCE	YES
15	DELAY TIME OFF	RCE	YES
16	SORTNO	RCE /RSE /RPE	YES
17	SORTNOALL	RCE /RSE /RPE	YES

Table 15-13 Partial Route with elements CSV File ["XXX.CSV]

Column	Heading	Concerns the following element	Changeable
1	INIT	RCE /RSE /RPE	NO
2	TABLE	RCE /RSE /RPE	NO
3	MODETABLE.NAME	RCE /RSE /RPE	YES

Column	Heading	Concerns the following element	Changeable
4	PARTIALROUTE.NAME	RCE /RSE /RPE	YES
5	AUTOMATIONSYSTEM.NAME	RCE /RSE /RPE	YES
6	CE.NAME	RCE /RSE /RPE	YES
7	MODE.NAME	RCE /RSE /RPE	YES
8	COMMAND_CE.NAME	RCE / RSE	YES
9	NOCOMMAND	RCE	YES
10	VALUE	RPE	YES
11	SUM	RPE	YES
12	PULSE TIME ON	RCE	YES
13	PULSE TIME OFF	RCE	YES
14	DELAY TIME ON	RCE	YES
15	DELAY TIME OFF	RCE	YES
16	SORTNO	RCE /RSE /RPE	YES
17	SORTNOALL	RCE /RSE /RPE	YES

15.3.1.3 CSV import function

After having configured CSV files, you import them into the RCS configuration database. With this action all inputs are deleted from the tables and replaced by the inputs from the CSV files.

i.e. by deleting the Control Elements, whose relations are removed from Partial Routes, which have already been configured and aren't created by transferring the CSV files.

You should therefore consider when importing the following CSV files:

- "ALLAS_CE.csv"
- "ALLAS_SE.csv"
- "ALLAS_PE.csv"
- "ALLAS_LE.csv"
- "ALL_MOD.csv"
- "ALL_LOC.csv"
- "ALL_PR.csv",

that CSV files "ALLROUEL.csv" should be imported if Partial Routes have already been configured with Route elements. The list of CSV files also displays the sequence for importing CSV files in order to avoid misconfigurations.

15.3.1.4 Warnings and Fault Messages

Error:

- Directory couldn't be created!
  - Possible reasons:
    No drive connection available.
    No administration rights for the user.
- Couldn't save Database!
  - Possible reasons:
    Database is also opened via a different database (ACCESS).
    Insufficient memory storage on the hard disk.
- No protocol file!
  - Possible reasons:
    The current protocol file has been opened by a different application. No overwriting possible.
    No administration rights for the user.
- Table couldn't be transferred!
  - Possible reasons:
    There is no database access.
    Database is opened via a different application (ACCESS).
- No Automation system found in this project!
  - Possible reasons:
    The necessary Automation Systems weren't created in the configuration database before starting the import function.
- Type not found!
  - Possible reasons:
    The possible CE, SE, PE, LE types weren't created in the configuration database before starting the import function.
- Name too long!
  - Possible reasons:
    The name which should be transferred into the CSV file is too long for the element.
- Number value isn't valid.
  - Possible reasons:
    The value in the CSV file isn't numeric.
- IgnoreError value isn't valid.
  - Possible reasons:
    The value which should be transferred for the error bit isn't numeric or does not have the value 0 or 1.

- External value isn't valid!
  - Possible reasons:
    The value which should be transferred for the external bit isn't numeric or does not have the value 0 or 1.
- Comment too long!
  - Possible reasons:
    The comment which should be transferred is too long
- Duplicate name for indexed field!
  - Possible reasons:

  The transferring element already exists in the database.
  Duplicate configuration in the CSV file.
- PartialRoute ID not found.
  - Possible reasons:
    The Partial Route for the RCE, RSE, RPE or RLE Route element to be transferred hasn't already been configured. The Partial Route CSV file wasn't read in (ALL_PR.CSV)
    The indicated Partial Route name wasn't written correctly.
- No Mode Table found in this project!
  - Possible reasons:
    The necessary Mode tables weren't created before the import function!
    Sequence for reading in CSV files wasn't considered.
- PartialRoute not a member of ModeTable!
  - Possible reasons:
    The Partial Route name in the CSV file hasn't been written correctly.
    The Partial Route isn't assigned to the ModeTable.
- No CE, SE, PE, LE elements found in this project!
  - Possible reasons:
    No CE, SE, PE or LE elements have been configured in the database.
    Sequence for reading in CSV files wasn't considered.
- No Commands found in this project!
  - Possible reasons:
    No Command masks were defined for the transferring element type.
    Control description in the CSV file isn't written correctly.
- No Modes found in this project!
  - Possible reasons:
    Necessary Mode tables weren't configured in the database.
    Sequence for reading in CSV files wasn't considered.
    Description of the Modes in the CSV file isn't correct.
- Mode not a member of ModeTable.
  - Possible reasons:
    Description of the Mode in the CSV file isn't correct.

- Command value isn't valid!
  - Possible reasons:
    No Command masks were defined for the transferring element type.
    Description of the control in the CSV file isn't correct.
- VisForSource value isn't valid.
  - Possible reasons:
    The transferring value for the bit "Source location" isn't numeric or does not have the value 0 or 1.
- VisForVia value isn't valid.
  - Possible reasons:
    The transferring value for the bit "Variants location" isn't numeric or does not have the value 0 or 1.
- VisForDest value isn't valid.
  - Possible reasons:
    The transferring value for the bit "Target location" isn't numeric or does not have the value 0 or 1.
- CheckRes value isn't valid.
  - Possible reasons:
    The transferring value for the bit "position of rest" isn't numeric or does not have the value 0 or 1.
- Pulse Time 'ON' isn't numeric.
  - Possible reasons:
    Description of the Pulse Time in the CSV file isn't correct.
- Pulse Time 'OFF' isn't numeric.
  - Possible reasons:
    Description of the Pulse Time in the CSV file isn't correct.
- Delay Time 'ON' isn't numeric.
  - Possible reasons:
    Description of the delay time in the CSV file isn't correct.
- Delay Time 'OFF' isn't numeric.
  - Possible reasons:
    Description of the delay time in the CSV file isn't correct.
- SortNo isn't numeric.
  - Possible reasons:
    Description of the SortNo in the CSV file isn't correct.
- Element couldn't update!
  - Note!
    General error, please consider other error messages.

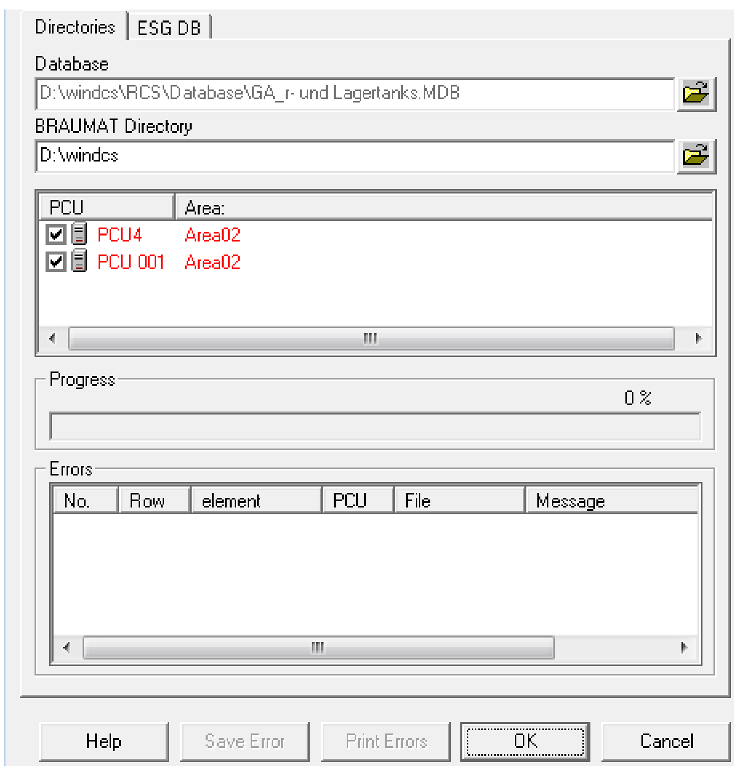
General Messages:

- Table transfer OK!
- Export successful!
- Import successful!
- Close CSV file!

15.3.2 Import/export system data

15.3.2.1 Import function

Basic principles:



By using the import function (under menu 'Database→Import ICMs') it is possible to include the PCUs configured in the system configuration and their ICMs in the RCS configuration.

- In so doing, the ICM text files (e.g. <proj-path>\PCU001\Texte<x>\esg1.txt ... esg4.txt) are read out from the active PCU directories and their entries are applied.
- The PCUs to be imported can be restricted using the selection box in column PCU.

- Further details about the ICM groups are shown in the 'ICM DB' tab. The selection of the ICM groups to be imported can be restricted in the Import column.
- During the import, a check is performed as to whether all available PCUs have been set up in the RCS configuration. If a required PCU is missing, it is created automatically in the engineering file.

Warnings and Fault Messages

- If an error occurs during import, it will be displayed in the lower list window. Usually a
- same term assignment for an ICM is used. The term's item and line number are displayed.
- DBxxx cannot be found.
- ESGx.TXT file not available.

Solving the problem:

- The identical ICM should be renamed and the import function can be executed again.
- Check if DB.xxx is present in PCU folder
- Check if ICMx.TXT file is present in PCU folder

15.3.2.2 Export function

In contrast to the import function, the terms of the determined locations can be transferred into the system configuration via the export function in the menu Database→Export Locations. The names are transferred to saved PCU-related text files (e.g. <proj-path>\pcu001\texte \RCS_Loc.txt).

15.3.2.3 Additional functions

'Database / Compact':

The configuration database has several relations and logic operations to different database tables. During the configuration phase elements will be created new, deleted etc. In that way there will be logic operations, which won't be released from the database. The memory capacity of the database will increase. Using "Compact" in the "Database" menu unnecessary relations and logic operations can be released.

'Configuration / Automatic Compact':

You can activate the automatic compression under the menu item 'Configuration'. By activating the automatic compression the database will be compressed automatically when closing the configuration tool.

15.4 Material and RCS

15.4.1 Overview

Engineering and application:

Engineering and application incorporate the following steps:

- Configuration of materials and material groups
From Version V7 this takes place in the recipe editor (see chapter Material and material groups (Page 585)).
- Definition of predecessor/successor relationships between materials and material groups
From Version V7 this also takes place in the recipe editor (see chapter Material assignment (Page 586)).
- Use of the material configuration during a route search (OS)
- Specification of materials to the route control system by the user program (AS)

RCS material sequences from Version V7

The rule in earlier system versions that materials of a material group are automatically allowed as successors is no longer valid.

The following new rules provide significantly greater flexibility:

- If the material's own group is configured as successor, these materials are allowed as successors.
- If the material's own group is not configured as a successor, these materials are not allowed as successors.

This enables the configuration engineer to create a suitable material sequence configuration according to requirements.

Note

Checking material groups and material sequences

- When the project is upgraded, the existing material definitions, material groups, and material sequence relationships are transferred 1:1 initially.
- These must be checked according to the project requirements in connection with the new rules and, if necessary, adapted. In particular, if materials of a group are to have automatic predecessor/successor relationships with one another, this group must be explicitly configured as a "successor to itself".
- This is done in the recipe editor in the new dialogs of the "Engineering / Material management..." menu.

15.4.2 Server (OS)

15.4.2.1 Load Server

RCS configuration sequence and loading on the server:

1. Route configuration with the application RCS configuration
`'...\rcs\database\xyz.mdb'` is generated or modified
2. Loading the RCS server to accept new RCS configurations.
The route control server interrupts the communication with the ASs.
The route control configuration is copied
`'...\rcs\database\xyz.mdb' → '...\rcs\runtime\rcs.mdb'`
3. The route control configuration is read into the Server.
4. After that the route control server communicates with the ASs again.

Note

For material engineering, the RCS server as of version V7 directly accesses the database `"<proj-path>\material\Database\material.mdb"` created using the recipe editor.

15.4.2.2 Route search and route list

With the route search or the route request a Material ID will be transferred which influences the route search (the algorithm). The Material ID 0 means that there is no material in the means of transport (pipe lines, conveyors, etc.). A Material ID between 1 and 1024 is assigned to a material.

During configuration the user can determine which materials may follow after each other or those which aren't allowed.

If it is assumed that there is a material transport A involving a connection element LE1 and with it the material 458 has been transported, then LE1 is allocated with the material 458. If the route has requested the route list for a second material transport B and searched, a check will be performed as to whether the material 789 may follow after 458. In this case the route will be found, the bit 'MaterialOk' will be set in the Master AS. If the new material hasn't been allowed the 'Bit MaterialErr' will be set.

15.4.3 Automation system (AS)

15.4.3.1 Interface between user program and route control

There are the following options for exchanging data between the user program and the route control system:

1. Calling one of the following FCs from the user program (typically in EOP-FCs). The Material ID can be set in each case via the MATERIAL parameter:
   - RC_ROUTE_CM_FC (FC830)
   - RC_CTRL (FC834)
   - RC_MAT (FC836)
2. RCS unit interface (see section RCS interface (Page 661)). The Material ID can be transferred via the structure RC_ROUTE_CM_UDT.

15.4.3.2 Overtaking the Material ID in the route

First the Material ID is only used and checked for the route search with request of a route if the material is allowed as successor (Flags 'MaterialOk' and 'MaterialError' in Route DB). The Material ID hasn't been accepted in the connection elements of the routes yet – neither with the start of the route (Input ON).

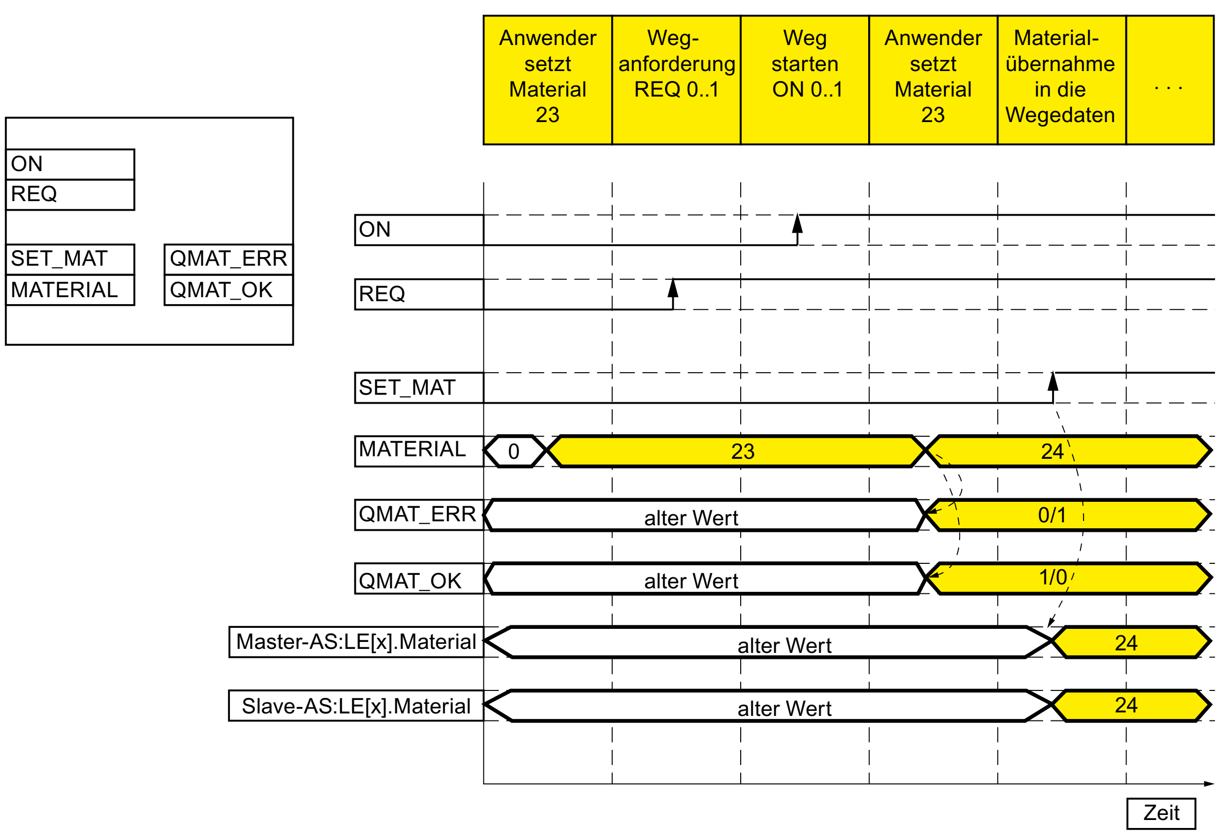


Image: Signal sequence of the input and output parameter relating to material at the route block.

Only with the positive transition of the input SET_MAT will the Material ID be accepted in the connection elements of the Master and the Slave AS.

15.5 Special functions

15.5.1 Dynamic Route ID allocation

15.5.1.1 Scope of use

If in a plant the number of material transports exceeds 300 then the user should ensure a coordination of the ID allocation or that no more solid IDs can be assigned to the single transport processes.

The more simple solution is to leave the route control system to the ID allocation. No more than 300 material transports may occur simultaneously.

The user program assigns the route control system an ID, controls the material transport and gives the ID back to the system at the end (ID is put back in the pool of the route IDs of an AS). The ID is available for another material transport.

15.5.1.2 Function

The dynamic Route ID allocation takes place only in the Automation System. The user program presets the ID 0 and receives an ID back as a return value between 1 and 300 if one ID had been free, otherwise 0.

DB100 (DB RC_CFG/Configuration block of the route control in an AS) determines which Route IDs of the Route control with regard to this AS are available. This information is written by starting the route control server in the AS (compare configuration "Assignment Route to AS").

The following blocks are able to search a free Route ID and to give this back:

- RC_ROUTE_CM (FB 800)
- RC_ROUTE_CM_FC (FC 830)
- RC_ORDER (FC 833)
- SEQ Block and UDT

The two most important parameters with the call of the so-called FCs are ID and DYN_ID.

15.5.1.3 Example SCL

```
MW2000 := 0;
CALL FC 830 (...,
 ID = MW2000;
 DYN_ID = MW2000;
...)
...
CALL FC 833 (...
```

ID = MW2000;

...)

// ID (In-Parameter) = 0 means dynamic route ID search

// DYN_ID (Out-Parameter) = here the found ID will be given back.

In the flag word MW2000 the ID is filed and can be transferred for any further calls by blocks of the Route control, e.g. RC_CTRL, transferred as ID.

15.5.2 Crossing routes

15.5.2.1 Scope of use

There are two main cases of use in which crossing routes may occur:

- Switch over from one source tank to another as soon as the first tank is empty
- Switch over from one destination to another as soon as the first tank is full.

Target is a smooth switchover without losing time, i.e. empty running times should be avoided. Such requests should be solved with crossing routes. There are two routes with the same function identifier whereby the second route is in the AS at the beginning before it will be switched from the first to the second route.

15.5.2.2 Blocks

The known interface blocks of the route control are used:

- RC_ORDER (FC833)
- RC_CTRL (FC834)
- RC_EXT_PE (FC835)
- RC_ROUTE_TIMES (FB813) or
- RC_TR_ROUTE_UNIT (FB830)
- RC_TR_ROUTE2_UNIT (FB850)

15.5.2.3 Example

A switch over should be performed from one source tank (Outlet valve LOC_ID=10) to another (LOC_ID=20) as soon as the first one becomes empty. The destination tank has the LOC_ID=30 at the inlet valve. The first source tank has a built-in sensor that outputs an empty message (Mode 3 is fulfilled) as soon as the source tank contains insufficient material. Mode 2 means for both routes that the material transport takes place (valves are open, pumps run).

Both participating routes

- have different route IDs (fixed or determined dynamically) and
- have the same function identifier (FUNC_ID).

Preparation (Routes are requested)

CALL RC_ORDER (..., ID = 1, FUNC_ID= 99, SOURCE = 10, DEST = 30, ...)

call RC_ORDER (..., ID = 2, FUNC_ID= 99, SOURCE = 20, DEST = 30,...)

if both routes have been loaded (recognizable at the exit QRETVAL):

call RC_CTRL (..., ID = 1, MODE = 2, ...)

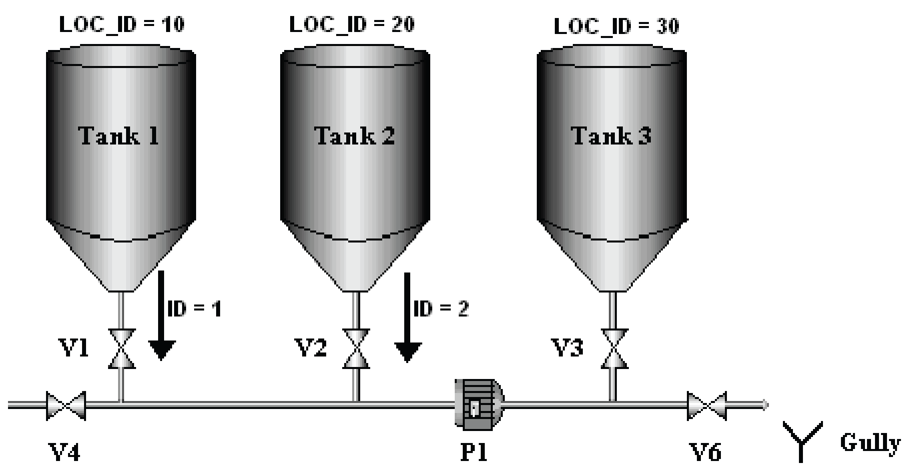
as soon as Mode3 has been fulfilled, the second route is switched on:

call RC_CTRL (..., ID = 2, MODE=2 & 3, ...)

as soon as Mode 2 of the second route has been fulfilled, the first route is switched off:

call RC_CTRL (..., ID = 1, MODE = 0, ...)

Now only the second route runs until an end criterion is reached (material amount reached, time etc.).



15.6 Route tester

15.6.1 Standard Route Test

Under 'Accessories' -> 'Route tester' you'll find the following dialog:

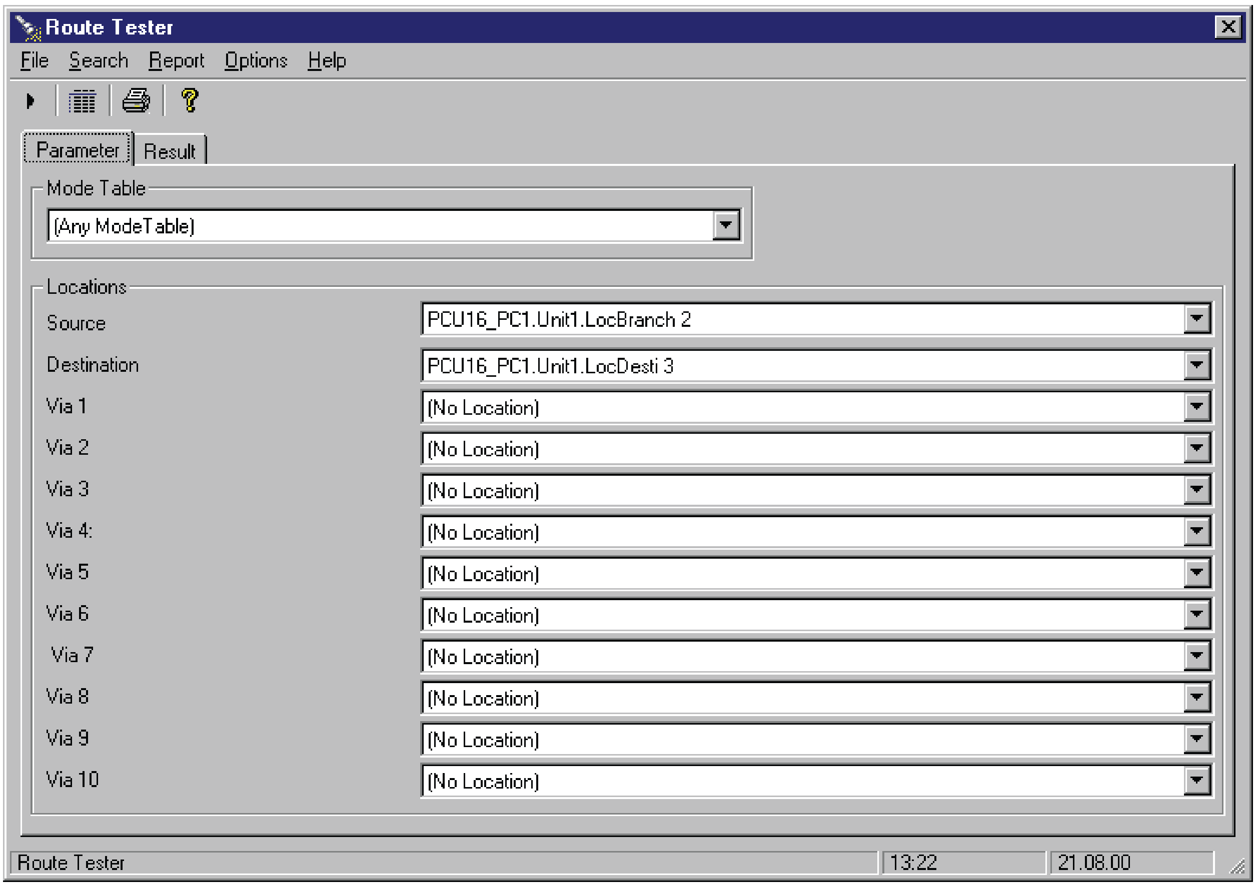


Image: Route tester

You can test Routes that have at least a Source and a Destination under "Locations". Furthermore you can include up to 10 nodes (Via). Duplicated values inside the locations are not allowed. After selecting the locations, you can either initialize the test by 'Search' -> 'Start' or by simply clicking on the icon "Route Search".
To display prio values, activate "Options \ Show prio". The Prio is defined for any Partial Route in the Mode table MT. The progress box shows whether one or several Routes have been found.

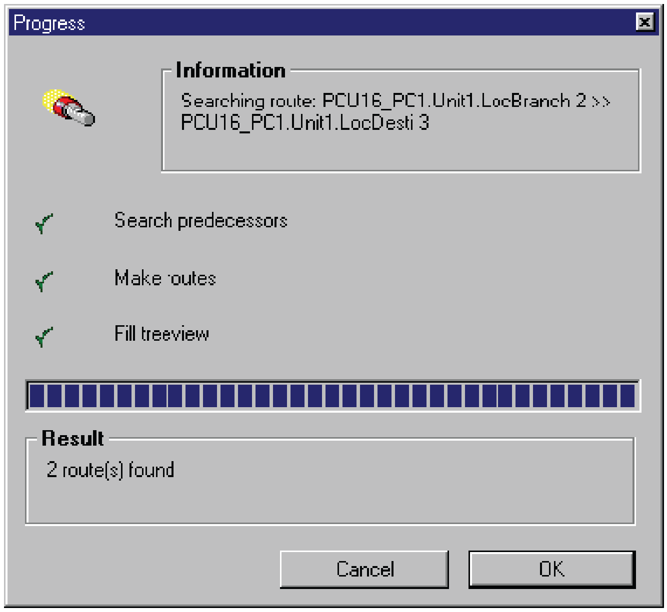


Image: Progress

If at least 1 route has been found for the current locations, the "Results" tab is focused on. The partial routes and all elements are listed in table form on this tab.

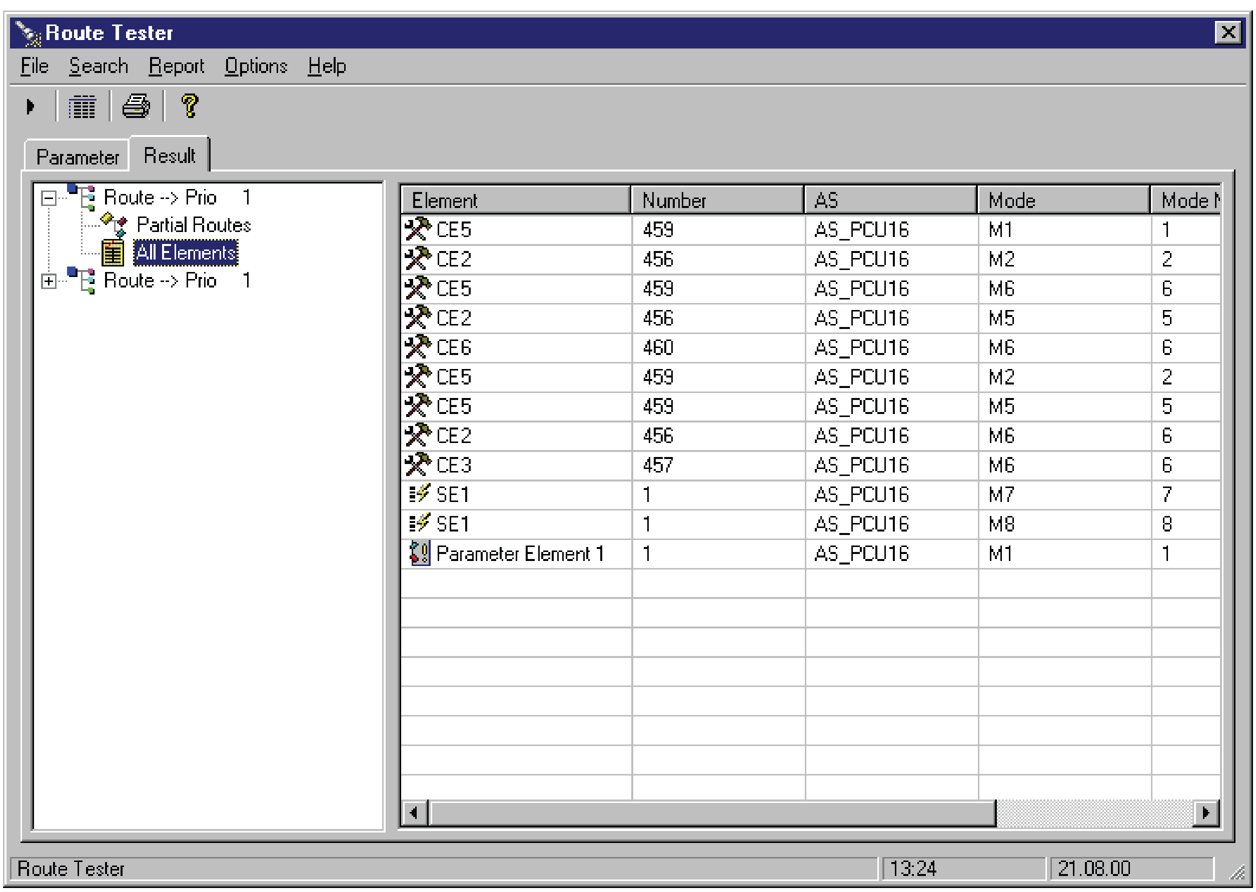


Image: Result

Note

Reaction to long runtime of the route search algorithm

If the runtime exceeds approximately 20 seconds, a dialog appears informing the user that the search algorithm will be busy for a long period of time. The user can then cancel the function. In this case, all routes found to date are listed, with the shortest route always at the top of the list.

Under 'Report' -> 'Print Report' with the Button 'Report'-> 'Display report' you can display the results in report form using the Button . You close the Route test with 'file' -> 'Close'.

15.7 Messages

15.7.1 Overview

Messages are created by the system in special cases:

- Route request (successful and erroneous)
- Diagnosis messages
- Operating and error messages by elements

The messages include:

- A reference to the batch.
- A reference to the unit.

15.7.2 Function in the AS

The batch and unit reference is created via the FC833 showing an input parameter TA. Via this TA number the batch data are read out from the sequencer data record (DB725) and written into the defined route block via ROUTE-ID. Here the batch data of all objects, which create the message, are available.

15.7.3 Messages of a route

Basics

Each route request is recorded as a message. The message is part of the message type 'B' (Operation message). If no valid route is found, more messages will be created that display the reason for the failed search of the route. The route request is always closed with a message.

The structure of a message row is made up of the following information fields (from left to right):

- Time
- Type (B=operation message; S=error message)
- Recipe category
- Order number
- Batch
- Unit PCU
- Unit ID
- Unit name
- PCU
- Block (RCS, CE, SE)

- Block number
- Block name
- Message text ... for examples refer to the following paragraphs

Start message

When a route is requested, a message with the message text "REQ start" is generated.

There is always a "request start" message (type B). This is then followed by one or more error messages. At the end there is either another B message with the successful request or an error message.

Causes of error messages of a route:

- The algorithm cannot find any route on which all partial routes are free.
- Error messages are caused by elements that
  - are disabled "ICM in maintenance" (is possible with CE and SE)
  - are in the "manual operation" mode (possible only with CE)
  - are in the "error" mode (possible only with CE)
  - are in the "affected by maintenance" mode (possible only with CE)
- No alternative route is found

Final message, error messages of the route (block=RCS)

Message text	Description
REQ OK	The route search was successful; in other words, a route was found
REQ ERROR	An error occurred in the route search, for example the route could not be found
Invalid locations	At least one of the node points for the source or destination of the route is invalid (does not exist = 0 or the source is the same as the destination)
Error transferring route list to AS	At least one of the route lists cannot be transferred to the intended AS
Element(s) or partial route(s) already in use	A partial route is already being used
Material compatibility error	The material supplied with the route request is not permitted as the follow-on material

Status messages of the elements (block=CE, SE)

A message is output under the following conditions:

- The route must be active
- If a CE element is included in two routes, two messages are generated. If more than two routes use an element, a message is generated for each route. A message is also generated for each route if the function ID is the same
- This error message is a duplicate of the ICM error message, but is related to the route.

Message text	Description
Route request: Element disabled / Element in maintenance	A route element (CE or SE) is not available (disabled)
Route request: Element is in manual mode	A control element (CE) is in the status "manual operation"
Route request: Element is disrupted	A control element (CE) is disrupted, regardless of the mode or whether the ICM is being used actively or with "NoCommand"
Route request: Element affected by maintenance	An element must not be activated, because it is affected by maintenance work

Error message position error (block=CE, SE):

A message is output under the following conditions:

- A message is generated for active and passive elements
- The route must be active
- If an element is included in two routes, two messages are generated.

Message text	Description
Switch on monitoring disruption incoming / outgoing	Mode edge on + feedback not in defined status + monitoring time elapsed Mode edge off + feedback not in defined status + monitoring time elapsed
Switch on error monitoring disruption incoming / outgoing	Mode on + feedback not in defined status + fault time elapsed

Error message rest position error (block=CE)

A message is output under the following conditions:

- A message is generated only for active elements
- The route must be active
- If a CE element is included in two routes, two messages are generated

Message text	Description
Rest position monitoring disruption incoming / outgoing	Mode off + feedback not in defined status + monitoring time elapsed
Rest position error monitoring disruption incoming / outgoing	Mode off + feedback not in defined status + fault (possibly new rest pos fault time) time elapsed

15.8 Application example

15.8.1 Configuration example "Filling and Cleaning Processes within breweries"

Description:

The small and simplified part from a brewery area includes 3 tanks. The material (beer) is carried, outgoing from the source reservoir tank 1, into the target tanks (Tank 2 or 3). The cleaning process is also important for the RCS, called *Cleaning In Process* (CIP), in addition to the filling process. Water is pumped through dirty pipes and directed out to a drain. In this example, only cleaning of the lines is described.

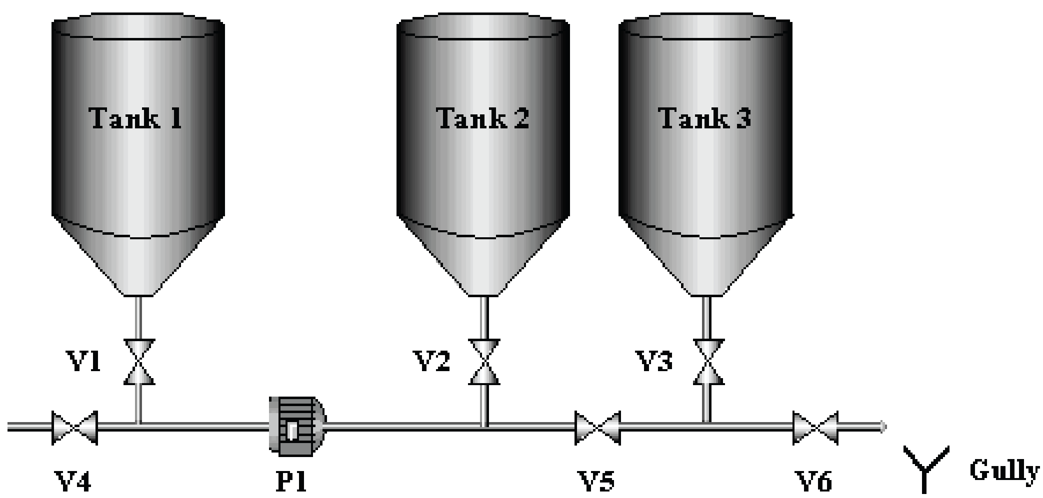


Image: Part of a brewery plant

- Locations
  Before the plant is subdivided into single Partial Routes, the defining locations or points should be determined. The three tanks, which are used as source and destination tank, are overtaken in the list as units. After the pump P1, a branch for tank 2 or tank 3 is possible. For this reason, the fourth location is a branch point and is defined and marked with VP Pump. Locations can be selected in a user-defined way – the setup that is displayed is only used as a guideline.

Table 15-14 Locations of the example plant

Location	Comment
Tank 1	Source tank Tank 1
VP pump	Branching point after Pump 1
Tank 2	Destination tank Tank 2
Tank 3	Destination tank Tank 3

- Control Elements
  The actuators of the plant that are currently active can be read directly:

Table 15-15 Control Elements of the plant as example

Control Element	Comment
V1...6	Valve
P1 Pump	Pump with two speeds

- Mode table
  The Mode Table covers special functions (Modes), which are assigned to a certain operational sequence. In the example below this Mode table is called *filling process* and contains the following Modes:

Table 15-16 Modes of the Mode Table "Filling process"

Modes	Comment
Initial Setting	Check conditions for start
Start	Start of filling over, Opening valves
Pump slowly	Turn on pump with slow speed
Pump Quickly	Go over to pump quickly
Switch off	End

- Partial Routes
  Now you have to configure the individual connections between the location in the form of Partial Routes.

Table 15-17 Partial Routes of the configuration as example

Partial route	From	To	Mode table
Source Route Tank 1	Tank 1	VP pump	Filling
Destination Route Tank 2	VP pump	Tank2	Filling
Destination Route Tank 3	VP pump	Tank3	Filling

The definition of the actual functionality which is linked via the assignment of the individual elements to the Partial Routes should be effected after these preparatory steps. In the following table these parameters are listed for all three Partial Routes. Attributes without any significance have not been listed.

Table 15-18 Control Elements of the Partial Routes

Partial route	Name	Type	Passive	Control	Modes
Source Route Tank 1	V4	CE	TRUE	CLOSE	Initial Setting
Source Route Tank 1	V1	CE	FALSE	OPEN	Start
Source Route Tank 1	P1	CE	FALSE	SPEED_1	Pump slowly

Partial route	Name	Type	Passive	Control	Modes
Source Route Tank 1	P1	CE	FALSE	SPEED_2	Pump Quickly
Destination Route Tank 2	V5	CE	TRUE	CLOSE	Initial Setting
Destination Route Tank 2	V2	CE	FALSE	OPEN	Start
Destination Route Tank 3	V5	CE	FALSE	OPEN	Start
Destination Route Tank 3	V2	CE	FALSE	CLOSE	Start
Destination Route Tank 3	V6	CE	TRUE	CLOSE	Initial Setting
Destination Route Tank 3	V3	CE	FALSE	OPEN	Start

It is obvious that Control Elements are used in many different ways as well as in different Partial Routes. The Route Request and the default of the Modes should be done via the call RC_ROUTE_CM_FC (FC830). In this way, you have to consider that parallel execution of different Modes is also allowed. The basic setting can be combined as an additional security query with all other functions.

In the second step the cleaning process should be integrated. This relates to process sequences whose setup and engineering in RCS are not simple.

Additional process information is also necessary, which is produced by adding further elements besides a changeable Mode Table. In contrast to this, the defined Partial Routes and locations aren't modified.

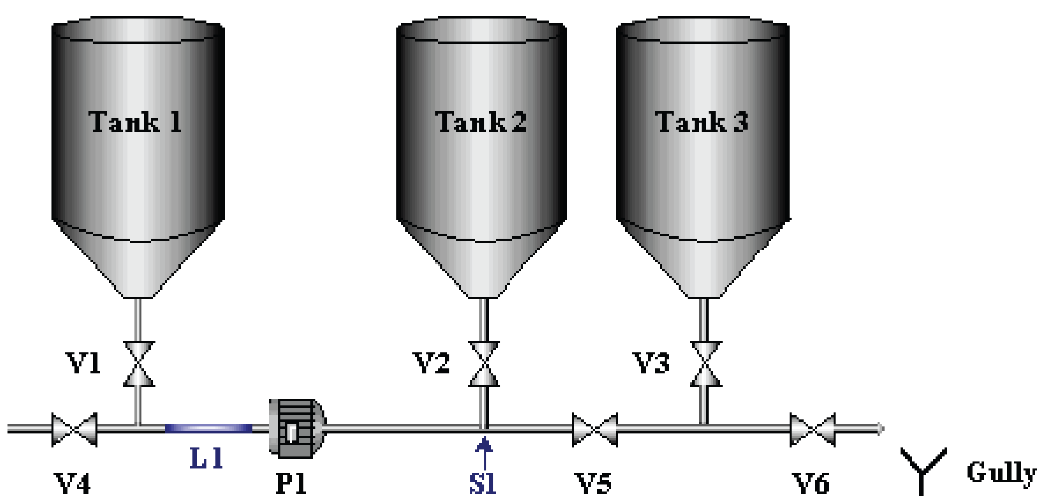


Image: Extended Area

The Mode table is extended as well as adapted by the process filling and named *Filling/CIP*.

Table 15-19 Modes of the Mode table "Decant/CIP"

Modes	Comment
Initial Setting	Check conditions for start
Open Source tank	Open valve to the Source tank
Filling	Filling Destination tanks
Emptying to drain	Emptying water to drain
Start Cleaning Process	Open valve for water supply
Pump slowly	Turn on pump with slow speed
Pump Quickly	Go over to pump quickly
Water	Test whether pipe contains water
Beer	Test whether pipe contains beer

- Process information
  Now you have to configure the individual connections between the location in the form of Partial Routes.
  The transition from CIP to the Filling process occurs without any interruptions, i.e. there is no separate switch-off. CIP liquid (acid, lye, disinfection) and beer are pumped successively through the pipes. The transfer controls a sensor (S1), which determines the current pipe content.
  In contrast to the Sensor Element, which supplies information via pipe state, the Link Element is used for the permanent storage of this information. Possible material combinations are tested using the Transition Table and are included in the Route search. Therefore the Link Element (L1) doesn't play any role in further sequences.
- Partial Routes
  The division of Partial Routes with the assigned indications relating to Source and Target don't change as mentioned before.

Table 15-20 Elements of Partial Routes with corresponding parameters

Partial Route	Name	TYPE	Passive	Control	Modes
Source Route Tank 1	V4	CE	TRUE	CLOSE	Open Source tank
Source Route Tank 1	V1	CE	FALSE	OPEN	Open Source tank
Source Route Tank 1	L1	LE	FALSE	----	---
Source Route Tank 1	P1	CE	FALSE	SPEED_1	Pump slowly
Source Route Tank 1	P1	CE	FALSE	SPEED_2	Pump Quickly
Destination Route Tank 2	V2	CE	FALSE	OPEN	Filling
Destination Route Tank 2	V5	CE	FALSE	OPEN	Emptying to drain
Destination Route Tank 2	V6	CE	FALSE	OPEN	Emptying to drain

Partial Route	Name	TYPE	Passive	Control	Modes
Destination Route Tank 2	S1	SE	FALSE	Level_1 [Conductivity Water]	Water
Destination Route Tank 2	V4	CE	FALSE	OPEN	Start Cleaning Process
Destination Route Tank 2	S1	SE	FALSE	Level_2 [Conductivity Wort]	Beer
Destination Route Tank 3	V3	CE	FALSE	OPEN	Filling
Destination route Tank 3	V5	CE	FALSE	OPEN	Initial Setting
Destination Route Tank 3	V6	CE	FALSE	OPEN	Emptying to drain
Destination Route Tank 3	S1	SE	FALSE	Level_1 [Conductivity Water]	Water
Destination Route Tank 3	V4	CE	FALSE	OPEN	Start Cleaning Process
Destination Route Tank 3	S1	SE	FALSE	Level_2 [Conductivity Wort]	Beer

In contrast to the first example, sequences don't result directly from the function definitions, but they derive from the combination of different Modes. Enormous flexibility can be achieved by separating complex tasks into special sections.

Table 15-21 Elements of Partial Routes with corresponding parameters

Modes	Step 1	Step 2	Step 3	Step 4	Step 5	End
Initial Setting	X	X	X	X	X	-
Open Source tank	-	X	X	-	-	-
Filling	-	-	X	X	-	-
Emptying to drain	X	X	-	-	X	-
Start Cleaning Process	-	-	-	X	X	-
Pump slowly	X	X	-	X	X	-
Pump Quickly	-	-	X	-	-	-
Water	X	-	-	-	-	-
Beer	-	-	-	X	X	-

Route control and monitoring process mode 16

16.1 Set up of Route Control

16.1.1 Settings after the first start

After having started up the Route Control Server for the first time, you have to connect to the current engineering database. The diagnosis window of the PCU Server is started either via the Menu '*Options'-> 'Route Control'* or via the illustrated icon in Fig. 1. Via the Menu *'Functions'-> 'Load database'* the engineering database has to be selected. After that, a start up takes place automatically in which the data are read in the selected engineering database. In order to read in further engineering changes, the function in the Offline tool should be used under the Menu *Tools-> 'Loader'.*

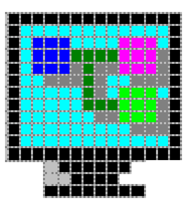


Figure 16-1 Icon for activating the diagnosis window in the toolbar of the PCU Server

In order to switch over from one server to another, the menu *'Program'-> 'Activate Route Control' should be selected.* You can click the icon from the toolbar which is displayed in the figure (Icon for switching over) as an alternative.

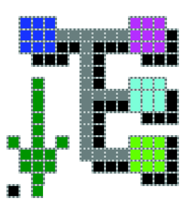


Figure 16-2 Icon for switching over

16.2 Operation and Control

16.2.1 RCS Server

The RCS Server has a diagnosis window which can either be activated via the Menu 'Options'-> 'Route Control' via the PCU Server or the corresponding icon in the toolbar.

16.2.1.1 Diagnosis functions

Basics

The diagnosis window of the Route Control displays information in tabs, which are listed subsequently. You have to consider that these data are available in compressed form.

Tab <Status>:

Value	Description
IOS	IOS number
Status	Status of local IOS (e.g. SERVER, STANDBY or general inquiry running)
Server	IOS number of the current server
Standby	IOS number of the current Standby Server
PCUs	List of the configured PCUs. Numbers in brackets mark either S5 or S7 modules which have no connections to the server.

Data of the tab <Status>

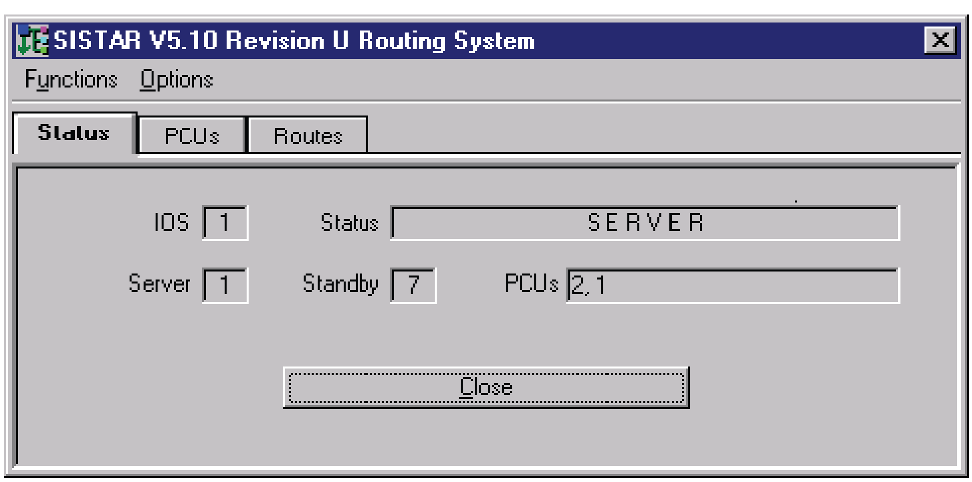


Figure 16-3 Tab <Status>

Tab <PCUs>

In the list view PCUs are listed which are configured in the Route Control configuration. You can see the significance of the individual columns in the following table.

Column	Description
PCU	Number of the PCU
Type	Only S7 type is used in this system version
Master For	List of Routes which are controlled by this PCU.
Supervising	This column gives information about so-called Alive messages. The four numeric values are to be interpreted as follows: Amount of received messages Amount of missing messages Number of messages that can be missing before an error is displayed. Cycle time in which messages can be transmitted.
FIFO	Used FIFO data record
Svr	IOS number that is the Server in this PCU at present
Stdby	IOS number that is in the Standby mode in this PCU
State	Status of the PCU (e.g. SERVER)
Last Error	Last error message

Meaning of the column within the PCU list view

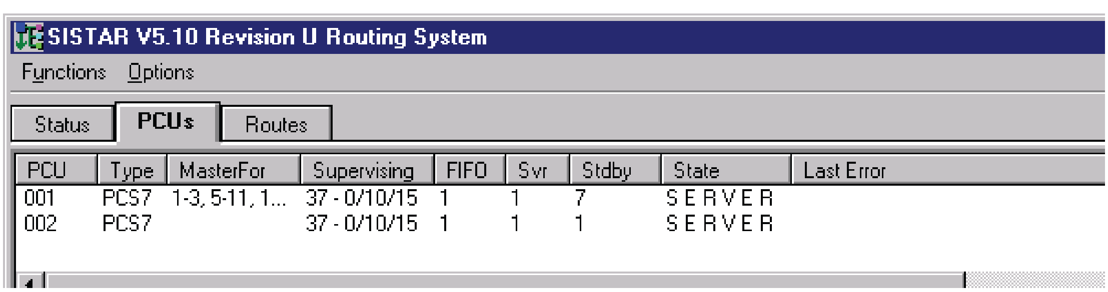


Figure 16-4 Tab <PCUs>

The PCU error messages/statuses entered in the "Last Error" column are listed and, if applicable, described in more detail in the following tables. If errors/warnings are involved, the color of the text turns red/blue.

Note

Not all error texts are multilingual.

Error text	Description
Could not get physical address for FIFO x	Cause: Incorrect OM32.dll or Param.pcu Version
Database error	Error when reading in the database -> convert database; remove write protection
Setting Master Bits failed!	Operation in the route DB100 failed -> possible PCU connection abort; DB is not available
Failed to initiate Update tele!	Writing in DB 100 failed; PCU connection abort/PCU in Stop; DB 100 is not available
Failed to initiate route info tele!	Writing in the AS failed; PCU connection abort/PCU in Stop
Failed to initiate element info!	Writing in the AS failed; PCU connection abort/PCU in Stop
Missing general request tele!	Message that is necessary for the general inquiry has not been received; PCU connection abort/PCU in Stop
Reading route information failed!	Read out the route info DBs (DB404) failed -> PCU connection abort/ PCU in Stop
Read element information failed!	Read out the route info DBs (DB404) failed -> PCU connection abort/ PCU in Stop
Processing element information failed!	Internal processing failed (no remedy possible)
Setting FIFOs failed!	Writing in DB670 or DB690 failed; PCU connection abort/PCU in Stop; DBs are not available
Killer message initiation failed!	Writing in the AS failed; PCU connection abort/PCU in Stop
Monitoring message failed!	Writing in the AS failed; PCU connection abort/PCU in Stop
Error when setting the Standby IOS	Writing in the DB 100 failed; PCU connection abort/PCU in Stop; DB 100 not available
Reading the Server IOS failed!	Reading the DB 100 failed; PCU connection abort/PCU in Stop; DB 100 not available
Monitoring message failed!	Monitoring message of AS missing; -> PCU connection abort/PCU in Stop or RCS Server failure
Wait for Killer message(s) ... Timeout	Monitoring message of the AS is missing; -> PCU connection abort/PCU in Stop

: Error messages

Status message	
Killer message received	
Route message received	
Element message received	
Start redundancy ...	
Synchronize database ...	
Open database ...	
Read materials ...	
Read data ...	
Start general inquiry ...	

Status message	
Set Master Bits ...	
Master Bits set	
Update message initiated ...	
Route message initiated ...	
Element message initiated...	
Read route information ...	
Repeat general inquiry	
Ready	
Read element information ...	
Set FIFOS	
Killer message initiated ...	
Server	
Standby	
Aborted	
Wait for Killer message(s)	

Statuses

Tab <Routes>

There are Route data in compressed form on the tab <Routes> :

Column	Description
RouteID	Number of the Route (1..300)
PCU	PCU number from which the Route is controlled
Cnt	Amount of Route requests altogether
Time	Summed up time of all Route Requests
Thrd	Request thread is active
FPAct	If the Faceplate Active Bit is set, all detailed data are sent to the server in order to visualize (e.g. RCS Online). The first of the two numerical values indicates whether the bit is set and the second stands for the number of registered Clients.
State	Number which admits conclusions to the state of the Route. This is helpful e.g. for cross-coupling problems.
FuncID	Number of the function identifier, which plays an important role in the Route request as the elements are occupied with this ID.
Mode	Value of the function (32-bit) in hexadecimal (higher and less qualified data word)
Unit	Unit which has started the Route
Year	Job year
Rtype	Recipe type
Order	Order number
Batch	Batch number
Source – VIAs – Dest	IDs of different locations

Column	Description
PRts	Numbers of the Partial Routes
Description	State of the Route in plain text or error information

Meaning of the columns in the Route view

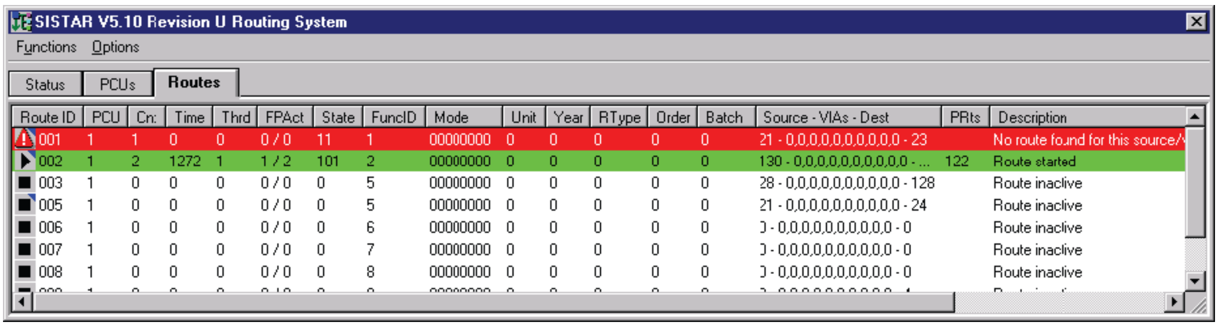


Figure 16-5 Tab <Routes>

Note

Information in this diagnosis tab is available only on the actual RCS server station. On the Standby server the message 'No Routes to show in this list' appears.

16.2.1.2 Settings options

You can display the parameters of configuration file ‘RCS_PORT.INI’ and edit them via the Menu 'Options' -> 'Settings' of the RCS Server application. The following different sections are available:

- [Global]
  - *SetPCS7Flag* -> This bit defines whether the system configuration should be downloaded automatically. By default, the system is filed in the Offline database and is identical for all PCUs. However, if the mixed operation is used, this bit should be deactivated and the corresponding bits have to be set directly in the PCUs.
  - In the current system version, *SetPCS7Flag* must always be set to "0"
- [Database]
  - *Synchronize*-> Database of the Server and Standby are synchronized during start up
  - *OfflineDB* -> Path to the Offline database which was loaded last.
- [Alarm]
  - *Enable* -> Messages are created by RCS (messages and faults)
  - *SendToStandby* -> Messages are also routed to the Standby IOS
  - *SetAlarmBit* -> Set Message bit in Unit data record
  - *OnlyErrors* -> 0 = view of all messages / 1 = faults/errors only

- [Supervising]
  - *Enable* -> Activate/Deactivate monitoring
  - *Cycle Time* -> Cycle time in which messages of the PCUs are expected.
  - *MaxMissCnt* -> Number of missing messages which will cause an error.
  - *AutoSwitch* -> Automatic Server switch in case of error.
  - *AutoRestart* -> Automatic Restart in case of error
  - *BroadcastMessage* -> Send Server failure to Clients
- [Rules]
  - These rules have an effect on the search algorithm. By default, definite status information causes an exclusion of elements from the search. You can modify these standard settings via the following bits.
  - *IgnoreManual* -> Ignore manual operation
  - *IgnoreMaint* -> Ignore if an element was removed
  - *IgnoreError* -> Ignore ICM error
  - *IgnoreDisabled* -> Ignore disable flag; this bit is used internally in order to e.g. disable the elements from PCUs which currently have no existing connection.
  - *IgnoreABM* -> ABM (Affected by Maintenance) means that elements are affected by other elements, which are removed. This bit can also be deactivated as a criterion within the Route search.
  - *SE_TYPE_ID ->* It may frequently be the case that special route combinations need to be excluded. This can be achieved by disabling the partial routes in the way that you set sensor elements in the state 'Disabled'. The search algorithms of the route control will include these elements in the list of the disabled elements if no routes can be found, as there is no differentiation as to whether this state was actively sought (configuration/ programming) or whether it indicates an error case. This is the problem as the user is confused by this information in case of error. Via the setting SE_TYPE_ID, the ID of the type is defined. Its sensor elements will not be included in the error list, so they can only be used as "alterable switches". If you want to use this function, an individual SE type should be created during the configuration of sensor elements that are used in the above-mentioned manner. (*Note:* SE_TYPE_ID=0 switches off the function.)
- [Material]
  - *Enable* -> Include the configured material sequence in the Route search.
  - *IgnoreUnknownIDs* -> If unknown Material IDs were used, this will cause a material error. This standard setting can be bypassed with the switch <IgnoreUnknownIDs>

16.2.1.3 Expanded functionality

The RCS Server via the menu Function offers the following possibilities besides the diagnosis windows:

- RCS Online -> Start Online Tool
- RCS Offline -> Start Engineering Tool

- Load database -> This function has to be executed for the first start of the Route control server, so that the engineering database is copied into the Online path. Any further loading processes should be activated via the configuration tool.
- Restart -> Restart of the Route control

16.2.2 RCS Online

16.2.2.1 Route View

One of the remarkable features of RCS Online is that there are different views, which supplement the content of each view. The following figure provides a total overview of the structure of the Online application, which makes it easier to assign single explanations.

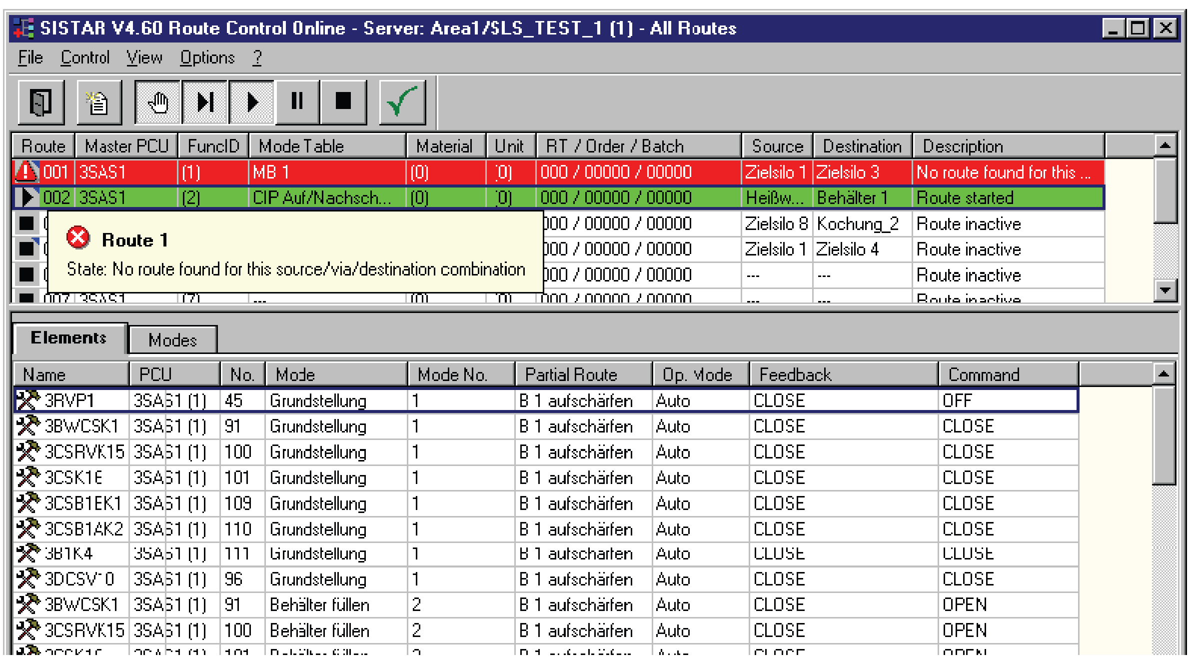


Figure 16-6 RCS Online

Display

The Route view offers an overview of Routes which are created in the engineering tool indicating the details listed below:

Route:

Number of the Route (1..300). The Route state or the corresponding error text, in case of error, will be inserted as a tooltip.

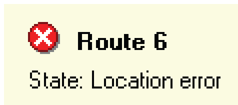


Figure 16-7 Tooltip <Route status>

Note

If the route is in manual mode, this indicated by a blue corner marking at the top right.

Master PCU:

A Route can only be controlled actively in one PCU This PCU is called the Master PCU for this Route and it is specified in the Offline engineering. This information is loaded in case of a general inquiry in the PCU

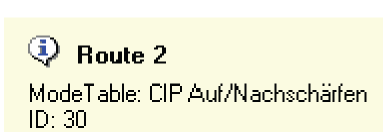
Function identifier:

Routes with the same function identifier may also use the same Elements/Partial Routes (key word concurrent Routes). The name of the function identifier is adjustable via the engineering. The name and the assigned number of the function appear as a tooltip.

Tooltip <Function identifier>

Mode table:

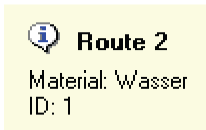
The mode table plays an important role during Route requests, as only the Partial Routes are accepted in the Route search which correspond to the mode table. If the ID is 0, any partial Route can be used for the route



Tooltip <Function catalog>

Material:

Information as to which material should be transported can be supplied via a Material ID. If Link elements are available in the used Partial Routes, their current material conditions will be compared with the material by the material configuration. Non-permitted sequence conditions will cause a material error.



Tooltip <Material>

Unit:

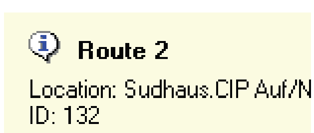
Unit which has started the Route.

Recipe type / Job / Batch (see Fig. "Tooltip <Route status>"):

Recipe type / Job number / Batch number

Source/Destination:

Name of the location of the actual Route Destination/Source. The tooltip displays the complete name of the location in the plant (in the context of the process cell/unit) under indication of the assigned ID (compare figure below).



Tooltip <Location>

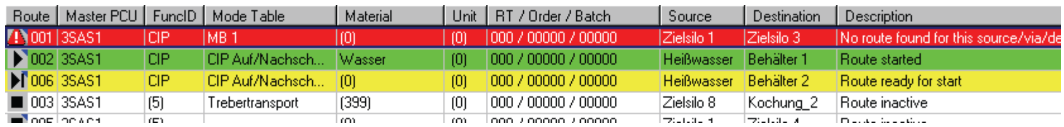
Description:

Status of the Route or, in case of an error, the corresponding error message.

Route	Master PCU	FuncID	Mode Table	Material	Unit	RT / Order / Batch	Source	Destination	Description
001	3SAS1	CIP	MB 1	(0)	(0)	000 / 00000 / 00000	Zielsilo 1	Zielsilo 3	No route found for this source/via/de
002	3SAS1	CIP	CIP Auf/Nachsch...	Wasser	(0)	000 / 00000 / 00000	Heißwasser	Behälter 1	Route started
006	3SAS1	CIP	CIP Auf/Nachsch...	(0)	(0)	000 / 00000 / 00000	Heißwasser	Behälter 2	Route ready for start
003	3SAS1	(5)	Trebertransport	(399)	(0)	000 / 00000 / 00000	Zielsilo 8	Kochung_2	Route inactive

Tooltip <Route status>

An icon is assigned to the individual Route depending on its state. In addition there is a change in color for special statuses (Route request, Route started and Route in error). The following list displays the different icons with the description of the Route status. The following figure illustrates three Routes in RCS Online; their statuses affect the background color.



'Route view' tooltip with three Routes, which affect the background color by their statuses.

Sorting:

Left-clicking on a column in the Route view will result in the routes being sorted by the value in that column.

In addition to this using the option dialog (see section Settings options (Page 1125)) you can decide whether there is automatic sorting. This setting only comes into effect if the sorting concerns a dynamic value. With sorting according to column <Description>, each change of state could cause a re-sorting. However, it should be noticed that the Route which was selected last remains visible.

Column	Sorting
Route	According to Route numbers (statistical value, i.e. no automatic sorting necessary)
Master PCU	According to the number of the Master PCU (static value, i.e. no automatic sorting necessary)
Function identifier	According to ID of the function identifier
Function catalog	According to ID of the function catalog
Material	According to ID of the material
Unit	According to Unit ID
RT / Job / Batch	According to Job number
Source	According to the ID of the location
Target	According to the ID of the location
Description	First according to Routes in error, according to running Routes and last according to Routes whose Route request is running

Operation

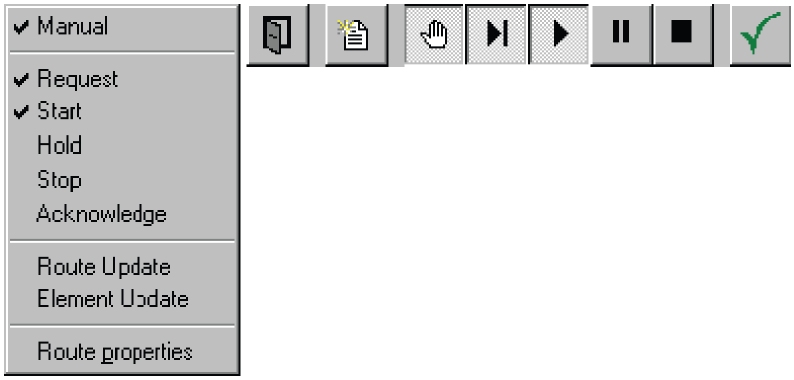
In principle, you divide between operations, which could be effected directly by the Route view, and values, which should be modified via a separate property dialog. A prerequisite is always the manual operation, as only operations are admissible.

Operation by the Route view:

Operation by the Route view can be selected via

- The Menu <Function> (-> select corresponding menu item),
- The context Menu (-> Select Route + right mouse key + corresponding menu item) or

- The toolbar (-> select corresponding button)
- . Different possibilities are combined in Fig. 14.



Tooltip menu <Function> in main or context menu and toolbar.

Commands

The individual commands are explained in the following:

 Internal/External switch (Manual/Automatic mode)

Prerequisite for the Route Control via RCS Online is the manual mode. Switching over to the internal operation is bump-free, e.g. the values preselected by the automatic are integrated in the manual mode. In contrast to that, switching back to the automatic mode is not bump-free, as the applied signals of the automatic program are overwritten immediately.

 Route request

A Route should be requested via this function, i.e., you search for a Partial Route combination, which fits to a source, target and Via1..10 and the assigned elements are occupied and loaded into the participating PCUs. Therefore, the elements and Partial Routes for Routes with a different function identifier are locked, without having started the Route. By resetting the request, the elements are released again. Setting or resetting the request input will only affect the actions if the start input is not set.

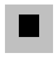
 Start Route

By setting this input, the Route is started, i.e. the individual functions which are involved are activated. If the operation occurs without any Route request, it will be effected implicitly.

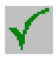
By resetting the route the functions are deactivated again. If the request input is not set, the Route will be released.

 Put Route on Hold

If a running Route is on Hold, the functions will be deactivated. In contrast to resetting the start input (with set Route Request), deactivation delay times are not considered.

 Stop Route

Stopping could be put down to the reset of inputs Route request, Start and Stop.

 Acknowledge

Acknowledge Route errors (e.g. Monitoring time error).

Update Route or elements respectively:

This menu item is used solely as an option for updating the Online view of this Route or their elements respectively and therefore has no influence on the Route control.

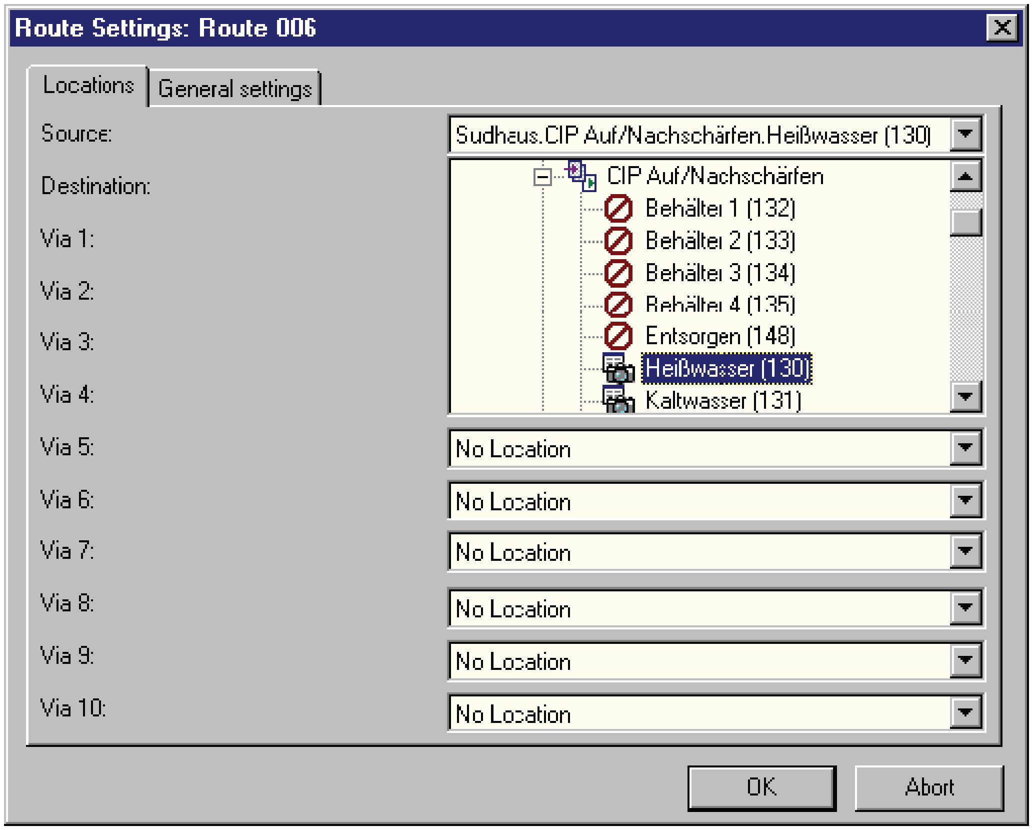
Operation via the property dialog.

Changes via the property dialog will only be allowed if the Route is in internal mode and in the position of rest. Otherwise the OK Button and the individual edit options are deactivated. This additional limitation besides the internal mode is necessary, as these values will be included in the Route request (e.g. source or destination) and a change to the running time would be meaningless.

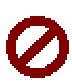
The property dialog is subdivided into two tabs:

Locations:

On this tab, parameters for source, Via 1..10 and destination are located. The display of the locations results in a tree view, which views the plant hierarchy and distinguishes between Process Cell/Unit-related locations and so-called general locations.



If no special locations are preselected (configuration) for source, Via1..10 or destination, these are displayed with the following icon and are, therefore, not selectable:



General settings:

The tab 'General settings' offers the possibility to change the following values (see also Fig. "Tab <Locations>" below):

Mode table:

Via a combination field which contains the list of all configured Mode tables and the input <Any Mode tables>, the requested input can be selected. The consequences for the Route request have already been described in chapter Display (Page 1106).

Function identifier:

The function identifier of a Route plays a role in the element and Partial Route allocation. This means that all Partial Routes and active elements with this function identifier are allocated and may only be used by Routes with the same function identifier. The input of the function identifier also occurs via a combination field, supporting the direct input of numbers, as the configuration of function identifiers is not mandatory. The input 0 is not admissible as this would be the same as avoiding the element allocation.

Material:

The input of the material can also be executed via a direct numerical value or by selecting an input from the combination field. The material at the Route affects the Route search, as all link elements which are included in the Route will be checked i.e., using the material configuration it will be checked whether the requested material may follow after the current material that is stored in the Link elements.

Ignore errors:

This input affects the properties of the Route in case of error. If this bit is set, there will be no automatic deactivation of the route. Furthermore, the error is only displayed by a running Route in case of error.

16.2.2.2 View of the participating elements of the Route.

RCS Online offers two different views of all included elements on the Route. You can change between both views via tabs.

Element view

General:

Individual elements and their commands are the main focus. Therefore, an input is to be effected for each use of an element and can be compared with the engineering tool by the display type.

The illustrated columns in the following figure have the following meaning:

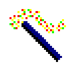
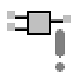
Name	PCU	No.	Mode	Mode No.	Partial Route	Op. Mode	Feedback
3CSGK4/1	3SA61 (1)	93	Behälter aufsch..	3	B 1 aufschärfen	Auto	CLOSE
3CSRVK15	3SA61 (1)	100	Behälter aufsch..	3	B 1 aufschärfen	Auto	CLOSE
3CSK¯6	3SA61 (1)	101	Behälter aufsch..	3	B 1 aufschärfen	Auto	CLOSE
3CSB¯EK1	3SA61 (1)	109	Behälter aufsch..	3	B 1 aufschärfen	Auto	CLOSE
3CSB¯AK2	3SA61 (1)	110	Behälter aufsch..	3	B 1 aufschärfen	Auto	CLOSE
3RVP1	3SA61 (1)	45	Grundstellung	1	B 1 aufschärfen	Auto	ON
3BWCSK1	3SA61 (1)	91	Grundstellung	1	B 1 aufschärfen	Auto	CLOSE
3BWCSK2	3SA61 (1)	92	Grundstellung	1	B 1 aufschärfen	Auto	CLOSE
3DCSV10	3SA61 (1)	96	Grundstellung	1	B 1 aufschärfen	Auto	CLOSE
3CSRVK15	3SA61 (1)	100	Grundstellung	1	B 1 aufschärfen	Auto	CLOSE
3CSK¯6	3SA61 (1)	101	Grundstellung	1	B 1 aufschärfen	Auto	CLOSE
3CSB3EK1	3SA61 (1)	106	Grundstellung	1	B 1 aufschärfen	Auto	CLOSE
3CSB3AK2	3SA61 (1)	107	Grundstellung	1	B 1 aufschärfen	Auto	CLOSE

3I
Rest P

Figure 16-8 Element view while Route running

Name:

Name of the element. The displayed icon depends on the element type:

Icon	Element type
	Control Element
	Sensor Element
	Parameter Element
	Link Element

PCU:

Name and the number of the PCU in brackets to which the element is assigned.

Number:

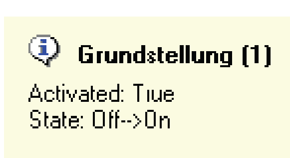
Number of the corresponding element (1..1024).

Function (not relevant for connection elements):

Name of the mode in which the element has been configured. The name depends on the mode table used. The background color of this column is combined with the status in order to enable a diagnosis with a better overview:

Status	Background color
Mode error (Monitoring time/Error tolerance time) run out	Red
Activate mode + feedback not yet OK	Yellow
Activate mode + feedback OK	Green
Deactivate mode + rest position not fulfilled	Gray
Deactivate mode	-

In the tooltip these details are also filed in plain text:



Function number (not relevant for connection elements):

Number of the function in which the element is used (activation/check) Background and tooltip correspond to the column 'Function'.

Partial route:

Special Partial Route in which the element has been configured. This information does not play any role for the running time, but is useful for diagnosis purposes.

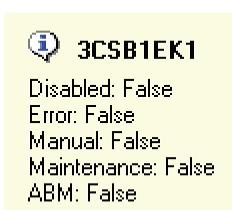
Operation type (is not relevant for Link elements)

The individual status details are evaluated in this column.

Condition	Meaning
Disabled	This is a general bit of the Route Control and is used to exclude elements from the Route Control. So the elements are marked as disabled if their PCU has no connection to the server.
Error	General error of the element
Manual Operation	Elements have been taken out of automatic mode and cannot be controlled by the Route.
Maintenance	Element is removed.
ABM	Element is affected by the maintenance of another element.
Automatic	If there are none of these signals, the element is in automatic mode.

If the element is not in automatic mode, this will be highlighted by the red background color.

In the tooltip these signals are listed individually, as a combination is also possible:



Feedback:

The current feedback is displayed in this column. If the mode is activated, a yellow change in color will be effected, if the feedback does not correspond to the requested commands/checks. With correspondence, the background color is set to green. If there is a mode error (see Column Control), the column will be colored red.

In the tooltip the rest position is displayed:

Note

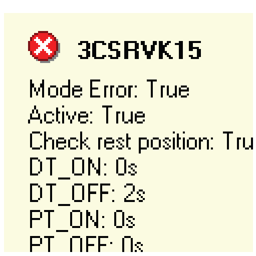
Parameter elements are not updated and the current material is displayed in the column with the link elements.

Command (is not relevant for Link elements):

This column contains information about the command of active or passive elements respectively. Parameter elements are displayed with value and unit.

Information	Description	Type
Mode Error	If an element was the cause of a mode error (monitoring time or error fault time run out), this property will be set.	CE, SE
Active	Sensor Elements are always passive, whereas Control Elements can be installed in an active (controlled) or passive way (only check).	CE
Rest position	Element is monitored if the mode has not been activated. This leads to a mode error, only relevant for active CEs.	Active CE
Delay time on/off	Time (in seconds) for a delayed switching ON or OFF.	Active CE
Pulse time On/Off	Time (in seconds), which changes the element from activated to deactivated state.	Active CE
Sum	Parameter Elements can be installed with the summation attribute, so that the values of all participating parameter elements of a type on the Route can be added. This can be e.g for the dynamic calculation (depending on the combination of the used Partial Routes) of flow rates or operating times.	PE

The following supplementary information that depends on the type of the element appear in the tooltip:



Operation:

This view also offers the possibility to (de)activate the individual modes in internal operation besides the diagnosis function. Via context menu (right mouse key), you can change the mode of the selected element.

- Activating the function x
- Deactivating the function x
- Switching to process picture

Note

Switching to process picture

The process picture in which the element is visualized can be called directly with this function.

Sorting:

A separate context menu is configured for sorting elements (activate right mouse key via any column) to help keep track of the numerous integrated elements. You can update dynamic values in the Route view (adjustable in the Option dialog). The individual sorting possibilities refer to three sections, which are explained in more detail in the following:

Element:

The individual sorting possibilities are self-explanatory with the menu (see Fig. 17). You have to note that the element in state error shows either a function error or a status error (e.g. element is maintained).

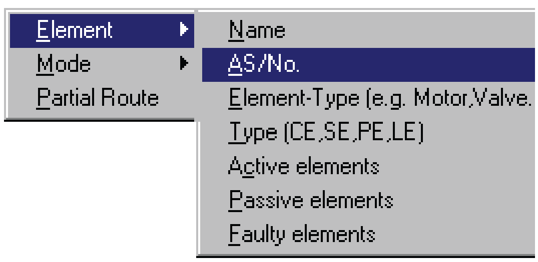
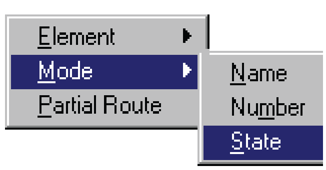


Figure 16-9 Sorting possibilities in reference to the element

Function:

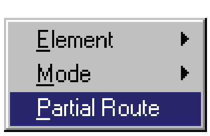
In addition to the sorting possibility according to name and number, you can sort according to the state of the mode, resulting in the following criteria for the sequence:

- Modes in error (monitoring time or error fault time ran out)
- Activated modes, where at least one expected feedback has not been fulfilled yet.
- Activated modes, with expected feedbacks OK.
- Deactivated modes



Partial route:

A sorting is to be effected according to Partial Routes in which the elements have been installed. This sorting is the best way to compare the available engineering, as the display corresponds to the engineering tool perspective.



Mode view

The mode view enables a "mode-oriented" view of the elements of the running Route. This means that all modes used separately by the Route are displayed in the column of the list view, i.e. in order to test the elements of a mode, you have to consider exactly 1 column. In contrast to the element view, each element appears only 1 time, even if it has been used in different modes. Data relating to the corresponding control or interrogation will be filed in the assigned mode column (compare following figure).

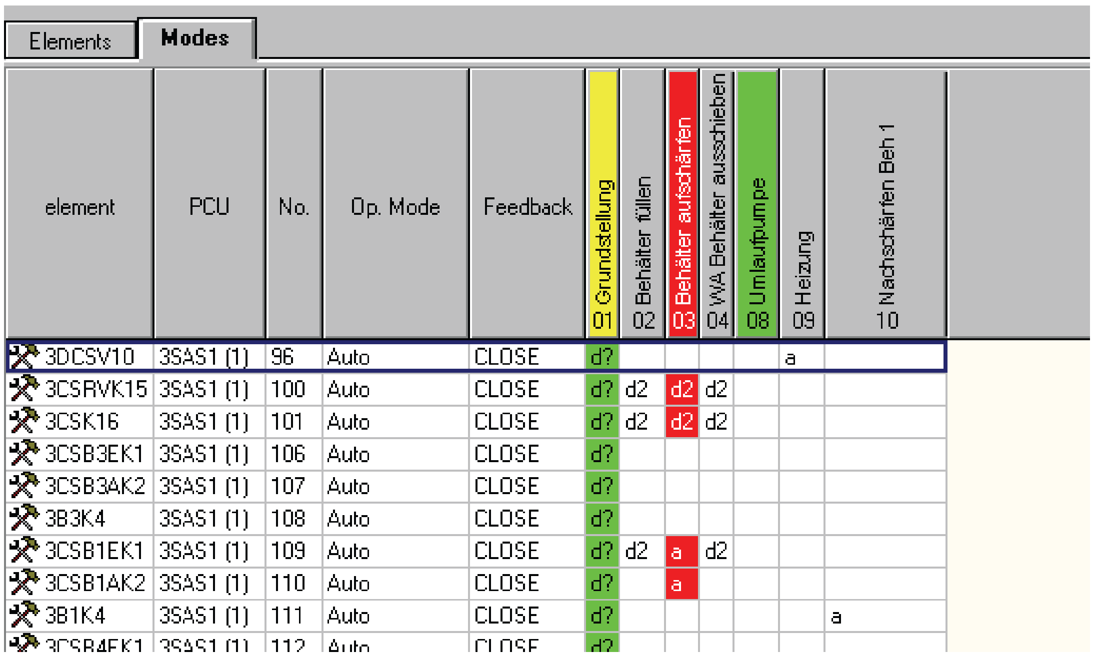
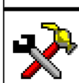
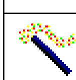
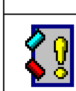
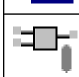


Figure 16-10 Mode view

The meaning of the columns is explained in detail:

Name:

Name of the element. The displayed icon depends on the element type:

Icon	Element type
	Control Element
	Sensor Element
	Parameter Element
	Link Element

PCU:

Name and the number of the PCU in brackets to which the element is assigned.

Number:

Number of the corresponding element (1..1024).

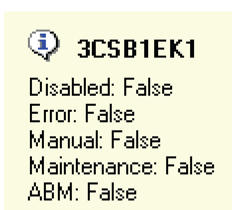
Operation type (is not relevant for Link elements)

The individual status details are evaluated in this column.

Condition	Meaning
Disabled	This is a general bit of the Route Control and is used to exclude elements from the Route Control. So the elements are marked as disabled if their PCU has no connection to the server.
Error	General error of the element.
Manual Operation	Elements have been taken out of automatic mode and cannot be controlled by the Route.
Maintenance	Element is removed.
ABM	Element is affected by the maintenance of another element.
Automatic	If there are none of these signals, the element is in automatic mode.

If the element is not in automatic mode, this will be highlighted by the red background color.

These signals are filed in plain text in the tooltip:



Feedback:

The current feedback is displayed in this column. If the mode is activated, a yellow change in color will be effected, if the feedback does not correspond to the requested commands/checks. In contrast to the element view, there is no change of color, as the feedback could refer to different commands/checks and as a consequence no explicit comparison is given.

In the tooltip the rest position is displayed:

Note

Parameter elements are not updated and the current material is displayed in the column with the link elements.

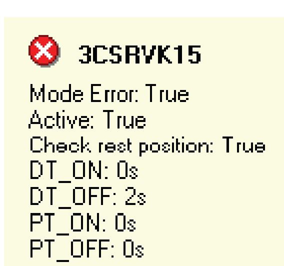
Mode columns (not relevant for link elements):

Depending on the amount of modes used, additional columns are added to the list view, i.e. for each mode a column, displaying the mode number horizontally and the assigned mode name vertically.

An abbreviation is filed for each type of command or check respectively (the value together with the assigned unit for Parameter Elements), which is configured via the 'Options' dialog. You can create a display with a more detailed overview and more information using relatively little space.

Additional information is displayed by a tooltip similar to the tooltip in the element view.

Information	Description	Type
Command	Type of command or check respectively, which is available in this mode. In this way, you can compare this value with the current feedback if a mode error occurs.	CE\SE\PE
Mode Error	If an element was the cause of a mode error (Monitoring time or fault time error), this property is set, consequently.	CE, SE
Active	Sensor Elements are always passive, whereas Control Elements can be installed in an active (controlled) or passive way (only check).	CE
Rest position	Element is monitored if the mode has not been activated. This leads to a mode error, only relevant for active CEs.	Active CE
Delay time on/off	Time (in seconds) for a delayed switching ON or OFF.	Active CE
Pulse time On/Off	Time (in seconds), which changes the element from activated to deactivated state.	Active CE
Sum	Parameter Elements can be installed with the summation attribute, so that the values of all participating parameter elements of a type on the Route can be added. This can be e.g for the dynamic calculation (depending on the combination of the used Partial Routes) of flow rates or operating times.	PE

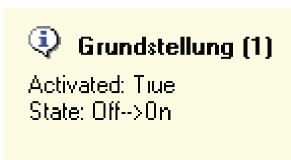


The background color of each individual command/check is adapted depending on the current feedback. An activated mode with a corresponding feedback leads to a green background color, otherwise (incorrect feedback) a yellow color is displayed. If there is a mode error for this particular command/check, the background will be colored red.

Besides the individual commands/checks you also adapt the mode columns depending on the condition:

Status	Background color
Mode error (Monitoring time/Fault time run out)	Red
Activate mode + feedback not yet OK	Yellow
Activate mode + feedback OK	Green
Deactivate mode + rest position not fulfilled	Gray
Deactivate mode	-

In the tooltip these details are also filed in plain text:



The internal mode offers the possibility to (de)activate the mode via the requested mode column (left-click). If an operation is possible (Manual mode), the mouse cursor will be changed to a hand.

Sorting:

The sorting possibilities are limited with the mode view and refer exclusively to the elements; see following figure. The menus and their sorting-related functions have already been mentioned in the element view.

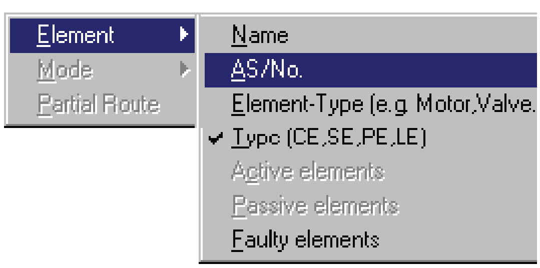


Figure 16-11 Sorting possibilities in the function view

16.2.2.3 Request details

The view 'Request detail' offers diagnosis functions for different error sources, which could occur within a Route request and additional information, e.g. listing of all assigned elements. If information is available within the request details (in case of error) the transfer to this view will be effected automatically indicating only diagnosis windows with contents. Switching to the special view can also be initiated manually via the menu (compare figure "Menu View/ Request details"). These lists remain up to the next Route request and represent the state of the elements at the time of the Route request, i.e. an element is, for instance, marked as 'occupied', but has been released in the meantime.

Figure 16-12 Menu View/Request details

Locked elements

Locked elements are general elements, which caused a Route request to fail. You divide the locked elements into two categories:

Element is occupied

Locked elements which are occupied with a different Route request, i.e. these elements have already been used by one route/several routes with a different function identifier (Note: A multiple allocation is admissible for routes with the same function identifier). In order to mark this in the view, you file the column function identifier red and the corresponding error text for the occupied element appears as a tooltip (compare figure below). In this list there is no Sensor element, as an occupation is to be effected only with parameter elements, active Control elements and link elements.

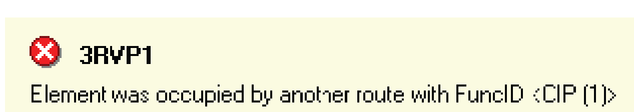


Figure 16-13 Tooltip for an occupied element

Element status

Elements which have been taken out from the Route search due to their status. The known state information is part of it e.g. Maintenance or manual mode. In the tooltip (compare following figure) this information is listed additionally, as several bits could be set at the same time. (In order to emphasize the non-conforming statuses, the column is filed in red color).

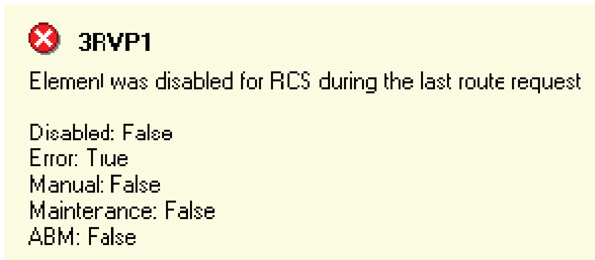


Figure 16-14 Tooltip due to state information of locked elements.

Locked Elements

Name	PCU	No.	Mode	Mode No.	Partial Route	Op. Mo...	FuncID
3RVP1	3SAS1 (1)	45	--	8	B 2 aufschärfen	Error	CIP (1)
3DCSV10	3SAS1 (1)	96	--	9	B 2 aufschärfen	Auto	CIP (1)

Figure 16-15 Locked elements

Additional data are still listed, which makes it easy to find the combination in the engineering (e.g. Name of the Partial Route), besides the columns which have already been mentioned.

Note

If such an error occurs with Route request with any Mode table, no mode name will be indicated, but only the mode number, as an explicit assignment is not possible.

Locked Partial Routes

A Partial Route is occupied and is, therefore, locked for Routes with a different function identifier, if it contains at least 1 active Control Element, Parameter Element or Link Element. The indication of a locked Partial Route is set equally with a list of all Parameter Elements or active Control Elements of this Partial Route. The column which is filed red (compare following figure) shows the function identifier, which occupies the Partial Route. This occupation can be derived from one or more Routes.

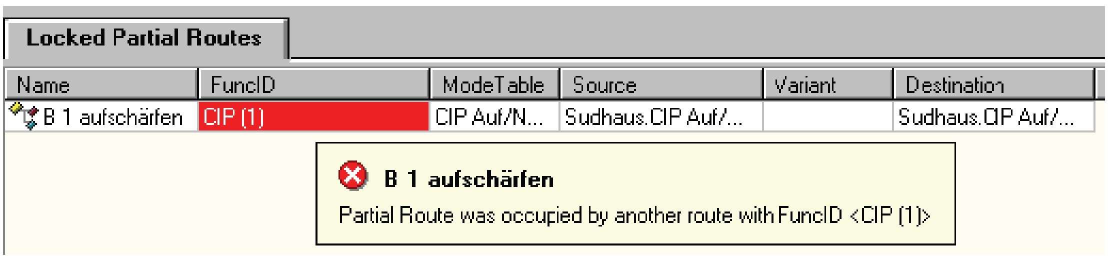


Figure 16-16 Locked Partial Routes with Tooltip

Inconsistent Commands

There are inconsistent commands if an element is used actively in one mode with different commands. Although this is guaranteed within a Partial Route in the configuration tool, it can occur in the dynamic combination of the Partial Route to a whole Route. The following figure shows examples of 2 elements with clashing commands. Via tooltip or background color, the necessary columns are emphasized. In addition to this, some data of the element or the use within the Partial Route is displayed, to find the corresponding elements in the engineering for further adjustments.

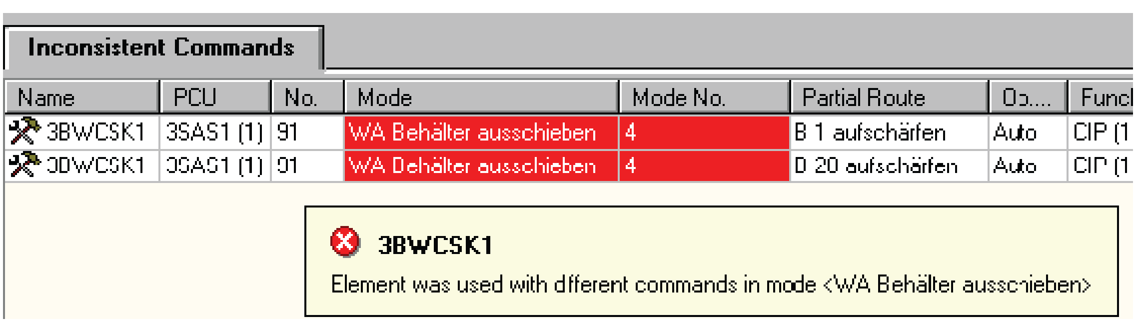


Figure 16-17 Inconsistent commands with tooltip

Material error

If link elements are used, the current material is included in the Route search. A check is performed to ascertain whether the material at the block is a reliable successor of the material which is stored in the Link element. This check takes place using the configured material database. In the display the Link element appears with the material that was filed in the Link element last (compare following figure).

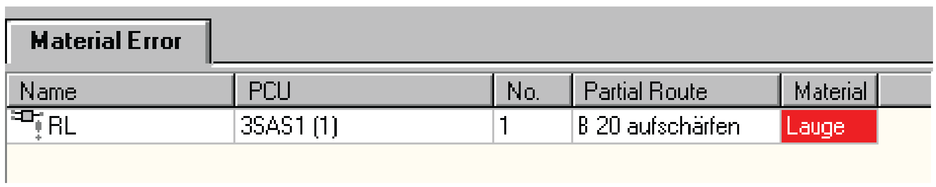


Figure 16-18 Material error

16.2.2.4 Settings options

You can make additional settings, which are explained in the following via an Option dialog (Menu 'Options' ->'Properties'):

Tab <General settings>: (compare figure "Tab <General settings> of the 'Options' dialog" below):

Tooltip:

- Time (in milliseconds) in which the tooltip is present
- Time (in milliseconds) until a tooltip will be displayed

Automatic Sorting:

You can activate or deactivate the automatic sorting for the different views. This setting only comes into effect if you assort according to dynamic values.

Note

If rapid changes to the sorting values occur, the view is frequently repainted and the display is always busy.

Enabling the start of multiple instances:

By setting this switch it will be possible to start several instances of RCS Online on one IOS.

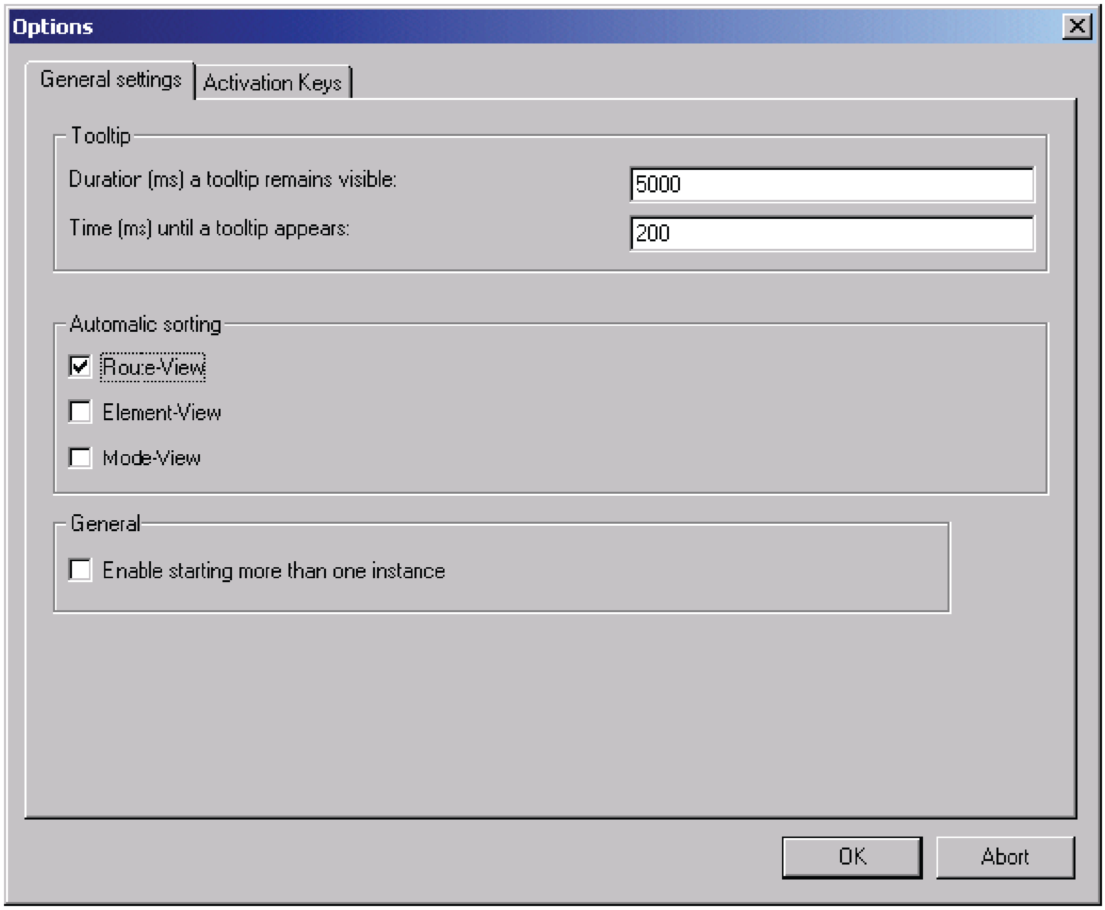
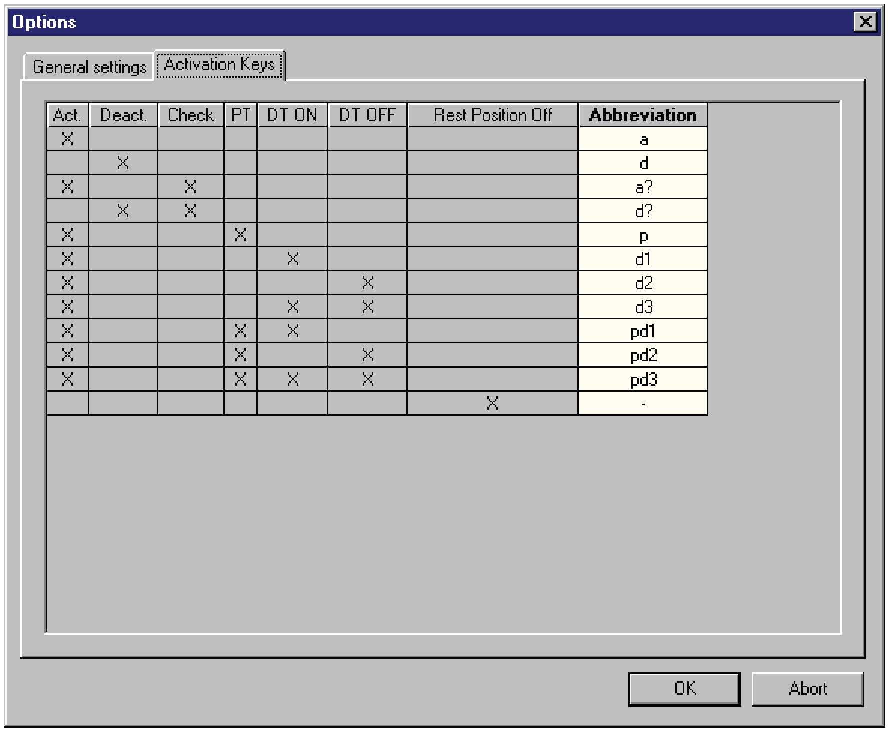


Figure 16-19 Tab <General settings> of the Options dialog

Tab <Activation Keys>:

For the mode view the complete command names are not displayed, but rather special abbreviations, in order to optimize the information of the view. These abbreviations (keys) can be configured individually, preselecting different combination possibilities in tables (see Figure 29).



Act.	Deact.	Check	PT	DT ON	DT OFF	Rest Position Off	Abbreviation
X							a
	X						d
X		X					a?
	X	X					d?
X			X				p
X				X			d1
X					X		d2
X				X	X		d3
X			X	X			pd1
X			X		X		pd2
X			X	X	X		pd3
						X	-

Figure 16-20 Tab <Activation Keys> of the Options dialog

Icon	Meaning
a	This element will be **activated in this mode level**
d	This element will be **deactivated in this mode level**
a?	This element will be **activated in this mode level**, and its actual position will be requested (checked)
d?	This element will be **deactivated in this mode level**, and its actual position will be re-quested (checked)
p	This element will be activated in this mode level in pulse mode.
d1	This element will be activated in this mode level with a switch-on delay time > 0
d2	This element will be activated in this mode level with a switch-off delay time > 0
d3	This element will be activated in this mode level with a switch-on and switch-off delay time > 0
pd1	Same as d1, but with pulse time
pd2	Same as d2, but with pulse time
pd3	Same as d3, but with pulse time
-	The idle state position will also be monitored When the mode level is deactivated, monitoring will be performed to determine whether the element is also switched off (motor) or closed (valve).

16.2.3 RCS Faceplate

The RCS Faceplate is designed for the user (in contrast to RCS Online) and offers different diagnosis functions which are integrated in different views.

Broadly speaking, the RCS faceplate is a control (OCX) that can be integrated in an ActiveX container, such as "Process diagram design". The following section shows an example of this application.

16.2.3.1 Configuration

The following steps are necessary to insert the RCS faceplate in a process picture:

Place Controls:

A dialog will appear in order to select a Control via the menu 'Variables'-> 'Control', or alternatively via the toolbar, and a click on the configured picture. Here you select the input <rcs faceplate>.

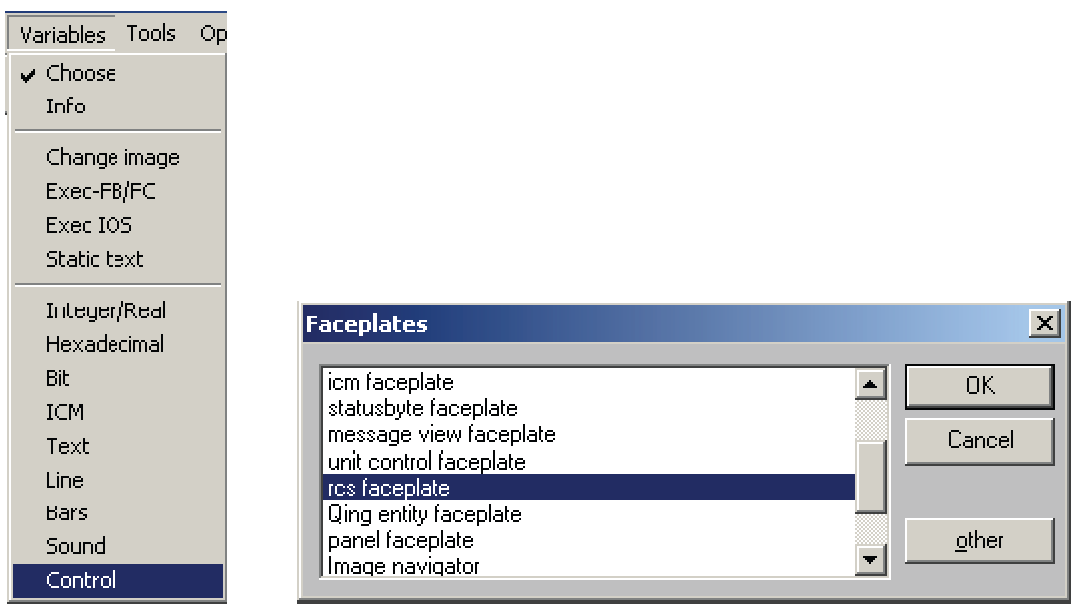


Figure 16-21 Procedure with configuration

Adapt properties of Control:

In reference to the Route control system, there is only 1 property which needs to be explained more precisely, namely the function identifier (compare following figure). The function identifier is a special ID, which gives a hint as to which "type" of Route is used. This function identifier is part of an important role, as active elements are occupied with this function identifier. Only Routes with identical function identifiers can use the same elements. The basic display of the control and the diagnosis possibilities which are involved refer to the Routes with this function identifier.

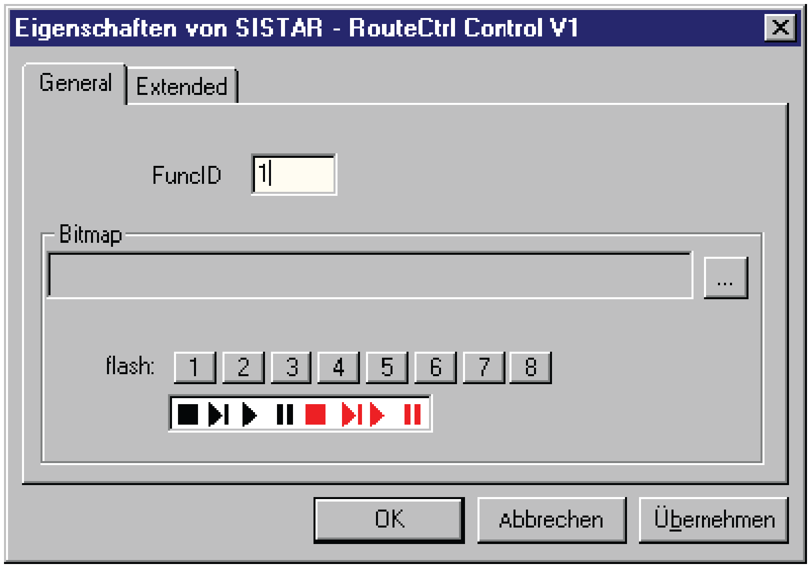


Figure 16-22 Tab <General> of the Properties Control

Note

Multiple instances of the faceplates provided by the system may be placed into the same process picture. For most faceplate types they may be connected to different AREAs. But, due to some system constraints, this feature is not supported with the RCS Faceplate described here.

16.2.3.2 Operation and Control

Compressed process information

In the process display the Control relates to the Route Control Server and shows the status of the Route with the adjusted function identifier by an icon. If there are more Routes with the same function identifier, a prioritization is to be effected, e.g. the display of the Control refers to a higher prioritized Route, where the prioritization takes place as follows:

- Running Route
- Route on Hold
- Routes with running Route request
- Inactive Routes

Moreover, an error icon is displayed, which will conform with the highest prioritized condition, if there is one error for one of these Routes (not only the Route that has the condition with the highest priority but all Routes with this function identifier). The error information is also displayed separately from the status information.

Example:

If there is a Route in the Route request with an error and a Route is running without any errors with the same function identifier, the symbol will be shown in red (Transfer errors from the other Route) for a running Route (highest priority), although there is no error in the running Route, directly.

Blinking of the Control is preset for faulty Routes, but it can be configured via the property dialog. You can also adapt the separate display of an icon in the manual mode via the properties. The icon below will be displayed, if the Route is executed in internal mode.


Manual operation icon

The tooltip, which displays the currently set function identifier and all running Routes with their sources, destinations and statuses, is used for a detailed diagnosis. If the status of a Route is faulty, the Tooltip will display an error icon.

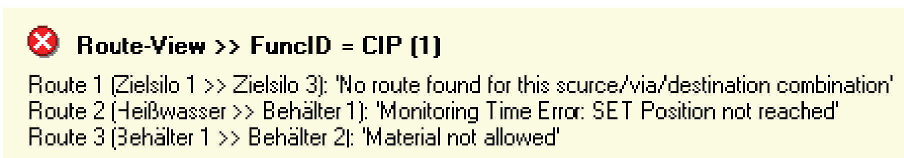


Figure 16-23 Tooltip of the active routes

Detailed Information

Core statement

You receive detailed information by left-clicking on the Control. A separate dialog appears, which contains the following diagnosis window.

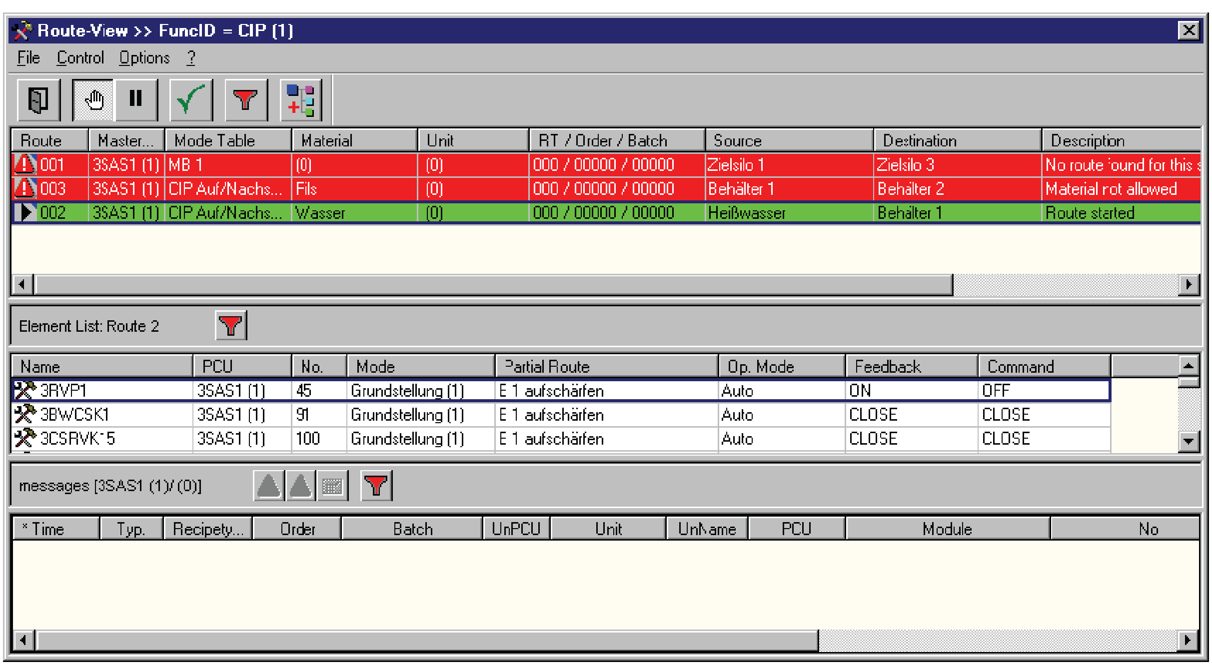


Figure 16-24 Detailed view of the faceplate

Route view:

Active Routes appear in the Route view, if their function identifier conforms with the selected ID. (Via a filter the display can be limited to Routes in state error. The filter from the toolbar is set in this case.) The display type refers mainly to RCS Online. You can see the significance of the individual columns in the section Display (Page 1106). There are two differences:

- There is no column 'function identifier' as this one is identical for all listed Routes and it is also listed separately in the title line.

The complete name (with plant reference) for Source, Destination and (if used) Via Parameter is displayed as Tooltip for all columns.

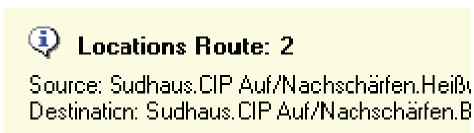


Figure 16-25 Selected locations as Route Tooltip

Operation possibilities:

It is possible to execute special functions via toolbar:

 Internal/External switch (Manual/Automatic mode)

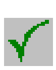
Prerequisite for the Route Control via RCS Online is the manual mode. Switching over to the internal operation is bump-free, e.g. the values preselected by the automatic are integrated in the manual mode. In contrast to that, switching back to the automatic mode is not bump-free, as the applied signals of the automatic program are overwritten immediately.

 Put Route on Hold

If a running Route is on Hold, the functions will be deactivated. In contrast to resetting the start input (with set Route Request), deactivation delay times are not considered.

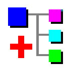
 Stop Route

Stopping a Route is put down to the resetting of the inputs 'Route Request', 'Start' and 'Hold'. i.e., the Route is set to the state "Inactive".

 Acknowledge

Acknowledge Route errors (e.g. Monitoring time error).

If added operations or diagnosis functions are used, RCS Online should be started. This can be effected by clicking on the RCS Online Symbols (see following figure) in the toolbar, selecting the currently displayed Route in the Faceplate.



Symbol <RCS Online> in the toolbar

Element view:

The element view corresponds to the view of RCS online (see section Display (Page 1106)), whereby the columns 'Function' and 'Function number' are combined. A filter can be set additionally in the same way as the Route view, in which only elements in state error are displayed. Included are all errors that concern the state type and mode errors.

Note

Switching to process picture

The process picture in which the element is visualized can be called directly with the shortcut menu (right mouse button).

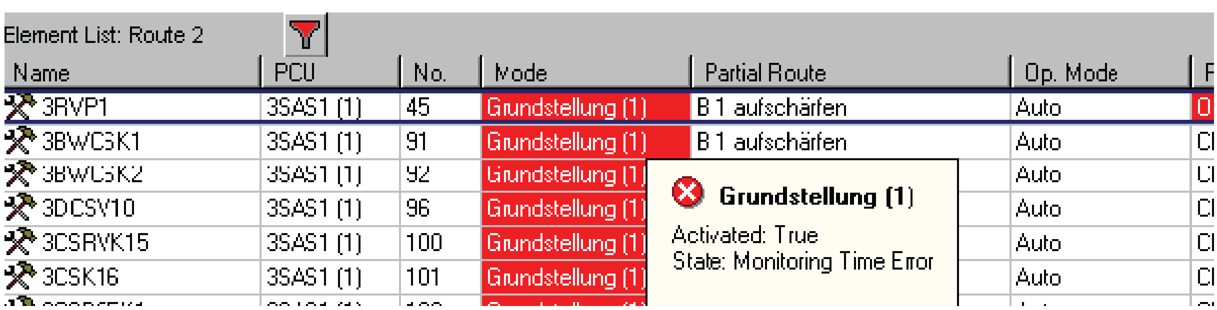


Figure 16-26 Element view with activated filter

Request detail:

The request details are also based on the representation in RCS online and more information is available in the section Request details (Page 1122). If data is available for the requested details, the view will be activated automatically.

Message window:

In the message window messages are listed which refer to the Unit of the preselected Route (see following figure). Via a filter you can reduce the display to the error messages.

Figure 16-27 Message window

Route control for AS blocks (S7-400) 17

17.1 Blocks Overview

Block	Representation	Function	Use
FB801	RC_ROUTE	Route Control basic function, processing of all configured route lists, calls further RC blocks	System
FB802	RC_ROUTE_XC_REC	Cross-coupling between partner PCUs	System
FB803	RC_ROUTE_RCE_ON	Monitor and control RC elements.	System
FB804	RC_ROUTE_RCE_OFF	Enable route elements	System
FB805	RC_ROUTE_TIME	Controls all RC times	System
FB806	RC_ROUTE_XC_SEND	Functions to monitor and transfer data to other ASs associated with the route	System
FB807	RC_ROUTE_STATE_OS	Functions to monitor and send data to OSs	System
FB808	RC_ROUTE_TELEGR	Sends telegram memory to OSs	System
FB810	RC_TIME_RCE	Processes timer addresses for RC Elements	System
FB811	RC_ROUTE_DBL	Expands/reduces Route DB (e.g. after a Route request)	System
FB812	RC_TG34_TG36	Assembles telegram(s) of type 34/36 and sends them to OS	System
FB813	RC_ROUTE_TIMES	Processes internal timer addresses	System
FB816	RC_ROUTE_XC_SND_OR-DER	Sends route order data to slave routes via cross-coupling	System
FB817	RC_ROUTE_XC_PE_ACTV	Sends actual values of external parameter ele-ments to master route via cross-coupling	System
FB818	RC_ROUTE_GET_EXT_PE	Triggers sending actual values of the external PEs from slaves	System
FB822	RC_IF_MOTOR	Motor interface	User
FB823	RC_IF_MOT_REV	Reverse motor interface	User
FB824	RC_IF_MOT_SPED	Interface for a motor with two speeds	User
FB825	RC_IF_VAL_MOT	Motorized valve interface	User
FB826	RC_IF_VALVE	Valve interface	User
FB827	RC_IF_CE_COMMON	Interface for controls/feedbacks of a freely de-fined Control Element	User
FB828	RC_IF_LE	Returns Link Element state	User
FB830	RC_TR_ROUTE_UNIT	Patches route request or status data from / to the sequencer DB "SEQ" (DB 725) - 1. Route interface	System
FB836	RC_ROUTE_MAT	Monitoring and execution of the material func-tions	System
FB850	RC_TR_ROUTE2_UNIT	Patches route request or status data from / to the sequencer DB "SEQ" (DB 725) - 2. Route interface	System

Block	Representation	Function	Use
FC800	RC_ENCODER	Encodes 32 separate Modebits to DWORD	User
FC801	RC_DECODER	Decode a DWORD into 32 bits	User
FC808	RC_IF_SENSOR	Interface between a sensor and route control	User
FC809	RC_IF_CONDUCT	Same as RC_IF_SENSOR, but with integrated limit value functionality in order to "digitize" a floating point actual value into four steps	User
FC810	RC_IF_PE	Interface between a setpoint parameter element and route control	User
FC811	RC_TG34_03	Internal function to send telegram 34 to the route control servers	System
FC812	RC_ROUTE_CE_ERR	Functions to send control element error messages to OS	System
FC813	RC_ROUTE_SE_ERR	Functions to send sensor element error messages to OS	System
FC817	RC_IF_SE_COMMON	User sensor element interface	User
FC818	RC_IF_SE_BM_SIG	Data exchange between RCS SEs and sensor signals	System
FC819	RC_IF_CE_BM_ICM	Data exchange between RCS CEs and ICMs	System
FC820	RC_CALL	Route control call interface	User (<V7) System (V7)
FC822	RC_ROUTE_PE_DGRAM	Sends datagram to OS if PE value changes	System
FC823	RC_UPD_CESEPE	If "boFaceplate" is set by the OS, this signal will be copied into the data records of the CE/SE/PE elements of the custom Route list	System
FC824	RC_UPD_CESEPE_EX	Creates the Update Requests (telegram 31) of CE/SE/PE elements	System
FC825	RC_LE_DGRAMM	Sends telegrams from the Link elements to the OS when the MAT-ID changes	System
FC826	RC_XFER_LE	Assembles LE Data records	System
FC828	RC_DB_CREATE	Automatically expands Route DBs if necessary	System
FC829	RC_XFER_MON_FLT	Patches the timeout setpoints from RC_ROUTE_CM instance-DB to the Route list	System
FC830	RC_ROUTE_CM_FC	Patches the inputs/outputs of a Route to the RCS system	User
FC831	RC_S5_S7_CALL	Calls further blocks (internally)	User
FC832	RC_ERR_FC	Returns error information relating to route	System
FC833	RC_ORDER	Alternative to FC 830; patches the constant data of a route to the RCS- system	User
FC834	RC_CTRL	Alternative to FC 830; patches the dynamic data of a route to the RCS-system	User
FC835	RC_EXT_PE	Alternative to FC 830; patches external PE data of a route to the RCS- system	User
FC836	RC_MAT	User interface for material functions	User
FC837	RC_HANDSHAKE	(Internal)	System
FC850	RC_S5_S7_XC_DB	(Internal)	System

17.2 User interface block

17.2.1 Control of the routes

17.2.1.1 Methods for control

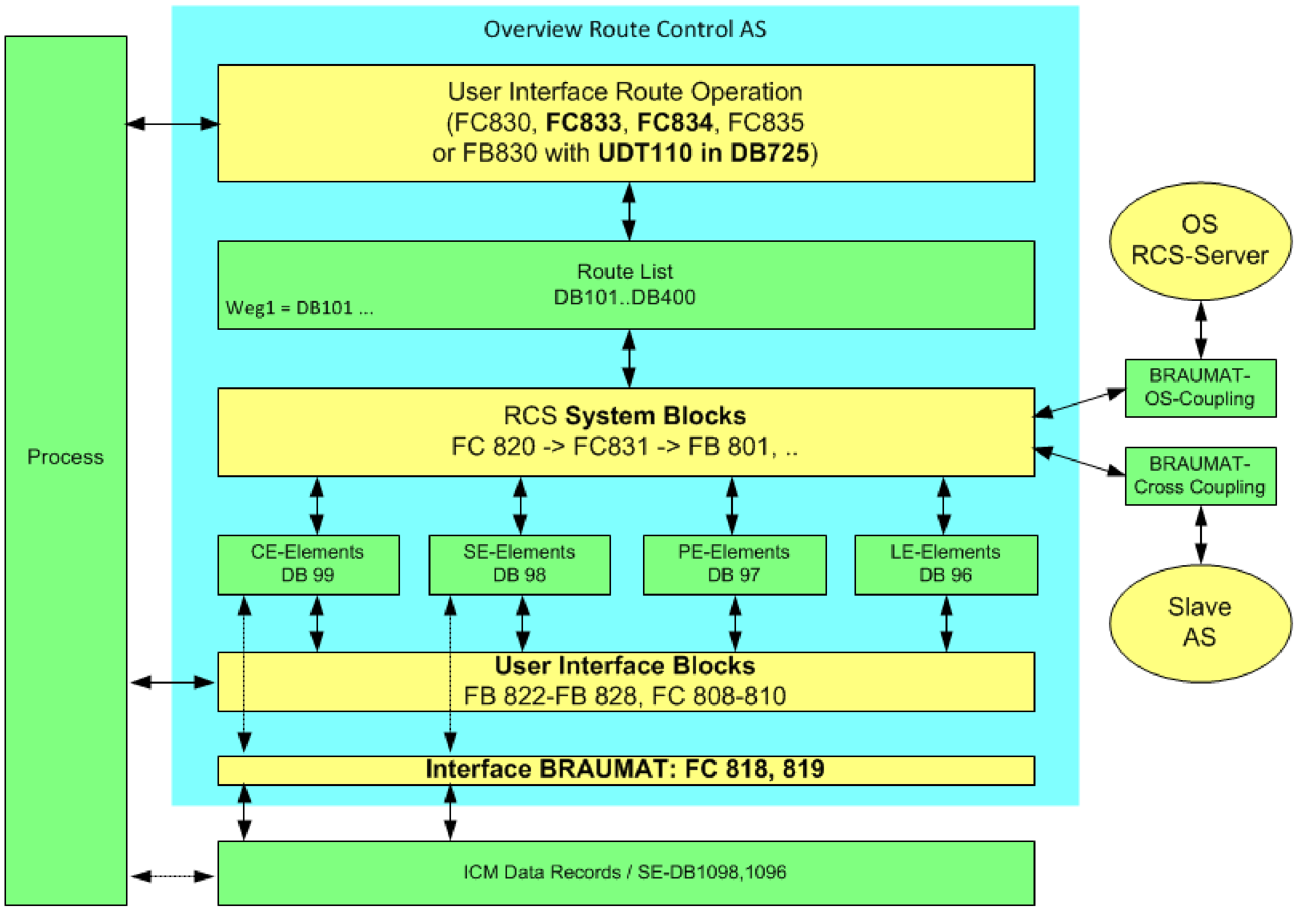
Basics

In the AS user program, the configuration engineer has the option of working with individual routes (requesting / starting / stopping routes) and to activate /deactivate the elements referenced in the route using the modes.

The resulting control of the CE elements (motor, valve) leads to the control of the corresponding ICM instances by the route control.

The second task of the AS is to implement the actual route control itself; in other words, to process the various route lists and to control the route and its elements according to the current status of the route and the control information and to trigger the required communication with the OS server or the partner AS. The communication itself is handled by the AS system blocks.

The following graphic shows the architecture and interfaces of the RCS system blocks in the automation system:



- The following section explains the possible variants of the user interface for route control in greater detail (upper yellow block).
- The interfaces to the process (lower yellow block) are explained in the section Standard interfaces (Page 1155).
- Calling the RCS system blocks (middle yellow block) is explained in the section System block Calls (Page 1159).

User interface for controlling a route

Each route used must be called via one of the 3 user interfaces in the user program (typically EOP-FC) of the master AS. This user interface has the following tasks:

- Initiating and feedback of the route request
- Controlling the route (start/stop route, control/query elements using mode bits)
- Interface for various route parameters
  (setting of setpoints for example for error monitoring; monitoring fault time etc.)
- Diagnostics of the current route status

Depending on the engineering environment, the following three alternatives are available for controlling a route as master in the AS (= master route function):

1. RC_ROUTE_CM_FC (FC 830)
   FC 830 contains the entire interface for controlling/checking a master route. For more details on this block, refer to the following section.
2. RC_ORDER and RC_CTRL (FC 833 and FC 834)
   Due to the large number of input/output parameters, online diagnostics on the AS is made more difficult and for this reason as an alternative to FC 830 it is possible to use a combination of the blocks FC 833 and FC 834. These blocks contain only the static part (RC_ORDER) as user interface or the dynamic part (RC_CTRL) of the route master function. For more details on these blocks, refer to the following section.
3. RCS interface in the sequencer dataset `"SEQ".uRCS, "SEQ".uRCS2`
   The following blocks are automatically called in the system that give users the option to implement the route master function for 2 routes over the header data areas in the DB725 sequencer data block.

- RC_TR_ROUTE_UNIT (FB 830) („uRCS“; UDT110) - 1. Route interface
- RC_TR_ROUTE2_UNIT (FB 850) („uRCS2“; UDT110) - 2. Route interface
  Without special block calls, this allows the user to control a maximum of two routes per unit simply by changing the data in the data area. The data exchange between the recipe control and route control is implemented by FB 830/FB 850 internally in the system. For more details on this interface, refer to the section RCS interface (Page 661).

Control of special functions

Separate user interfaces are available for the following special functions:

- External parameter elements RC_EXT_PE (FC 835)
  The seldom required external parameter elements, if they exist, are patched from/to the root list by the FC 835 block.
- Extended material function RC_MAT (FC 836)
  To store the transported material in the link elements of the route, FC 836 must be called in the user program. If the unit Interface „uRCS“ UDT 110 is used, this call is omitted because the material interface is already integrated there.

17.2.1.2 RC_ROUTE_CM_FC (FC830)

Description

This block routes the inputs/outputs of a route to the RCS system. The block must be called up for each route once in one of the following blocks:

- either in the OB1 cycle
  BmUsrCycleBeginFB (FB1220)
- or in the OB35 cycle
  BmUsrSeqxxx (FB1001 – FB1128), or BmUsrEopxxxx (FC1001 – FC2999)

As this block has a lot of input/output parameters that are not required for all application cases, you have the alternative option of replacing the FC830 call with a FC833 or FC834 call and, if required, also with the FC835 call.

Monitoring Time

For each of the 32 modes there is an on and off monitoring time (MON_TMxx , xx stands for the corresponding mode). The Monitoring Time starts with the rising edge of ModeXX and runs to the rising edge of QmodeXX (:= Mode condition is met, all line elements return a corresponding feedback signal). The Monitoring Time is also started with a falling edge of ModeXX and runs to the falling edge of QmodeXX (:= Qmode is not met). Hence, the function monitors the on/off time of a route mode.

Note

Important for ICMs:

If ICMs are configured with on/off delays, this additional delay time is ignored by the RCS. The user must take this into account when configuring the monitoring time.

The following must therefore apply:
RCS MON_TMxx > MAX((ESG x (MON_TM + ESG_SE))
The monitoring time to be set must be greater than the sum of runtime plus on delay of an ICM used in this mode.

Fault Time

All of the 32 modes are individually assigned a fault time(FLT_TMxx).
In contrast to the monitoring time, the fault time monitors the stable of the on/off situation. If the mode was activated and has been reached once, a CE response error would start this operating monitoring time. This feature is therefore primarily intended to be used for suppressing line disturbances caused, for example, by the "flutter" of valves.

Inputs

Parameters	Description	Data type
`ID`	**Number of the route (1 ... 300, 0 = dynamic route ID assignment)** The number must be unique within a system. It can be assigned only once and is also not permitted to be used a second time in other participating controllers of the system. Therefore the Route list is occupied in the whole plant.	Int
`INTERN`	Not currently used	Bool

Parameters	Description	Data type
`FUNC_ID`	**Function identification** Normally this parameter is equal to the Route list number. Only if two Route lists want to access to the same elements, must the same FUNC_IDs be allocated. Manually assigned IDs should be above those assigned by the system FUNC_IDs, in other words > 300. This makes diagnostics at runtime easier because it can be recognized immediately whether a system function identifier or a user function identifier is involved. **Example:** Material is taken from one tank via the same valves and is transported to different destinations. If the FUNC_ID of the Routes is not the same, the request is rejected by the OS.	Int
`MON_TIME`	**Default On/Off delay monitoring setpoint time [sec]** The default time is used when no time was explicitly set for a mode.	Int
`FLT_TIME`	**Default operating monitoring setpoint [sec]** The default time is used when no time was explicitly set for a mode.	Int
`REQ`	**Route request** With the signal TRUE , the Route is requested from the OS. This route is only released if REQ = FALSE and ON = FALSE.	Bool
`ON`	**Route On** • With the signal TRUE , the selected Route is started. If it was not loaded with REQ , the route is requested from the OS before the start. • With the signal FALSE , the selected Route is switched off. • With the signal FALSE for ON as well as for REQ , the route for the RCS server OS is released again.	Bool
`HOLD`	**Hold** Signal TRUE causes a quick stop, i.e. switching off the route without any configured Off delay.	Bool
`MODE`	**Modes 1 - 32** A maximum of 32 different modes are possible per route. By setting and resetting the individual bits, the user can define different modes of operation for the route. This 32 bit value is an OR logic operation of 32 binary signals with each binary signal standing for a mode.	DWord
`SOURCE`	**Source of the Route** Starting point of the Route. According to the Location ID from the RCS Offline configuration.	DInt

`VIA_1`
`...`
`VIA_10`
	Route via fixed route point
	The user can define via which fixed route points the route must travel. According to the Location ID from the RCS Offline configuration.
		DInt

`DEST`	**Destination of the Route** End point of the Route. According to the Location ID from the RCS Offline configuration.	Dint

Parame-ters	Description	Data type
MODE_TBL	**Mode Table ID** According to the Mode Table ID from the RCS offline configuration. In different Mode Tables different combinations of modes can be defined for a route. This corresponds to a different behavior dependent on the selected Mode Table. **Example** The route with the same source and destination has different modes in the Mode Table "Production" compared with those in the Mode Table "Cleaning".	DInt
MATERIAL	**Material** Current product number. When changing the material, QMAT_ERR is created if the next material is not permissible. Value range: 0..1024, 0 means: Transport path does not contain any material in the sense of the material sequence check. See also RC_MAT (FC 836).	DInt
RESERVE	Not currently used	Bool
IGN_ERR	**Ignore error** With signal 1, the route elements are not terminated in the case of an error (on/off monitoring/operation monitoring).	Bool
CE_FIELD	Not currently used This input will receive the same functionality in later Versions as 'SOLID' in FB 800.	Bool
ACK	**Acknowledgment** Errors relating to the On/Off monitoring and the operating monitoring time can be acknowledged. **Note:** RCS works internally with the rising edge of an acknowledgment as if the mode was switched on new. Therefore an artificial edge of Mode On or Off is created internally . This means that in case of error "Operating monitoring time", it is not the operating monitoring time that is started again, but the On/Off monitoring time after response and further errors.	Bool
TA	Unit (1..128) in which the route is processed.	Int

Outputs

Parameters	Description	Data type
QRET_VAL	Return value • 8FF2 route data block doesn't exist • 8FF4 ID out of range • 8FF8 RC_ROUTE_CM is running on slave route return value by function block "RC_ROUTE" • 8000 OS was not able to send the route list to all PLCs • 8001 Route request time out, OS didn't send the route list in time • 8002 No route found for the given source/via/dest. values • 8003 Control element(s) or PR(s) used by another route • 8004 Location error • 8005 Error in route list: number of elements or record length wrong • 80FF Route with this material not allowed • 8011 route data block doesn't exist • 8012 function block "RC_ROUTE" isn't called in OB 1 • 8013 function state isn't defined • 8014 error by call of function block "ATTRIB_PTR_FC" • 8015 receive buffer doesn't exist return value by function block "RC_ROUTE_XC_REC" • active route • 8021 route identification in receive buffer is wrong • 8022 monitoring receive time is ended • passive route • 8025 receive buffer doesn't exist • 8026 no data in receive buffer • 8027 route identification in receive buffer is wrong • 8028 monitoring receive time is finished return value by function block "RC_ROUTE_RCE_ON" • 8031 error by call of function block "ATTRIB_PTR_FC" (RCE)	Word
QREQ_RC	**Response of the state of the Route request** Returned value of the block "RC_ROUTE_CM"	Word
	• 0000 | Idling	
	• 0002 | Order route request running	
	• 0004 | Closed without any errors, Route list received	
	• 8000 | Transmission error	
	• 8001 | OS has not responded within the monitoring time	
	• 8002 | Route is not defined in OS	
	• 8008 | Route request could not be sent to OS	
	• 80FF | Material not allowed	

Parameters	Description	Data type
QREQ	**Response to Route request (REQ = TRUE or ON = TRUE)** • FALSE: No error • TRUE: Error	Bool
QON	**Response to Route ON (ON = TRUE)** • FALSE: Modes cannot be activated • TRUE: Modes can now be activated	Bool
QHOLD	**Status Hold** • FALSE:Route is not on hold • TRUE: Route held. All CE elements are reset. You reach the status Hold by the input parameter Hold = TRUE or by OS RCS online operation.	Bool
QMODE_OF	**Rest position mode 1 - 32** One bit per Mode. That means QMODE_OF1 to QMODE_OF32. • TRUE: All active control elements (CE) of the corresponding mode are in rest position, i.e. are set to off (setpoint value) and give back the feedback value "is off".	DWord
QMODE_AC	**Acknowledgment Mode 1 – 32 fulfilled** One bit per Mode. That means QMODE_AC1 to QMODE_AC32. • FALSE: ModeXX Conditions not met • TRUE: ModeXX Conditions met The evaluation of the mode response does not depend on whether the corresponding mode is activated or not. **Example:** Only SEs are built with Mode X. In order to query the mode (= evaluation QMODE_ACX) the mode does not need to be reset, i.e. it is not necessary to set MODEX to TRUE . **Tip:** FC 801 (RC_Decoder) can be included afterwards, if necessary, in order to achieve a conversion of DWORD in 32 individual bits.	DWord
QMODE_ER	**Response error mode 1 – 32** One bit per Mode. That means QMODE_ER1 to QMODE_ER32.	DWord
QMON_TIM	**Default On/Off delay monitoring actual time [sec]** There is only one internal MON_TIM for each mode. The lowest (= the most critical) is displayed on QMON_TIM .	Int
QFLT_TIM	**Operating monitoring actual time [sec]** There is only one internal FLT_TIM for each mode. The lowest (= the most critical) is displayed on QFLT_TIM .	Int
QMON_TOU	**Group error On/Off monitoring time expired mode 1 - 32** Or connections from QMON_ERR1 to QMON_ERR32.	Bool
QFLT_TOU	**Group error operating monitoring time expired mode 1 - 32** Or connections from QFLT_ERR1 to QFLT_ERR32.	Bool
QERR	**Group error of the Route**	Bool

Parameters	Description	Data type
QMAT_ERR	**Material error** The following material may not be requested from the RCS offline configu-ration due to the material transitions table. 0: No error 1: Material error, material does not tolerate previous material	Bool
QACTIV	**Status of the Route** 1 : Route is requested and active. 0 : The route has been terminated completely and all elements are released again. Only then is it useful to request a new Route.	Bool

Inputs/Outputs

Parameters	Description
DYN_ID	**Dynamic route identifier** The RCS server automatically assigns a free ROUTE ID when there is a route request.

See also

Dynamic Route ID allocation (Page 1084)

17.2.1.3 RC_ORDER (FC833)

Description

This block can be used with the FC834 as an alternative to the FC830 call. The reason for this distinction is that the FC833 allocates the constant data (locations, MODE_TBL, ...), and the FC834 allocates the dynamic data (REQ, ON, MODE, ...). The seldom required external parameter elements, if they exist, are patched from/to the root list by calling FC835 .

Inputs

Parameters	Description	Data type
ROUTE_ID	**Route list numberr (1 ... 300, 0 = dynamic route ID assignment)** The number of the route list must be unique in a plant. This number can be assigned only once and must not be used a second time in other controllers operating in the plant. Therefore the route list is occupied in the whole plant.	Int
FUNC_ID	**Function identification** Normally this parameter is equal to the route list number. Only if two route lists want to access to the same elements, the same FUNC_IDs be assigned in both. Manually assigned IDs should be above those assigned by the system FUNC_IDs, in other words > 300. This makes diagnostics at runtime easier because it can be recognized immediately whether a system function identifier or a user function identifier is involved. **Example:** Material is taken from one tank via the same valves and is transported to different destinations. If the FUNC_ID of the Routes is not the same, the request is rejected by the OS.	Int
SOURCE	**Source of the route** Starting point of the route. According to the Location ID from the RCS Offline configuration.	DInt
VIA_1 bis VIA_10	**Route via fixed route point** The user can define via which fixed route points the route must travel. According to the Location ID from the RCS Offline configuration.	DInt
DEST	**Destination of the route** End point of the Route. According to the Location ID from the der RCS Offline configuration.	DInt
MODE_TBL	**Mode Table ID** According to the Mode Table ID from the RCS Offline configuration. In different Mode Tables different combinations of modes can be defined for a route. This corresponds to a different behavior dependent on the selected Mode Table. **Example** The route with the same source and destination has different modes in the Mode Table "Production" compared with those in the Mode Table "Cleaning".	DInt
IGN_ERR	**Ignore error** With signal 1, the route elements are not terminated in the case of an error (on/off monitoring/operation monitoring).	Bool
SOLID	Not currently used (Route for the solid material transport)	Bool
TA	Unit (1..128) in which the route is processed.	Int

Outputs

Parameters	Description	Data type
iRetVAL	0 = OK, <0 =error • -28680 (W#16#8FF8) Attempt to make operator input in slave route • -28684 (W#16#8FF4) Gen. error due to parameter 1: ID is outside the valid range • -28686 (W#16#8FF2) Route DB not available • -28687 (W#16#8FF1) Route set to internal	Int
BoError	0 = OK, 1 = error	Bool

Inputs/Outputs

Parameters	Description	Data type
DYN_ROUTE_ID	**Dynamic route identifier** The RCS server automatically assigns a free ROUTE ID when there is a route request.	Int

See also

Feedback values from RCS blocks (Page 1160)

17.2.1.4 RC_CTRL (FC834)

Description

This block can be used with the FC833 as an alternative to the FC830 call. The reason for this distinction is that the FC833 allocates the constant data (locations, MODE_TBL, ...), and the FC834 allocates the dynamic data (REQ, ON, MODE, ...). The seldom required external parameter elements, if they exist, are patched from/to the root list by calling FC835 .

Inputs

Parameters	Description	Data type
ROUTE_ID	**Route list numberr (1 ... 300, 0 = dynamic route ID assignment)** The number of the route list must be unique in a plant. This number can be assigned only once and must not be used a second time in other controllers operating in the plant. Therefore the route list is occupied in the whole plant.	Int
REQ	**Route request** With the signal TRUE , the route is requested from the OS. This route is only released if REQ = FALSE and ON = FALSE.	Bool

Parameters	Description	Data type
ON	**Route on** • With the signal TRUE , the selected route is started. If it was not loaded with REQ , the route is requested from the OS before the start. • With the signal FALSE , the selected route is switched off. • With the signal FALSE for ON as well as for REQ , the route for the RCS server OS is released again.	
HOLD	**Hold** Signal TRUE causes a quick stop, i.e. switching off the route without any configured Off delay.	Bool
MODE	**Modes 1 - 32** A maximum of 32 different modes are possible per route. By setting and resetting the individual bits, the user can define different modes of operation for the route. This 32 bit value is an OR logic operation of 32 binary signals with each binary signal standing for a mode.	DWord
MATERIAL	**Material** Current product number. When changing the material, QMAT_ERR is created if the next material is not permissible. Value range: 0..1024, 0 means: Transport path does not contain any material in the sense of the material sequence check. See also RC_MAT (FC 836).	DInt
SET_MAT	**Adopt next material** This input is now ineffective (reserved) – see RC_MAT (FC836)	Bool
ACK	**Acknowledgment** Errors relating to the On/Off monitoring and the operating monitoring time can be acknowledged. **Note:** RCS works internally with the rising edge of an acknowledgment as if the mode was switched on new. Therefore an artificial edge of Mode On or Off is created internally . This means that in case of error "Operating monitoring time", it is not the operating monitoring time that is started again, but the On/Off monitoring time after response and further errors.	Bool
GETXPE	Taking external parameter element actual values	Bool

Outputs

Parameters	Description	Data type
`iRouteState`	Status of the route (FB801) • -1 Initial status • 0 Ready • 9 Send batch name • 10 Send route request • 11 Wait for route list • 12 Route list loaded, initializations • 13 Wait for order data via cross coupling • 14 Route editing possible • 15 Send start frame • 101 Control rote • 102 Terminate route • 103 Route is terminated • 112 Terminate route, error • 113 Route is terminated, error • 200 Route is terminated due to cross coupling error	Int
`QINTERN`	Not relevant for users Block is used internally / No reconnection of the inputs	Bool
`QRET_VAL`	Return value • 8FF8 Attempt to make operator input in slave route • 8FF4 Gen. error due to parameter 1: ID is outside the valid range • 8FF2 Route DB not available • 8FF1 Route set to internal	Word
`QDIAG`	Diagnosis Information • 8021 Route identifier of slave route wrong • 8027 Route identifier in receive buffer wrong • 8066 Slave route: Error in XC_JOB_USER_IF_FC • 8112 Error in DB length change • 8161 Error in XC_JOB_USER_IF_FC Order not sent • 8182 XC order could not be sent	Word

Parameters	Description	Data type
QREQ_RC	Response of the state of the Route request Return value of the block "RC_ROUTE_CM" • 0000 Idle • 0002 Order route request running • 0004 Ended without errors, route list received • 8000 Transfer error • 8001 OS has not responded within the monitoring time • 8002 Route is not defined in the OS • 8008 Route request could not be sent to the OS • 80FF Material not permitted	Word
QREQ	Response to Route request (REQ = TRUE or ON = TRUE) • FALSE: No error • TRUE: Error	Bool
QON	Response to route ON (ON = TRUE) • FALSE: Modes cannot be activated • TRUE: Modes can now be activated	Bool
QHOLD	Status Hold • FALSE: Route is not on hold • TRUE: Route held. All CE elements are reset. You reach the status Hold by the input parameter Hold = TRUE or by OS RCS online operation.	Bool
QMODE_AC	Acknowledgment Mode 1 – 32 fulfilled One bit per Mode. That means QMODE_AC1 to QMODE_AC32. Mode bit x = 1 : Conditions met Mode bit x = 0 : Conditions not met The evaluation of the mode response does not depend on whether the corresponding mode is activated or not. **Example:** Only SEs are built with Mode X. In order to query the mode (= evaluation QMODE_ACX) the mode does not need to be reset, i.e. it is not necessary to set MODEX to TRUE . **Tip:** FC 801 (RC_Decoder) can be included afterwards, if necessary, in order to achieve a conversion of DWORD in 32 individual bits.	DWord
QRESTPOS	Rest position mode 1 - 32 All CEs used actively in a mode are in rest position Mode bit x = 1 : Rest position is satisfied. Mode bit x = 0 : Rest position is not satisfied.	DWord
QRPOSERR	Rest position error mode 1 - 32 Not all CEs used actively are in rest position Mode bit x = 1 : Rest position error is satisfied (pending) Mode bit x = 0 : No rest position error	DWord

Parameters	Description	Data type
QMON_ERR	On/Off monitoring time expired mode 1 - 32: All elements of a mode have reached their set position in good time Mode bit x = 1 : Set position not reached in good time. Mode bit x = 0 : All elements have adopted their set position in good time.	DWord
QFLT_ERR	Operation monitoring time expired mode 1 - 32 After adopting their set positions elements of a mode have left the set position for a time longer than the operation monitoring time Mode bit x = 1 : End position was exited. Mode bit x = 0 : All elements are still in the set position (activated status).	DWord
QMON_TOU	Group error On/Off monitoring time expired 0: No error, time has not yet elapsed 1: Time has elapsed.	Bool
QFLT_TOU	Group error operation monitoring time expired 0: No error, time has not yet elapsed 1: Time has elapsed.	Bool
QERR	Group error of the Route 0: No error 1: Group error in route block; Further information in Response QRET_VAL	Bool
QMAT_ERR	Material error The selected material may not be requested from the RCS offline configuration due to the material transitions table. 0: No error 1: Material error, material does not tolerate previous material	Bool
QMAT_OK	Next material OK 1: No error 0: Material error, material does not tolerate previous material	Bool
QACTIVE	Status of the Route 1 : Route is requested and active. 0 : The route has been terminated completely and all elements are released again. Only then is it useful to request a new route.	Bool
GETXSTS	Status of actual values of the external parameter elements 0: Values are invalid 1: Values are valid	Byte
GETXDIAG	Diagnostics information on invalid actual values of external PEs: • 8180 No external PE • 8181 No external PEs in slaves • 8182 XCJob	Word

Parameters	Description	Data type
iRetVal	0 = OK, <0 =error • -28680 (W#16#8FF8) Attempt to make operator input in slave route • -28684 (W#16#8FF4) Gen. error due to parameter 1: ID is outside the valid range • -28686 (W#16#8FF2) Route DB not available • -28687 (W#16#8FF1) Route set to internal	Int
boError	0 = OK, 1 = error	Bool

17.2.1.5 RC_EXT_PE (FC835)

Description

This block serves to allocate the setpoint and actual values for the external parameter elements for the route DB

Note

Allocating external parameter elements

The allocation of external parameter elements from the route DB to the "RC_PE_FIELD" (DB97) must be explicitly triggered by the user via the function "RC_IF_PE" (FC810). It is reasonable to perform this in the user program directly after calling the function "RC_EXT_PE" (FC835).

Inputs

Parameters	Description	Data type
ROUTE_ID	**Route list numberr (1 ... 300, 0 = dynamic route ID assignment)** The number of the route list must be unique in a plant. This number can be assigned only once and must not be used a second time in other controllers operating in the plant. Therefore the route list is occupied in the whole plant.	Int
EXT_1_I to EXT_24_I	Setpoint 1-24 for external parameter elements	Real

Outputs

Parameters	Description	Data type
EXT_1_O to EXT_24_O	Actual value 1-24 for external parameter elements	Real
QRET_VAL	Return value • 8FF4 ID outside limits • 8FF2 Route DB not available	Word
ERROR	0 = OK, 1 = error	Bool

17.2.1.6 RC_MAT (FC836)

Description

This block is the user interface block for the execution of the material functions for a Route.

- Access::
  The call of the block RC_MAT occurs in connection with the interface blocks of the route control (FC830 / RC_ROUTE_CM_FC and/or its derivatives FC833 / RC_ORDER and FC834 / RC_CTRL) in the sequence FB or in the EOPs (OB1 or OB35)
- Function:
  - The block RC_MAT carries the current Material ID, the requested material function (cf. section "Function") and the timeout setpoint for the monitoring of the material function in the Route DB.
  - With increasing flank of the input SETMAT , the requested function is carried out and the monitoring time is started. At the exit of the block the successful execution of the function and/or an occurred error is reported.
  - With the input MATCHK_DIS , the customer can prevent an inspection of the Material ID by the OS.
  - With increasing flank of the input ROLL_BCK , the last material input in the LE element will be undone. In conjunction with the current material function the execution of the roll-back function refers to all link elements of the route or only on the link elements with RouteID. entered

Inputs

Parameters	Meaning
`ID`	**Value range:** • 1..max Material function can be carried out if the Route DB exists and the Route is in the state "Ready for Start" at least. • 0 Material function cannot be carried out Value < 0 or > max -> Error message
`MATERIAL`	**Identification for Material** Requested product number **Note** **The value of the material ID must be identical to the MaterialID in FC 830 ... Otherwise, the route DB may contain inconsistent data.** Unfortunately this could not be avoided, as for reasons of backwards compatibility with projects in which the FC 836 is not used, the Material ID at the other interface blocks for the Route request is still required.

<table>
<tr><th>Parameters</th><th colspan="5">Meaning</th></tr>
<tr><td rowspan="9">SWITCH</td><td colspan="5">Function identifier
Determines the material function which is carried out:</td></tr>
<tr><td>SWITCH #</td><td>Material Check</td><td colspan="3">Input of the Link element</td></tr>
<tr><td></td><td></td><td>RouteID</td><td>FuncID</td><td>MaterialID</td></tr>
<tr><td>0 = Reset all RouteID</td><td>Yes *</td><td>0</td><td>0</td><td>--</td></tr>
<tr><td>1 = Set all RouteID</td><td>Yes *</td><td>= RouteID of the Route</td><td>= RouteID of the Route</td><td>--</td></tr>
<tr><td>2 = Partial Reset RouteID</td><td>No</td><td>0</td><td>0</td><td>--</td></tr>
<tr><td>3 =Set MaterialID</td><td>Yes *</td><td>--</td><td>--</td><td>= MaterialID of the route</td></tr>
<tr><td>4 =Force Set MaterialID</td><td>Yes *</td><td>--</td><td>--</td><td>= MaterialID of the route</td></tr>
<tr><td colspan="5">* Material check YES: The input into the LE elements is carried out only according to positive response of the material check; Exception: Input MATCHK_DIS =1; (Material check is deactivated).
The functions 0, 1 and 4 refer in each case to all LE elements of the Route; the functions 2 and 3 refer only to the LE elements, in which the entered RouteID corresponds to the RouteID of the Route.</td></tr>
<tr><td>SETMAT</td><td colspan="5">Starts the material function
If this parameter switches from 0 to 1, the function entered in the Parameter SWITCH will be executed.
SWITCH can be in the area 0..4 when SETMAT is called.</td></tr>
<tr><td>ROLLBCK</td><td colspan="5">Withdraw the last Material ID input
If this parameter changes from 0 to 1, the last entered MaterialID will be set up in the LE elements again.
SWITCH can be with call of ROLBCK in the area 3..4 and refers either to all LE blocks or only to the Link elements whose entered RouteID corresponds with the RouteID of the Route.</td></tr>
<tr><td>MATCHK_DIS</td><td colspan="5">Disabling material check
If this parameter is set to 1 no material check will be triggered by the OS; instead of this the material inputs are carried out with the material function immediately.</td></tr>
<tr><td>TIME_VAL</td><td colspan="5">Monitoring setpoint [sec]</td></tr>
</table>

Outputs

Parameters	Meaning
QCHK_OK	Positive result material check The requested material can be carried by the material that is entered on the Link elements of the Route according to the configured Material Transition Table with this Route.
QCHK_ERR	Negative result material check The requested material cannot be carried by the material that is entered on the Link elements of the Route according to the configured Material Transition Table with this Route.

Parameters	Meaning	
QCHK_OK and QCHK_ERR at the same time = 0	Material check is active If both outputs QCHK_OK and QCHK_ERR are 0 at the same time and the input SETMAT is 1, a material check will be executed by the OS.	
QXFER_OK	Positive result material input The requested material function was executed successfully for the selected Link elements.	
QXFER_ERR	Negative result material input The requested material function was not executed successfully for the selected Link elements.	
QXFER_OK and QFER_ERR at the same time = 0	Material input is active If both outputs QFER_OK and QXFER_ERR are 0 at the same time and the input SETMAT is 1, a material input will be executed.	
iRetVal	Return value of the last material function Result of the material function as Hex Diagnosis word:	
	• 0000	Material input executed without errors.
	• 0001	Material check active by OS
	• 0002	Material input active in Link elements
	• 0003	Material input in Master AS finished; Material input Slave is still active
	• 8361	Route DB is not available
	• 8362	RouteID is outside of the available area
	• 8363	Timeout Material check by OS
	• 8364	Timeout Material input in Link elements
	• 8365	SWITCH is outside the permitted range (0..4)
	• 8366	Route is not Master (call FC RC_MAT in slave PCU)
	• 8367	MATERIAL is outside the permitted range (0..1024)
	• 80FF	Continuation message is not allowed

17.2.1.7 RC_ENCODER (FC800)

Name

FC800

Description

This block groups 32 individual mode bits in a shared DWORD . This gives the project engineer the possibility of controlling the mode of a route bit-by-bit (= one input bit for each mode bit).

Inputs

Parameters	Meaning
MODE_01	Set bit mode 1 active / inactive
...	
MODE_32	Set bit mode 32 active / inactive

Outputs

Parameters	Meaning
QMODE	MODE_01 ... MODE_32 encoded according to DWORD

17.2.1.8 RC_DECODER (FC801)

Name

FC801

Description

This block is the counterpart to "RC_Encoder" (FC800).
The project engineer can use it to decode bits which are assembled in a DWORD into 32 individual bits.

Inputs

Parameters	Meaning
MODE	MODE_01 ... MODE_32 encoded according to DWORD

Outputs

Parameters	Meaning
QMODE_01	Set bit mode 1 active / inactive
...	
QMODE_32	Set bit mode 32 active / inactive

17.2.2 Process interface of the route elements

17.2.2.1 Standard interfaces

Linking the route elements to the process

The following interfaces are provided for linking the RCS elements to the process:

- CE interface
  In a process cell project, the system transfers the control of the CE elements to the data records of the ICMs. The interface of the elements to the process is handled by the system (refer to the section ICM - Individual Control Modules (ICM1 ICM2 ICM3 ICM4) (Page 269)). Only the CE types 1 and 5 (motor, valve) are transferred to the ICM datasets.
- SE interface
  The two following data blocks make up the user interface for the SE elements:
  - RC_SISTAR_SE (DB1098) with 1024 bits for the feedback of 1024 binary sensors
  - RC_SE_ERR_SISTAR (DB 1096) with 1024 bits for the corresponding error messages of these sensors

  Only SE type 1 (binary sensor) is transferred to the SE instance RC_SE_FIELD (DB98). For other SE types the following additional interface blocks should be used (for a description, see the next section):
  - RC_IF_SENSOR (FC 808)
  - RC_IF_CONDUCT (FC 809)
- PE interface
  The RC_IF_PE (FC810) block serves as the user interface between an external parameter element and the RCS. If applicable, the function transfers external actual values to the PE instance DB RC_PE_FIELD (DB97) and extracts certain signals from the DB and makes them available at block outputs for further processing.
- LE interface
  By assigning Link Elements to partial routes (→ Route LE) the LEs used represent the material transferred on the active route.
  The RC_IF_LE (FB828) block extracts the attributes of a Link Element from the LE instance DB RC_LE_FIELD (DB96) and makes them available for further processing at block outputs.

17.2.2.2 RC_IF_PE (FC810)

Description

This block is the interface between a setpoint parameter element and route control. The function transfers external parameter elements, if any, to the "RC_PE_FIELD" data block (DB97) and extracts the signals specified below from this to its own block outputs for further processing.

Inputs

`ID`	PE number
`FACTOR`	`QVALUE_D := RND(FACTOR*QVALUE_R)`
`DIS_ACTV`	Disables "Write actual value"
`ACT_VAL`	Actual value, e.g. EPAR_REAL
`DEF_VAL`	Initial value at inactive or controlled PEs
`EN_DEF`	Release for the initial value

Outputs

`QRET_VAL`	Return value
	• 8102 PE number is out of range (1-DB97.iAct)
	• 8103 External setpoint index is outside the range of values (1-24)
	• 8104 External setpoint route ID is out of range
`QVALUE_I`	INT (QVALUE_D)
`QVALUE_D`	RND(FACTOR*QVALUE_R)
`QVALUE_R`	Output value (REAL)
`QINDEX`	Index for external PE setpoint
`QEXT`	External setpoints • TRUE: setpoint comes from EXT_1-8_I of block "ROUTE_CM"
`QSUMMED`	Summation value
`QERR`	Error
`QBA_ID`	Batch number
`QROUTE`	Route ID of the route that is currently using this parameter element.
`QFUNC_ID`	Function ID of the Route
`QMODE_NO`	Mode Number (1-32)

17.2.2.3 RC_IF_SENSOR (FC808)

Name

RC_IF_SENSOR (FC808)

Description

This block is the interface between a sensor and the Route Control System. The function transfers the signals from a sensor element to the dataset of a sensor element in the "RCS_SE_FIELD" (DB98) data block.

Inputs

ID	SE number
FB_00	Feedback
ERROR	Error
DISABLED	Locked (e.g. by maintenance work) Partial routes with this element cannot be used by RCS, i.e. for a Route request the RCS server attempts to calculate the total route from Source to Destination via different Route nodes.

Outputs

QRET_VAL	Return value • 0801: SE number is out of range • 0803: Wrong SE type (only binary sensors are permitted)
QFB_00	Feedback
QERR	Error
QBA_ID	Batch number
QROUTE	Route ID of the route that currently uses this sensor element.
QFUNC_ID	Function ID of the Route

17.2.2.4 RC_IF_CONDUCT (FC809)

Description

Similar to "RC_IF_SENSOR" (FC808), but with an integrated limit value functionality to "digitize" a real actual value (e.g. measured value of a master value measurement) into four steps. I.e. in contrast to "RC_IF_SENSOR" which only provides a binary signal (e.g. manual flap open/closed), this function supplies 4 bits (e.g. for master value measurement: water, beer, acid, caustic).

Inputs

ID	SE number
LEVEL_1	Limiting value 1
LEVEL_2	Limiting value 2
LEVEL_3	Limiting value 3
LEVEL_4	Limiting value 4
HYS	Hysteresis
ACTUAL	Current actual value
ERROR	Error
DISABLED	Locked (e.g. by maintenance work) Partial routes with this element cannot be used by RCS, i.e. for a Route request the RCS server attempts to calculate the total route from Source to Destination via different Route nodes.

Outputs

QRET_VAL	Return value	
	• 8091 SE number is out of range	
	• 8093 SE type is not permitted (only type 2 is permitted: "Master value")	
QERR	Error	
QBA_ID	Batch number	
QROUTE	Route ID of the route that currently uses this sensor element.	
QFUNC_ID	Function ID of the Route	
QLEVEL1	exceeding limit value 1	
QLEVEL2	exceeding limit value 2	
QLEVEL3	exceeding limit value 3	
QLEVEL4	exceeding limit value 4	

17.2.2.5 RC_IF_LE (FB828)

Name

FB828

Description

This block returns the state of a Link Element (LE).

Inputs

Parameters	Meaning
ID	LE number

Outputs

Parameters	Meaning	
QRET_VAL	Return value	
	• 8281	LE ID is out of permitted range
QERR	Error	
QMAT_I	Material ID as INT	
QMAT_DI	Material ID as DINT	
QBA_ID	Batch number	
QROUTE	Route ID of the Route which is currently using this Link Element.	
QFUNC_ID	Function ID of the Route	

17.3 System block Calls

Basics

The Route Control System undertakes the processing of the route lists in the AS and thus the activation of elements and evaluation of feedback. The user must activate this system by calling „RC_CALL“ (FC 820) in the cyclic program (OB1) and in the 1 s interval (OB 32). This block in turn calls the other blocks of the route control that process the existing route control elements (RCEs) in a loop covering all route lists.

In accordance with the prescribed configuration and control of the mode bits by the user, the system controls the active elements and evaluates the feedback from all elements.

With active route control, the current states of the elements are transferred to the route list for the CEs and SEs, the feedback is formed for each individual mode. If the respective mode is addressed, the parameterized control is transferred from the RCEs to the corresponding elements for the CEs and PEs.

The embedding of the RCS runtime environment differs significantly between the AS versions:

PCU < V7

The following explicit FC820 ("RC_CALL") calls are necessary:

- in the interface blocks of the start-up OBs (FB1200, FB1201)
- in the interface block of the free cycle (FB1220 or FB1221)
- in the interface block of the 1s cyclic interrupt (FB1225)

Example:

```
CALL "RC_CALL"
  TR_CE_ICM := TRUE
```

The "TR_CE_ICM" parameter decides whether data routing between ICMs and CEs is to take place on the system side. This must always be set to "TRUE" here.

Note

Call of the RCS runtime system with PCU V7 or higher

The embedding of the RCS system blocks takes place automatically as long as RCS is identified as being active.

For this to occur, the system attribute "boRCSActive" = TRUE during project engineering in the "Parameterization" application / block class "RC_CNF" / Global data.

See also

AS Installation (Page 146)

17.3.1 Feedback values from RCS blocks

Here is a list of all possible error codes and measures for removing errors for the corresponding RCS blocks:

All value indications are in Hex format (W#16#8XXX)

Returned value of the block "RC_ROUTE" (FB801)

- **8000** Error when transferring a route list to the AS:
  In at least one of the automation systems on the path the route list could not be transferred.
  - Remedy: Abort Route request with REQ=0 and ON=0, repeat if necessary, and check why the transfer failed.
    Maybe the AS is in STOP or the connection to the RCS server OS has aborted.
- **8001** Timeout during route request.
  RCS server OS did not send the route list within 120 sec.
  - Remedy: Wait or abort with ON = 0 and REQ = 0.
- **8002** No free or available route for the specified Source/Via/Destination combination found.
  - Remedy: Check locations, then repeat Route request.
- **8003** Control element(s) or partial route(s) are already being used by a different route.
  - Remedy: Wait (> 60 sec) and request new or abort.
- **8004** Location error. At least one indication for Source, Via, Destination is false.
  - Remedy: Check the specified location in FB call parameter and in the RCS offline configuration,
    correct and request the route again.
- **8008** Route request message cannot be entered in the FIFO for sending to the RCS Server OS.
  - Remedy: Wait. RCS constantly repeats the sending of the message or abort route request with REQ = FALSE and ON = FALSE .
- **8010** Incorrect version (generation identifier) of the DB "RC_CFG" (DB100)
  - Remedy: Use the DB "RC_CFG" from the same S7 RCS library as the other blocks.
- **8013** Unknown state machine condition/Function status is not defined.
  - Remedy: Internal error, solved by RCS itself.

- **8015** Cross-coupling receive buffer does not exist.
  - Remedy: No user reaction necessary.
- **80FF** After a route request, the RCS server OS signals a material compatibility error, i.e. the specified new material is not allowed as
  the follow-on material.
  - Remedy: Change material identifier and request Route again.

Returned value of the block "RC_ROUTE_XC_REC" (FB802)

For an active Route (Master):

- **8021** Route: Wrong route ID in the receive buffer.
  At least one slave signals an incorrect route ID.
  The responses of these slaves are ignored.QDIAG remains saved
  until an error acknowledgement takes place.
  - Remedy: There is no user reaction necessary, or error response.
- **8022** For at least one slave the receive monitoring time ran out.
  - Remedy: Check whether all slave automation systems involved are in RUN and the cross-coupling connection to the master is working.

For a passive Route (Slave):

- **8027** Route ID in the receive buffer is wrong.
  For the sake of safety, the route is switched off.
- **8028** Monitoring receive time elapsed.
  For the sake of safety, the route will be switched off.

Returned value of the block "RC_ROUTE_RCE_ON" (FB803)

- **8030** CE (Control Element) device error.
  In other words, a control element (ICM) itself is disrupted.
  This is an external error, and does not come from RCS!
  Remedy: Check in RCS Online which elements return incorrect messages
  and clarify why this is happening.
- **8031** Monitoring time ran out. After
  switching on/off, the monitoring time elapsed for at least one mode, without the return messages of all elements being correct.
  With IGN_ERR = TRUE the route remains switched on, with IGN_ERR=0 it will be switched off.
  - Remedy: Check in RCS Online which elements return incorrect messages
    and clarify why this is happening.
    Expand, if necessary, the monitoring time of the corresponding modes.
    If the error is eliminated, this can be acknowledged with a positive edge at
    ACK or by OS operator input in RCS Online.
    With IGN_ERR = FALSE , the route is switched on again and the
    monitoring time is restarted.
    With IGN_ERR = TRUE , the error display disappears only if the error
    has been eliminated.

- **8032** Operating monitoring time ran out. A return message error occurred for at least one Mode after switching on/off successfully during operation and this did not disappear within the error monitoring time.
  with IGN_ERR = TRUE the route remains switched on; with IGN_ERR=0 it will be switched off.
  – Remedy: Check in RCS Online which elements return incorrect messages and clarify why this is the case.
    If the error is eliminated, this is acknowledged with a positive edge at ACK or by OS operator input RCS Online.
    With IGN_ERR = FALSE the route is switched on again and the On/Off monitoring time is started.
    With IGN_ERR = TRUE the error display disappears only if the error has been eliminated.
- **8033** Rest position monitoring time elapsed.
  For at least one mode, after switching off, some elements did not signal the rest position within the monitoring time.
  – Remedy: In RCS Online, check which elements are returning incorrect feedback and clarify why this is the case.
    If the error has been eliminated, this must be acknowledged by a positive edge at ACK or by OS operator input in RCS Online.
- **8034** The DB for the control elements ("RC_CE_FIELD", DB99) is not available.
  – Remedy: Check the AS project and reload the DB.
- **8035** Error when calling the block "ATTRIB_PTR_FC".
  The internal data structures in the DB "RC_CE_FIELD" (DB99) are wrong.
  – Remedy: If the process allows, the DB must be reloaded from the AS offline project .
- **8036** The DB for the sensor elements ("RC_SE_FIELD", DB98) is not available.
  – Remedy: Check the AS project and reload the DB .
- **8037** Error when calling the block "ATTRIB_PTR_FC" (SE).
  The internal data structures in the DB "RC_SE_FIELD" (DB98) are wrong.
  – Remedy: If the process allows, the DB must be reloaded from the AS offline project .
- **8038** The DB for the parameter elements ("RC_PE_FIELD", DB97) is not available.
  – Remedy: Check the AS project and reload the DB .
- **8039** Error when calling the block "ATTRIB_PTR_FC"
  The internal data structures in the DB "RC_PE_FIELD" (DB97) are wrong.
  – Remedy: If the process allows, the DB must be reloaded from the AS offline project .

- **803A** The DB for the link elements ("RC_LE_FIELD", DB96) is not available.
  - Remedy: Check the AS project and reload the DB.
- **803B** Error when calling the block "ATTRIB_PTR_FC"
  The internal data structures in the DB "RC_LE_FIELD" (DB96) are wrong.
  - Remedy: If the process allows, the DB must be reloaded from the AS offline project .

Returned value of the block "RC_ROUTE_RCE_OFF" (FB804)

- **8044** The DB "RC_CE_FIELD" (DB 99 , DB for the control elements) is not available.
  - Remedy: Check the AS project and reload the DB .
- **8045** Error when calling the block "ATTRIB_PTR_FC".
  The internal data structures in the DB "RC_CE_FIELD" (DB99) are wrong.
  - Remedy: If the process allows, the DB must be reloaded from the AS Offline project .
- **8046** The DB "RC_SE_FIELD" (DB 98, DB for the sensor elements) is not available.
  Remedy: Check the AS project and reload the DB.
- **8047** Error when calling the block "ATTRIB_PTR_FC".
  The internal data structures in the DB "RC_SE_FIELD" (DB98) are wrong.
  - Remedy: If the process allows, the DB must be reloaded from the AS offline project .
- **8048** The DB "RC_PE_FIELD"(DB 97 , DB for the parameter elements) is not available.
  - Remedy: Check the AS project and reload the DB.
- **8049** Error when calling the block "ATTRIB_PTR_FC".
  The internal data structures in the DB "RC_PE_FIELD" (DB97) are wrong.
  - Remedy: If the process allows, the DB must be reloaded from the AS Offline project .

Returned value of the block "RC_ROUTE_XC_SEND" (FB806)

- **8061** Master Route: Error when entering a cross-coupling job
(= Error when calling the block "XC_JOB_USER_IF_FC").
QDIAG remains stored until the error is acknowledged.
  – Remedy:
Change target AS to STOP : ->
- And into RUN again
or
- Cross-coupling order buffer full :-> Check coupling overload
- **8066** Slave Route: Error when entering a cross-coupling job
(= Error when calling the block "XC_JOB_USER_IF_FC").
QDIAG remains stored until the error is acknowledged.
  – Remedy:
Change target AS to STOP : ->
- And into RUN again
or
- Cross-coupling order buffer full :-> Check coupling overload

Returned value of the block "RC_ROUTE_STATE_OS" (FB807)

- **8071** Status message to OS could not be sent.
Error when calling the block "FIFO_INPUT_FC".
FIFO full or coupling to OS disrupted.
  – Remedy: Check coupling condition, or abort Route.

Returned value of the block "RC_ROUTE_CE_ERR" (FC812)

- **8124** Telegram to OS could not be sent.
Error when calling the block "FIFO_INPUT_FC".

Returned value of the block "RC_ROUTE_SE_ERR" (FC813)

- **8134** Telegram to OS could not be sent.
Error when calling the block "FIFO_INPUT_FC".

Returned value of the block "RC_ROUTE_CM_FC" (FC830)

- **8FF8** Route is not master

Returned value of the block "RC_CTRL" (FC834)

- **8FF8** Route is not master

Return value of the block "RC_TR_ROUTE_UNIT" (FB830) / "RC_TR_ROUTE2_UNIT" (FB850)

- **8FF8** Route is not master

Returned value of the block "RC_ROUTE_MAT" (FB836)

- 0000 Material input executed without errors.
- 0001 Material check active by OS
- 0002 Material input active in Link elements
- 0003 Material input in Master AS finished; Material input Slave is still active
- 8361 Route DB is not available
- 8362RouteID is outside the permissible area
- 8363 Timeout Material check by OS
- 8364 Timeout Material input in Link elements
- 8365 SWITCH is outside the permissible area (0..4)
- 8366 Route is not master (call FC RC_MAT in Slave PCU)
- 8367 MATERIAL is outside the permissible area (0..1024)
- 80FF Continuation material is not allowed

17.4 RCS remote elements

17.4.1 Using REMOTE elements

Application

REMOTE elements are used where two production areas "come into contact". Each production area belongs to another AREA and thus also to another Route Control project. From the perspective of a control system, certain elements, such as a valve, are associated with both areas because they act as a link between them.

REMOTE elements are only required for project-spanning communication between automation systems.

Note

- The remote use of a V element of an AREA B is only possible from an AREA A. Additional use of a further AREA might lead to double allocation of the element!
- With plant-spanning routes, there is **no** synchronization between the two Route Control projects. Multiple usage of an element in two or more process cells (local and remote) is therefore not possible.
- When requesting or starting a route, there is no check to determine whether any of the remote elements used in this route (CE, PE, SE) are already being used in a route in another plant.
- In this case, the route only ends with an error when one route requests a control that is currently used by another route.

Basic Principle

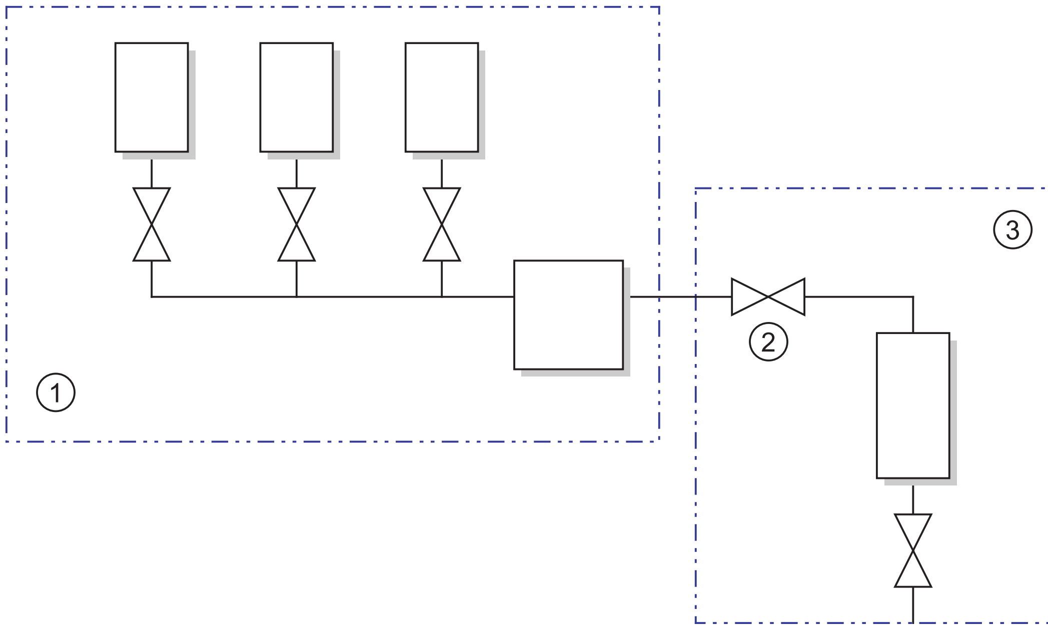
In the example below, Route Control project A contains a REMOTE control element which is designed to activate valve V in an AS of project B.

Note

Because communication via REMOTE elements places a load on cross-coupling (and thus on the entire system), it is only intended for use in a limited number of elements.

Diagram

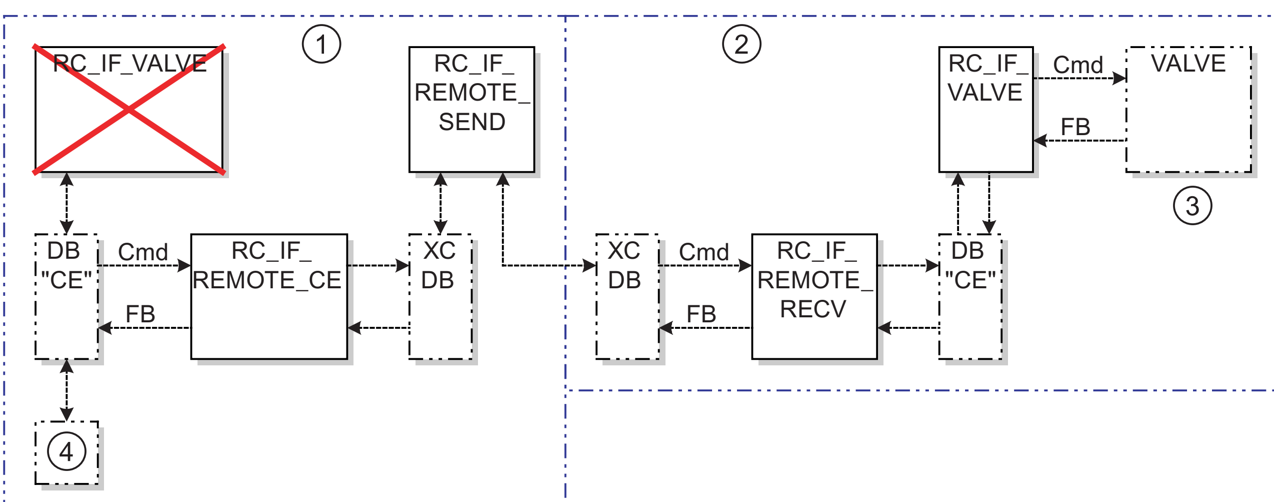
The figure below shows how the example is structured:



1	Route Control project A (master AS)
2	Valve V
3	Route Control project B (slave AS)

Affected blocks and communication paths

The following figure shows the interaction of blocks:



1	Route Control project A
2	Route Control project B

3	Valve V
4	Route

17.4.2 Configuring REMOTE Elements

Definitions

The AS containing the element to be activated is referred to below as a slave AS or peer AS.

The AS that wants to activate this element is referred to as a master AS or local AS.

Requirements

A valve is to be activated as an example. The element ID (= ESG instance ID 1 ... 1024) is known.

Notes on configuration

When configuring the connections, bear in mind the following:

- "Active connection establishment" is selected for only **one** of the connections.
- The local IDs are identical in both projects.
- The partner address and connection resource (TSAP) are entered under the address details.
- Identical S7 subnet IDs ("Ethernet" properties) are set in both projects.

Overview of the procedure

Step	Procedure
1	Establish the NetPro connections.
2	Call of an RC_IF_REMOTE_RECV block in the user interface blocks of the slave AS
3	Configure the CON_ID, SNDRCVDB block inputs
4	Call of an RC_IF_REMOTE_SEND block in the user interface blocks of the master AS
5	Configure the block inputs CON_ID, SNDRCVDB Use the same values as those at the receive block in the slave AS.
6	Call of an RC_IF_REMOTE block in the user interface blocks of the master AS In the example: RC_IF_REMOTE_CE
7	Configure the block inputs REM_CE, SNDRCVDB, ALT_VAL
8	During runtime: Set/reset NEW_GEN at the send block in the master AS
9	Control the route.

Call all blocks involved in remote processing in the same task. The blocks RC_IF_REMOTE_SEND (FB831) and RC_IF_REMOTE_RECV (FB833) and the element proxy blocks RC_IF_REMOTE_CE/_PE/_SE are affected.

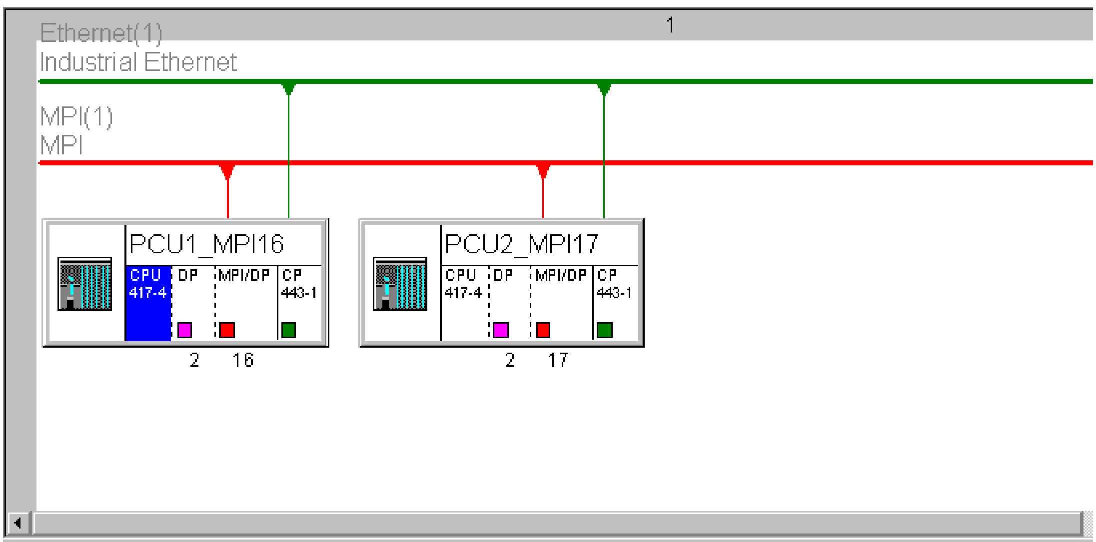
Use the task of OB32 (FB 1225 (BmUsr1000msFB) with a clock of 1 sec.

In the event of data transfer errors, error messages are output at the STATUS outputs of the blocks RC_IF_REMOTE_SEND (FB831) and RC_IF_REMOTE_RECV (FB833). These may originate directly from the system blocks BSEND/BRCV used for data transfer. Their meaning can be found in the description of these blocks.

NetPro

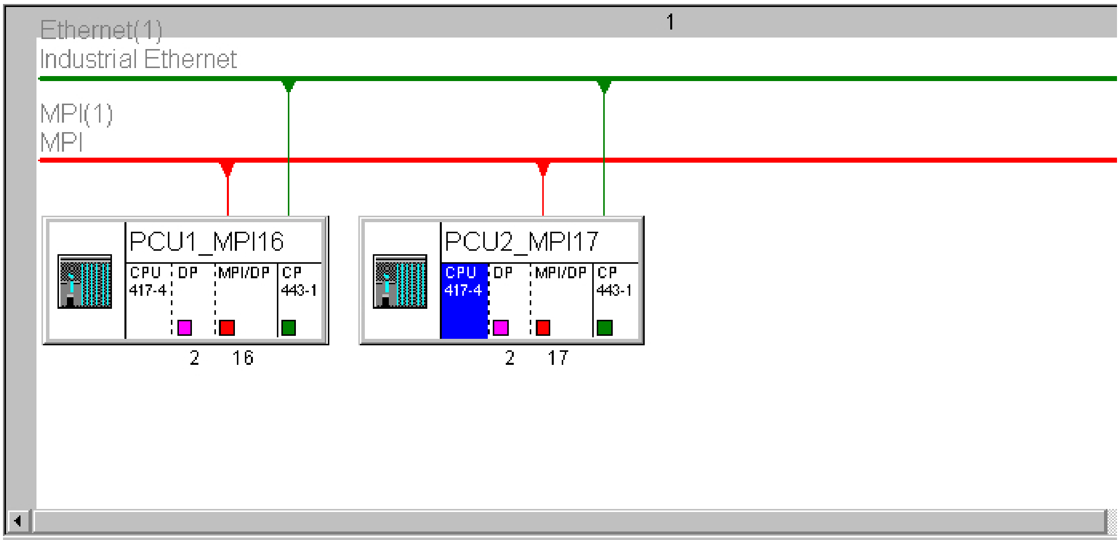
In NetPro, establish an unspecified connection to the MAC address of the peer AS and, in turn, an unspecified connection to the local AS with the same connection ID in the peer AS.

NetPro configuration in the local AS:



Lokal ID	Partner ID	Partner	Type	Aktive	Send operating mode	Subnet	Partner Sc	Local address
50	50	TO REMOTE-SLAVE	S7-connection	yes	no	Ethernet(1) (IE)		00-AB-BA-AF-FE-CF

NetPro configuration in the partner AS



Lokal ID	Partner ID	Partner	Type	Aktive	Send operating mode	Subnet	Partner Sc	Local address
50	50	FROM REMOTE MASTER	S7-connection	no	no	Ethernet(1) (IE)		00-AB-BA-AF-FE-DA

Note

In addition, check the address details of the connections in NetPro.
Use the same connection resources.

Procedure for configuring the slave AS

1. Call an RC_IF_REMOTE_RECV block in the following user interface blocks:
   FB 1200 (BmUsrWarmRestartFB)
   FB 1201 (BmUsrHotRestartFB)
   FB 1220 (BmUsrCycleBeginFB)
   FB 1225 (BmUsr1000msFB)
   Call example:
   ```
   CALL "RC_IF_REMOTE_RECV" , DB3833
   CON_ID :=W#16#50
   SNDRCVDB:=91
   QERR :=#RC_QERR
   S_STATUS:=#RC_S_STATUS
   R_STATUS:=#RC_R_STATUS
   C_STATUS:=#RC_C_STATUS
   ```
   Such a call is required for each logical connection to a master AS from which elements are to be addressed.
2. Configure the connection ID which specifies the connection to the master AS at the CON_ID input. You can see the connection ID in NetPro.
   You also need this ID for configuring the associated send block in the master AS.
3. At the SNDRCVDB input, configure the number of the data block (DB 91 ... 95) to be used for the data transfer.
   You also need this number for parameter assignments on the master AS.

Procedure for Configuring the Master AS

1. Call an RC_IF_REMOTE_SEND block in the following user interface blocks.
   FB 1200 (BmUsrWarmRestartFB)
   FB 1201 (BmUsrHotRestartFB)
   FB 1220 (BmUsrCycleBeginFB)
   FB 1225 (BmUsr1000msFB)
   Call example:
   ```
   CALL "RC_IF_REMOTE_SEND" , DB3831
   CON_ID :=W#16#50
   SNDRCVDB:=91
   NEW_GEN :=M5000.0
   QERR :=#RC_QERR
   S_STATUS:=#RC_S_STATUS
   R_STATUS:=#RC_R_STATUS
   C_STATUS:=#RC_C_STATUS
   QNEW_GEN:=#RC_QNEW_GEN
   QGEN_ACT:=#RC_QGEN_ACT
   ```
   Such a call is required for each logical connection to a slave AS in which elements are to be addressed.
2. Configure the connection ID which specifies the connection to the slave AS at the CON_ID input.
   Specify the same ID as at the associated receive block of the slave AS.
3. Configure the number of the data block to be used for data transfer at the SNDRCVDB input. Specify the same number as at the associated receive block of the slave AS.
4. The NEW_GEN parameter is set with the M5000.0 bit memory in the example.
   This must be set once from 0→1 and from 1→0 (edge transition) during runtime at the send block in the master AS.
5. Call the appropriate REMOTE interface block in the same user interface blocks as above.
   Call example:
   ```
   CALL "RC_IF_REMOTE_CE" , DB3821
   LOC_CE :=1010
   REM_CE :=1020
   SNDRVCDB:=91
   ALT_VAL :=W#16#1
   QCOMM_ID:=#RC_QCOMM_ID
   QCOMMAND:=#RC_QCOMMAND
   Q_FBACK :=#RC_QFBACK
   QRET_VAL:=#RC_QRETVAL
   QERR :=#RC_QERROR
   ```
   In the example, the removed element is a control element (CE). Therefore, the RC_IF_REMOTE_CE block is suitable here.
6. Configure the local ID of the block (1 ... 1024) at the LOC_CE input.
7. Configure the ID of the remote interface block at the REM_CE input. In the example, this is the ID of the block in the slave AS.
8. At the SNDRCVDB input, configure the same DB number as at the send block.
9. Configure the alternate value that is to be used if communication is interrupted at the ALT_VAL input.

Special points to consider when configuring remote connections

With AREA-spanning routes, it can occasionally happen that a route is rejected with the message "Element(s) or partial route(s) already in use", although none of the elements used are in another route.

With remote elements, the information about whether an element is allocated or not must be requested and received via the BSEND/BRCV coupling for the remote AREA. If the requesting partner does not receive a response within a certain time, this **also** leads to the above-indicated message because the lack of information means that **no** statement can be made about the allocation of the element(s). This behavior can occur more or less frequently on account of the different loading of the coupling path.

See also

Overview of REMOTE blocks (Page 1172)

Block overview (Page 673)

17.4.3 Blocks

17.4.3.1 Overview of REMOTE blocks

Note

These REMOTE blocks are only required for communication between automation systems from different AREAs.

REMOTE blocks

Blocks for REMOTE elements are used when route elements from another project are to be used.

The following REMOTE blocks are available for elements:

- RC_IF_REMOTE_CE (FB821) (Page 1173) for control elements
- RC_IF_REMOTE_SE (FB842) (Page 1175) for sensor elements
- RC_IF_REMOTE_PE (FB 843) (Page 1177) for parameter elements

In addition to the data exchange with the foreign AS in which the removed elements are located, the following communication blocks are required:

- RC_IF_REMOTE_SEND (FB 831) (Page 1179)
- RC_REMOTE_SEND (FB832) (Page 1180) (only used internally in RCS)
- RC_IF_REMOTE_RECV (FB833) (Page 1181)
- RC_REMOTE_RECV (FB834) (Page 1182) (only used internally in RCS)

For remote operation of controls, parameter and sensor elements, only the most important element data is transmitted between the controlling and connected remote element. Consequently, in contrast to local mode, not all outputs can be controlled on the connected remote element.

Note

In the instance data set of the connected remote element (RC_CE, RC_LE, RC_PE), the number of the DB used for data transmission is multiplied by -1 is output (-91 ... -95) at the `iRouteId, iFuncId` attribute as an identifier for the remote control.

17.4.3.2 RC_IF_REMOTE_CE (FB 821)

Description

Block RC_IF_REMOTE_CE is a REMOTE interface block that is used to activate a control element (CE) in another Route Control project in another AS. Such a block must be called in the user interface blocks FB 12xx for every foreign control element (see Configuring REMOTE Elements (Page 1168)).

Note

Call "RC_IF_REMOTE_CE" is removed again by user.

The remote flag of the CE used in DB99 is set with the call. When the FB_call is deactivated or deleted again later, the remote flag remains 1 and the CE is still locked for normal operation. This must be corrected manually by the user as follows:

- Reload DB99 from delivery state again
- Initiate RCS server restart

Inputs and outputs

Parameter	Meaning	Type	Default	Type	Permitted values
LOC_CE	ID of the control element in the local AS	INT	-	I	1 ...1024
REM_CE	ID of the control element in the remote AS	INT	-	I	1 ...1024
SNDRCVDB	Number of the data block in the local AS used for data exchange between the local and remote elements	INT	-	I	91 ... 95
ALT_VAL	Alternate value entered in the element in case of communication problems with the remote AS or with the remote element. You can find additional information under the QRET_VAL parameter.	WORD	W#16#0000	I	Examples of ALT_VAL: 0: Element receives feedback "undefined" H8000: Element receives feedback "fault".

Parameter	Meaning	Type	Default	Type	Permitted values
QCOMM_ID	Serial number of the element in the central send data block (you can find additional information under SNDRCVDB)	INT	-	O	-
QCOMMAND	Diagnostic information: 16-bit value of the command transferred to the remote element	WORD	-	O	-
Q_FBACK	Feedback value of the remote control element	WORD	-	O	-
QRET_VAL Permitted values 8211-8216	Return value for block	WORD	-	O	**8211**: The specified CE ID is too small. **8212**: The specified CE ID is too large. **8213**: The CE ID of the partner element is too small. **8214**: The CE ID of the partner element is too large. **8215**: The number of the send/receive block (SNDRVCDB) is too low and outside the permissible range. **8216**: The number of the send/receive block (SNDRVCDB) is too high and outside the permissible range.
QRET_VAL Permitted values 8217-821A		WORD	-	O	**8217**: An attempt is being made to transmit the data of more control elements than is permitted. *The total number of all CEs, SEs and PEs* that can be linked remotely cannot exceed 1,024. **8218**: Timeout or receive error in the send/receive program. The alternate value (ALT_VAL) is used. **8219**: The number of elements to be transferred does not match in the two partner AS. Solution: Trigger a positive edge at the NEW_GEN input at the RC_IF_REMOTE_SEND (FB 831) (Page 1179) block. This performs an initialization. **821A**: Generation running (QCOMM_ID = invalid)
QERR	You can find additional information under the QRET_VAL parameter.	BOOL	-	O	1: Error 0: OK

17.4.3.3 RC_IF_REMOTE_SE (FB 842)

Description

Block RC_IF_REMOTE_SE is a REMOTE interface block that is used to reference a sensor element (SE) in another Route Control project in another AS. Such a block must be called in the user interface blocks FB 12xx for every foreign control element (see Configuring REMOTE Elements (Page 1168)).

Note

Call "RC_IF_REMOTE_SE" is removed again by user.

The remote flag of the SE used in DB98 is set with the call. When the FB_call is deactivated or deleted again later, the remote flag remains 1 and the SE is still locked for normal operation. This must be corrected manually by the user as follows:

- Reload DB98 from delivery state again
- Initiate RCS server restart

Inputs and outputs

Parameter	Meaning	Type	Default	Type	Permitted values
LOC_SE	ID of the sensor element in the local AS	INT	-	I	1 ...1024
REM_SE	ID of the sensor element in the remote AS	INT	-	I	1 ...1024
SNDRVCDB	Number of the data block in the local AS used for data exchange between the local and remote elements	INT	-	I	91 ... 95
ALT_VAL	Alternate value entered in the element in case of communication problems with the remote AS or with the remote element. You can find additional information under the QRET_VAL parameter.	WORD	W#16#0000	I	Examples of ALT_VAL: 0: Element receives feedback "false" 1: Element receives feedback "true" H8000: Element receives feedback "fault".
QCOMM_ID	Serial number of the element in the central send data block (you can find additional information under SNDRCVDB)	INT	-	O	-
Q_FBACK	Feedback value of the remote sensor element	WORD	-	O	-

Parameter	Meaning	Type	Default	Type	Permitted values
QRET_VAL Permitted values 8421-8426	Return value for block	WORD	-	O	**8421**: The specified SE ID is too small. **8422**: The specified SE ID is too large. **8423**: The SE ID of the partner element is too small. **8424**: The SE ID of the partner element is too large. **8425**: The number of the send/receive block (SNDRVCDB) is too low and outside the permissible range. **8426**: The number of the send/receive block (SNDRVCDB) is too high and outside the permissible range.
QRET_VAL Permitted values 8427-842A	Return value for block	WORD	-	O	**8427**: An attempt is being made to transmit the data of more control elements than is permitted. *The total number of all CEs, SEs and PEs* that can be linked remotely cannot exceed 1,024. **8428**: Timeout or receive error in the send/receive program. The substitute value (ALT_VAL) is used instead. **8429**: The number of elements to be transferred does not match in the two partner AS. Solution: Trigger a positive edge at the NEW_GEN input at the RC_IF_REMOTE_SEND (FB 831) (Page 1179) block. This performs an initialization. **842A**: Generation running (QCOMM_ID = invalid)
QERR	You can find additional information under the QRET_VAL parameter.	BOOL	-	O	1: Error 0: OK

17.4.3.4 RC_IF_REMOTE_PE (FB 843)

Description

Block RC_IF_REMOTE_PE is a REMOTE interface block that is used to reference a parameter element (PE) in another Route Control project in another AS. Such a block must be called in the user interface blocks FB 12xx for every foreign control element (see Configuring REMOTE Elements (Page 1168)).

Organization Block

This block can be inserted into any organization block. Please note that all REMOTE elements of a connection (i.e. with an identical SNDRCVDB) must be inserted into the same runtime group as the associated RC_IF_REMOTE_SND block.

Inputs and outputs

Parameter	Meaning	Type	Default	Type	Permitted values
LOC_PE	ID of the parameter element in the local AS	INT	-	I	1 ... 1024
REM_PE	ID of the parameter element in the remote AS	INT	-	I	1 ... 1024
SNDRVCDB	Number of the data block in the local AS used for data exchange between the local and remote elements	INT	-	I	91 ... 95
ALT_VAL	Alternate value entered in the element in case of communication problems with the remote AS or with the remote element. You can find additional information under the QRET_VAL parameter. Any feedback of type Real is allowed.	REAL	-	I	-
QCOMM_ID	Serial number of the element in the central send data block (you can find additional information under SNDRCVDB)	INT	-	O	-

Parameter	Meaning	Type	Default	Type	Permitted values
QRET_VAL Permitted values 8431-8436	Return value for block	WORD	-	O	**8431**: The specified PE ID is too small. **8432**: The specified PE ID is too large. **8433**: The PE ID of the partner element is too small. **8434**: The PE ID of the partner element is too large. **8435**: The number of the send/receive block (SNDRVCDB) is too low and outside the permissible range. **8436**: The number of the send/receive block (SNDRVCDB) is too high and outside the permissible range.
QRET_VAL Permitted values 8437-843B	Return value for block	WORD	-	O	**8437**: An attempt is being made to transmit the data of more control elements than is permitted. The total number of all CEs, SEs and PEs that can be linked remotely cannot exceed 1,024. **8438**: Timeout or receive error in the send/receive program. The substitute value (ALT_VAL) is used instead. **8439**: The number of elements to be transferred does not match in the two partner AS. Solution: Trigger a positive edge at the NEW_GEN input at the RC_IF_REMOTE_SEND (FB 831) (Page 1179) block. This performs an initialization. **843A**: Generation running (QCOMM_ID = invalid) **843B**: Incorrect index for external parameter element
QERR	You can find additional information under the QRET_VAL parameter.	BOOL	-	O	1: Error 0: OK
QSPVALUE	Setpoint sent to the remote parameter element	REAL	-	O	-
QACTV	Actual value of the remote parameter element	REAL	-	O	-
QINDEX	If this is an external parameter element, the index value is output here.	BYTE	-	O	0: No external PE Greater than 0: external PE, value range: 1 ... 24

Parameter	Meaning	Type	Default	Type	Permitted values
QEXT	External: This is an external parameter element of the route. The parameter-element setpoint was configured indirectly (by means of the index). Internal: The parameter-element setpoint was configured directly.	BOOL	-	O	0: Internal parameter element 1: External parameter element
QSUMMED	This is a summation parameter element.	BOOL	-	O	0: Not a summation parameter element 1: Summation parameter element
QROUTE	ID of the route that is actively using this parameter element	INT	-	O	-
QFUNC_ID	Function ID (FUNC_ID) of the route that is actively using this parameter element	INT	-	O	-
QMODE_NO	Mode level (MODE) of the route that is actively using this parameter element	INT	-	O	-

17.4.3.5 RC_IF_REMOTE_SEND (FB 831)

Description

This block provides the interface for assigning parameters to the cross-coupling connection between Route Control automation systems on the sending end of available REMOTE elements. This block is the counterpart to the RC_IF_REMOTE_RECV (FB 833) (Page 1181) block in the peer AS. Such a block must be called in the user interface blocks FB 12xx for each remote connection (i.e. with an identical SNDRCVDB) (see Configuring REMOTE Elements (Page 1168)).

Inputs and outputs

Parameter	Meaning	Type	Default	Type	Attr.	OCM	Permitted values
CON_ID	Local communication ID for accessing the peer AS (from NetPro connection configuration)	WORD	-	I	VQ	-	-
SNDRCVDB	Number of the data block to be used for communication and data exchange with the peer AS	INT	91	I	VQ	-	-
NEW_GEN	Positive edge triggers a request for a new communication ID for all REMOTE element blocks.	BOOL	-	I	VQ	-	0 --> 1: Trigger request

Parameter	Meaning	Type	Default	Type	Attr.	OC M	Permitted values
QERR	Indicates whether or not a block error is pending	BOOL	-	O	VQ	-	1: An error has occurred in the block. 0: No pending errors in the block. You can find additional information under S_STATUS, R_STATUS, C_STATUS.
S_STATUS	The return value of the internal BSEND call is passed through to the outside.	INT	-	O	VQ	-	You can find additional information about return values under the BSEND block.
R_STATUS	The return value of the internal BRCV call is passed through to the outside.	INT	-	O	VQ	-	You can find additional information about return values under the BRCV block.
C_STATUS	Return value for block	INT	-	O	VQ	-	0: No error 100: Error timeout for receive (RCV) 101: Error timeout for send (SND)
QNEW_GEN	Initialization of communication ID for all remote blocks has begun.	BOOL	-	O	VQ	-	0: Initialization not requested 1: Initialization requested
QGEN_ACT	Initialization of communication ID for all REMOTE blocks is active.	BOOL	-	O	VQ	-	0: Initialization in progress 1: Initialization not active

17.4.3.6 RC_REMOTE_SEND (FB832)

This block is only used internally in RCS, no call from the user program is required.

Description

This block communicates at the sending end with up to 5 other automation systems via the coupling DBs DB91 ... DB95.

The data to be sent is read according to its type PE/SE/CE and ID (1 to 1024) from DB97 (PE data), DB98 (SE data) and DB99 (CE data) and copied to the send area of DB91 ... DB95. The type and the ID are copied to the receive area of the DB91 ... DB95 so that the actual values can be written back in an analog manner to the DB 97/98/99 when received.

With BSEND, the setpoints are transferred in the time pattern of "iRateSendSPV" (start value=2 seconds).

The actual values are received with BRCV .

BSEND errors are output in the instance DB in S_Error and S_Status .

BRCV errors are output in the instance DB in R_Error and R_Status .

Errors of FB 832 are output in C_ERROR and C_Status :

C_Error = 1:

- C_Status = 100 → RCV Timeout
- C_Status = 101 → SEND Timeout (→ BSEND reset)

17.4.3.7 RC_IF_REMOTE_RECV (FB 833)

Note

This block is only required for communication between automation systems from different projects.

Description

The RC_IF_REMOTE_RECV block is the interface for assigning parameters to the cross-coupling connection between Route Control automation systems on the receiving end of available REMOTE elements. This block is the counterpart to the RC_IF_REMOTE_SEND (FB 831) (Page 1179) block in the peer AS. Such a block must be called in the user interface blocks FB 12xx for each remote connection (i.e. with an identical SNDRCVDB) (see Configuring REMOTE Elements (Page 1168)).

Inputs and outputs

Parameter	Meaning	Type	Default	Type	Attr.	OCM	Permitted values
CON_ID	Local communication ID for accessing the peer AS (from NetPro connection configuration)	WORD	-	I	VQ	-	-
SNDRCVDB	Number of the data block to be used for communication and data exchange with the peer AS	INT	91	I	VQ	-	-
QERR	Positive edge triggers a request for a new communication ID for all REMOTE element blocks.	BOOL	-	O	VQ	-	-
S_STATUS	The return value of the internal BSEND call is passed through to the outside.	INT	-	O	VQ	-	You can find additional information about return values under the BSEND block.

Parameter	Meaning	Type	Default	Type	Attr.	OC M	Permitted values
R_STATUS	The return value of the internal BRCV call is passed through to the outside.	INT	-	O	VQ	-	You can find additional information about return values under the BRCV block.
C_STATUS	Return value for block	INT	-	O	VQ	-	0: No error 100: Error timeout for receive (RCV) 101: Error timeout for send (SND)

17.4.3.8 RC_REMOTE_RECV (FB834)

This block is only used internally in RCS, no call from the user program is required.

Description

This block communicates at the receiving end with up to 5 other automation systems via the coupling DBs DB91 ... DB95.

With BRCV calls, the setpoints are received from the partner AS. The received data is entered according to its type PE/SE/CE and the ID (1 to 1024) in the DB97 (PE data), DB98 (SE data) or DB99 (CE data). The type and the ID are copied to the send area so that the actual values can be read from the DB 97/98/99 in an analog manner during sending and can be transferred to the send area.

With BSEND , the actual values are transferred in the time pattern of "iRateSendSPV" (start value=2 seconds).

BSEND Errors are output in the instance DB in S_Error and S_Status .

BRCV Errors are output in the instance DB in R_Error and R_Status .

Errors of FB834 are output in C_ERROR and C_Status :

If C_Error = 1:

- C_Status = 100 → RCV Timeout
- C_Status = 101 → SEND Timeout (→ BSEND reset)
- C_Status = 102 → Error in RCV frame (Type: Faulty)
- C_Status = 103 → Error transmission message (Type: Faulty)

Cross coupling AS-AS (S7-400) 18

18.1 General:

Cross coupling for 32 PCUs

This cross coupling provides the following features:

- Data exchange from one PCU up to a maximum of 31 different PCUs
- Redundant bus structure with two CP443 is possible
- Only for S7 - to S7 - PLCs
- Maximum data volume of 400 bytes per job

MAC address assignment

- Industrial Ethernet address = 08.00.06.01.07.xy
  (xy = PCU number in hexa format, e.g. for PCU16: 08.00.06.01.07.10).

18.2 Cross Coupling S7-S7 for 32 PLCs

18.2.1 General

Cross coupling is used for data interchange from a PCU to a maximum of 31 other PCUs via the process bus, SIMATIC NET Industrial Ethernet.

The hardware interfaces and S7 connections, via which the data is exchanged, are specified in the Step 7 project using the tools 'HW-Config' and 'NetPro'.

So that the coupling between the PCUs can be structured correctly, it is necessary for the configuration to have been performed already with these two tools and for it to have been loaded to the AS involved.

This is normally the case when supplying a new standard system.

However, if the project is expanded by the addition of further PCUs or communication CPs, this must be checked, whereby it is important to ensure that the connection table is edited again in order to prevent inconsistencies (see chapter 'Connection table in SIMATIC NetPro (Page 1184)').

Note

Not for PCU type S7-1500!

The "Cross coupling S7-S7" functionality is not available for PCU type S7-1500. However, on the application side, blocks can be transferred between S7-400 and S7-1500 using the BSEND/ BRECEIVE system blocks.

18.2.2 Connection table in SIMATIC NetPro

Basics

The connection table specifies between which AS a connection is to be set up. In the case of the SIMATIC S7 station CPU, this connection table is configured using the "NetPro" SIMATIC Manager tool (configure network).

Note

Connection configuration with NetPro

For further information about connection configuration, please refer to the STEP 7 documentation.

In the case of the SIMATIC S7, communication is carried out directly via the firmware. Jobs are assigned between the software and the firmware via the local and the partner ID of the connection.

The respective delivered project (CPU 416 or CPU 417) includes a functional connection table for a standard configuration. This connection table must be expanded to the actual number of ASs.

For activation of the cross coupling, the partner PCU numbers must be entered in the XC_PCU_32 parameter assignment.

The standard configuration of the connection table is configured according to the following rules:

- The connections are set up as bilaterally configured connections. These include a connection ID at both end points and have to be configured with both the active and passive connection partner.
- To be configured for 16 AS and therefore for 15 connections per AS.
- The local connection IDs in an AS must be different, which is why the PCU number of the partner AS is used here.
- The active connection setup (active = "Yes") is parameterized respectively for the device with the higher connection ID. This is then done passively for the partner (active="No").
- Example: AS13 (local ID = 5) couples to AS 05 (local ID = D)
  - AS13 makes the connection active → active="Yes"
  - AS05 makes the connection passive → active="No"

Structure of the Ethernet address:

- 08.00.06.xx.yy.zz
- xx = Bus number 01 (for redundant process bus 01 or 02
- yy= 07 = identifier for S7
- zz= PCU number as hexa number

Table 18-1 Connection-Id structure

1st connection:	ID =PCU number in hexa
2nd connection (redundant):	ID= PCU number in hexa + 200 hexa

Example:

AS 05 should couple to AS 13 via redundant process bus with two CP443.

The following connections in AS13 are used:

Local ID	Partner ID	Partner	Type	Active
5	D	AS_05_416-3	S7 connection	Yes
25	2D	AS_05_416-3	S7 connection	Yes

The following connections in AS 05 are required:

Local ID	Partner ID	Partner	Type	Active
D	5	AS_13_416-3	S7 connection	No
2D	25	AS_13_416-3	S7 connection	No

Information on NetPro

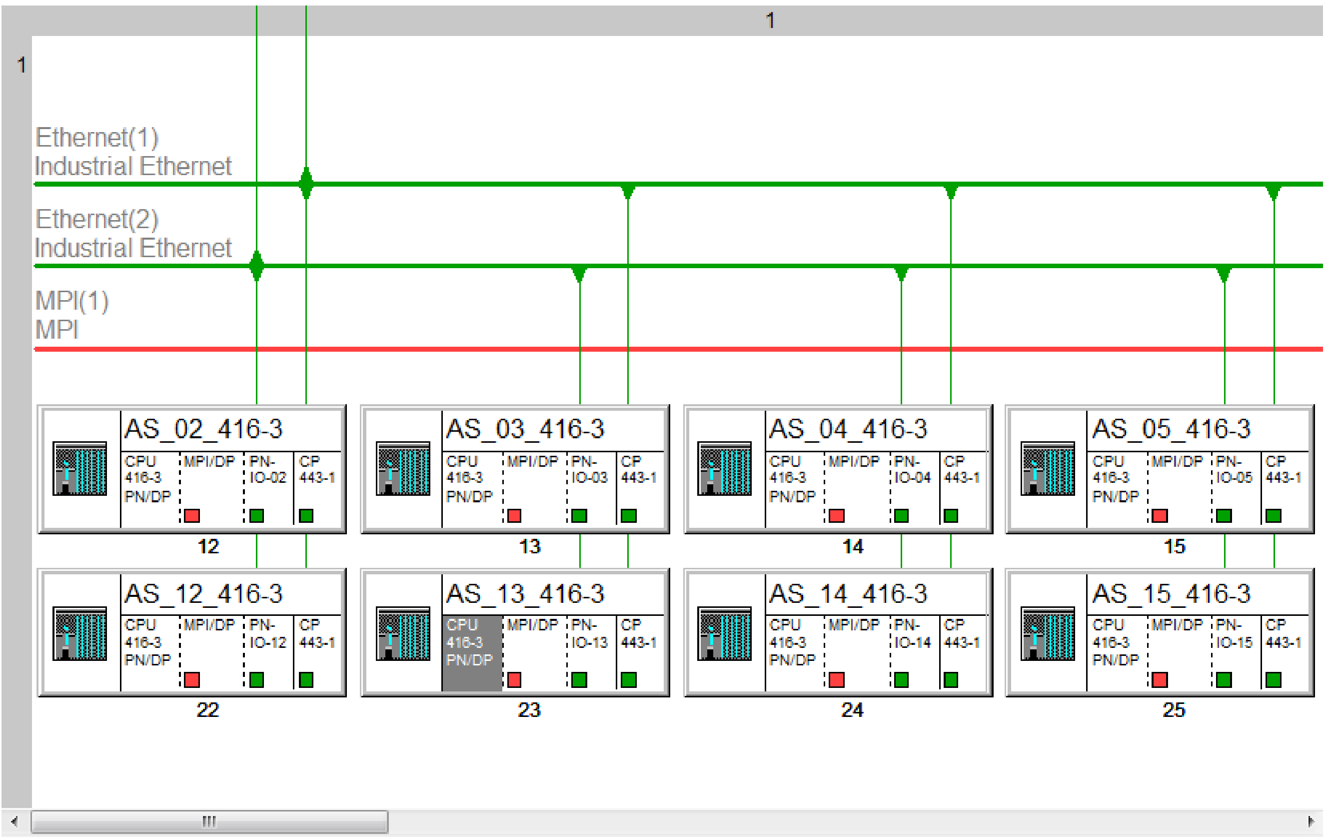
- In the example of the above-described S7 connection configuration and in the AS delivery project, all the stations are located in a single S7 project.
- NetPro contains a connection table with all configured communication connections.
- For new connections, the remote end point of the created S7 connection is automatically included in the partner's connection table.
- All S7 connections, which have a connection ID at both end points, are bilaterally configured S7 connections.
- You can also create an S7 connection to an unspecified partner, wherein the end points of these connections may be in different projects. Further information on this topic in the can be found in the STEP 7 online help at:
  - Configuring interproject connections
  - Connections types to partners in other projects
  - Creating interproject subnets

Important:

For a bilaterally configured S7 connection, the configuration has to be loaded to both connection partners after the S7 connection has been created.

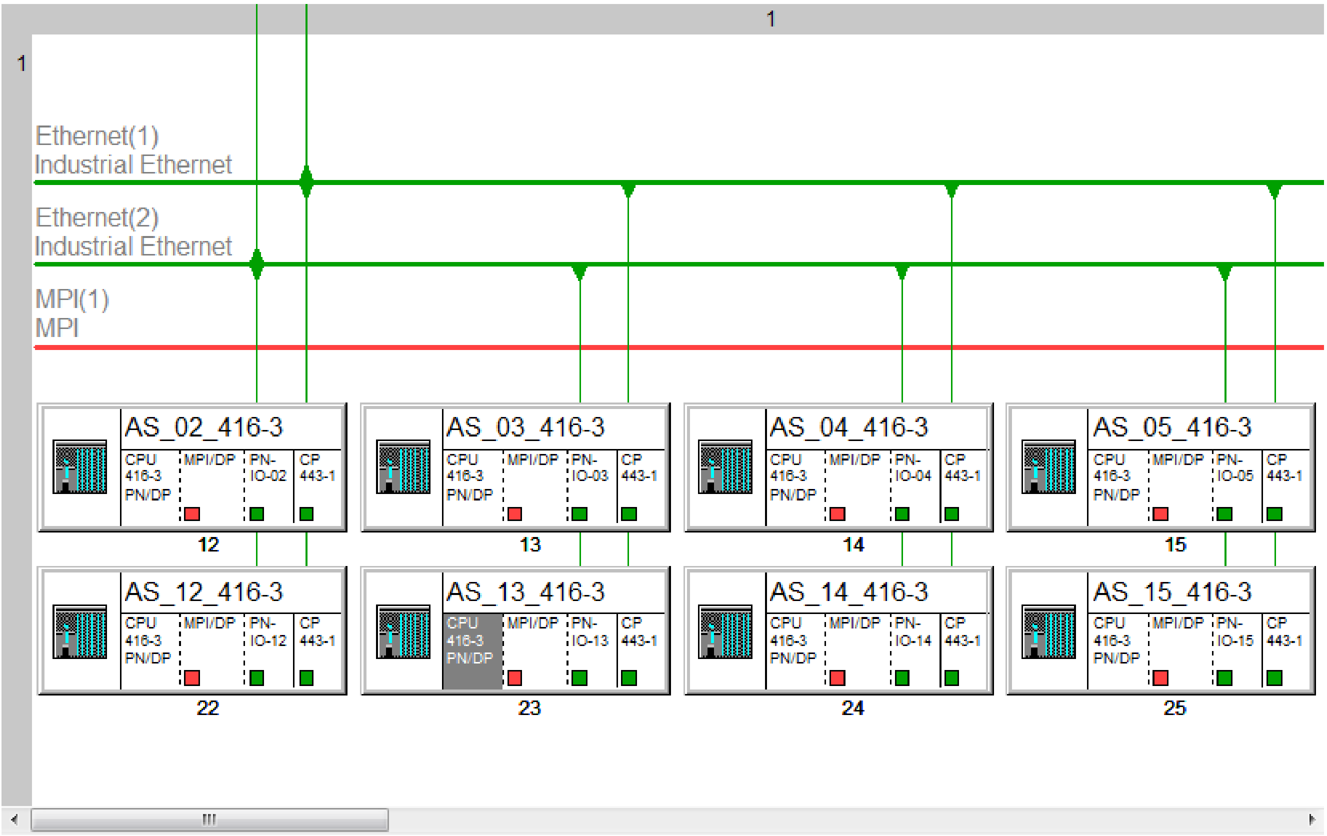
Examples of connection tables in the delivery project

The picture shows the NetPro view for AS 13 without redundant process bus:



Local ID	Partner ID	Partner	Type	Active connection	Subnet
1	D	AS_01_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
2	D	AS_02_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
3	D	AS_03_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
4	D	AS_04_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
5	D	AS_05_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
6	D	AS_06_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
7	D	AS_07_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
8	D	AS_08_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
9	D	AS_09_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
A	D	AS_10_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
B	D	AS_11_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
C	D	AS_12_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
E	D	AS_14_416-3 / CPU 416-3 PN/DP	S7 connection	No	Ethernet(1) [IE]
F	D	AS_15_416-3 / CPU 416-3 PN/DP	S7 connection	No	Ethernet(1) [IE]
10	D	AS_16_416-3 / CPU 416-3 PN/DP	S7 connection	No	Ethernet(1) [IE]

The following picture shows the NetPro view for AS 05 without redundant process bus:



Local ID	Partner ID	Partner	Type	Active connection	Subnet
1	D	AS_01_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
2	D	AS_02_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
3	D	AS_03_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
4	D	AS_04_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
5	D	AS_05_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
6	D	AS_06_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
7	D	AS_07_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
8	D	AS_08_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
9	D	AS_09_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
A	D	AS_10_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
B	D	AS_11_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
C	D	AS_12_416-3 / CPU 416-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
E	D	AS_14_416-3 / CPU 416-3 PN/DP	S7 connection	No	Ethernet(1) [IE]
F	D	AS_15_416-3 / CPU 416-3 PN/DP	S7 connection	No	Ethernet(1) [IE]
10	D	AS_16_416-3 / CPU 416-3 PN/DP	S7 connection	No	Ethernet(1) [IE]

18.2.3 Parameterization of Connections (XC_PCU_32)

18.2.3.1 Parameterization XC_PCU_32

Basics

Module XC_PCU_32 lets you specify to which partner PCUs a connection should be set up.

To do this, you must enter the number of the partner PCU in the parameter PCU. It must be ensured here that a connection to this AS (=PCU) is also configured in the 'NetPro' connection table.

If a PCU no. is changed, it is essential to restart the system Only then is the changed PCU No. transferred (this is a system property of SIMATIC S7 (see system manual):

- The addressing parameters are only evaluated when the module is first called up.
- Thus, the first call establishes the communication relationship (connection) to the remote partner until the next restart.

Note

Not for PCU type S7-1500!

The S7-S7 cross coupling and the XC_PCU_32 parameter assignment is not available for PCU type S7-1500!

Parameterization XC_PCU_32 / DB 704

PCU parameter set (max. 31 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
PCU	NL	INT	ENG	0	PCU number
Enable	HI	BOOL	ENG	TRUE	Release of PCU
ConnID_1	HI	WORD	ENG	W#16#0000	ID of connection 1
ConnID_2	HI	WORD	ENG	W#16#0000	ID of connection 2
InstanceDB	HI	INT	ENG	751	Instance DB of GET/PUT/STATUS
ErrorCode	HI	INT	RT	-3	0=OK -1 = DB missing -2 = incorrect -3=PCU<=0

Name	Access	D.Type	A.Type	Value	Comment
PCU_State	HI	INT	RT	-1	-1=Verb.error 0=STOP 2=RUN 1=C.START 3=W.START 4=HOLD 5=DEFECT
PCU_State_ Conn_1	HI	INT	RT	-1	-1=Verb.error 0=STOP 2=RUN 1=C.START 3=W.START 4=HOLD 5=DEFECT
PCU_State_ Conn_2	HI	INT	RT	-1	-1=Verb.error 0=STOP 2=RUN 1=C.START 3=W.START 4=HOLD 5=DEFECT
ParamError	NL	BOOL	RT	FALSE	PCU No. has an incorrect DS
HBParamErr	HI	BOOL	SYS	FALSE	Auxiliary bit PCU No. has an incorrect DS

Note

- PCU_State_Conn_1 and PCU_State_Conn_2 receive the operation condition of the partner PCU from SFB22. It will be slid on the right side in order to read this condition better. If the partner PCU is in the condition Run, this condition will be changed therefore from 512 to 2 in a cyclic way.
- For diagnostic purposes, further information is available to the user on the state of each connection.
- This information can be found in the application 'Parameterization → Block XC_PCU_32', and then looking under 'Options → Extended attributes' for which the information is required.

PCU numbers above 16

As standard, PCU numbers in the range from 1 to 32 are used. 31 connections can be supplied simultaneously via the cross coupling. However, it is possible for PCU numbers in the range from 1 to 200 to be used. Consequently, PCU numbers from this range can also be allocated to the 31 connections.

In delivery project, the connections for a maximum of 16 AS (PCUs) are prescribed. An extension to 32 PCUs (= quantity structure of the XC_PCU_32) is possible at any time by using the Simatic Manager tools 'HW-Config' and 'NetPro'.

If 16 ASs is an adequate number in a real system and PCU Nos. 17 ... 200 are to be used, it is recommended that an unneeded connection be reconfigured as appropriate (Local_ID, Partner_ID, interfaces, MAC addresses).

General points to remember:

A 1:1 allocation of AS No. (=Local_ID) to PCU No. is not essential, but is recommended for reasons of clarity. The following should be noted:

- The connection identified by the Local_ID must address the correct partner AS.
- In the XC_PCU_32 data set, the parameters 'PCU' of the partner PCU and 'ConnID' of the 'Local_ID' of the connection partner must correspond to the connection partner in the NetPro connection table.

18.2.3.2 PCU allocation to data set

Parameter assignment 'XC_PCU_32'

The 'XC_PCU_32' data sets must be occupied with a particular pattern in order to guarantee that this can also be used by the option pack 'Route Control System':

- For each connection, a data set is occupied for both connection partners, i.e. for a maximum of 32 PCUs, data sets DS 1 ... DS 31 are used
- For a particular connection, the same data set has to be used in both partner PCUs.
- The PCU attribute (= Partner PCU) must be unique across all 'XC_PCU_32' data sets of a PCU, i.e. may only be used once.
- In order to guarantee this, the PCU attribute of a particular data set should be calculated for both partner PCUs from the following (sub) tables. The procedure is described in a following example.

Table for calculating the PCU value of the connection partners

Table 18-2 Attribute value PCU (= connection partner) in PCU No 1 - 16

Data record	Buffer DB *)	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
DS 01	XC_PUTX_1 (DB451)	02	01	31	30	29	28	27	26	25	24	23	22	21	20	19	18
DS 02	XC_PUTX_2 (DB452)	03	32	01	31	30	29	28	27	26	25	24	23	22	21	20	19
DS 03	XC_PUTX_3 (DB453)	04	03	02	01	31	30	29	28	27	26	25	24	23	22	21	20
DS 04	XC_PUTX_4 (DB454)	05	04	32	02	01	31	30	29	28	27	26	25	24	23	22	21

Data re-cord	Buffer DB *)	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
DS 05	XC_PUTX_5 (DB455)	06	05	04	03	02	01	31	30	29	28	27	26	25	24	23	22
DS 06	XC_PUTX_6 (DB456)	07	06	05	32	03	02	01	31	30	29	28	27	26	25	24	23
DS 07	XC_PUTX_7 (DB457)	08	07	06	05	04	03	02	01	31	30	29	28	27	26	25	24
DS 08	XC_PUTX_8 (DB458)	09	08	07	06	32	04	03	02	01	31	30	29	28	27	26	25
DS 09	XC_PUTX_9 (DB459)	10	09	08	07	06	05	04	03	02	01	31	30	29	28	27	26
DS 10	XC_PUTX_10 (DB460)	11	10	09	08	07	32	05	04	03	02	01	31	30	29	28	27
DS 11	XC_PUTX_11 (DB461)	12	11	10	09	08	07	06	05	04	03	02	01	31	30	29	28
DS 12	XC_PUTX_12 (DB462)	13	12	11	10	09	08	32	06	05	04	03	02	01	31	30	29
DS 13	XC_PUTX_13 (DB463)	14	13	12	11	10	09	08	07	06	05	04	03	02	01	31	30
DS 14	XC_PUTX_14 (DB464)	15	14	13	12	11	10	09	32	07	06	05	04	03	02	01	31
DS 15	XC_PUTX_15 (DB465)	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01
DS 16	XC_PUTX_16 (DB466)	17	16	15	14	13	12	11	10	32	08	07	06	05	04	03	02
DS 17	XC_PUTX_17 (DB467)	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03
DS 18	XC_PUTX_18 (DB468)	19	18	17	16	15	14	13	12	11	32	09	08	07	06	05	04
DS 19	XC_PUTX_19 (DB469)	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05
DS 20	XC_PUTX_20 (DB470)	21	20	19	18	17	16	15	14	13	12	32	10	09	08	07	06
DS 21	XC_PUTX_21 (DB471)	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07
DS 22	XC_PUTX_22 (DB472)	23	22	21	20	19	18	17	16	15	14	13	32	11	10	09	08
DS 23	XC_PUTX_23 (DB473)	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09
DS 24	XC_PUTX_24 (DB474)	25	24	23	22	21	20	19	18	17	16	15	14	32	12	11	10
DS 25	XC_PUTX_25 (DB475)	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11
DS 26	XC_PUTX_26 (DB476)	27	26	25	24	23	22	21	20	19	18	17	16	15	32	13	12
DS 27	XC_PUTX_27 (DB477)	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13

Data record	Buffer DB *)	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
DS 28	XC_PUTX_28 (DB478)	29	28	27	26	25	24	23	22	21	20	19	18	17	16	32	14
DS 29	XC_PUTX_29 (DB479)	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15
DS 30	XC_PUTX_30 (DB480)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	32
DS 31	XC_PUTX_31 (DB481)	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17

Table 18-3 Attribute value PCU (= connection partner) in PCU No 17 - 32

Data record	Buffer	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
DS 01	DB451	32	16	15	14	13	12	11	10	09	08	07	06	05	04	03	17
DS 02	DB452	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	02
DS 03	DB453	19	32	17	16	15	14	13	12	11	10	09	08	07	06	05	18
DS 04	DB454	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	03
DS 05	DB455	21	20	32	18	17	16	15	14	13	12	11	10	09	08	07	19
DS 06	DB456	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	04
DS 07	DB457	23	22	21	32	19	18	17	16	15	14	13	12	11	10	09	20
DS 08	DB458	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	05
DS 09	DB459	25	24	23	22	32	20	19	18	17	16	15	14	13	12	11	21
DS 10	DB460	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	06
DS 11	DB461	27	26	25	24	23	32	21	20	19	18	17	16	15	14	13	22
DS 12	DB462	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	07
DS 13	DB463	29	28	27	26	25	24	32	22	21	20	19	18	17	16	15	23
DS 14	DB464	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	08
DS 15	DB465	31	30	29	28	27	26	25	32	23	22	21	20	19	18	17	24
DS 16	DB466	01	31	30	29	28	27	26	25	24	23	22	21	20	19	18	09
DS 17	DB467	02	01	31	30	29	28	27	26	32	24	23	22	21	20	19	25
DS 18	DB468	03	02	01	31	30	29	28	27	26	25	24	23	22	21	20	10
DS 19	DB469	04	03	02	01	31	30	29	28	27	32	25	24	23	22	21	26
DS 20	DB470	05	04	03	02	01	31	30	29	28	27	26	25	24	23	22	11
DS 21	DB471	06	05	04	03	02	01	31	30	29	28	32	26	25	24	23	27
DS 22	DB472	07	06	05	04	03	02	01	31	30	29	28	27	26	25	24	12
DS 23	DB473	08	07	06	05	04	03	02	01	31	30	29	32	27	26	25	28
DS 24	DB474	09	08	07	06	05	04	03	02	01	31	30	29	28	27	26	13
DS 25	DB475	10	09	08	07	06	05	04	03	02	01	31	30	32	28	27	29
DS 26	DB476	11	10	09	08	07	06	05	04	03	02	01	31	30	29	28	14
DS 27	DB477	12	11	10	09	08	07	06	05	04	03	02	01	31	32	29	30
DS 28	DB478	13	12	11	10	09	08	07	06	05	04	03	02	01	31	30	15
DS 29	DB479	14	13	12	11	10	09	08	07	06	05	04	03	02	01	32	31

Data record	Buffer	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
DS 30	DB480	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	16
DS 31	DB481	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01

Note

Buffer DB

This information is only used for diagnostics if a possible coupling problem cannot be localized with the XC_PCU_32 attributes 'ErrorCode' and 'PCU_State_xxx'. This column is not relevant for the initial configuration.

Procedure with an example:

A connection shall be established between PCU 3 and PCU 4.

- For a connection partner (e.g. PCU 3), the accompanying column (column 3) of the table applies, the partner PCU (here: PCU 4) has to be looked for here.
- The line found as a result (DS 05) precisely corresponds to the data set, in which the connection has to be configured at both sides. From this, it follows:
- In DS 05, 3 = 4 must be entered for PCU
- In DS 05, 4 = 3 must be entered for PCU

18.2.4 Job Parameterization (XC_JOB_32)

18.2.4.1 Parameterization XC_JOB_32

Basics

- A maximum of 512 coupling jobs are available in each PCU. A maximum of 400 bytes can be transferred with each job.
- The cross coupling is not active if data block DB 704 (XC_PCU_32) has been deleted or the number of partner PCUs is equal to zero. In this case, processing is interrupted by the function block for the cross coupling.
- A job is enabled when a PCU no. is entered and the enable bit is set to 1. The job is then initiated each time the parameterized time has elapsed.
- If the PCU no. is zero, this job is not assigned and can be used by the system for a temporary job.
- A temporary job is only initiated once. Examples include jobs such as: an SEQS initiation for the start of a plant section in another PCU or the user executing a one-off job using function FC 685 (see chapter Temporary Job (Page 1201)).

- If all jobs are assigned, it is not possible to initiate a temporary job, i.e. it is not possible to start plant sections of another PCU via the ASTA module.
- GET jobs (SFB 14), PUT jobs (SFB 15), and time jobs (PUT Time) are possible.
- The areas peripherals, inputs, outputs, flags, and data blocks can be used as storage areas for the transmitted and received data. The coding corresponds to the definition for SIMATIC S7:

Note

Not for PCU type S7-1500!

The S7-S7 cross coupling and the XC_JOB_32 parameter assignment is not available for PCU type S7-1500!

Table 18-4 Coding of memory areas (parameters: SrcArea / DestArea / FltSrcArea)

Hex-Code	Description
80	Storage area peripherals
81	Storage area inputs
82	Storage area outputs
83	Storage area flags
84	Data block

The job is processed when the actual time is 0.

If an error is detected when processing the job, the 'PCU_Error' bit is set.
If the job has been incorrectly parameterized, this is indicated to the user by the setting of the bit 'ParamError'.

Default values can be specified for GET jobs. These values are transferred to the local storage area if the connection fails (see chapter Temporary Job (Page 1201)).

A job is parameterized in the PCU with the application 'Parameterization', select 'Block XC_JOB_32'. The following screen form appears:

Parameterization XC_JOB_32 / DB 705

PCU parameter set (max. 512 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
PCU	NL	INT	ENG	0	PCU number
Type	NL	USINT	ENG	14	14=GET 15=PUT 16=PUT-TIME
Enable	NL	BOOL	ENG	FALSE	Enables JOB
Time	NL	INT	ENG	10	Job time in seconds
ActTime	HI	INT	RT	0	Current time value
JobTime	HI	INT	RT	0	Job time

Name	Access	D.Type	A.Type	Value	Comment
Src	NL	INT	ENG	200	Source: 0=P/E/A/M 1..8191=DB
SrcArea	NL	BYTE	ENG	B#16#84	Source: (Storage area, see table above)
SrcByte	NL	DINT	ENG	L#0	Source-Byte-Address
NumBytes	NL	INT	ENG	400	Number of bytes to be transmitted
Dest	NL	INT	ENG	200	Target: 0=P/E/A/M 1..8191=DB
DestArea	NL	BYTE	ENG	B#16#84	Target: (Storage area, see table above)
DestByte	NL	DINT	ENG	L#0	Target-Byte-Address
PCU_State	HI	INT	RT	0	-1=Verb.error 0=STOP 2=RUN 4=HOLD 5=DEFECT
LastJob Result	HI	INT	RT	0	0=Idle 2=Run 4=OK 8=Error
LastGetPut Result	HI	INT	RT	0	-1=CPU STOP 0=OK Rest corresponds to SFB GET/PUT
FltTime	NL	INT	ENG	120	Waiting time until default value is transferred
FltActTime	HI	INT	RT	120	Current time value until default value is transferred
FltSrc	NL	INT	ENG	201	Source default values 0=P/E/A/M 1..8191=DB
FltSrcArea	NL	BYTE	ENG	B#16#00	Source default values: (Storage area, see table above)
FltSrcByte	NL	DINT	ENG	L#0	Substitution values byte address
BLKMOV_ RET_VAL	HI	INT	RT	0	Substitution values: Block transfer 0=O.K.
TmpJob	HI	BOOL	SYS	FALSE	Temp. JOB, if finished PCU=0 Enable=0
CONN_1	NL	BOOL	RT	FALSE	Transfer to connection 1 successful
CONN_2	NL	BOOL	RT	FALSE	Transfer to connection 2 successful

Name	Access	D.Type	A.Type	Value	Comment
PCU_Error	NL	BOOL	RT	FALSE	Remote PCU is not in the run or connection error
ParamError	NL	BOOL	RT	FALSE	GET/PUT/BLKMOV: Parametrization error

Job diagnostics

If a job is not executed, but is exited with an error, the user is provided with information designed to facilitate error location. This information can be displayed via the button "Advanced attributes".

Diagnostics: Of PCU_State

These IDs of the PCU state are passed on by calling the system function block SFB 22 (status).

The ID in the PCU_State parameter has the following meaning:

PCU_State	Description
-1	Connection error
0	PCU is in the operating state Stop
2	PCU is in the operating state Run
4	PCU is in the operating state Halt
5	PCU is in the operating state Defective

Diagnostics: States of parameter 'LastGetPutResult'

These states are passed on when calling the system function blocks SFB 14 (GET) and SFB 15 (PUT).

In addition, the statuses -1 and 0 are created by the cross coupling function block.

The states have the following meaning:

Status	Description
-1	PCU is in the operating state Stop
0	OK
1	Communication problems, e.g.: • - Connection description not loaded • (local or remote) • - Connection interrupted (e.g. cable, CPU switched off, CP in STOP)
2	Negative acknowledgement from partner device, the function cannot be executed
4	Error in the receive area pointers RD_i relating to data length or data type
8	Access error in the case of the partner CPU
10	Cannot access local user memory, (e.g. accessed DB has been deleted)
11	Warning: New job is ineffective as previous job has not been completed

Status	Description
12	When calling the SFB • an instance DB was specified that does not belong to SFB 14 • the specified DB is a global and not an instance DB • - cannot find instance DB (load new instance DB from PU)
20	Not enough memory

18.2.4.2 GET Job

Basic principles

- A GET job is used to retrieve data from another PCU and write it to the local storage area.
- Default values can be parameterized for this job type. If the connection between the PCUs is interrupted, or if the job is aborted for any other reasons (not a parameterization error), the default values are transferred to the destination data on expiry of a specified time.
- It is therefore possible to run the PCU in a safe state in the event of a coupling data failure, e.g. by setting setpoint values to zero or resetting the control.
- This is the main advantage of a GET job compared with a PUT job. In the case of a PUT job, an error is also detected when the job is aborted, whereby the destination data is in the partner PCU. The PCU which detects the error can no longer access the destination data.
- As the PUT job in the partner PCU is processed by the firmware, the PCU has no interface for detecting an error. It is therefore essential for users to monitor this type of job themselves, or to send only 'unimportant data' which cannot endanger operation in the event of failure.
- If possible, always use a GET job instead of a PUT job.
- If a GET job is aborted, the FltTime is started in the parameter record of the XC_JOB_32 module. When this time has elapsed, the default values are transferred.
- As the job is repeatedly triggered in the parameterized time (Parameter Time), the default values are also repeatedly transferred if a job is aborted and the 'FltTime' time has elapsed.
- If the 'FltSrcArea' parameter is set to 0, there are no default values.

Example:

- Using a GET job, 200 data bytes are to be retrieved every 5 seconds from PCU 4, data block DB 200, from DBW 800 onwards.
- The received data is to be stored in data block DB 400, from DBW 200 onwards.
- If the connection fails, the default values are to be transferred 30 seconds later from data block DB 100, from DBW 500 onwards to data block DB 400.

XC_JOB_32 (DB 705)

NAME	TYPE	Value	Comment
PCU	INT	4	Get data from PCU 4
Type	USINT	14	Job type: 14=GET
Enable	BOOL	TRUE	Enables job
Time	INT	5	Get data every 5 seconds
Src	INT	200	Source DB 200
SrcArea	BYTE	B#16#84	Source storage area DB
SrcByte	DINT	L#800	Byte address 800 in source DB 200
NumByte	INT	200	Transfer 200 bytes
Dest	INT	400	Destination DB 400
DestArea	BYTE	B#16#84	Destination storage area DB
DestByte	DINT	L#200	Byte address 200 in destination DB 400
FltTime	INT	30	GET: Waiting time until default value is transferred
FltSrc	INT	100	GET: Source default values DB 100
FltSrcArea	BYTE	B#16#84	GET: Default value area DB
FltSrcByte	DINT	L#500	GET: Default value byte address 500 in DB 100
PCU_Error	BOOL	FALSE	Partner PCU is not in the run or connection error
ParamError	BOOL	FALSE	Parameter error

18.2.4.3 PUT Job

Basic principles

A PUT job is used to write data from a local PCU to another PCU.

Default value transfer is not possible with this job type as the destination data is in the partner PCU in which the job was aborted.

The PUT job is processed by the firmware in the partner PCU. There is no interface to signal an error when data is no longer being received.

For this reason, users must monitor this job in the partner PCU themselves and take the appropriate measures in the event of connection failure, e.g. set setpoint values to zero, reset controls, ...

However, in order to ensure that the PUT job in the partner PCU is monitored and the received data are written in a safe state, the easiest solution is to use a GET job in the partner PCU.

Example:

- Using a PUT job, 360 data bytes are to be written every 8 seconds to PCU 2, in data block DB 400, from DBW 200 onwards.
- The data to be sent is stored in data block DB 200, from DBW 400 onwards.

XC_JOB_32 (DB 705)

NAME	TYPE	Value	Comment
PCU	INT	2	Send data in PCU 2
Type	USINT	15	Job type: 15=PUT
Enable	BOOL	TRUE	Enables job
Time	INT	8	Send data every 8 seconds
Src	INT	200	Source DB 200
SrcArea	BYTE	B#16#84	Source storage area DB
SrcByte	DINT	L#400	Byte address 400 in source DB 200
NumByte	INT	360	Transfer 360 bytes
Dest	INT	400	Destination DB 400
DestArea	BYTE	B#16#84	Destination storage area DB
DestByte	DINT	L#200	Byte address 200 in destination DB 400
FltTime	INT	0	No default values
FltSrc	INT	0	No default values
FltSrcArea	BYTE	B#16#0	No default values
FltSrcByte	DINT	L#0	No default values
PCU_Error	BOOL	FALSE	Partner PCU is not in the run or connection error
ParamError	BOOL	FALSE	Parameter error

18.2.4.4 Time Job

Basic principles

- With a time job, the current time and date of system data block DB 701 in the PCU of the parameterized job is written to system data block DB 701 of the partner PCU so that its time and date are set.
- If a time job (type=16) is parameterized, an entry is automatically made for a PUT job to set the time and date in the other PCU. The addresses in the job are preset for system data block DB 701 in accordance with the SIMATIC S7 standard.
- The ID for the time job is then changed to type=17. The user can now adjust the addresses if the current time and date are made available via another data interface.

Example

- Using a time job, the current time and date are to be sent to PCU 8 every 60 seconds Here, the transmitted values are to be used to set the time.
- You can find how to assign system data block DB 701 the current time, date and values required to set the time in the data description of DB 701.

XC_JOB_32 (DB 705)

NAME	TYPE	Value	Comment
PCU	INT	8	Send data in PCU 8
Type	USINT	17	Job type: 17=time job after presetting
Enable	BOOL	TRUE	Enables job
Time	INT	60	Send time every 60 seconds
Src	INT	701	Source DB 701
SrcArea	BYTE	B#16#84	Source storage area DB
SrcByte	INT	182	Byte address 182 in source DB 701: Time, date
NumByte	INT	8	Transfer 8 bytes
Dest	INT	701	Destination DB 701
DestArea	BYTE	B#16#84	Destination storage area DB
DestByte	DINT	L#190	Byte address 190 in destination DB 701: Time, date
FltTime	INT	0	No default values
FltSrc	INT	0	No default values
FltSrcArea	BYTE	B#16#0	No default values
FltSrcByte	DINT	L#0	No default values
PCU_Error	BOOL	FALSE	Partner PCU is not in the run or connection error
ParamError	BOOL	FALSE	Parameter error

18.2.4.5 Temporary Job

In contrast to normal orders, which are repeatedly initiated on expiry of the parameterized time, a temporary order is only initiated once.

An example of a system function which is handled via a temporary order is the start of a plant section in another PCU via the ASTA module.

Users can initiate a temporary order by calling up function FC 685 'XC_JOB_USER_IF_FC' with iMode=1. The order in FC 685 is assigned the same parameters used for the parameterization of an order in module ‘XC_JOB_32’.

If a temporary order is to be initiated, the orders are searched for free data records. An order is free if the PCU no. of a data record is set to zero. The temporary order is entered in the first free data record.

Note

XC_JOB_32 data records to be kept free

Temporary orders are only executed when at least fifty XC_JOB_32 are free and activated. This has to be taken into account when setting the number of data records.

If all data records are assigned, a temporary order cannot be initiated, i.e. it is also not possible to start the plant section of another PCU via the ASTA module.

Once the temporary order has been initiated, the PCU no. is set to zero again and the enable bit is reset. This means that the order is free again.

If processing is error-free, function FC 685 is exited with RLO=0, if an error occurred, this setting is RLO=1.

If an error occurred, the error number is passed on to the 'iRetVal' parameter. If execution was error-free, the 'iRetVal' parameter is assigned the number of the processed order. Analysis of the 'iRetVal' parameter is carried out by the user.

iRetVal parameter

The values of the 'iRetVal' parameter have the following meaning:

IretVal	Description
> 0	Number of processed order
< 0	Error
- 1	Cannot find free order
- 2	'XC_JOB' has no data records
- 3	Incorrect iMode (not 1 or 2)
- 4	PCU not in ‘XC_PCU’

Module FC 685 also has a second function (iMode=2) which lets you set the actual time to zero while an order is running, i.e. an order can be initiated immediately. This enables event-dependent triggering of orders.

Example: Initiating a temporary order (iMode=1)

A temporary order is to be initiated.

Function FC 685 is called up for this purpose. FC 685 is assigned the same parameters which would be used to parameterize this order in module 'XC_JOB'.

The following order is supposed for our example:

Using a GET order, 240 data bytes are to be retrieved from PCU 4, data block DB 200, from DBW 800 onwards.

The received data is to be stored in data block DB 400, from DBW 150 onwards.

CALL FC685		Call FC 685 'XC_JOB_USER_IF_FC'
Imode	:=1	Initiate temporary order
IJob	:=0	Find free order
IPCU	:=4	Get data from PCU 4
IType	:=14	Order type 14: GET
BoEnable	:=1	Enables order
BoTmpJob	:=1	Enter temporary order
iSrc_Dtype	:=2	Data type always 2 = byte
iSrc_Area	:=84	Source storage area DB200
iSrc_DB	:=200	Source DB 200
ISrcByte	:=800	Byte address 800 in source DB 200
ISrcBit	:=0	Bit address in source DB always 0
iSrc_Num	:=240	Number of bytes to be transmitted
iDest_Dtype	:=2	Data type always 2 = byte
iDest_Area	:=84	Destination storage area DB400

iDest_DB	:=400	Destination DB 400
iDest_Byte	:=150	Byte address 150 in destination DB 200
iDest_Bit	:=0	Bit address in destination DB always 0
ITime	:=0	Do not enter setpoint time as temporary
IActTime	:=0	Current time = 0, initiate immediately
IRetVal	:=0	Return value of FC 685

Analysis of the RLO and the 'iRetVal' value tells the user whether processing of the FC 685 call was error-free.

If an error has occurred, the 'iRetVal' value contains the ID of the cause of the error.

Example: Initiating a running order (iMode=2)

A certain event has occurred which means that an existing order, normally repeatedly initiated at parameterized time intervals, is to be initiated immediately and not after the time period.

To do this, function FC 685 is called up when the event occurs Only the 'iMode', 'iJob' and 'iPCU' are relevant in FC 685. Specification of all other parameters in FC 685 is irrelevant.

The following order is supposed for our example:

Order 44 has been parameterized in the PCU to interchange data with PCU 7 every 60 seconds. Due to the occurrence of a specific event, this order needs to be initiated immediately and not when the set time period has elapsed.

Function FC 685 is called up when this event occurs and the 'iMode', 'iJob' and 'iPCU' parameters are assigned.

CALL FC685		Call FC 685 'XC_JOB_USER_IF_FC'
Imode	:=2	Set time to zero
IJob	:=44	Process order 44
IPCU	:=7	Order for PCU 7
IType	:=0	Irrelevant
BoEnable	:=0	Irrelevant
BoTmpJob	:=0	Irrelevant
iSrc_Dtype	:=0	Irrelevant
iSrc_Area	:=0	Irrelevant
iSrc_DB	:=0	Irrelevant
ISrcByte	:=0	Irrelevant
ISrcBit	:=0	Irrelevant
iSrc_Num	:=0	Irrelevant
iDest_Dtype	:=0	Irrelevant
iDest_Area	:=0	Irrelevant
iDest_DB	:=0	Irrelevant
iDest_Byte	:=0	Irrelevant
iDest_Bit	:=0	Irrelevant
ITime	:=0	Irrelevant
IActTime	:=0	Irrelevant
IRetVal	:=0	Irrelevant

Analysis of the RLO and the 'iRetVal' value tells the user whether processing of the FC 685 call was error-free.

If an error has occurred, the 'iRetVal' value contains the ID of the cause of the error.

18.2.5 Starting a Plant Section via Cross Coupling

Basic principles

The ASTA module starts a plant section. The function is specified by parameterizing the data record (see chapter, ‘Blocks’-> ’ASTA’). The plant section is started by setting the corresponding flag bit (F 672.0 to F 683.7).

If ASTA types 9,10 or 12 are used, the plant section of a remote PCU is started.

For this plant section start in a different PCU, the cross coupling must be established between the participating PCUs.

If the start of a plant section is performed in a different PCU, there will be a search for a free job set in the block 'XC_JOB_32'. A temporary job is entered in this free job set for the different PCU. The start parameters are transformed to the different PCU via cross coupling. If no free job is available, no start is performed, but an error is displayed.

The data for starting the plant section is received in the data record assigned to this plant section in module 'XC_ASTA_RCV' of the remote PCU. Data record 1 is assigned to plant section 1, data record 2 to plant section 2, ..., data record 64 to plant section 64.

Once the data is received and the start conditions have been checked (see chapter, ‘Blocks’-> ’ASTA’), the plant section is started.

If the conditions for the start are not fulfilled, the start doesn’t occur.

The acknowledgment, either positive or negative, is sent back to the source ASTA in the source PCU where it can be appropriately analyzed by the user.

By using the parameterization application and selecting module 'XC_ASTA_RCV', you can select the last received start for any plant section.

18.2.6 XC_ASTA_RCV - Cross Coupling ASTA Receive

Important:

- For this function the user need not produce any configuration. The objects of the XC_ASTA_RCV class are fully managed by the system. Therefore the following description is for diagnostic purposes only.
- In this block the plant section start commands for the 64 possible plant sections are received, which are initiated in conjunction with the ASTA block and cross coupling.
- If the ASTA for a plant section is initiated in one PCU, the data is transferred to another PCU by cross coupling
- where this data is received at the data sets of the XC_ASTA_RCV block.

- After receiving the data and examination of the start conditions, (see block ASTA), the plant section is started.
  If the conditions for the start are not fulfilled, the start doesn't occur.
- The acknowledge (pos. or neg.) is sent back to the source ASTA.

Note

Not for PCU type S7-1500!

The S7-S7 cross coupling and the XC_ASTA_RCV parameter assignment is not available for PCU type S7-1500!

Parameterization XC_SEQS_RCV / DB 740

PCU parameter set (max. 64 per PCU)

Name	Access	D.Type	A.Type	Value	Comment
SrcPCU	NL	USINT	ENG	0	Source PCU
Status	NL	USINT	RT	0	0 calm 1..200 active 255 parameter incorrect
SrcASTA	NL	USINT	ENG	0	SEQS in source PCU
DestSEQ	NL	USINT	ENG	0	Destination unit
Year	NL	USINT	ENG	0	Year
RecipeType	NL	USINT	ENG	0	Recipe type
Recipe	NL	INT	ENG	0	Recipe number
Order	NL	INT	ENG	0	Order number
Batch	NL	INT	ENG	0	Batch number

18.2.7 Datagram type 25 of the route control

Minimization of data traffic between the controls is achieved by the implementation of datagram type 25 for the route control. The function attempts to form a group message consisting of such frames in DB451 – DB481. A station receiving such a datagram returns an acknowledgement to the transmitting station.

- To ensure proper functioning of these group datagrams, the appropriate dataset must contain the PCU number.
- This configuration is based on the table in section "PCU allocation to data set (Page 1191)".
- Block mode with acknowledgment must be enabled, To do this, the attribute "ModeBlock" (DBB 14) = 2 must be set in the parameterization "XC_PCU_32" (DB705).
- This setting can be made with the parameterization "XC_PCU_32 → Global data → Advanced Attr.".

18.3 Redundant Bus

18.3.1 General

- There can be built up two separated bus systems. To do this, two CP443 must be inserted into Simatic S7-400.
- In the configuration of the PCUs there is entered a normal as well as a reserve interface.
- If an error is found in an interface, it will automatically trigger transfer to a different interface.
- This interface is active until an error occurs here too and a transfer is performed to the first interface.

18.3.2 Parameterization of the Get/Put jobs

For S7 the following entries for the reserve interface are to be entered with configuration under XC_PCU_32.

Enable = 1	Release Bus 1 und 2
	Bus 1 only processes, if ConnID <> 0
ConnID_1 = 0	Bus 1 is processed. ConnID is formed internally from the PCU number. The hexa identifier of the bus number is not displayed.
ConnID_2 = Yxx	0=Bus 2 not processed
If <> 0, Bus 2 will be processed	

Declaration:			
Yxx = Hexa identifier			
Y = Bus identifier	e.g.:	2	i.e. Bus 2
xx = PCU number	e.g.:	20	i.e. PCU=32 (as 32 = 20Hexa)

For both bus connections the following connections must be entered:

Table 18-5 Example: Connections for PCU4 to PCU32

S7 bus	Local ID	Partner ID	Partner	Type	Active
1	4	20	AS_4_MPI_14	S7 conn.	Yes
2	204	220	AS_4_MPI_14	S7 conn.	Yes

Partner ID = ConnID in the parameter assignment		
Active:	Yes	If partner ID bigger than local ID
Active:	No	If partner ID smaller than local ID

Note

There can also be used the local ID 104 and partner ID 120 for bus 1, i.e. there must be entered these values with the configuration for ConnID_1.

IDs 4 and 20 are used automatically, i.e. no values have to be entered for ConnID_1.

Appendix A

A.1 Call parameters of the applications

A.1.1 Call parameters of the applications

The following sub-chapters contain descriptions of the call parameters of distinct applications. With these, it is possible for example to start the application within a process picture by means of the ‘Exec PC’ object and also establishing a defined workspace.

A.1.2 Call parameters of ‘BaliEdit.exe’

Baliedit.exe <View> <Year> <Order type> <Order number> <Order status> <Batch number> <Batch status>

- <View>
  Order ->The 'Order list' view is opened
  Batch ->The 'Batch list' view is opened
  BaMon ->The 'Batch monitoring' view is opened
- <Year>
  nn -> Only data records for this calendar year are selected
- <Order type> default: 0
  0 -> The selection of data records occurs independently of the recipe type
  Number -> Only data records with this recipe type are selected
- <Order number> default: 0
  0 -> The selection of data records occurs independently of the order number
  Number -> Only data records with this order number are selected
- <Order state> default: x
  x -> the display of the data records occurs independently of the order state
  S -> (Started) only those data records are selected which have the order state 'started'
- <Batch number> default: 0
  0 -> The selection of data records occurs independently of the batch number
  Number -> Only data records with this batch number are selected
- <Batch state> default: x
  x -> the data sets are displayed independently of batch state
  R -> (Ready) only those data records are selected which have the batch state 'ready to start'
  S -> (Started) only data records are selected, which have the batch state 'started'

Example:

```
baliedit.exe Batch 16 13 20 x 0 x
```

The batch list application is opened with the following selection criteria:

- “Batch list” view
- Year = 2016
- Order type = 13
- Order number = 20
- Order status = all
- Batch number = all
- Batch status = all

Note

No area may be selected via the command line. The own area is always taken here.

A.1.3 Call parameters for BM_BatMan.exe

Baliedit.exe <View> <Year> <Order type> <Order number> <Order status> <Batch number> <Batch status> <AREA=x>

- <View>
  Batch → The ‘Batch list’ view opens
  BaMon → (Batch monitoring) The ‘Unit allocation’ view opens
- <Year>
  nn → Only data records for this calendar year are selected
- <Order type> default: 0
  0 → The selection of data records occurs independently of the recipe type
  Number → Only data records with this recipe type are selected
- <Order number> default: 0
  0 → The selection of data records occurs independently of the order number
  Number → Only data records with this order number are selected
- <Order state> default: x
  x → The display of data records occurs independently of the order status
- <Batch number> default: 0
  0 → The selection of data records occurs independently of the batch number
  Number -> Only data records with this batch number are selected
- <Batch state> default: x
  x → The display of data records occurs independently of the batch status
- <AREA=n> default: Local AREA
  n → The view opens for this AREA

Example:

```
BM_BatMan.exe Batch 16 1 18 x 0 x AREA=3
```

Batch management opens with the following selection criteria:

- “Batch list” view
- Year = 2016
- Order type = 1
- Order number = 18
- Order status = all
- Batch number = all
- Batch status = all
- AREA = 3

A.1.4 Call parameters of BlrEdit.exe

Fundamentals

BLREdit.exe can be used with the following call parameters:

- /oid <PCU number>, <Name of class> (e.g. "BLR1"), <Instance number>
  Example:
  BlrEdit.exe /oid 7,BLR2,42
- /online - sets the online mode
- /faceplate - representation as faceplate (automatically online)

Main application case:

BlrEdit.exe /oid 7,BLR2,42 /faceplate

A.1.5 Call parameters of ‘Controlrec.exe’

Case 1: Controlrec.exe <p1> <p2> <p3>

<p1>	AREA=x (x=Area number)
<p2>	PCU x (x=PCU number … note space character)
<p3>	Unit number

Example: `controlrec.exe AREA=1 PCU 1 64`

Case 2: Controlrec.exe <p1> <p2> <p3> <p4> <p5>

<p1>	AREA=x (x=Area number)
<p2>	Recipe type
<p3>	Year

<p4>	Order number
<p5>	Batch number

Example: `controlrec.exe AREA=1 5 7 46 1`

A.1.6 Call parameters of ‘Dataconn.exe’

Case 1: dataconn.exe <p1> <p2> <p3>

- <p1>PCU=x (x=Number of PCU)
- <p2>CID=x (x=Number of class, must be present together with IID)
- <p3>IID=x (x=Number of instance, must be present together with CID)

Example: dataconn.exe PCU=1 CID=2 IID=1

Case 2: dataconn.exe <p1> <p2>

- <p1>“PCU=x” (Number of PCU)
- <p2>Name=”Name of data connection”. (Note: Quotation marks are required)

Example: dataconn.exe PCU=1 Name="1 SIV 101"

A.1.7 Call parameters of ‘EDITREC.exe’

Case 1: Editrec.exe <p1> <p2> <p3> <p4>

<p1>AREA=x (x=Area number)

<p2>true/false (distinction as to whether the master recipe or the recipe procedure should be opened (true=MR/false=RP)

<p3>recipe type number on MR or recipe procedure number on RP

<p4>recipe number on MR or empty on RP

Example: editrec.exe AREA=1 true 5 1

Case 2: editrec.exe <p1> <p2> <p3>

Example: editrec.exe AREA=1 false 16

A.1.8 Call parameters of ‘Kurven.exe’

Case 1: kurven.exe <p1> <p2> <p3> ... <p13>

- <p1> Display type
  c: application is started 'normally' with menu
  p: application window without menu
- <p2>: Image name
  Name of the trend image without path but with file extension (Weeks -> *.ltw, Batches -> *.ltb or Short-term picture -> *.st)

- <p3>: Year
  Specification of the year for batches or weekly archive. For short-term archives the value is not necessary. If '-' then the actual year is substituted
- <p4>: Week
  Only relevant for weekly archives. for other archive types use 0. If '-' then the actual calendar week is substituted
- <p5>: Order number
  Only relevant for batch archives. for other archive types use 0. Substitution with $Variable name for actual order number is possible.
- <p6>: Batch number
  Only relevant for batch archives. for other archive types use 0. Substitution with $Variable name for actual batch number is possible.
- <P7>: Recipe category
  Only relevant for batch archives. for other archive types use 0. Substitution with $Variable name for current recipe category is possible.
- <p8>: Area selection
  AREA=x (x=area number)

More optional parameters for mode "application without menu".

- <p9>: title (maximum length 28 characters)
- <p10>: horizontal (x) position of the upper left corner of the app window
- <p11>: vertical (y) position of the upper left corner of the app window
- <p12>: window width
- <p13>: window height

Example: kurven.exe p T_PCU1LZ.ltw 7 7 (only 4 parameters because weekly archive)

Case 2: Substitution with '$Variable name':

- A variable is defined in the process picture which contains the value for the substitution and a name is configured for this variable
- The variable name is given for the parameter as $Variable name. The actual parameter value is substituted with variable value by the runtime system

Example: Kurven.exe p Test.ltb - 0 $Auftrag $Charge $Rkat

- An integer variable with the name 'batch' has to be created in the process picture which shows the batch number from the sequencer record. This variable is called "Batch".
- In the same way a variable named 'order' for the Order number has to be created

A.1.9 Call parameters of 'Lzsys.exe'

LZSYS.exe <p1> .. <p?>

<p1> … <p?>

- /X:x (X-coordinate of main window)
- /Y:x (Y-coordinate of main window)
- /T → Display main window TopMost
- /F → Activate process picture in full screen mode without menu bar/toolbar/status bar
- /N → Change picture, do not start a new instance
- /A:x (x=Area for MultiClient operation)
- <p?> picture file name (*.bik)

Example: lzsys.exe /X:100 /Y:100 /T /N /A:1 t_startmenu.bik

A.1.10 Call parameters of ‘PARAM.exe’

PARAM.exe <p1> <p2> <p3> <p4>

- <p1> PCU number
- <p2>1/0 -> Online/Offline
- <p3>Block Class name
- <p4> instance number

Example: param.exe 1 1 ANA 23

A.1.11 Call parameters of ‘PARATXT.exe’

PARATXT.exe <p1> <p2> <p3> <p4>

- <p1> PCU number
- <p2>Language number
- <p3> File name (Important: without extension ‘txt’)
- <p4>Line number

Example: paratxt.exe 1 10 Sondwert 120

A.1.12 Call parameters of ‘PASSWCHK.exe’

Passwchk.exe <p1> <p2>

- <p1>min. required Password Level
- <p2>Application to be started

Example: passwchk.exe 10 biko.exe

A.1.13 Call parameters of ‘PCU_SERV.exe’

pcu_serv.exe <p1>

- <p1>not present or ‘–mini’ -> view/window size normal or minimized

Example: pcu_serv.exe –mini

A.1.14 Call parameters of ‘RCS_Online.exe’

Case 1: RCS_Online.exe <p1> <p2> <p3>

- <p1>Ax (x=Area)
- <p2> PCU Number
- <p3>Unit number

Example: RCS_Online.exe A1 1 64

Case 2: RCS_Online.exe <p1> <p2>

- <p1>Ax (x=Area)
- <p2>Route ID

Example: RCS_Online.exe A1 1

A.1.15 Call parameters of ‘REGLER.exe’

Regler.exe <p1> <p2> <p3> <p4> <p5> <p6> <p7>

- <p1> PCU number
- <p2>controller type 3PK or PID
- <p3>Instance number
- <p4>Dimension description (max. 10 chars)
- <p5>decimal places
- <p6>X-Position
- <p7>Y-Position

Example: regler.exe 1 3PK 1 Test 3 100 200

A.1.16 Call parameters of ‘SeqCtrl.exe’

Seqctrl.exe <p1> <p2>

- <p1>AREA=x (x=Area number)
- <p2>y

Example: seqctrl.exe AREA=1

The value y (1..3 chars) indicates the number of the kpos file (path ..\windcs\etc\kposnnn.ini). A further instance of the application is started by the call depending on the INI switch in file ..\windcs\sys\seqctrl.ini (see Manual 12_Operation and control of batches …).

A.1.17 Call parameters of ‘SRPNEU.exe’

Note

The command line serves for protocol printout only. No dialog window is created.

Srpneu.exe <p1> <p2> <p3> <p4> <p5> <p6>

- <p1>year
- <p2>Recipe type number
- <p3>Recipe name
- <p4>Order number
- <p5>Batch number
- <p6>Unit number

Example: Srpneu.exe 7 5 rcs_bali 100 234 64

A.1.18 Call parameters for Status_S7.exe

Status_S7.exe <p1> <p2> <p3> <p4>

- <p1> PCU number
- <p2>Block type + Block number (e.g. FB720 or FC1020)
- <p3>Block network number
- <p4> 1/0 -> Online/Offline

Example: Status_S7.exe 1 FC1413 3 1

A.1.19 Call parameters of transbst.exe

transbst.exe <p1> <p2>

- <p1> /U:<file name> .... Upload with config file <file name>
- <p1> /D:<file name> .... Download with config file <file name>
- <p2> /S ….Program execution with hidden window

Examples:

Call	Function
transbst /u:upload	transbst.exe executes the commands in the file `<poj-path>/trans/upload.trs`. The application window of "transbst.exe" is visible during the execution.
transbst /u:upload /s	transbst.exe executes the commands in the file `<poj-path>/trans/upload.trs`. The application window of transbst.exe is hidden during the execution.

See also section Starting automatic block transfer at the command line prompt (Page 172)

A.1.20 Call parameters of ‘WARTDAT.exe’

Wartdat.exe <p1> <p2> <p3>

<p1>	PCU number
<p2>	ICM Group
<p3>	1 = Switch cycle counter or 2 = Operating hours counter

Example: `wartdat.exe 1 1 1`

A.2 USERSTART.EXE

A.2.1 Functionality

The application makes it possible to start any application with call parameters on any IOS through a PCU event, for example modification of a data word.

As a main use case the call of process images in accordance with a PCU variable is described here - although other use cases are possible of course.

Here a PCU data word is assigned to the IOS application "LZSYS.EXE". During the setting of this data word through a PCU user block the application changes into the related picture. The allocation of the value of variables to the process diagrams occurs via INI file.

This function can be individually activated for every IOS (Client or Server).

A.2.2 Configuration

A.2.2.1 Configuration file "userstart.ini"

Basics

The file is located in the !`<proj-path>\sys`' system directory.

Example:

Process images on IOS 102 should be called automatically, dependent on the value change of a Data-TAG in PCU 1, DB709 (Special values), Data word 2.

Contents of file ‘USERSTART.INI’:

```
[IOS102]

DataSource = 1,0,709,2

StartApplication = LZSYS.EXE

StartList = userstart_ios102.img.txt

beenden = 1

ErrorHandling = 1

Default = 2
```

Parameters	Description
[IOSx]	Beginning of IOSx Section x is IOS Number. Several sections are possible in the INI file, so all participating IOS stations may share identical files, which, however, must be copied manually after changing Note: In case of incorrect IOS number assignment the complete section is ignored
DataSource=PCU,DBType,DBNo,DWNo	Data TAG Address – comma-separated, serving as an index for the text file with command line parameters at runtime Possible values for DBType are: 0: Data block
StartApplication=	Application to be started by value change of the data word
StartList=	Text file which contains the calling parameters (see example file below) Directory "`<proj-path>\sys`"
End=	Activating (=1) or deactivating (=0) the Application ‘Close button’
ErrorHandling=	Behavior when the TAG value is outside the range of available text line numbers 0 = nothing happens, 1 = MsgBox, 2 = Default Line index is used (as defined)
Default=	Default Line index, if variable value is outside the range of available text line numbers

A.2.2.2 Text file for calling parameters

The file is located in the "…\windcs\sys" system folder.

Example

Call process images via “LZSYS.EXE” application (see above)

Structure of the configuration file with the ‘LZSYS’ call parameter 'image name'.

Line No.	Content of ‘userstart_ios102.img.txt’	Assigned TAG value
1	<..blank line..>	0
2	Sudhaus\CIP.BIK	1
3	Sudhaus\LT.BIK	2
4	Sudhaus\MALT-1.BIK	3
5	Sudhaus\MALT-2.BIK	4
6	Sudhaus\MALT-6.BIK	5
7	Sudhaus\MST.BIK	6

Line No.	Content of ‘userstart_ios102.img.txt’	Assigned TAG value
8	Sudhaus\MTK.BIK	7
9	Sudhaus\PRT.BIK	8
10	Sudhaus\WATER.BIK	9
11	Sudhaus\WHP.BIK	10
12	Filtration\KEG.BIK	11
13	Filtration\PALFILT.BIK	12
14	Filtration\PASTEUR1.BIK	13
15	Filtration\TFF-CIP.BIK	14

A.2.3 Starting and Closing the Application

The application should be auto-started at system startup which is achieved via the appropriate entry in the ‘…\windcs\sys\prosched.ini’ configuration file (see manual ‘*Administration*’ chapter 7 ‘*Scheduler*’ for this).

Example of the entries within Prosched.ini:

```
[Global]
Processes=...,Userstart,...
...
...
[userstart]
Application=d:\windcs\sys\userstart.exe
;Window caption to identify mainwindow
Window=userstart
;Start this application if not running and check for running
Type=PERMANENT
;At own shutdown, shutdown although this application
ShutDown=1
```

The application UserStart.exe may be closed in the following ways:

- Application ‘Close button’ (if enabled)
- By closing the scheduler ProSched.exe (if configured)
- By closing the PCU server process

A.3 Trace Server Diagnostics

A.3.1 General

The trace server installed with the system is not intended for operating personnel. Rather, it is a special tool for detailed diagnostics by experts, e.g. commissioning personnel. The information displayed is complex but will be used by the hotline for diagnostics and fast identification of an error cause. With some practice you can gain valuable information for the commissioning phase, which is why the following chapter briefly explains this tool.

The Trace Server application always works locally rather than through the network. To run diagnostics the application must therefore always be started on the processor, which contains the application to be scanned.

A.3.2 Procedure

- Start the trace server via the basic menu / Diagnosis ... .application "TraceServer"
- Maximize the window after the first startup to make it easier to manage.
  - In the selection field choose one of the applications on which you would like to run the diagnostics. The Trace Server will only display currently active applications, i.e. applications currently stored in the main memory and registered with the Trace Server.
  - Define the scope of messages to be displayed (diagnostics, information, warnings, errors)

A.3.3 Meaning of display elements

Menu item	Notes
PIN needle	If selected, the Trace Server will always appear as the top application (topmost)
Trace	Recording of events ("tracing") ON/OFF
Scroll	Automatic scroll ON/OFF Switching on the "Scroll" function will ensure that the latest events always appear at the bottom of the screen moving older events up.
File	Recording of events in a file ON/OFF (Hotline, system control centers and developers often ask for such a file to be forwarded to them in packed format ...)
Clear	Delete export list
All Off	Reset all module event types assigned, i.e.: stop logging

Menu item	Notes
[Combobox]	Displays all modules that can deliver Trace outputs. You can select them individually and use the following four buttons to assign the event types to be monitored. The first entry in the Combo box <tracesvr [xxxx]> - <ALL [yyyy]> allows you to address all modules at the same time
Diagnostics, Infos, Warnings, Errors	Event types to be monitored **Note** Diagnostics levels 3 and 4 should not occur in normal operation (warnings, errors)

A.3.3.1 Module selection list

Displays the following information

Left:	Name of application (EXE), as it appears in Task manager, including its process ID
Right:	Name of module within the application

For example, if you have started up the "Plant overview" application (internally: seqctrl.exe), the selection list will display all modules and libraries that are related to it and can be found in the memory. You will see that "SEQCTRL" will load several other modules. You can configure their monitoring individually. This is optional. If you are equally interested in all events of a main module (here: SEQCTRL), you can use the sub-entry with the same name as the main module (here: SEQCTRL [1824] – SEQCTRL).

A.3.3.2 Main list

A list of incoming Trace messages will be displayed in the top part of the main window.

A.3.3.3 Filter list

If the settings contain an entry for the filter list, a sub-list will appear in the lower part of the main window. It contains filtered lines from the main list. This will allow you to detect rare Trace messages immediately.

Click on a line in the filter list to automatically jump to the corresponding line in the main list.

A.3.3.4 Status line

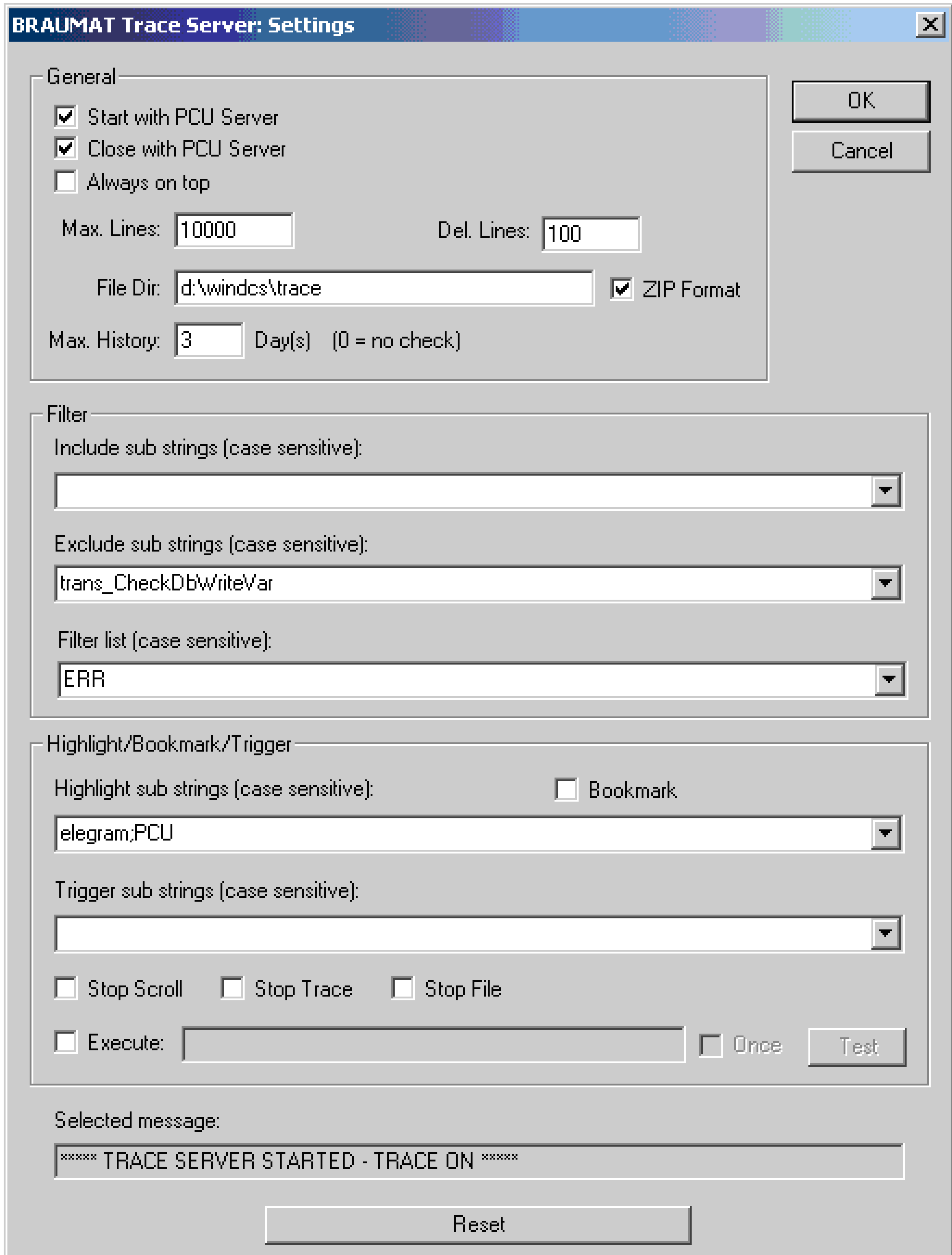
Entry (left to right)	Meaning
55 / 10000	Current/maximum number of possible lines
44[0000]	Bookmarked line [Event types each ON/OFF: 1/0]

Entry (left to right)	Meaning
File: C:\WINDCS\tmp\TS021119.txt (max:0)	Trace file currently in use If you are saving in a file, this will be the file the Support Service will ask you to send to them in packed format ...
Profile: USER	Currently active profile (see below)
IOS 1 19.11.02 09:56:54	(Superordinate information)

A.3.4 Settings

The setting options described in the following chapter provide the tool with flexibility in relation to filter, bookmarking and trigger options. For example, you can highlight messages with a particular string in color to make them easier to distinguish from other messages or even start the process of a defined application.

You can open the configuration dialog by double-clicking or via the context menu in the message window.



Element	'General' section
Start / Close with PCU Server	TraceSvr starts/stops automatically
Max. Lines	Max. number of lines for display
Del Lines	Number of lines to be deleted, when Max. Lines is reached
File Dir	Path for saving Trace files. A new file is created for each day.

Element	'General' section
ZIP-Format	ZIP compression option during writing process.
Max. History	Max. number of held Trace files. Older Trace files are deleted automatically. If 0 is preset, no Trace files will be deleted. Note: Depending on the filter settings, Trace files can become very large if not ZIP compressed (1GB per day)!

Element	'Filter' section
Include sub strings	If a message contains the string, the line will be recorded.
Exclude sub strings	If a message contains the string, the line will be discarded.
Filter list	If a message contains the string, the line will be displayed in a sub-window.
Element	'Highlight/Bookmark/Trigger' section
Highlight sub strings	If a message contains the strings 'Telegram' or 'PCU', the line will be highlighted in color.
Bookmark	In addition to 'Highlight', 'Bookmark' will also make it easier to find particular items.
Trigger sub strings	If a message contains the string, the following options will be carried out.
Stop Scroll	Trigger: Switch off 'Scroll' to ensure that the message remains visible.
Stop Trace	Trigger: Switch off Trace
Stop File	Trigger: Switch off writing to file
Execute	Trigger: Execute specified action, e.g. send message to another processor via batch.
Once	Carry out 'Execute action' only for the first trigger
Test	Carry out 'Execute action' immediately for test purposes

All Filter/Highlight/Trigger strings:

- Several strings may be entered separated by a semicolon.
- Case sensitive!

Element	Selected Message
	Currently selected Trace entry to be copied and inserted as filter text.

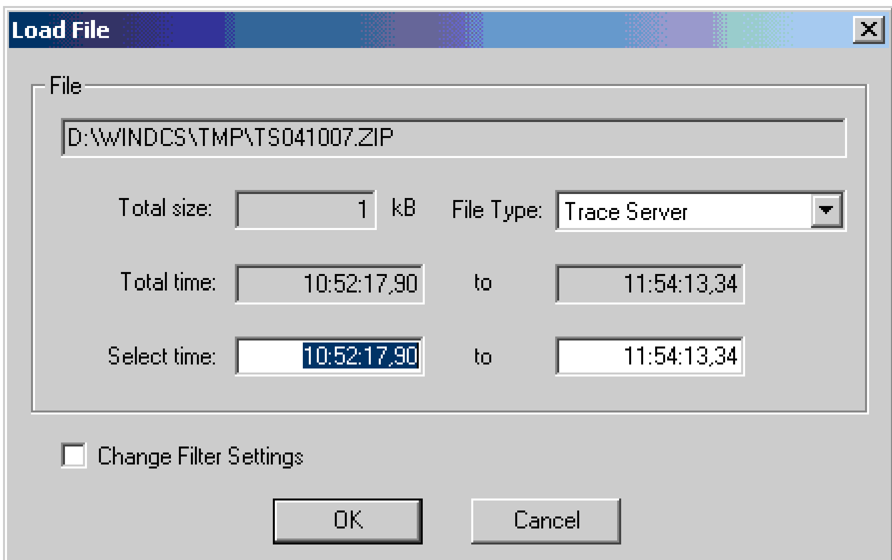
A.3.4.1 Trace Server: Profile

You can save all settings for modules and their events, as well as your chosen filters in a "Profile" and load this profile when required. Right-clicking will guide you to the appropriate menu (position is irrelevant). All menu entries are largely self-explanatory.

A.3.4.2 Trace Viewer (TRACEVWR.EXE)

This tool is very similar to the Trace server described above. The only difference is that it cannot perform Trace recordings, but only load recorded Trace files. Accordingly, the above-mentioned buttons in relation to recording options (Trace, Scroll, All Off, Diagnostics, Infos, Warnings, Errors) are deactivated.

To ensure efficient browsing of trace files, the start and end time for the required messages can be specified when opening the tool.



Element	'File' section
Total size	Size of the entire Trace file
Total time	Timeframe for all included Trace entries
Select time	Timeframe for required Trace entries You can also enter a partial string
File Type	The Trace Viewer can also display message archives and old PCU Server Traces. The type must be right for these details to be displayed in the right format. However, usually it will be determined automatically on the basis of the file name. If this fails, e.g. because the file name was changed, it can be entered manually.
Element	Change Filter Settings
	If this option is selected, a filter dialog will appear even before the Trace lines are imported.

Note

You can branch off to the Trace Viewer by using the context menu in the Trace Server. The current Trace file (if it has been saved before) will load automatically and will be displayed with the last valid filter settings.

A.4 Recipe system file storage and Ini switch

A.4.1 Process parameters, file location

Process parameter DBF definitions

The definition of the process parameters DBF is stored in '<proj-path>\recipe\def\mainrxxx.def' (xxx number of the recipe type).

Process parameters definition file

The definition of the process parameters is stored in the file '<proj-path>\recipe\mainrec \mainpara.xxx'.

This definition file is created and edited with the application Recipe editor / <select the recipe category> / Edit / Process parameter definition.

Process parameter values

The process parameters themselves are stored in the "<proj-path>\recipe\mainrec \mainrxxx.dbf" file.

This file is created and edited with the application Recipe editor / <select the basic recipe> in the parameter window.

A.4.2 Activating the process input list using Ini switches

The functionality 'process input list' may be disabled by setting the following configuration file entry: ‘windcs\recipe\project\plant.ini’ [Componentlist] Enable=0.

A.4.3 Recipe process input lists, file location

The process input lists are stored in the directory ‘windcs\recipe\complist\cl_rez\rez_xxx’. Every process input list is stored in one dBase file (‘crec_yyy.dbf’).

This list may be downloaded from a CIS system to the IOS and uploaded again. See also 'connection component list – recipe'.

A.4.4 Recipe unit procedure sequence

A CAS data record must be generated for each unit, which is used by the Order/Batch system.

Typically the same number of CAS data records are created as unit records exists.

The sequential order of the units is stored in files ‘windcs\recipe\basicrec.001\rexxxxx \soryyyyy.zzz’.

Where:

- xxxxx is the recipe procedure number (always combined in blocks of 50)
- yyyyy is the recipe procedure number (always combined in blocks of 50)
- zzz the line number (only the reference line may be edited), the default number of the reference line is 1

A.4.5 Synchronization

The synchronizations are stored in files ‘windcs\recipe\basicrec.001\rexxxxx\synyyyyy.zzz’.

Where:

- xxxxx is the recipe procedure number (always combined in blocks of 50)
- yyyyy the recipe procedure number
- zzz the line number (only the reference line may be edited), the default number of the reference line is 1

The file is created automatically by storing the recipe procedure. It may not be edited (influences recipe execution, changes are overwritten when saving a recipe procedure).

A.4.6 Alternatives

The alternatives are stored in files ‘windcs\recipe\basicrec.001\rexxxxx\altyyyyy.zzz’.

Where:

- xxxxx is the recipe procedure number (always combined in blocks of 50)
- yyyyy the recipe procedure number
- zzz the line number (only the reference line may be edited), the default number of the reference line is 1

A.5 Setpoint definition type ‘COND’

The setpoint is replaced by text from text lists described here.

The function makes it possible to select elements from groups again. The value in the AS is represented as a 32Bit value, which comprises the group and element number:

HiWord=<Group>

LoWord=<Element>

In the setpoint definition dialog the file name is defined without extension which makes it possible to assign two files with it here. The delivery state contains the following files as an example:

- `…<proj-path>\pcu.001\recipe\COND.cnd` → Definition of the subgroups (Level 2)
- `…<proj-path>\pcu.001\texte\COND.txt` → Group names (Level 1)

These may be defined per setpoint instance by the user. The element names (Level 2) are saved in custom files which are named "`…<proj-path>\pcu.001\texte\Tgroupx.txt`" with factory defaults.

See also

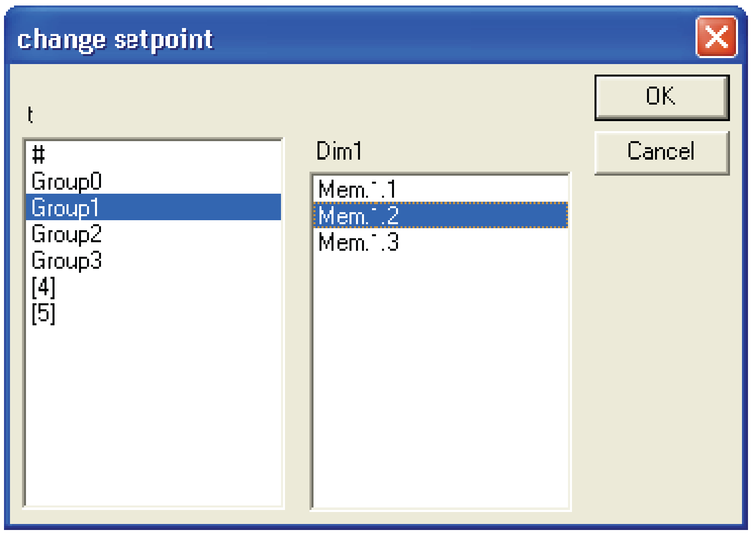
'Edit DFM definitions' dialog (Page 581)

A.5.1 Configuring the text files (contents = delivery state)

File:	Contents
…<proj-path>\pcu.001\recipe\COND.cnd	Syntax: <Type>, <Dim>, <Min>, <Max>, <Instancefile> TEXT,Dim1,1,3,TGroup1 TEXT,Dim2,1,2,TGroup2
…<proj-path>\pcu.001\texte\COND.txt	Syntax: <group-identifier> Group0 Group1 Group2 Group3
…<proj-path>\pcu.001\texte\TGroup1.txt	Syntax: <element-identifier> Mem.1.1 Mem.1.2 Mem.1.3
…<proj-path>\pcu.001\texte\TGroup2.txt	Syntax: <element-identifier> Mem.2.1 Mem.2.2 Mem.2.3

A.5.2 Change setpoint in ROP

With a double-click on the setpoint in the recipe procedure view or 'Sequence control', the following dialog is opened:



After selecting group/element and quitting with OK button, the setpoint is stored. At the same time the group index is stored as high word and the instance index as low word of the 32bit setpoint value.

Note on the delivery state:

- Selecting Group0 and Group3 > right column (instance) is empty

A.6 Converting recipes from earlier system versions

A.6.1 Recipe of Version V2 and Version V3

The recipe system V5 has a considerably extended range of functions and controls compared to the recipe system of the Versions V3 and V2.

The current system version supports only the V5 recipe system.

- Recipes of version V3 can be converted to the V5 recipe system using a conversion tool. To convert definition files for equipment modules (previously BOPs and SW), create unit classes, convert recipe procedures, and expand material management, you can use the "conv4to5.exe" conversion tool, which is available separately. This performs the named steps autonomously, for the most part. It converts the elements of recipe system version V3 into recipe system elements of version V5.
- Recipes of version V2 have to be converted into V3 recipes first using the previous master recipe editor and manual steps. There is no direct conversion from V2 to V5.

For both cases, you can obtain additional information via 'Siemens Customer Support'. You will find contact details in the preface of this manual.

A.6.2 Conversion of recipes created with system version V5.2

When you upgrade to the latest system version, the system converts the recipe files automatically when they are accessed for the first time. Instead of deleting the V5.2 files, the system renames these. The following files are affected by this automatic conversion process:

- Basic recipe header definition file
  "...\windcs\recipe\def\Rec_head.def" renamed "...\windcs\recipe\def\Rec_head.de~"
- Basic recipe header database files
  "...\windcs\recipe\head_XXX.dbf"renamed "...\windcs\recipe\head_XXX.db~"

As of V5.3, the following new definition files are used:

- Basic recipe header definition file "...\windcs\recipe\def\MR_HEADER.def"
- Basic recipe header database files "...\windcs\recipe\mheadXXX.dbf"

A.7 Weighing and weighers (S7-400)

A.7.1 Basics

A.7.1.1 Overview

Note

Note on CPU type SIMATIC S7-1500:

With the current system version, the system also supports the SIMATIC S7-1500 controller family.

The release is initially subject to the following limitations in this stage.

- GF weigher is not supported (modern solution for future version planned)
- TF weigher with SIWAREX M is not supported (outdated technology)
- SILO TANK data is not supported (modern solution for future version planned). The "Route Control System" option is not supported (modern solution for future version planned)
- For this reason, the following description applies only to use with the **SIMATIC S7-400** controller family.

Note

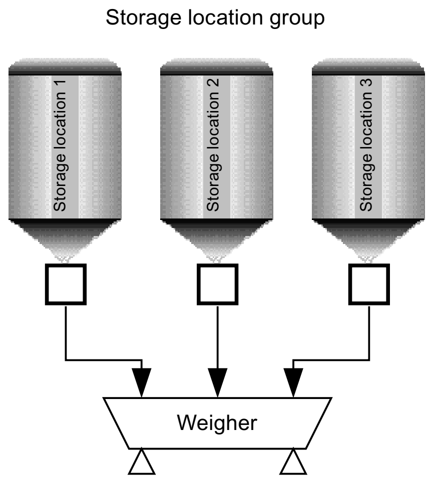
The 'process input list" functionality and component dosing is deactivated in the factory settings.

Starting with system version V7, the functionality of the process input list in connection with the component and storage management is deactivated by default. The options available in earlier system versions "Component management" with the sub view "Components" in the basic recipes and the corresponding recipe editor dialogs "Engineering/Process input" previously "Material stock" and "Engineering/Process input (previously "Depot locations" can only be used, when necessary after setting the following INI switch:

File `'<proj-path>\recipe\project\plant.ini'`

```
[Componentlist]

Enable=1
```

It is necessary to define storage locations and weighers for the automatic material weighing via the recipe system. In this regard you start with the following basic structure.



- The weigher object is defined in the system. A weigher is a unit in the sense of the recipe system.
- Material can be taken from one or several storage locations.
- A storage location group is assigned to a weigher. On one hand the silo group is a structured data block in the PCU and a file that consists of settings data for the weigher and for different materials.
- More materials can be assigned to a storage location group than storage locations are available.
- Materials can be enabled and disabled for a storage location group.
- Materials can have priority within the storage location group.
- The assignment of material and storage location is formed in two steps:
- Step 1 (Create control recipe)
  The weigher is determined by the defined and enabled materials. Thereby, as many weighing operations are added to the control recipe as weighings need to be executed. The weighing operations are inserted in the corresponding unit procedure.
- Step 2 (Processing)
  The weighing operation (EOP) searches by the material number in all storage locations of the storage location group and takes the material from the storage location with the highest priority.
  The value 1 is the highest priority, the value 255 the lowest one.

Note

In a few plants it may occur that several weighers are taken from one storage group. This case is also displayed on the above-demonstrated model. In the engineering, it is possible to put the storage silo groups together (see also the section "Several weighers on a storage group (Page 1266)").

A.7.1.2 Quantified project scope

- A maximum of 127 storage groups can be configured per stand-alone section.
- Each storage group can have a maximum of 255 storages.
- Priorities from 1 to 255 can be assigned in order to control the removal of material from storage locations with the same material.
- A maximum of 65,536 offline adjustment data records can be filed for each storage location group.
- A maximum of 16 weighers can be configured per PCU. Through user configuration the number can be increased to a maximum of 64.

A.7.1.3 Data flow

In the image below, the data flow between the individual parts which are participating in a single dosing are displayed.

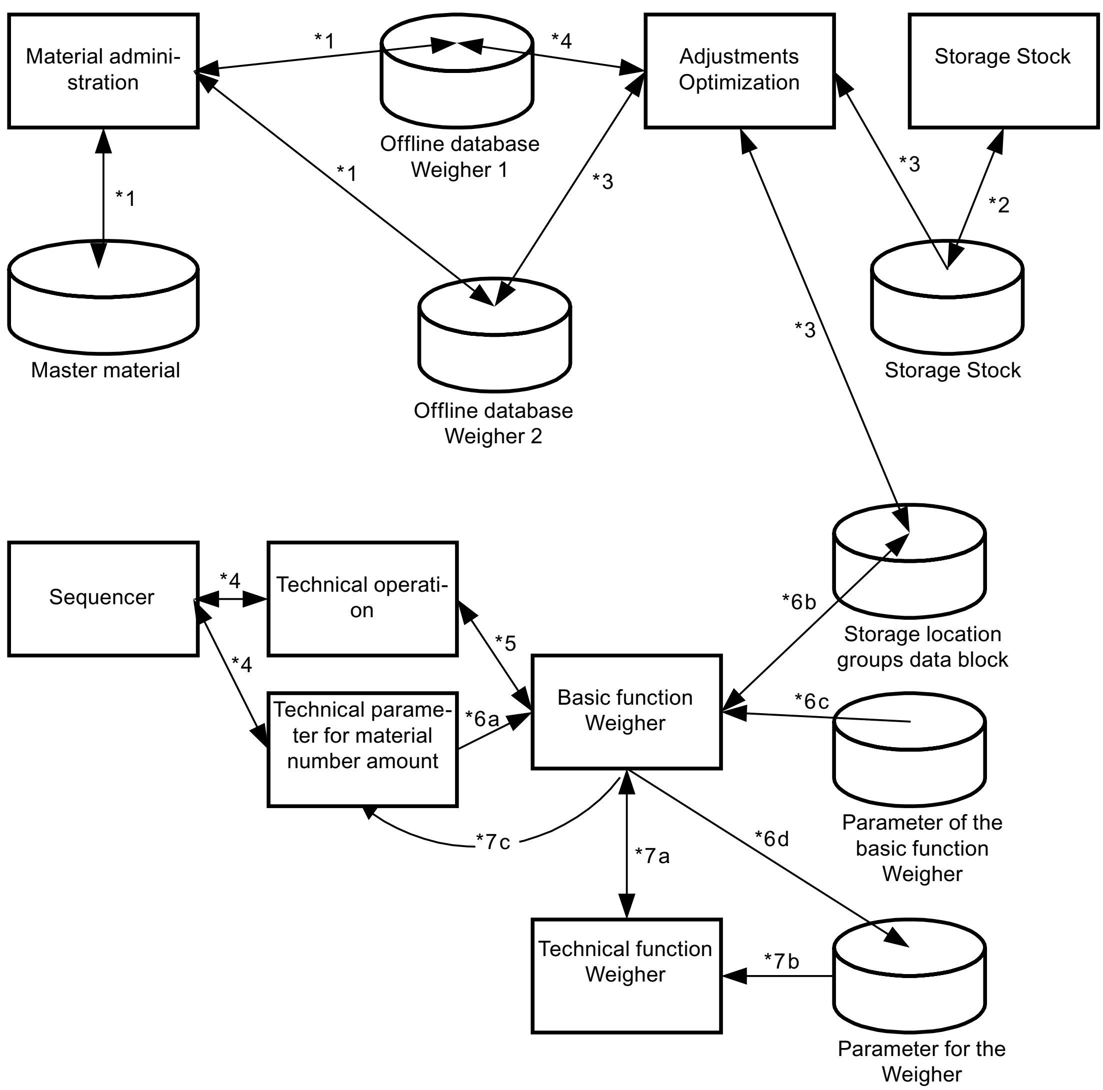


Table A-1 Explanation

*1	• Using the process input engineering in the recipe editor ("Engineering/Process input"), a material stock can be created and modified in the "<proj-path>\recipe\COMPLIST\MATERIAL.DBF" file. • The assignment of process input to the weigher is also defined here. An offline database is created for each weigher for the settings in the files "<proj-path>\recipe\COMPLIST\comp<x>.dbf".
*2	With the storage stock management in the recipe editor ("Engineering/Storage stock data", storage stock groups, weighers and storage stocks are configured in the "<proj-path>\recipe\COMPLIST\SILON.INI" file.
*3	• Parameters in the offline database "<proj-path>\recipe\COMPLIST\comp<x>.dbf" can be changed using the buttons "Settings" and "Optimize weighers". • A download or upload of the storage stock data blocks in the PCU is also possible. Data of the storage location configuration are used for adjustments. The material is imported into the storage location through the download of an adjustment data record.
*4	The sequencer calls the EOP which is part of a unit procedure and passes the setpoints to the technological parameters.
*5	An EOP calls the basic function 'Weigher'.
*6	The basic function (BF) weigher searches in the storage location group, which is assigned to the weigher for the material number, which was passed by the sequencer (*6a/ 6b). The assignment of the equipment parameters is read from the parameters for the BF weighers (6c). The adjustment data record which was found is transferred to the parameter for the technical function (TF) (*6d).
*7	The BF weigher calls the TF weigher and waits for the acknowledgment that the dosing has been terminated (*7a). The TF weigher uses the parameters which are loaded by BF (*7b) for dosing. The basic function transfers the actual process values into the technological parameters (*7c) and signals end of dosing to the EOP.

A.7.1.4 Material number and Material ID

A distinction is made between the material ID and the material number in the system.

- The Material ID consists of an ASCIII ID with 24 characters. This ID is only controlled on the OS.
- Material number:
  Because it is difficult to handle the material ID as a string within the S7, a material number is introduced in addition. This number is allocated relatively within the storage group. This material number is controlled in the control recipe and is in the storage location group data block.

A.7.2 Configure weighers

A.7.2.1 Overview - Weigher configuration

The following steps are necessary to configure a weigher:

- Define Recipe Category
- Define Unit
- Define adjustment for storage group
- Configure Weigher function

- Configure material and storage stock
- Create materials
- Create and optimize adjustment data record
- Import material

A.7.2.2 Define Recipe Category

Why Recipe Category?

The storage location group communication with OS is based on the mechanism for free protocols. On the other hand the free protocols need a recipe category.

These recipe categories are not identical to the categories which contain the recipes for running the weighing steps. These are instead recipe categories which the system requires to record stock data and the stock history by means of the free protocols, or to update these data by means of datagrams. The RCats are neither generated, nor configured in the Editrec-Tree-View.

Area for the Recipe Category

The area for recipe categories which could be used for the component administration is occupied by the system from 100 – 255. This value can only be changed via an ASCII editor.

Data file: '`<proj-path>\recipe\complist\silon.ini`'
entry: [Global]
Key: MinRecType=100
Key: MaxRecType=255

Name of the storage location group

In the component administration, a storage location group is set equally with the weigher. This is possible due to the 1:1 assignment. The name of the silo group must be assigned using the "Text parametrization" application. The names are stored in the '`<proj-path>\texte.x\rtyp.txt`' file.

If the file has fewer lines than necessary, the files can be lengthened as necessary.

A.7.2.3 Define Unit for a weigher

A unit must be configured for a weigher. The configuration of a unit is described in the section "Parameter assignment of technological objects".

A.7.2.4 Define adjustments for a storage location group

An adjustment parameter data record must be configured for the weigher. This configuration consists of:

- Configure a data block with associated data type (UDT)
- Configure a description data file for the data management on OS.

Since the files are weigher specific, these are not supplied with the standard system.

In case of questions about the integration of current SIWAREX weighers, contact SIMATIC Customer Support.

Data block for storage group

This block must have the following standard structure.

- The block consists of header data and an array for a UDT.
- The UDT describes the adjustment for the weigher with the help of STEP 7. As the adjustments may be different from material to material one data record is created for each storage location.
- In this data record a data record is loaded via the component administration. Besides the adjustments for a material the material number is also loaded.
- This data record should have at least the parameter 'Material number'. Any further parameters are optional.
- The material number is the key for a download and upload of the corresponding adjustment parameters.

UDT for storage groups

The adjustments for the weigher are assembled in a UDT. The UDT is created via the SIMATIC Manager function.

DB description for storage groups

So that the system can access data of the block, a description of the data block must be created for the object manager.

This description is a PCU-specific file (param.pcu) and needs to be copied to the subdirectory '`<proj-path>\pcu.xxx`' (xxx stands for the number of the PCU in which the weigher is located).

The file extension must be '.pcu' (see also section: User- and Option blocks (Page 389)).

Definition data file for offline database

Basics

An adjustment data record should be filed on the OS for each material which should be scaled. The adjustments are filed in dBase format. In this document, this definition is referred to as offline data structure for the weigher dosing. For the dBase the system uses a description data file (Def data file, see 'System description' 'dBase definition data file').

The structure of this data file is weigher-specific. The first four sections must be available for all weighers.

Offset	Section	Type	Length	Comment
1	MATID	CHAR	24	Material ID
25	COMPNO	CINT	5	Material Number
30	DOSGRP	CINT	3	Dosing group
33	ENABLE	CINT	1	Enabling
34	PRIORITY	CINT	3	Priority at dis-charge

Data name 'compx.def'

The data file must have the name 'compx.def'. For the value x the storage location group must be entered. If more than 1 weigher is connected for a storage location group the number for x can differ from the storage location group.

Path: '`<proj-path>\recipe\complist`'

A.7.2.5 Interrelations between definitions

The previous chapters deal with three definitions

- UDT for the data block
- Description data file for the data block
- Offline database structure

The following rules apply for the data exchange between the offline database and the data block in the PCU.

- The data block structure in S7 and the structure of UDT should conform with the block description.
- It is not necessary to accept all attributes in the block description.
- Only attributes with the same names (Name in block description and Offline data structure) are transferred with uploads and downloads of the adjustment parameters.
- There are attributes which are only found in the offline data structure. These parameters are not overwritten with the upload of the parameter. 'Enable' is such an attribute.

- There are attributes which are only found in the block description. These ones are not overwritten with a download. These attributes do not change the value with a download of the adjustments. As a result, these parameters are independent of the material.
- The name for the material number must conform with the offline data structure and in the data block description. If this is not the case, the upload will not be carried out.

A.7.2.6 Define weigher function

You should configure for the weigher:

- Weighing (TF)
- Weighing operation (Basic function)

.

The function is split into three parts.

- BF weigher
  This function represents the interface between recipe system and weigher.
- TF Weigher
  This function implements the weigher-specific operations.
- Search storage
  Search in the storage DB for the material.

A.7.3 Integration in the system (example SIWAREX)

A.7.3.1 S7 Data block

Basic function Weigher (BF/German GF)

Basics

The part of the PCU software that is not dependent on the weigher is known as "GF Weigher" and it is formed by the following blocks:

- FC645 WEIGHER_GF_FC
- FB645 WEIGHER_GF_FB
- DB645 WEIGHER_GF

The blocks are in the relevant STEP 7 project supplied in the "BM_PCU_STEP7" folder.

Interconnection of the GF weigher data block in the project.

All above-mentioned building blocks should be transferred to the AG/Project.

Configuring GF Weigher (DB645)

- The variable "WEIGHER_GF".iAct is set to the number of weighers actually required. Maximum is 16. The "Parameter assignment" application or the S7 software can be used for this.
- Do not enable more weighers for processing than you need. The weigher function demands a high CPU performance. Furthermore there are periphery access errors in AG and bus access errors on the DP Profibus.

Technological function (TF)

Basics

The blocks of the TF weigher form the weigher-dependent part of the PCU software.

Note

TF blocks based on the example of SIWAREX

The blocks and functions listed below relate to a project-specific implementation based on the SIWAREX weighing systems (SIWAREX-M, SIWAREX-FTA) and are not part of the supplied STEP 7 project. The following explanations are intended only to clarify the relationships.

The following blocks belong to TF-SIWAREX:

FC646	WEIGHER_SIWA_M_FC
FC649	WEIGHER_SIWA_M_TEXT_FC
FB647	WEIGHER_SIWA_M_FB
FB648	WEIGHER_SIWA_M_USR_FB
FB649	WEIGHER_SIWA_M_MSGS_FB
FB650	WEIGHER_SIWA_M_DIAG_FB
DB646	WEIGHER_SIWA_M_VECTOR
DB830, 832-892	WEIGHER_SIWA_M_1, _2 - _32
DB831, 833-893	WEIGHER_SIWA_RUN_1, _2, _32
DB920 - 951	WEIGHER_SETTING_VAL_W1 - W32

Integration of the TF-SIWAREX blocks in the project

All FCs/FBs from the list above as well as DB646 are included, regardless of the number of SIWAREX weighing systems.

The following is also required for each installed weighing system:

- One block from the sequence DB830, 832, 834...
- One block from the sequence DB831, 833, 835...
- One block from the sequence DB920, 921, 922...

You should always start with the lower block numbers.

Example: 3 SIWAREX weighing systems are to be used.

We transfer FC646, FC649, FB647-650 and DB646.

Further for

- Weigher 1: DB830, DB831 and DB920
- Weigher 2: DB832, DB833 and DB921
- Weigher 3: DB834, DB835 and DB922

A.7.3.2 Configuration

S7 Manager: HW Configuration

By double-clicking on the hardware symbol of an S7 station in your S7 project, you start the tool 'HW configuration'.

In this way the used weighers are located decentrally in the substations of the DP Profibus.

The following has been specified for the I/O address assignment in the system standard:

I/O address	**= (n-1) *16+ 1024; n=1, 2, etc.**
	= 1024, 1040, 1056, 1070, etc.

These addresses are also entered in the original SIWAREX DBs. If you cannot use the standard address assignment, do not forget to make the necessary changes to the SIWAREX DBs during parameter assignment.

Interaction between the S7 software components

The following components are involved in the weighing:

- Recipe System
- Silo/Tank data administration
- Unit-Sequence control
- BF Weigher + User program
- TF Weigher

Their interaction is briefly explained here:

- The process (Basic recipe) for the production of a product is filed in the recipe system. Further components for this product are taken from a product-specific list.
- Via silo/tank administration you define the assignment of the components to the unit (Unit "Weigher..."), where the dosing starts. Several weighers/units can participate in dosing components; they work time-parallel. On one weigher, it is possible to dose several components sequentially (multiple component dosing).

- In general, the process of a dosing/multiple component dosing is realized in several steps by sequential control. In the basic recipe a basic structure is filed per weigher/unit. It consists of all component-independent steps (Equipment operation: EOPs), as well as a macro EOP, the "Dosing EOP".
- When creating the control recipe (in other words the instructions for the PLC) this dosing EOP is expanded: For each component to be dosed, a line of this EOP is inserted instead of the macro EOP. If the capacity of the weigher would be exceeded by the addition of the next component, the system generates an Emptying EOP. If there is no component found for this weigher, the complete process can be eliminated.
- The processing of the control recipe applies to the unit control software in the AS. The EOPs are worked through in the specified sequence. The configuration engineer selects the actions and parameters for each EOP. The dosing EOP needs the components (number) and the weight setpoint as mandatory parameters. It also requires tolerance and storage location as possible parameters.
- The GF weigher interface is called in the dosing EOP. This interface and therefore also the user program is not dependent on the weigher.
- At the start of the EOP, the GF determines the storage location of the component based on the component information and the tank/silo datasets belonging to the weigher and transfers this information to the setpoint of a DFM decoder configured for this purpose. The user program interprets the decoder bits and sets the route from the storage location to the weigher.
- The TF works in the background and is coupled to the BF via a data interface. Via this interface, commands to TF, status messages from TF and setpoint/process values for weight and tolerance are communicated. Furthermore with dosing start the TF receives a note relating to the determined component data record in the tank/silo DB, from which the adjustment data for the weighing of this component are taken. Before sending the dosing command to the SIWAREX weighing system, all necessary component-dependent data is transferred to the module. Data which are not plausible are refused with "data error" (e.g. if the setpoint is less than the tolerance value). In such a case no dosing can be started.
- If everything is correct, the dosing starts. The BF communicates the status of dosing via bit interfaces to the user program. With an error, the dosing is put on hold (Unit Sequence is set "-" (HOLD)). The user can also stop a running dosing any time with this operation. With the manual transfer from "-" to "+“ the dosing is continued without balancing. In case of error this is the acknowledgement command to GF/TF. In case of a still-active error, the unit sequence is stopped again.
- By leaving the step, a running dosing is canceled.
- If this dosing could not be closed, display bits signal to the user program the end and the result of the dosing procedure (see 'programming').

BF Configuration

Requirement:

The number of datasets (i.e. weighers) was set during configuration in the 'WEIGHER_GF' data block (DB645) in the variable "WEIGHER_GF.iAct".

Configure now for each GF data record :

- Weigher name (Shift F5)
- DFM for component number
- DFM for weight setpoint/process value
- DFM for tolerance setpoint/process value (is only used for logging)
- Unit assignment of the weigher
- Weigher type (Presetting 0=SIWAREX-M)
- TF DB (Presetting DB830, 832 etc.)
- DB for silo/tank data (dosing adjustment data of the components)
- DFM for partial weight setpoint/process value
- DFM for storage location (if this one can be different than the component number)
- DFM for Partial Route

Any further indications are used for diagnosis and do not need to be configured.

Silo/tank data (component adjustment data)

To allow the classes SILO_W1, .. SILO_Wn (DB920, 921 ..) to be set or even read in, you need to copy a class definition file (part of the project-specific TF weigher implementation) to the corresponding PCU folder "`<proj-path>\pcu.nnn`". After system startup, the classes are then read in and displayed.

Note

You may copy only one of the class files:

- `SILO_GR.PCU`: Example of the weighing data records under PCU=S7, Rec.Version=3
- `SILO_GR_S1.PCU`: Example of the weighing data records under PCU=S7, Rec.Version=5

The data record consists of a weigher-independent part and a weigher-specific part.

The weigher-independent part consists of:

- Component number (1..n)
  - Actual inventory [g]
  - Maximum inventory [g]
  - Priority with dosing
  - Inventory record keeping yes/no
  - Inventory control yes/no
  - Further Status and Control bits
  - Specific weight

Here, the weigher-dependent part is adapted to the SIWAREX weighing system. The parameter identifications are taken from the instruction manual. There you can read their identifications.

MatFlowCtrl	Data flow monitoring ON
OptimizeFine	Fine optimization ON
TolPlus	Tolerance Plus [g]
TolMinus	Tolerance Minus [g]
CoarseVal	Coarse Val-Break value [g]
FineVal	Fine Val-Break value [g]
SettlingTime	Damping time [s]
InchingTime	Inching time [s]
MonInchRepropTime	Monitoring time Inch operation [s]
FlowCoarseTime	Monitoring time flow coarse [s]
FlowCoarse	Minimum flow coarse [g]
FlowFineTime	Monitoring time flow fine [s]
FlowFine	Minimum flow fine [g]
FlowCoarseDelTime	Delay time Flow monitoring coarse [s]
FlowFineDelTime	Delay time Flow monitoring fine [s]

Control the preset values and adjust them according to your requirements, if necessary.

TF Configuration

- Classes SIWAREX_W_1, .. SIWAREX_Wn (DB830, 832, 834 ...)
  If the I/O address assignment was selected as outlined above, no parameters need to be set here.
- Otherwise adapt the following:
  MOD_ADDR_IN → I/O address of the weigher according to the HW configuration.
  For the installation you will find useful diagnosis information here. The significance of the parameters can be taken from the SIWAREX-M instruction manual.
- Handling of the application compartments:
  You communicate with SIWAREX-M via so-called application compartments. 2 writing, 1 command and 2 reading compartments are available. The usage is executed in such a way that the number of the first data record, which should be written and read, is transferred in Parameter ‘DRNO’. If you set the bit 0 in the parameter 'BITFIELD', this data record is transferred and the result of the job is displayed in 'OK' or 'ERR', respectively. The bit 1...15 of 'BITFIELD' is relevant for the data records following after 'DRNO'. Therefore, it is possible to transfer up to 16 data records with 1 job.
- Tip:
  As the TF software uses above-mentioned application compartments during running operation, you should never have direct write access to these compartments.
  For this reason USR application compartments WR/CMD/RD have been installed. Here jobs can be filed in a manual way or through the program and results can be read. The TF software checks these compartments cyclically. If they contain job data it executes these jobs with higher priority than the others. Therefore a rapid intervention is guaranteed.
  If you know the meaning of the SIWAREX datasets, you can address the device very efficiently using this method.

A.7.3.3 Programming

Note

DFMs used for dosing purposes are pure dosing parameters. The call of DFM processing in the Sequencer FB, as required for normal DFMs, could produce errors and should be omitted for such DFMs.

TF

The processing of the SIWAREX DBs must be integrated in OB1 and may only be called once per cycle.

OB1 is integrated using FB1220 or FB1221:

```
NETWORK
TITLE = SIWAREX-M
 CALL "WEIGHER_SIWA_M_FC"
 (iRetVal := iVarTemp
 );
```

The called FC searches for all SIWAREX weighing systems in DB645, takes the number of the SIWAREX DB from the corresponding GF dataset and processes this. On the user‘s side, no further precautions are required.

GF

Basics

The WEIGHER_GF_FC function forms the (only allowed) call interface of the weigher function.

This function is not permanently integrated in an execution level (OB). Rather, the call is always made on an as-needed basis from the user program within the unit control, thus either from a user unit FB or an EOP.

Example:

TA-User-FB, Call of the weigher 1, Function 16 (Status Query)

```
    UN ATL;
    SPB NRUN:
    CALL WEIGHER_GF_FC Only processing, if unit works
    (
    iWeigher := 1,
    iMode := 16,
```

```
    RET_VAL := #iRetVal
    );
NUN: NOP 0;
```

Parameter:

```
iWeigher    : Number of the desired weigher (GF data record) corresponds
to the unit number
iMode       : Processing mode:
       0, 16: Status inquiry
           1: Upw. dosing with tare
           2: Downw. dosing with balance
           4: Man. upw. dosing with tare without display
           5: inching prop. upw.          with tare
           6: inching prop. downw.      with tare
          21: prop. upw.     without tare
          22: prop. downw. without tare
          24: manual prop. upw. without tare
          32: tare
RET_VAL    0: OK
           1: call after BOP (basic operation)
           2: #iWeigher > "WEIGHER_GF".iAct
           3: record copy
          -2: #iMode wrong
          -3: tank DB: "TEST_DB" returned error
          -4: write error: DFM weight actual value
          -5: write error: DFM tolerance actual value
          -6: read error: DFM weight setpoint value
          -7: read error: DFM component number
          -8: read error: #iLen of tank DB
          -9: read error: #iOffset of tank DB
        -10: read error: #iAct of tank DB/no component records available
         -11: read error: component record
     -12,-19: read error: component control bits
     -13,-20: read error: component status bits
         -14: read error: component priority
     -16,-21: read error: tank content
         -17: read error: path number
         -18: access error: DFM weight actual value
         -22: access error: max. tank content
```

Note

FB645 is a sub-program of FC645.

This must not be called directly in the user program!

A.7.4 Storage administration

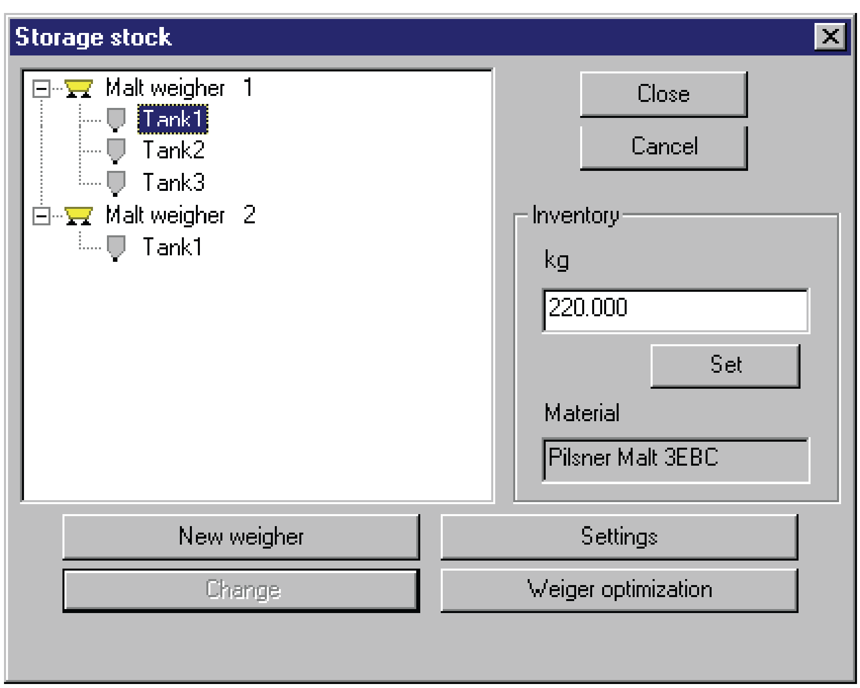
A.7.4.1 Overview - Storage Stock

After the weigher has been configured in the user program the stock must be configured in the recipe system.

The call is executed in the recipe editor in the menu "Engineering / Depot locations". This menu item brings you to the configuration dialog for the depot locations.

A.7.4.2 Dialog 'Storage stock'

This dialog displays the overview for the storage location groups which are configured for this section. In the system the storage location group and weigher are set equally. Therefore they are used for headlines and texts equally.



New Weigher

A new weigher/storage location group is created via this button. The property dialog of a weigher is opened before creating. After having left this dialog with 'OK', the weigher is created.

Change

The property dialog of the selected weigher is opened via this button.

Settings

The dialog for the adjustments of the weigher parameter for a storage location is opened by this button. In the dialog, the values, which are filed in the offline database, are displayed and can be specified.

The dialog will only be opened if there is also a known material number available. If this is not the case, an error message will be displayed that there is no offline adjustment data record available for the material.

This is the normal process after having created a weigher. The material must be created or selected and transferred in the storage location by the material stock.

Weigher Optimization

The dialog for optimizing the weigher adjustment is opened for a storage location. In the dialog, the values of the offline database and in the storage location data block are displayed in two columns.

The dialog will only be opened if there is also a known material number available.

Stock

The displays in this control element group are only enabled when stock management is enabled for the weigher selected from the structure view.

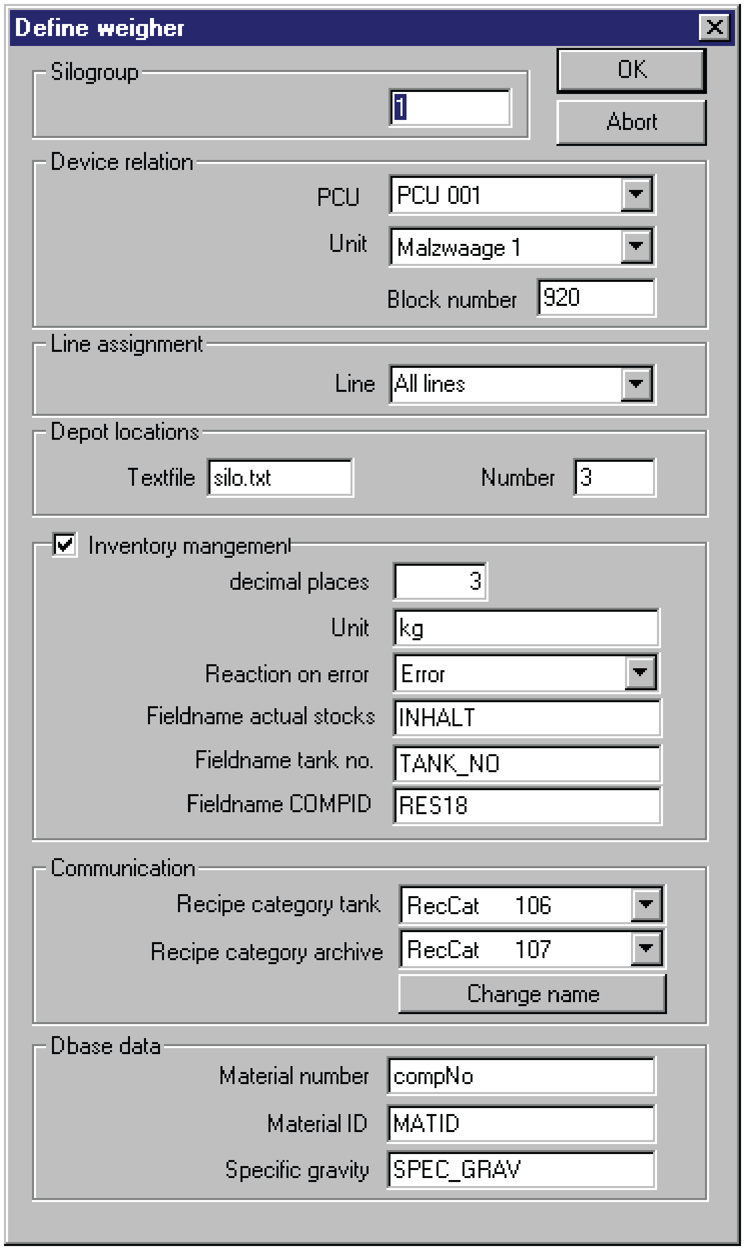
Further information, and information on the configuration of "Free Protocols" for stock management can be found in chapter Stock management (Page 1260).

The top input box shows the current stock of a tank/silo in the field above the specified unit. You can click “Set” to write the current stock value in the upper input box to the database table of the "Free Protocol". This operation is protected by means of password level ‘Mat_SetActSt’.

The ‘Material’ field shows the material name and the current material stock.

A.7.4.3 Dialog 'Define Weigher'

In the view of the storage stock, all important properties are adjusted in this dialog.



Hardware import

The hardware import consists of the indication of a unit. At this point, the unit sequence which is used for the weigher should be declared to the storage stock. A unit is defined by the indication of the PCU and the unit.

Furthermore, the defined data block in the PCU for this storage group must be indicated.

Name of the storage location group

The name of the storage location group is identical to the name of the recipe category which is used for the communication. The name should be edited via text configuration.

Line assignment

The weigher can be assigned to a line via this adjustment. This adjustment is only relevant for plants which have several product lines and automate those via line recipes.

A weigher can also be assigned to all lines (Line=0).

Storage location

Two adjustments must be taken into consideration. The number for the storage locations and the text file under which the names for the storage locations are filed. The text file is expected in the text folder for the corresponding PCU.

Stock management

Set the "Stock management" check box to enable this functionality. Operator control of the other control elements of this group is not enabled unless this function is set.

Decimal places

Number of decimal places for the value of the current stock volume. The value is saved internally as integer and computed with this value.

Unit

Text-based input of the measurement unit. This is text only and is not used for calculation.

Reaction to Errors

Error response when enabling jobs. When a material shortage is detected (e.g. in a tank or silo), the system either reports an error (‘Error’),

tries to locate a standby tank or silo for dosing (‘Search another tank’), or ignores the error (‘Ignore’).

Field name Actual Stocks

Name of the column of the Dbase table that contains the data of the actual stock. Default name is 'ACTSTOCKS'. The column may take every position of the Dbase table, save for the first two columns. Its data type is CINT, without decimal places. The length must be longer than the number of decimal places by at least 1.

Field name TANK_NO

Name of the first column of the Dbase table for the tank number. Default value is 'TANK_NO'.

Field name CompID

Name of the column in the Dbase table that contains the CompID of the actual stock. Default value is 'ACTCOMPID'. The column may take every position of the Dbase table, save for the first two columns.

Communication

The recipe categories for the communication with the automation system are selected from these two combo boxes. The recipe category must be assigned, because the communication functions are effected by means of the mechanism of the free protocols.

The free protocols are used to transfer data from the PLC to the IOS. Since these data are usually job- or batch-related, they are stored in a hierarchical order determined by recipe category, order number and batch number. For stock management, the recipe category links the data to a certain weigher and thus to a certain storage location group. The job number encodes the tank number. The batch number is not required; its default value is "1".

The "tank recipe category" defines the category that is used to transfer the value of the stock volume in a tank or silo, while historic data are saved to the "recipe category archive".

Class attributes name for dBase data record

For compatibility with weigher adjustments which are available for the version V4, the field name for the material number must be adapted. The presetting is 'COMPNO'. The indication should conform to the indication in the block description.

Field name for the ID

For compatibility with the weigher adjustments which are available for the version V4, the field name can be adapted for the material ID. The presetting is 'MATID'.

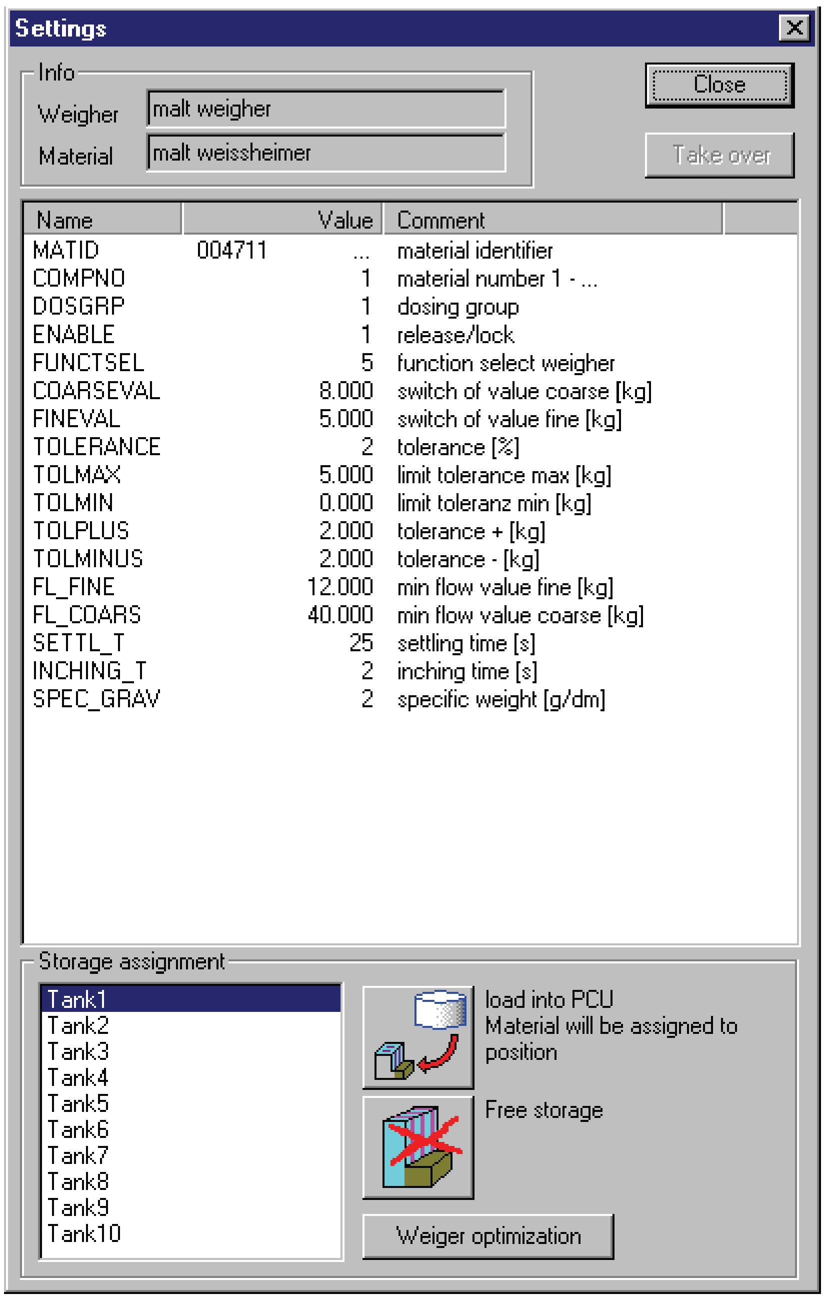
Storage location group

Storage location groups are charged with a running number. It is necessary to change the number for the access to the offline database for the functionality that more than 1 weigher can access a storage location group. The presetting is identical to the storage location group number.

A.7.4.4 Dialog 'Settings'

Functions

- There is the possibility to change offline adjustments for a material within a storage location group.
- Offline adjustments can be transferred into the PCU (Download). Download defines that there is a special material in the storage.



Info

The weigher and the selected material is displayed in the information fields.

Close

The dialog is closed via this button. If a parameter was changed, a query as to whether the changes should be accepted is issued in advance.

Transfer

The changes are transferred via this button.

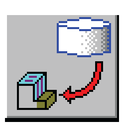
List of adjustment parameters

All parameters are displayed which were defined in the offline structure. The values can be changed directly in the list. Since these parameters are highly specific to the corresponding materials and also depend on the type of weigher, we shall go into details relating to their significance at this point. Information on definitions and the correlation between adjustment parameter sets are found in chapter Define adjustments for a storage location group (Page 1237) and Interrelations between definitions (Page 1238).

Storage location assignment

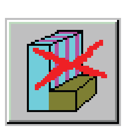
All storage locations for this storage location group are displayed in this list.

Button



All adjustments which are filed in the storage location data block are loaded via this button. The adjustments should be checked before the download, especially the enabling. If a retrieval from the storage location is also planned, the enabling should have the value 1.

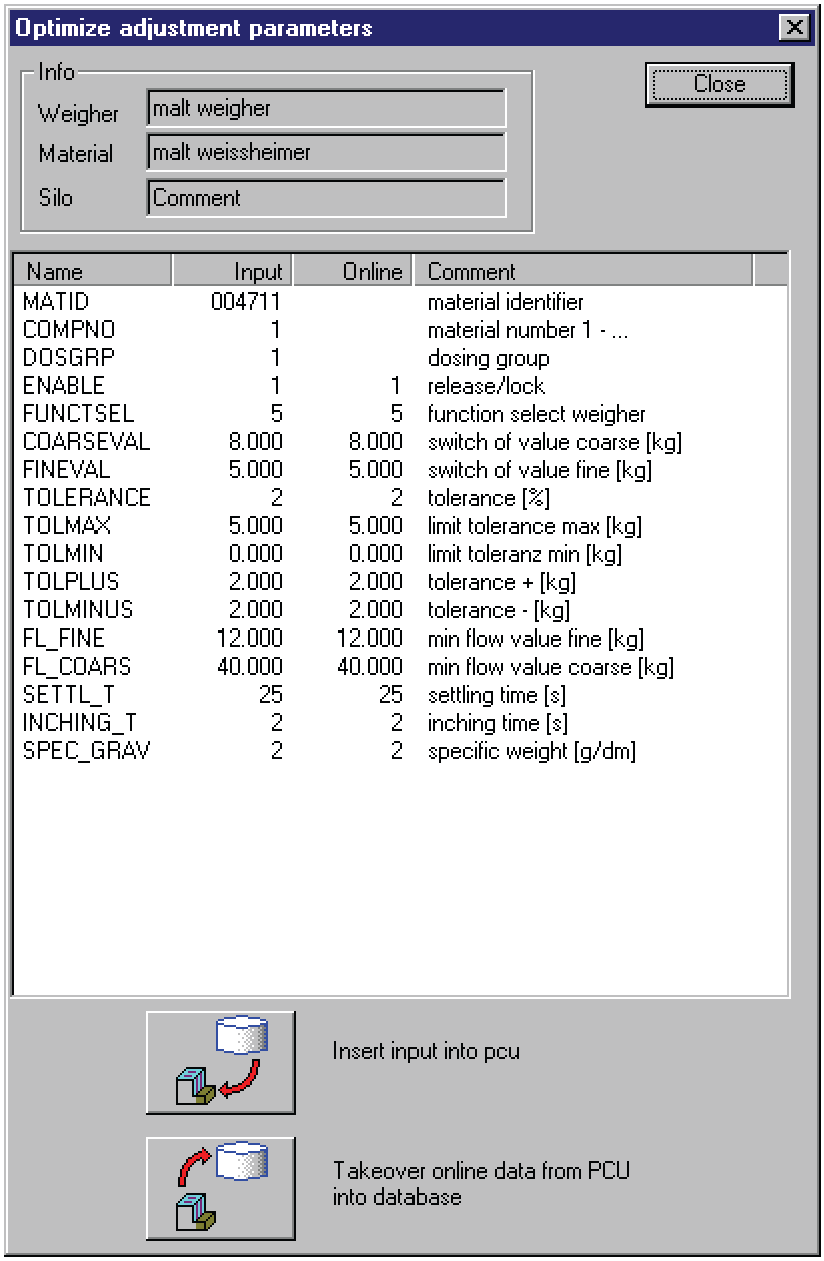
Button



Button for deleting the settings in the storage location DB.

A.7.4.5 Dialog 'Optimize adjustment parameters'

It is possible to optimize the weigher adjustments via this dialog. In the dialog a comparison of the parameters, which are used Offline and Online, is displayed. There is also the possibility to upload and download the adjustment data sets.



Info

The weigher, the material and the storage location (Silo) are displayed in this section.

Close

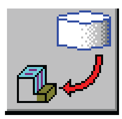
The dialog is left via this button.

List

The adjustment values for the Offline column can be edited in the list.

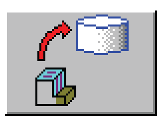
The values are not transferred directly into the Offline data management!

Button



The adjustments which are displayed in the column 'Offline' are loaded into the storage location data block. All parameters are written (Consistency)

Button



The values which are filed in the storage location data block are transferred into the offline data set via this button.

A.7.5 Component management

A.7.5.1 Overview - material administration

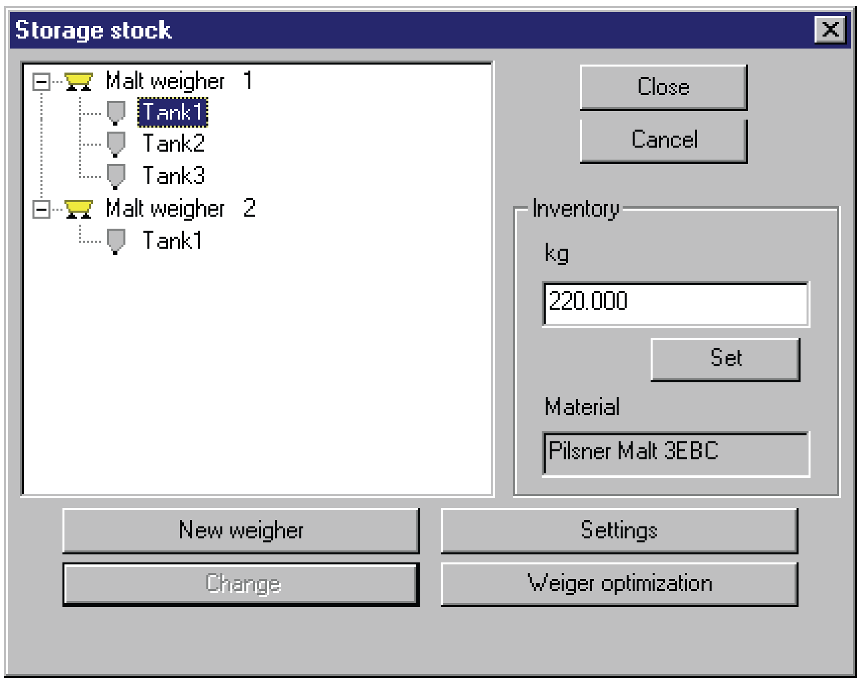
After having configured the weigher and the storage location administration, the material should be configured in the recipe system.

The corresponding configuration dialog is called in the recipe editor with the "Engineering / Components" menu item.

A.7.5.2 Dialog 'Material stock'

Functions:

Material can be created, changed and deleted via this dialog. Starting with the master material, the materials can be assigned to a storage location group and the offline data record can be created and changed.



Tree view

There are three main branches for displaying materials.

These are:

- Material types
  Materials are listed according to types.
- All materials
  All materials are listed.
- Weighers
  All materials are listed according to storage location groups. Materials are only found after the assignment to a weigher.

Material types

- Number
  With this input field the number of material types can be set. The number of material types is stored in the file "<proj-path>\recipe\complist\silon.ini'.
- Edit names
  With this button the text parametrization with the file '<proj-path>\texte.x\mattyp.txt‘ is opened.

Material

- ID
  The Material ID is entered via this field. The ID can only be entered by creating a new material. It is not possible to change it afterwards as the ID is a key within the material database table. The ID has a maximum of 24 characters (letters and digits). In special cases the ID can also be shorter with maintenance of V4.6 configuration
- Type
  The material can be assigned to a material type via this selection. This selection can also be changed later.
- Name
  A name can be allocated for the material via this input field. The name can have a maximum of 50 characters.
- Unit
  A dimension name can be allocated for the material via this input field. The name can have a maximum of 16 characters.
- Create new
  Creating a new material is enabled by pressing this button. After having pressed the button, the input section for the Material ID is enabled and you can enter a new Material ID. The input of a new material is completed with the button 'Transfer'.
- Transfer
  A new material or a change to an existing material is finished and the specification is accepted via this button.
- Delete
  The selected material is deleted with this button. The function is enabled only for materials which have not been assigned to any weigher or storage location.

Material/weigher assignment

- A material of a weigher/storage location group can be assigned via fields and buttons. There are two indications necessary; these are weigher and material. Both can be selected via the tree overview, but only step for step. The selection is transferred into both indication fields 'Weigher‘ and 'Material‘.
- Weigher
  No direct input can be executed in this field. The weigher which was selected last is displayed in this field.

- Material
  No direct input can be executed in this field. In this field is displayed the weigher which was selected last.
- Settings
  The dialog 'Settings' is opened via this button (see dialog 'Settings'). Offline adjustments are displayed for combinations between weigher and material which have been taken into consideration before. The button is only operator-controllable if data records are available for the combination.
- Assign material to the weigher
  A data record is created in the selected storage location group for the selected material via this button. The data record is allocated with default values. Only after this assignment can the data record be edited.
- Delete assignment
  The assignment for material and weigher is deleted via this button. The data record in the offline data management is deleted.

  Note

  When deleting, no check is performed to ascertain whether material is still available at the storage location. A batch which has already been created is not influenced. The material is no longer found in this weigher for new batches.

A.7.5.3 Dosing group

The dosing group is an additional attribute, which can be set at three places.

- At the material in the offline database
- In the process input list
- as setpoint at the EOP 'Dosing‘

What is the purpose of the parameter?

The parameter is used to influence the sequence of weighings within a weigher, regardless of the input sequence in the process input list.

With the release of the batch, the dosing group is entered in the batch list. Either it comes from the recipe process input list or from the adjustment data records of the material.

Allocate dosing group recipe-specific

The indication in the process input list can be executed in an optional way. You can allocate the dosing group for the material in a recipe-specific way with an input in the process input list.

Note

The function uses an additional column in the file of the recipe process input list. You have to check whether the column is also available in the definition file. The input 'DOS_GR' with the length of 5 characters and the data type CINT is necessary. A corresponding file is delivered by the system. There is also a file without a dosing group available for purposes of compatibility.

Changing the definition file has the effect that current recipe process input lists cannot be read. These process input lists should be converted with a dBase tool.

Allocate dosing group material-specific

If there is no dosing group entered, the dosing group which was defined for the material is valid.

What influence do the indications have?

When creating the control recipe, wildcards for the recipe operations are built in n times in the control recipe. If and how often the operations are built is determined in the process input list and the offline database for the weighers.

All inputs of the batch process input list are checked for each recipe operation.

- If the input for "my" weigher is defined,
- the dosing groups for the operations and process input list input conform, or if one of both values is 0

Only if both criteria are met will the weighing operations be transferred into the control recipe.

Example

There are two types of materials: A and B. The materials for the type A must be proportioned before type B.

- Unit procedure
  Two recipe operations are built into the Unit procedure, which both refer to the technological operation 'dosing' of this weigher. For the first recipe operation, the value for the setpoint 'dosing group‘ is allocated with the value 1 and for the second recipe operation the value 2 is allocated.
- Adjustment data record for materials

- For the materials of type A, 1 is entered for the parameter for the dosing group (the field is allocated with the name 'DOSGRP' by default). For the materials of type B, the value 2 is allocated.
- When creating the control recipe, all materials are entered in place of the first recipe operations that have used the storage location group for the weigher and the dosing group 1.
  For the second recipe operations, this occurs for the dosing group 2 in the same way.

A.7.5.4 Display materials

Basics

You can change the material number to text in order to display materials in process images. For this, two text data files are led in parallel to the offline database of the material administration.

Material ID

The material ID is stored in the text file `<proj-path>\pcu.nnn\texte\comp#x.txt` (nnn=PCU no.; x=silo group).

It is structured line oriented. The line position corresponds to the material number.

Material Name

The material name is stored in the text file ' `<proj-path>\pcu.nnn\texte\compx.txt`' (nnn=PCU no.; x=silo group).

It is structured line oriented. The line position corresponds to the material number.

A.7.5.5 Stock management

The current stock of a tank/silo is managed by means of the free protocol mechanism. To do so, the automation device sends datagrams of type 6, which are converted into the entries in a Dbase table. The recipe editor, the dialogs for storage location management and the order system can access this Dbase table in order to compare the reference values with the process values.

Configuration of the Dbase table

Definition file

A Dbase table is required for storing the entries in the datagram type 6 files. You also need a definition file in order to create this table. The columns used for stock management can be assigned any name other than the default, but must then be made known to the system in the weigher definition dialog. Also note the defaults for the silo/tank data functions and free protocols.

Defaults

TOPIC

Field name for actual stocks	ACTSTOCKS
Field name for tank number	TANK_NO
Field name for the CompID of stored materials	ACTCOMPID

In contrast to the order- and batch-related free protocols, the tank/silo management system uses only the tank number in the column with the default name ‘TANK_NO’ as primary key. However, the free protocol mechanism requires a second column (in the sample below:

‘Dummy’) for the function. The user may not write-access this column, and it always contains a "1".

```
0 Free Protocol/structure of received/datagram Type 6
 514
 1 TANK_NO CINT 5 Tank-NO
 6 DUMMY CINT 5 Dummy
 11 SZ_JAHR    CINT  2    start time year
 13 SZ_MONAT   CINT  2    start time month
 15 SZ_TAG     CINT  2    start time day
 17 SZ_STUNDE  CINT  2    start time hour
 19 SZ_MINUTE  CINT  2    start time minute
 21 SZ_SEKUNDE CINT  2    start time second
 23 BEREICH    CHAR 16    recipe type name
 39 REZEPT     CHAR 16    recipe name
 55 TANKNAME CHAR 16 [2,1,0,1,"NAMEHT.TXT"]
 71 AUFT_NR CINT 6 [2,1,1,1]
 77 AUFT_NR1 CINT 6 [2,1,2,1]
 83 LFD_TANKNR CINT 6 [2,1,3,1]
 89 LFD_TANKN1 CINT 6 [2,1,4,1]
 95 NAME CHAR 16 [2,2,0,2]
 111 NAME1 CHAR 16 [2,2,2,1]
 127 INHALT_L CINT 12 [2,2,3,2]
 139 INHALT CINT 6 [2,3,0,1]
 145 VOLLMENGE CINT 6 [2,3,1,1]
 151 STAT_ZAEHL CINT 6 [2,3,2,1]
 157 TANKSTATUS CINT 6 [2,3,3,1]
 163 TANKZUST CINT 6 [2,3,4,1]
 169 DRUCK CINT 7.2 [2,4,0,1]
 176 TEMPERATUR CINT 7.1 [2,4,1,1]
 183 ACTSTOCKS CINT 7.2 [2,4,2,1]
 190 ACTCOMPID CINT 6 [2,4,3,1]
 196 STATUS CHAR 16 [2,4,4,1,"STATUSHT.TXT"]

 (...)
 400 BEF_DATUM  CHAR 16    [2,11,0,3,"DT1"]  HT was filled initially
 416 VOL_DATUM  CHAR 16    [2,12,0,3,"DT1"]  HT full
 432 ENT_DATUM  CHAR 16    [2,13,0,3,"DT1"]  HT was emptied
 448 CIP_DATUM  CHAR 16    [2,14,0,3,"DT1"]  HT was cleaned
 464 NOTIZ CHAR 50
```

Configuration of the datagram type 6

Basics

To ensure proper functioning of stock management, you need to configure the datagram type 6. The configuration data is located in the file `'<proj-path>\sys\prot_006.ini'`.

- You configure in the section '[Rez_XXX]' ('XXX'= number of the recipe category). This recipe category is set in the weigher definition dialog.
- As already described, the tank/silo data is coded only according to the tank number. This means that the path system must be set correctly as the first entry under this section '[Rez_XXX]': 'New_Path=1'
- With this entry, the datagrams are stored in successive order according to the tank number (corresponds with the order number in the order-/batch-related free protocols) and dummy batch number ‘1’.
- It is useful to adapt the next entry, namely the ‘Block_Order’, to the number of tanks/silos available in the storage location group, so that all data of this group are stored in a single Dbase table.

Example:

‘Malt weigher’ is assigned the recipe category for stock management as recipe type 106. The corresponding silo group consists of five silos.

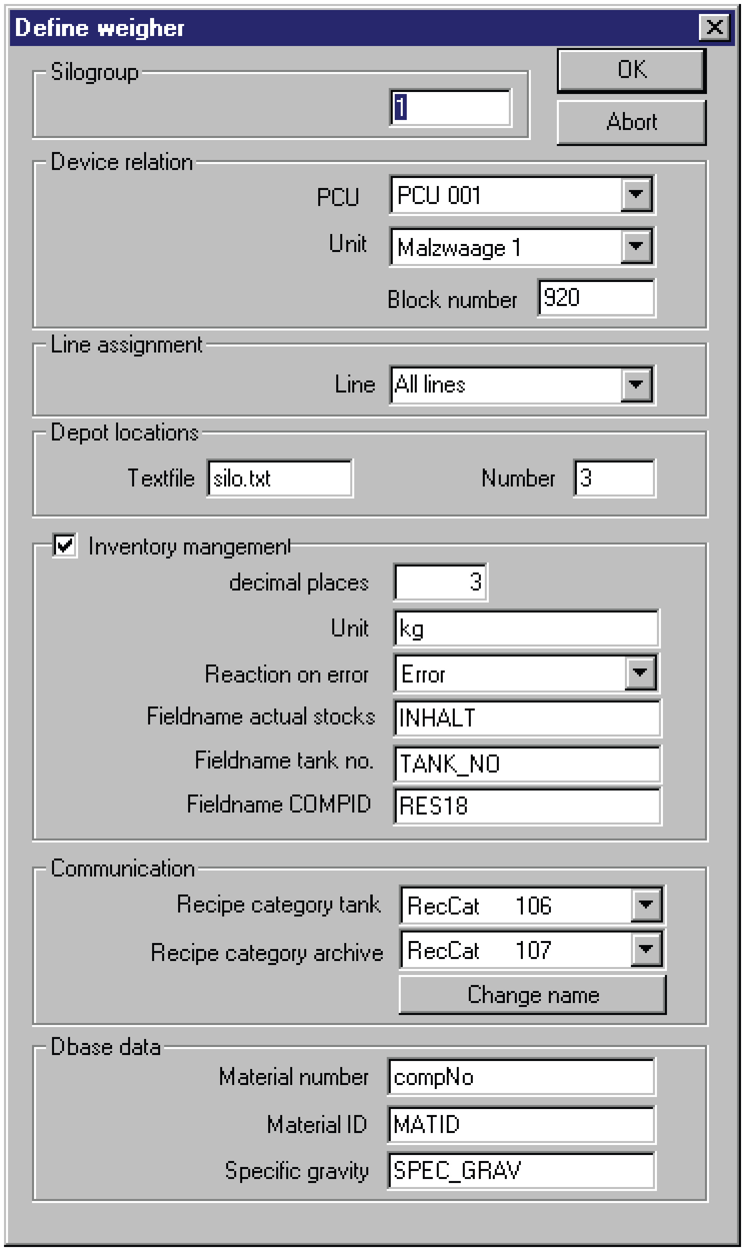
'prot_006.ini'

```
(...)
[REZ_106]
;Only for recipe type <xxx>. Look at [GLOBAL] New_Path
New_Path=1
;Only for recipe type <xxx>. Look at [GLOBAL] Block_Order
Block_Order=5
(...)
```

The Dbase table is stored under "windcs\frprot\frja_XX\rez_106\anr00005\Fe_00050.dbf". "XX" defines the last two numbers of the current year.

See also section: "Logging / free protocols".

Dialog 'Define Weigher'



Inventory management is enabled in the weigher definition dialog box by setting the "Inventory management" check box. This enables operator control of all control elements of the control elements group.

The meaning of the input boxes is described in the chapter '>Define Weigher< dialog box'.

A.7.6 Rights administration

A.7.6.1 General

The rights are edited via the application 'User administration' (see 'Administration').

A.7.6.2 Right to access 'Rec_Proj‘

Key:	Rec_Proj
Description:	Recipe system configuration

It protects the functions:

- Create storage location groups
- Change storage location groups

A.7.6.3 Right to access 'Mat_Man‘

Key:	Mat_Man
Description:	Material administration

- Optimization of adjustment parameters
- Edit offline adjustment parameters
- Delete adjustment data records
- Upload adjustment parameters
- Create new materials
- Assignment of materials to storage location groups
- Delete assignment
- Create material types

A.7.6.4 Right to access 'Mat_Ass‘

Key:	Mat_Ass
Description:	Check material in/out

- Download adjustment parameters
- Adjustment data record enabled and disabled
- Change priority of the material in the storage

A.7.6.5 Right to access 'Paratxt'

Key:	Paratxt
Description:	Text parameterization - Change/Store

- Change names for recipe categories
- Changes names for material types

A.7.6.6 Right to access 'Mat_SetActSt'

Key: Mat_SetActSt

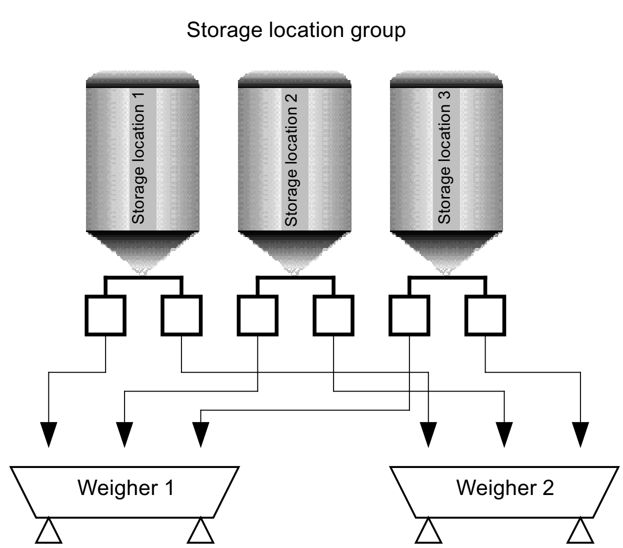
Description: Dialog 'Storage stock management': Setting the actual stocks

A.7.7 Cases for use

A.7.7.1 Several weighers on a storage group

The system only supports directly a 1:1 assignment for weigher and storage location group. In many plants there is more than 1 weigher to dose from a storage location group.

Example:



Note

This solution is only suitable for the use of line recipes.

Configuration

For the above-demonstrated example, the following configuration can be selected.

- Two storage location groups are configured.
- The same storage location group data block is entered for both storage location groups.
- The same storage location group number is allocated for both storage location groups.
- The same text data file of the storage location is declared for both storage location groups.
- The same number of storage locations is declared for both storage location groups.
- Both weighers are assigned to different lines.

Material assignment, adjustments

- A material storage location group assignment must be made for both storage location groups.
- Alternatively the adjustments can be made via one weigher or the other weigher, as both weighers refer to the same offline database.
- Alternatively a material storage location assignment can be made by each weigher, as both weighers refer to the same storage location data block.

Note

The material storage location assignment must be deleted for both weighers. Even for the first call, the data record is deleted in the offline database. By deleting the assignment for the second weigher, an error message appears, which can be ignored.

A.7.7.2 Create new material

To create a new material, you have to take the following steps:

- Open the material stock dialog via the recipe editor.
- By pressing the button 'New‘ initiate the procedure.
- Enter the new ID in the Material ID field (the ID cannot be changed afterwards).
- Fill out the fields for material name and material type.
- Finish creating via the button 'Transfer'.

A.7.7.3 Assign material to a storage location

Materials can only be assigned to a storage location if there is an adjustment data record for the material on the weigher. It will be defined for the system via the assignment that the material is available for this weigher.

Create adjustment data record

To create a new material, you have to take the following steps:

- Open the material stock dialog via the recipe editor.
- Selection of a material. This can take place via the partial tree 'Types‘, 'All‘ or 'Weigher‘.
- Selection of a weigher. A weigher must be selected in the partial tree 'Weigher‘.
- In the below sections of the fields 'Weigher/Material assignment‘, the selected material as well as the weigher are displayed.

If the material has already been assigned to the weigher, the button 'Assign Material to weigher' has been disabled, otherwise the assignment is created by the button.

Note

To be on the safe side, a data record which has been newly created should not be enabled. Enabling should be executed only before the storage location is assigned.

Assign storage location

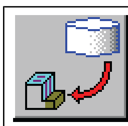
The adjustment data record for a material must be selected for the weigher. Selection in the partial tree of the weigher.

Via the button "Adjustment“ you reach the dialog for the adjustment of the data record.

There the material for the weigher should be enabled.

Enabling must be filed via the button 'Transfer‘.

The storage location must be selected within the storage location group.

	By pressing the button, the adjustment data record with the material is transferred. The assignment is finished.

Note

Enabling material for a weigher can be useful before the assignment.

Case of use:

At the beginning of the day all batches which should be produced are entered and enabled. It is known, that the delivery of the process input list comes in the course of the day. The production of the batches which use this process input list has not begun at this stage.

A.7.7.4 Remove storage location assignment

The storage location assignment deletes the material number (write value 0) in the storage location group data block of the storage location.

- Open the material stock dialog with the "Engineering / Components" menu in the recipe editor.
- Selection of the material for which the storage location assignment should be deleted.
- Via the button "Adjustment“ you reach the dialog for the adjustment of the data record. There, the material for the weigher should be disabled.
- Enabling must be filed via the button 'Transfer‘.
- The storage location within the storage location group must be selected.
- By pressing the button 'Enable storage location‘ the assignment is deleted.

A.7.7.5 Delete Offline Data record

You have to take the following steps in order to delete a data record in the offline database

- Open the material stock dialog via the recipe editor.
- Selection of the material for which the adjustment data record should be deleted.
- Pressing the button 'Delete assignment‘.

Note

When deleting the offline data record, no check is performed to ascertain whether the material is really available in the storage location.

Deleting has no effect on batches which have already been created and enabled. This means that the material is still expected in the storage location group.

A.7.7.6 Optimize adjustments

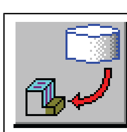
The values of the data block are compared with the values of the offline data set in order to optimize the adjustments for the weigher.

The dialog can be opened via two routes. Via the master material dialog and via the storage location dialog.

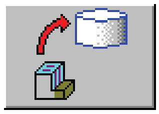
Changes can only be made via the column 'Input‘.

Inputs are only held in the dialog.

By pressing the button 'Insert input into PCU‘, the values are transferred into the data block.

	By pressing the button 'Insert input into PCU‘, the values are transferred into the data block.

The new values can be used for dosing.

	If the adjustments are optimized by the button 'Transfer online data from PCU into database', online data can be re-read and filed.

A.7.7.7 Automatic intake and outtake

Automatic intake and outtake

Via the telegrams type 21 an automatic intake and outtake can be realized. Via this telegram the release, the lock and the download of weigher settings can be performed.

This is only possible when the corresponding materials are created and the weigher datasets are stored in the material stock data.

Intake

For one storage group a batch (recipe), which executes the intake. In the last step of this procedure the weigher setting for the stored material is requested via the telegram 21. The engineer can decide to release the material at this point or later.

Outtake

When a silo becomes empty or other reasons exist for disabling a material in a silo, the material can be locked by sending a specific telegram type 21. So this material in the silo is not used for the next batches.

A.7.8 Compatibility with V4.x

A.7.8.1 General

The configuration of the Version V4.x can be transferred. You have to consider the following:

A.7.8.2 Material database

There is no master material database in the V4.x version. This database can be created using a separately available conversion tool ("conv4to5.exe"; for information about where you can obtain this tool, please refer to the manual preface). All existing materials stored in the component files ("compx.dbf") are transferred to the material stock database.

A.7.8.3 Material types

There are no material types in the V4.x version. These need to be newly configured. The accepted materials should be assigned to material types.

→ Material administration

A.7.8.4 Number of storage locations

In version V4.x the number of storage locations was defined by the length of the appropriate text file of the storage locations. You should configure them explicitly in Version V5.

→ Storage administration

A.7.8.5 Storage location group

The adjustment database can be addressed indirectly for the Version V5. Normally, the number of the adjustment database as well as the number of the storage location group are identical. You use a subscribing for the function 'Several Weighers to a storage location group'.

A.7.8.6 Fields of the adjustment database

The fields for material number, material ID and specific weight are adjustable in order to enable an adaptation to current component files.

A.7.8.7 Download and Upload of parameters

It was necessary to configure the upload and download of an adjustment parameter explicitly in Version V4.x. This configuration was made in the 'silon.ini' .

For Version V5 the upload and download of adjustment parameters is made via columns whose names are the same in the adjustment database (compx.def) and the corresponding configuration of the data block description (xx.pcu). Only parameters which are equal will be transferred.

The corresponding inputs in the 'silon.ini' will not be used any more.

A.7.9 Relation between Recipe and Batch

A.7.9.1 Configure weighing operation

A weighing operation consists of the operation and equipment parameters which are part of the weigher.

Opening the configuration dialog

Weighing operations which are configured in the PCU have to be configured in the IOS as well. The dialogs for this is opened using the menu item 'Engineering / Equipment data' in the recipe editor (see 'Recipe system' 'Equipment data dialog').

In the dialog's tree control, the operations/DFMs which were chosen for the weighers can be selected and configured.

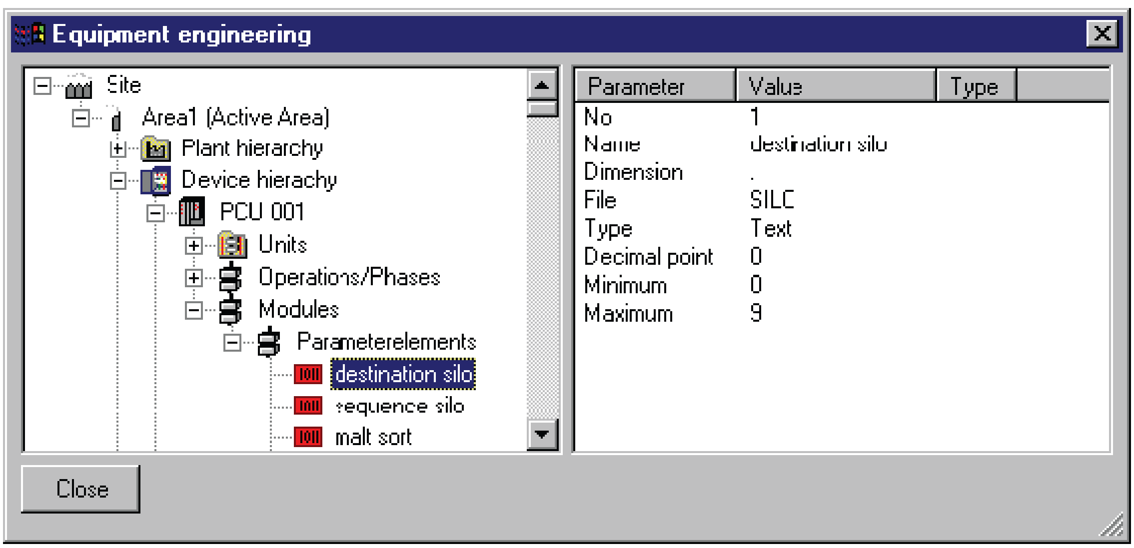
Equipment parameters of the operations

Basics

The technical parameters must be configured as a first step. There are 5 parameters in total required for a weighing operation. Three of the parameters are used in the PCU and have already been filed in the configuration of the basic function 'Weigher‘. The configuration should be completed for these parameters in the recipe system.

There are 2 more parameters for handling this operation in the recipe system ('Dosing group' and the 'reaction for a empty process input list').

Right-clicking will open the property dialog of technological parameters. See 'Recipe system' Technological Parameters.



For the weighing operation the properties for the parameter should be allocated as follows:

Material number

This parameter is configured as a TEXT parameter. The names are material names or IDs filed by the storage stock.

Name	Can be freely assigned
Unit	- (no)
Type	TEXT
Minimum	0
Maximum	The maximum value should conform to the number of materials which are defined in the offline database for this weigher/storage location group.
File	compx.txt: The name for the material is displayed (Process cell overview, Step protocol)

	comp#x.txt The ID is displayed for the material, x stands for the storage location group
Decimal point	Cannot be allocated with TEXT

Quantity

The quantity is generally assigned in kilograms with a decimal point of 3. Specification in grams is thus possible.

Name	Can be freely assigned
Unit	Kg
Type	SP
Minimum	0
Maximum	Indication depends on the capacity of the weigher. Indications are made in grams (decimal point clean value).
File	-
Decimal point	3 (input and output in grams)

Tolerance

Both the setpoint as well as the process values are allocated by the weigher for the parameter 'tolerance'. The parameter is used for logging the weighing result.

Name	Can be freely assigned
Unit	Kg
Type	SP
Minimum	0
Maximum	Not relevant (no input is possible)
File	-
Decimal point	3

Dosing group

This parameter is only used in IOS. The parameter is not accepted in the control recipe and does not appear in any step protocol.

See 'Control recipe generation'.

Name	Can be freely assigned
Unit	-x
Type	SP (for direct specification) TEXT (if a classification using names is required and possible)
Minimum	0
Maximum	Depends on the engineering
File	Is only necessary for the TEXT type.
Decimal point	0

Reaction with empty list

This parameter is only used in IOS. The parameter is not accepted in the control recipe and does not appear in any step protocol.

See 'control recipe generation'.

Name	Can be freely assigned
Unit	-
Type	TEXT
Minimum	1
Maximum	3
File	Dosreact
Decimal point	0

Content of the text file 'dosreact':

Line 0:	empty
Line 1:	Ignore
Line 2:	TA - Remote
Line 3:	Recipe load error

Used Tank

This parameter is set in the batch process input list. In the recipe it is write-protected and stored in the step protocol.

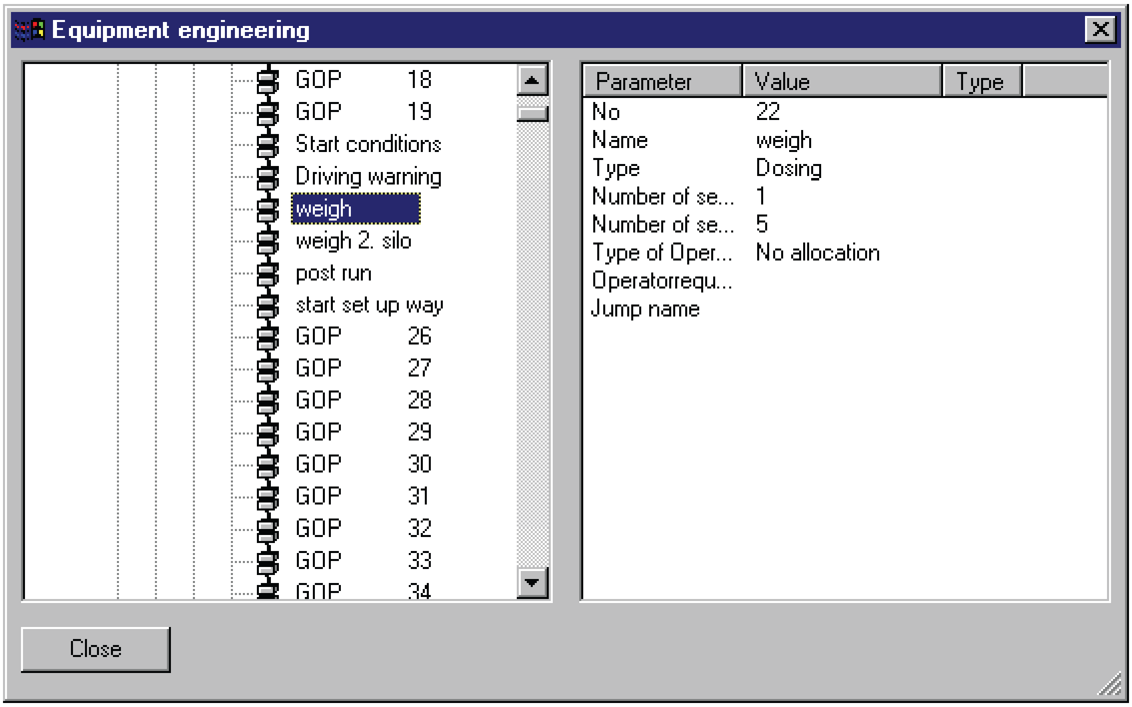
Name	Can be freely assigned
Unit	-
Type	SW (for direct input)
Minimum	0
Maximum	Depends on the engineering
File	-
Decimal point	0

Additional setpoints

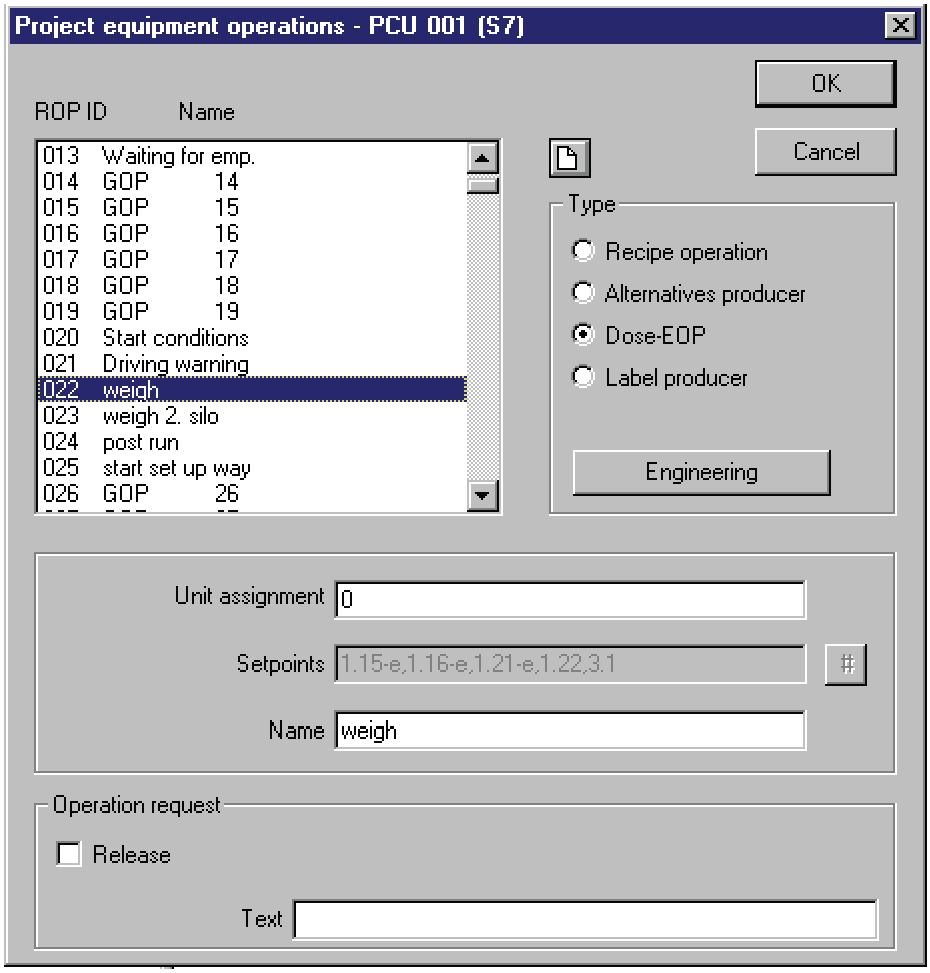
Additional setpoints are used as containers for the weighing parameter. They should be stored in the step protocol, and do not influence the weighing process. In the recipe they are write-protected ('-e'), and can be freely assigned and configured.

Configure equipment operation

In the figure below, you can see the navigation to the used operation via the device hierarchy.



Right-clicking opens the 'Project equipment operations' dialog.

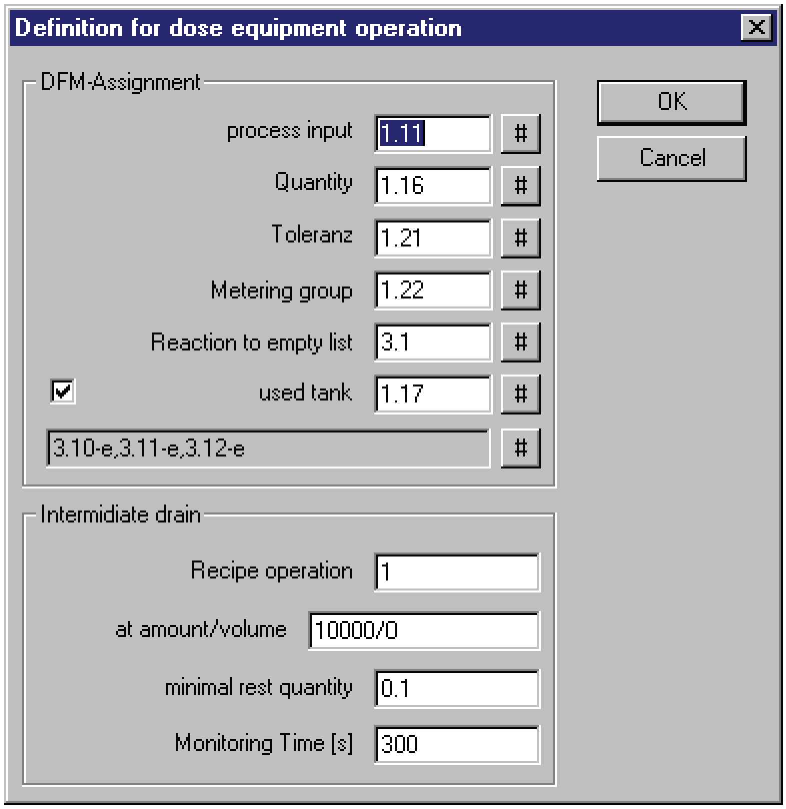


The following inputs are to be made for the EOP:

- The type of the operation must be set to 'Dose EOP'.
- Unit assignment: the appropriate unit needs to be entered here.
- The setpoints are configured via a specific dialog. This dialog is opened with the button '#'.
- A name for the operation must be allocated. The name can have a maximum length of 32 characters.
- If the operation which is configured in the PCU sets an operation request, the enabling and the operation text can be configured.

Configuration dialog of EOP 'Dosing'

Any further adjustments for a dosing operation are made in the dialog.



DFM assignment

- Material, volume, tolerance, metering group, and response to empty list
- "Used Tank" check box and input field for additional setpoint
- Setting the check box enables further input boxes for operator input. Here the user can program the DFM for the used tank, and additional DFMs which are write-protected in the recipe.
- The tank to be used can be selected only from the batch input materials list. The batch status must be locked.
- The batch input materials for the silos can be preset in a master system.

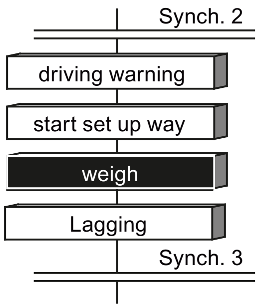
Data for the intermediate drain

- EOP for the intermediate drain.
- Number which should be added for an intermediate drain. It is possible to make indications for the weight and volumes. If the value for one of the two indications is 0, it will only be used for calculation. If both indications are set, reaching a limit value results in adding an intermediate discharge.

- Minimum rest number (see 'Control recipe generation').
- Monitoring time in seconds.

A.7.9.2 Create recipe unit procedure

The unit procedure for a weigher is created as usual. The dosings for a weigher are displayed via the dosing operation. All dosing events are added at the place where the dosing operation is.



The dosing operation is marked in the list display with blue color. The first three setpoints are not editable as those are set via the process input list.

Step	ROP ID	Name	Time	Setpoint 1	Setpoint 2	Setpoint 3	Setpoint 4	Setpoint 5
4	22	weigh	Time	process input[-]	Quartity[kg]	Toleranz[kg]	Metering group[-]	empty list[-]
			00:00:00	#	#	#	1	Rezept-Lade-Fehl

The meaning of the setpoints 4 and 5 is described with the control recipe generation. See 'control recipe generation'.

A.7.9.3 Create process input list for batch

Recipe process input list

All materials which are used for the production of a batch of this recipe are entered in the process input list. The quantity input is based on a standardized batch size. There is more information in the recipe system manual. See recipe system process input list.

Here, only the aspects which should be considered in relation to dosing are discussed.

Batch process input list

Enabling the batch will result in a batch process input list being compiled from the recipe process input list. In this process the storage location administration is used to define storage locations for the materials.

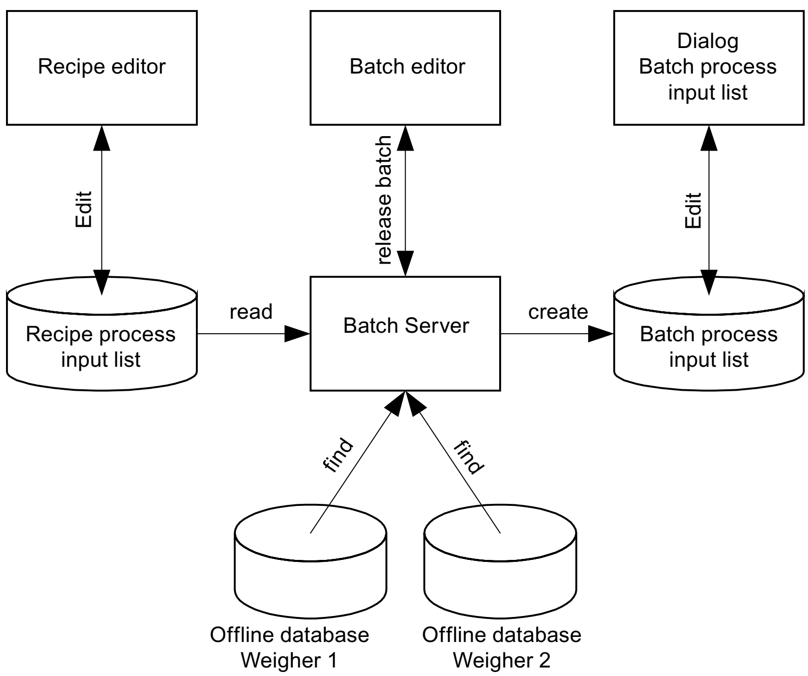
When the process input list for the "used tank" contains a value unequal to "automatic" (= 0), the system tries to locate the corresponding tank in the storage location group. The system responds based on the weigher configuration data if it does not find a tank under the specified number or if the tank contains insufficient or the wrong material:

- It reports an error,
- It tries to locate another tank/silo with the corresponding material in sufficient quantity, or
- It does not respond.

When the "automatic" parameter for the tank number is set, the system tries to locate a suitable tank as described below ("search order").

When a specific tank is selected for weighing from the batch process input list, the function checks the current volume of this silo.

See also section: "Batch processing / Batch list editor / Dialogs of the menus / Batch process input list"



Search sequence

You take the following steps when searching for storage locations:

You search for the sequence in the storage location groups.

You search for the sequence of the material number within the storage location group.

An input is found,

- if the Material ID conforms
- the input is not disabled and
- the line assignment of the storage group/weigher conforms to the batch.

The unit which is assigned to the storage location/weigher is entered in the process input list for the storage location which has been found.

The following inputs are listed in the batch process input list at the end:

- Material number
- Quantity
- Unit (consisting of PCU number and Unit number) and
- Dosing group.

Quantity adaptation

For creating the batch process input list, the quantity indications of the recipe process input list are adapted to the batch size. The nominal batch size is in the basic recipe header.

A linear scaling of the quantities is carried out.

Dosing group

When creating the batch process input list, the dosing group of each substance is determined.

If there is a dosing group in the recipe process input list, it will be transferred to the batch process input list.

If there are no inputs about the dosing group, the dosing group will be taken from the adjustment data record of the material.

Editing the batch process input list

Via a dialog the batch process input list can be edited.

You can open the dialog via

- The batch list
- The control recipe editor or
- The plant overview

.

For more information, refer to the section "Batch processing / ...".

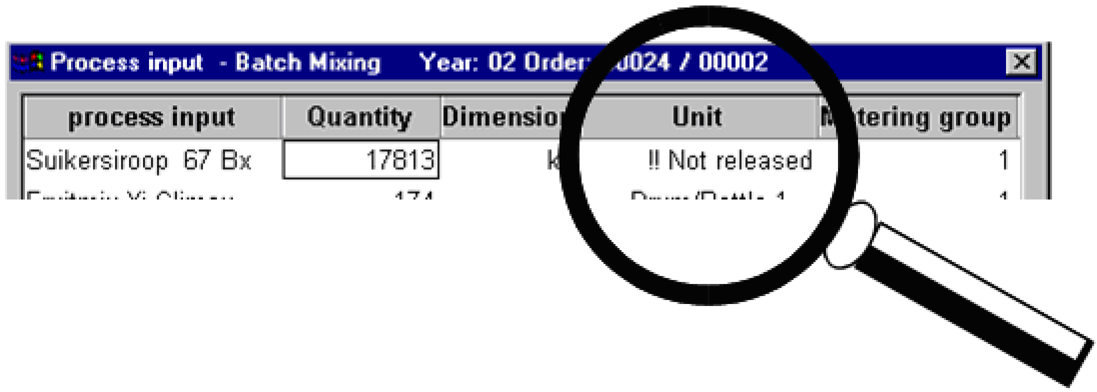
Error create batch process input list

Standard behavior

In the standard system the batch is set to error only when the complete list cannot be converted.

Error by one missing material

The behavior can be changed, so that one missing (no conversion possible) material causes the error of the batch.



The missing entries in the list are displayed in the unit column of the batch process input list. The list can be opened via the batch list editor (F8).

Release of the option

The following entry is required for this option:

Data file:	`<proj-path>\sys\complist.ini`
Key:	`[GLOBAL]`
Input:	`ErrByOneMissingMat=1`

A.7.9.4 Generating control recipes

Control recipe generation

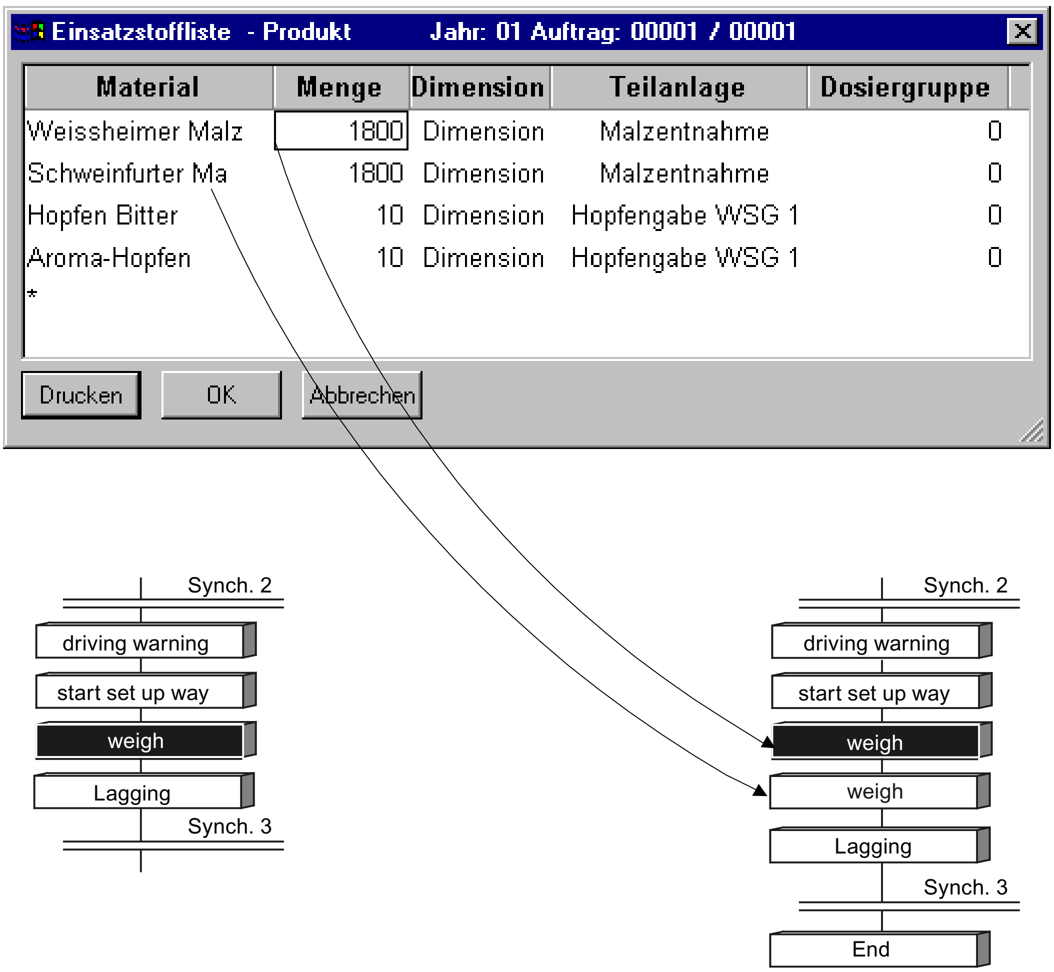
When creating the unit procedures of the control recipe, the recipe operations which are marked as recipe operations are treated in a special way.

A recipe operation is added for each suitable input of the process input list at the place where a dosing operation is.

An input is correct for the weigher if the unit conforms and the dosing group is correct.

Implementation

Example without any intermediate drain



Unit procedure of basic recipe

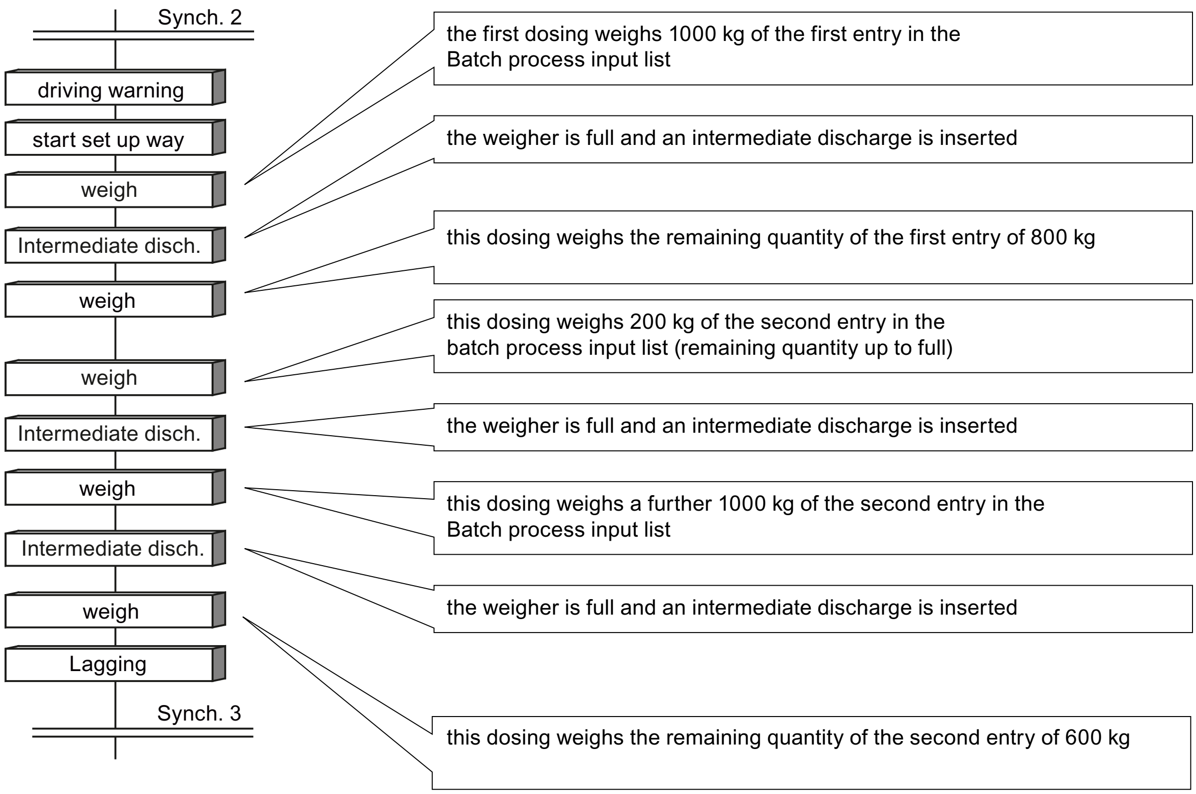
Unit procedure of control recipe

In the example, the two malt dosing events are converted into two dosing operations. The capacity of the weigher is bigger than both dosings. Therefore it is not necessary to insert an intermediate discharge.

Intermediate discharge

If the capacity of the weigher is reached, an intermediate discharge is inserted automatically. When adding, you ensure that the rest number does not fall below the configured minimum. The configured EOP is added for the intermediate discharge.

In the example the capacity was reduced to 1000 kg, so that the intermediate discharge can be added.



Intermediate discharge via volume

During configuration of the offline data structure, a field for the specific weight of the material should be allocated. With the structure for the SIWAREX weigher the data structure is the field 'SpecGrav'. The name of this field should be recorded as adjustment for the storage location group, if a different name is used.

Influence of the dosing group

With the search for inputs in the process input list, the dosing group is also considered besides the unit. A weighing operation will only be added if the dosing groups match. The value of the dosing group in the process input list and the value of the setpoint are equal.

Value 0

The value 0 has a special position. The value 0 as setpoint of the weighing operation adds all inputs from the process input list for this unit.

Reaction with empty list

If no input for the weigher is found, the reaction can be influenced via the last setpoint of the weighing operation.

Possible reactions are:

Ignore **(value 1)**	No weighing EOP is added. The control recipe is created without the weighing operation and transferred to the PCU.
Example:	• For cascaded weighers (weigher 1 discharges into weigher 2), weigher 2 should always be available in the control recipe. • Manual inputs within a 'normal' unit are made or not made.
TA - remove **(value 3)**	An empty control recipe is created and transferred to the PCU; this unit is removed from all synchronizations and alternatives. This means that the unit is removed from the control recipe.
Example:	Several products work together with a common recipe procedure. For one part of the products, process inputs for the weigher X are used, but not for the other part. In this case, it is not possible to remove the weigher X from the control recipes in which the weigher is no longer used. In this way you reach 2 targets: • If the weigher is not available due to repairs or errors, all products which do not use the process inputs of this weigher can be produced without manual intervention. • If a different process step is designed in parallel, this weigher can be used for another batch. The capacity utilization of this plant will be improved.
Recipe load error **(value 2)**	The control recipe is not created and not transferred to the PCU. The sequence control reacts with recipe load error. The processing of the control recipe is interrupted. An entry is logged in the message archive.
Example:	A weigher Y must pre-fill a hybrid container with wet components. Other weighers then dose dry substances onto them. If the dry substances were dosed directly into the hybrid container, they would become lumpy and the batch would not be useable. In this case, a dosing must take place in weigher Y; otherwise you cannot continue with the process.

A.7.10 What to do with errors?

A.7.10.1 General

Problem	Reason
You cannot select a batch process input list after having created a batch.	Is the process input list enabled?
The status 'error' is displayed when creating batches.	A fault has occurred while converting the recipe process input list into the batch process input list. There is no conforming storage location for any indicated materials. See also section: "Batch processing ..."
You cannot load a sequence with a weighing operation (recipe load error).	You should not weigh any material for this sequence. The following reaction is configured with 'Recipe load error'. You can check this in the diagnostic window of the recipe server. See also section: "Batch processing ..."

A.8 RCS CE-IF blocks (S7-400)

A.8.1 RC_IF_MOTOR (FB822)

Name

FB822

Description

This block is an interface between an RCS-CE and a motor provided by the user. The process is integrated (call, supply to the interface) via the user program.

Note

This block cannot be used in place of an ICM because the associated CE type (here: 1) is being used as an identifier for a motor within the scope of the RCS process data routing.

Inputs

`ID`	Control element number
`FB_ON`	Feedback ON
`MONITOR`	Monitor feedback • FALSE: The internal RCS feedback is simulated from QAUTO_ON . • TRUE: FB_ON is used as internal RCS feedback.
`MAN`	Manual operation is active
`LOCK`	1 = motor is locked
`ERROR`	Motor signals a fault
`DISABLED`	Locked (e.g. by maintenance work) Partial Routes with this element cannot be used by RCS. For a Route request the RCS server OS attempts to calculate the total route from Source to Destination via different Route no-des.

Outputs

`QRET_VAL`	Return value • 8223 The type of the control element is false (type <> 1 "MOTOR") • 8221 Control element ID is out of permitted range
`QAUTO_ON`	Automatic command ON
`QERR`	Error
`QBA_ID`	Batch number
`QSTEP_N1`	Step number

`QBA_NA`	Chargenname
`QROUTE`	Route ID of the Route which is currently using this Control Element.
`QFUNC_ID`	Function ID of the Route
`QMODE_NO`	Mode Number (1...32)

A.8.2 RC_IF_MOT_REV (FB823)

Name

FB823

Description

This block is an interface between an RCS-CE and a reverse motor provided by the user. The process is integrated (call, supply to the interface) via the user program.

Additional inputs (otherwise as for "RC_IF_MOTOR")

`FB_DIR` Direction of rotation feedback

- FALSE: forward
- TRUE: backward

Additional outputs

`QAUT_DIR` Automatic setting of the direction of rotation

- FALSE: forward
- TRUE: backward

A.8.3 RC_IF_MOT_SPED (FB824)

Name

FB824

Description

This block is an interface between an RCS-CE and a two-speed motor provided by the user. The process is integrated (call, supply to the interface) via the user program.

Additional inputs (otherwise as for "RC_IF_MOTOR")

`FB_SPEED`	Rotation speed feedback • FALSE: slow • TRUE: fast

Additional outputs

`QAUT_SPD`	Automatic setting of the rotation speed • FALSE: slow • TRUE: fast

A.8.4 RC_IF_VAL_MOT (FB825)

Name

FB825

Description

This block is an interface between an RCS-CE and a motorized valve provided by the user. The process is integrated (call, supply to the interface) via the user program.

Additional inputs (otherwise as for "RC_IF_MOTOR")

`FB_CLOSE`	Feedback OFF

Additional outputs

`QAUTO_OC`	Automatic setting open or closed • FALSE: closed • TRUE: open

A.8.5 RC_IF_VALVE (FB826)

Name

FB826

Description

This block is an interface between an RCS-CE and a valve provided by the user. The process is integrated (call, supply to the interface) via the user program.

Note

This block cannot be used in place of an ICM because the associated CE type (here: 5) is being used as an identifier for a valve within the scope of the RCS process data routing.

Additional inputs (otherwise as for "RC_IF_MOTOR")

`FB_CLOSE`	Feedback closed

A.8.6 RC_IF_CE_COMMON (FB827)

Name

FB827

Description

This block is an interface between an RCS-CE and a drive provided by the user. The process is integrated (call, supply to the interface) via the user program.

Inputs

`ID`	Control element number
`Active`	Drive is currently active, so not in neutral position
`FB_00 - FB12`	Feedback signals
`MONITOR`	Monitor feedback • FALSE: The internal RCS feedback is simulated from QAUTO_ON . • TRUE: FB_ON is used as internal RCS feedback.
`MAN`	Manual operation is active
`LOCK`	1 = motor is locked
`ERROR`	Motor signals a fault
`DISABLED`	Locked (e.g. by maintenance work) Partial Routes with this element cannot be used by RCS. For a Route request the RCS server OS attempts to calculate the total route from Source to Destination via different Route no-des.

Outputs

QRET_VAL	Return value • 8001 CE interface DB is not available • 8003 Invalid CE number • 8004 Invalid CE type
Q_C_00 - Q_C_15	Command signals
QERR	Error
QBA_ID	Batch number
QSTEP_N1	Step number
QBA_NA	Chargenname
QROUTE	Route ID of the Route which is currently using this Control Element.
QFUNC_ID	Function ID of the Route
QMODE_NO	Mode Number (1...32)

Index