SIEMENS

SIMATIC

PCS 7/APACS+ OS Channel Users Guide for V6.1

Manual

Edition 11/2005
A5E00687511-01

Safety Guidelines

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 to property damage only have no safety alert symbol. The 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**
with a safety alert symbol indicates that minor personal injury can result if proper precautions are not taken.

Caution
without a safety alert symbol indicates that property damage can result if proper precautions are not taken.

Notice
indicates that an unintended result or situation can occur if the corresponding notice is not taken into account.

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel.** Within the context of the safety notices in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

⚠ **Warning**
This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.
Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the 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
Automation and Drives
Postfach 4848
90437 NÜRNBERG
GERMANY

A5E00687511-01

Siemens AG 2005
Technical data subject to change.

Preface

Purpose of the Manual

This manual provides information necessary use to the Channel DLL interface that connects an APACS+ system to a PCS 7 OS V6.1.

Required Basic Knowledge

Readers are presumed to be expert in the use of APACS+ HMI software.

Where is this Manual valid?

This manual is valid for the software package PCS 7/APACS+ OS Options V6.1.

Training Centers

Siemens Technical Training Center provides extensive training for all levels of plant personnel to ensure optimal performance from APACS+ and PCS 7 control systems. Classes include extensive hands-on activities using appropriate equipment, making the training directly and immediately applicable.

On-line information is available: http://www.sea.siemens.com/sitrain

Siemens also 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:

Telephone: +49 (911) 895-3200.

Internet: http://www.sitrain.com

A&D Technical Support

Worldwide, available 24 hours a day:

United States: Johnson City, TN	**Worldwide:** Nürnberg	**Asia / Australia**: Beijing
Technical Support and Authorization Local time: Monday to Friday 8:00 AM to 5:00 PM Telephone:+1 (423) 262 2522 or +1 (800) 333-7421 (USA only) Fax:+1 (423) 262 2289 Mail to: techsupport.sea@siemens.com GMT: -5:00	**Technical Support** 24 hours a day, 365 days a year Phone:+49 (180) 5050-222 Fax:+49 (180) 5050-223 E-Mail: ad.support@siemens.com GMT:+1:00 **Authorization** Local time: Monday to Friday 8:00 AM to 5:00 PM Phone: +49 (180) 5050-222 Fax: +49 (180) 5050-223 Mail to: ad.support@siemens.com GMT: +1:00	**Technical Support and Authorization** Local time: Monday to Friday 8:00 AM to 5:00 PM Phone:+86 10 64 75 75 75 Fax:+86 10 64 74 74 74 Mail to:ad.support.asia@siemens.com GMT:+8:00
Automation and Drives Service and Support International http://www.siemens.com/automation/service&support		
The languages of the SIMATIC Hotlines and the authorization hotline are generally German and English.		

Contents

1 Editing APACS+ Channel Properties

1. To display the APACS+ System Parameters properties:
2. Select **Start > Simatic > WinCC > Window Control Center**.
   The WinCC Explorer opens.
3. Open the desired project if it has not opened automatically.
4. In the left pane, double-click **Tag Management**.
   APACS+ appears.
5. Double-click **APACS+**.
   APACS+ Unit appears.
6. Right-click **APACS+ Unit** and select **System Parameters**.
   The APACS+ System Parameters property page appears.

The following text describes the parameters:

1.1 Interface to APACS+ Controllers

The "Reset All Statistics" button will NOT work if the channel is not in runtime mode. The commands invoked write to a specific registry key that the channel monitors. When the key is set it clears all the statistic tags and updates the registry.
The default settings are valid for most user applications. Contact product support as described in the Preface to this manual.

1.2 Total Operations

Limits per resource - This is the number of operations (reads, writes, normal updates) that can be outstanding between the controller and the channel at any time. If there are more operations pending in the channel, they are waiting for operations at the controller to complete.

Limits per Interface - This is the total number of operations (reads, writes and updates) that can be outstanding between all of the controllers and the channel at any time.

In progress - This is the total number of operations that are currently outstanding between all controllers and the channel.

Queued - This is the number of operations that are waiting to be processed by an APACS+ controller because there are currently operations outstanding, and these operations would put the number above one of the limits.

Cumulative Total - This is the total number of reads and writes that have occurred to the APACS+ controllers. This can be reset in order to get statistical information about the channel (i.e., the number of operations over time.).

Cumulative Failed - This is the total number of read/write operations to the APACS+ controllers that failed. This can be reset in order to get statistical information about the channel (i.e., the number of operations over time.).

1.3 Single Tag Reads

Limit per Resource - This is the total number of reads (Asynch Read or Synch Read) that can be outstanding between an APACS+ controller and the channel. This number should be well below the Total Operations limit in order to keep read requests from preventing other data from updating.

Limit per Interface - This is the total number of reads (Asynch Read or Synch Read) that can be outstanding between all APACS+ controllers and the channel. This number should be well below the Total Operations limit in order to keep read requests from preventing other data from updating.

In Progress - This is the number of single tag reads currently outstanding between an APACS+ controller and the channel.

Queued - This is the number of single tag reads that are waiting to be processed by an APACS+ controller because there are currently single tag reads outstanding, and these reads would put the number above one of the limits.

Cumulative Total - This is the total number of single tag reads that have occurred between the channel and the APACS+ controllers. This can be reset in order to get statistical information about the channel (i.e., the number of reads over time.).

Cumulative Failed - This is the total number of single tag reads that have failed between the channel and the APACS+ controllers. This can be reset in order to get statistical information about the channel (i.e., the number of reads that failed over time.).

1.4 Single Tag Writes

Limit per Resource - This is the total number of writes (Asynch Write or Synch Write) that can be outstanding between an APACS+ controller and the channel. This number should be well below the Total Operations limit to keep write requests from preventing other data from updating.

Limit per Interface - This is the total number of writes (Asynch Write or Synch Write) that can be outstanding between all APACS+ controllers and the channel. This number should be well below the Total Operations limit to keep write requests from preventing other data from updating.

In Progress - This is the number of single tag writes currently outstanding between an APACS+ controller and the channel.

Queued - This is the number of single tag writes that are waiting to be processed by an APACS+ controller because there are currently single tag writes outstanding, and these writes would put the number above one of the limits.

Cumulative Total - This is the total number of single tag writes that have occurred between the channel and the APACS+ controllers. This can be reset to get statistical information about the channel (i.e., the number of writes over time.).

Cumulative Failed - This is the total number of single tag writes that have failed between the channel and the APACS+ controllers. This can be reset to get statistical information about the channel (i.e., the number of writes that failed over time.).

1.5 Tag List Reads

Limit per Resource - This is the total number of update operations (normal background processing) that can be outstanding between an APACS+ controller and the channel. This number should be on the order of the Total Operations limit in order to keep the data updating as fast as possible. The data tags are broken up into multiple lists for updating at the controller to optimize performance.

Limit per Interface - This is the total number of update operations (normal background processing) that can be outstanding between all APACS+ controllers and the channel. This number should be on the order of the Total Operations limit to keep the data updating as fast as possible.

In Progress - This is the number of update operations currently outstanding between an APACS+ controller and the channel.

Queued - This is the number of update operations that are waiting to be processed by an APACS+ controller because there are currently update operations outstanding, and these update operations would put the number above one of the limits.

Cumulative Total - This is the total number of update operations that have occurred between the channel and the APACS+ controllers. This can be reset to get statistical information about the channel (i.e., the number of updates over time.).

Cumulative Failed - This is the total number of update operations that have failed between the channel and the APACS+ controllers. This can be reset in order to get statistical information about the channel (i.e., the number of updates that failed over time.).

Poll Time - This is the most recent Loop time observed for exec thread loop, in milliseconds.

Last Poll Time - This is the last loop time in milliseconds observed for the APACS API execution thread.

Long Loop Trigger Time - This is the maximum time in milliseconds expected for good execution of the loop. When this is exceeded the total tag list overruns is incremented.

Poll Delay Time - This is the umber of milliseconds to delay between API polling loops

Maximum Last Poll Loop Time - This is the maximum `LastPollLoopTime`

Total Cumulative Long Loops - Yhis is the count of number of loops which exceeded `Limits.PollLoopTime` to execute, indicating excessive API delays, API problems or unexpected resolving of tags synchronously in Async calls

Total Tag List Overruns - This is the number of taglists which have exceeded their scan rate (could partially be due to delays in starting the scan due to limits on active operations). This indicates whether the scan rate can support the APACS+ workload.

1.6 Tag Counts

Active Tags - This is the total number of active APACS+ tags. If two OPC clients are reading the same APACS+ tag, this count will only increment by one.

Recent Inactive Tags - This is the number of APACS+ tags that have become inactive within the last 12 hours.

Inactive Tags - This is the number of APACS+ tags that have been inactive for more than 12 hours. These need to be re-verified by the APACS+ controller.

Total Tag Lists - This is the total number of APACS+ tag lists. The tags are broken up into multiple tag lists to optimize read performance.

Total Tags - This is the total number of APACS+ tags that are currently created in the channel.

Maximum Tag List Length - This is the maximum number of tags allowed in a tag list. This number can be changed to further optimize tag update performance.

Combine Single Tag Reads - This setting allows multiple single tag reads to be combined into a larger list in order to optimize performance of single tag reads.

Force Tag List Compare - This is the number of updates from the APACS+ controller where the value is actually read from the APACS+ controller whether it has changed or not, and injected into the channel. Normally, data updated from an APACS+ controller only occurs when a value has changed.

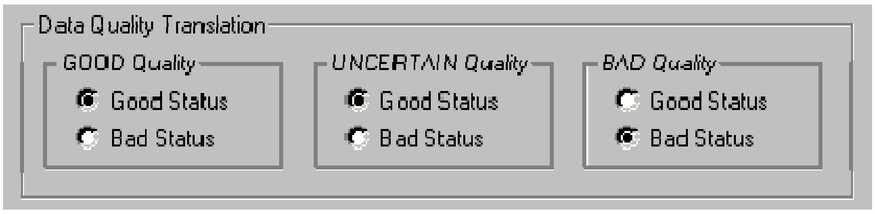
Tag Cache Aging Rate - When a tag is created in the channel, then deleted, information about the tag is stored in a cache so that future operations to the same tag can be quicker. This value represents the time, in milliseconds, that a tag will be held in the cache.

Total Lookups to Tag Cache - This is the total number of times that the tag cache has been checked for a tag that was previously used. This can be reset to get statistical information (i.e., the number of tag look-ups over time.)

Total Hits on Tag Cache - This is the total number of times that the tag cache was checked, and the requested item was found in it. This can be reset in order to get statistical information (i.e., the number of successful tag look-ups over time).

1.7 Data Quality Translation

APACS+ data has the characteristics of *good, uncertain, and bad.* The PCS7 environment regards data as either *good* or *bad*. The **Data Quality Translation** checkboxes provide an opportunity to determine how data is translated from the APACS+ system. The default translation is shown in the illustration below.



Data Quality Translation

GOOD Quality: ● Good Status ○ Bad Status

UNCERTAIN Quality: ● Good Status ○ Bad Status

BAD Quality: ○ Good Status ● Bad Status

2 Editing APACS+ Tag Parameters

The software channel connecting the APACS+ system and the OS includes a mechanism for manually creating OS tags and linking them to APACS+ data structures. This is done in the **APACS+ Tag Parameter** window.

Tags are ordinarily created automatically by the DBA utility, but this manual mechanism provides an opportunity to intervene.

1. Select **Start > Simatic > WinCC > Window Control Center**.
   The WinCC Explorer opens.
2. Open the desired project if it has not opened automatically.
3. In the left pane, double-click **Tag Management**.
   APACS+ appears.
4. Double-click **APACS+**.
   APACS+ Unit appears.
5. Right-click **APACS+ Unit** and select **New Tag**.
   The **Tag Properties Window** opens, with the default name *NewTag* and a defaullt datatype of *Binary Tag.* (The new tag is an OS tag. It is linked to an APACS+ tag in a later step.)
6. Assign an appropriate name and datatype to the tag. Assign a length if necessary. Click the **Select** button.
7. The window contains an **Address** tab and provides, for reference, the **OS Tag Name**. The **Item Name** field is editable. Enter the APACS+ tag name you wish to associate with the new OS tag.
8. Click the **OK** button to finish.

3 Examining the APACS+ Connection Parameter

The read-only APACS+ Connection Parameter indicates the current attributes of the APACS+ Unit connection. The default name is *NewConnection*, which can be changed when the connection is created.

Here is one way to display the parameter:

1. Select **Start > Simatic > WinCC > Window Control Center**.
   The WinCC Explorer opens.
2. Open the desired project if it has not opened automatically.
3. In the left pane, double-click **Tag Management**.
   APACS+ appears.
4. Double-click **APACS+**.
   APACS+ Unit appears.
5. Expand the **APACS+ Unit** entry.
   The current connections are listed.
6. Right-click a connection name of the connection of interest.
   A menu appears.
7. Select **Properties**.
   The **Connection Properties** window apppears.
8. Click the **Properties** button on this window.
   The **APACS+** Connection Parameter opens.

Since the parameter is read-only, you can close the window by clicking the **OK** or **Cancel** button.

4 Mapping Data Types

The DBA utility attempts to convert APACS+ data types to their PCS 7 equivalents. If you use the DBA utility, you should confirm that it has converted your APACS+ data types properly. If you use another method of supplying APACS+ data types in WinCC, the table provides the default and recommended conversions:

APACS+ Type	Recommended OS Data Type	Recommended Conversion
BOOL	BIT (BOOL)	NA
BYTE	BYTE	ByteToUnsignedByte
USINT	BYTE	ByteToUnsignedByte
SINT	SCHAR	CharToSignedByte
INT	SWORD	ShortToSignedWord
UINT	WORD	WordToUnsignedWord
WORD	WORD	WordToUnsignedWord
DINT	SDWORD	LongToSignedDword
UDINT	DWORD	DwordToUnsignedDword
DWORD	DWORD	DwordToUnsignedDword
TIME	DWORD	DwordToUnsignedDword
TIMEOFDAY	DWORD	DwordToUnsignedDword
DATE	DWORD	DwordToUnsignedDword
DATETIME	DWORD	DwordToUnsignedDword
ERROR_SUMMARY	DWORD	DwordToUnsignedDword
REAL	FLOAT	FloatToFloat
LREAL	DOUBLE	DoubleToDouble
STRING	TEXT_8	NA

5 Performance Optimization

The APACS API Subsystem includes many statistics and parameters to facilitate fine tuning the product and its load on the MBus/Controllers.

As a starting point, the default configuration should be used, as this will offer a reasonably performing configuration.

5.1 Checking for Overruns

Overruns can be detected using the $System.Apacs.Statistics.TotalOverruns value. In addition, $System.Apacs.Statistics.MaxReadListTime, and $System.Apacs.Statistics.MaxScanTime can be used to monitor the performance of taglist operations within the API. Similar statistics are also available on a per-controller basis.

Overruns can be caused by a number of factors:

- The controller may not have adequate processing time to process APACS communications. In this case the controller or other loads on the controller should be investigated. This would manifest itself as a high value for MaxReadListTime.
- The PC may not have adequate CPU time available. Check the performance monitor and confirm your system's CPU utilization. To be deterministic, the average should be less than 70% with only occasional peaks of 90-100%. If your PC’s CPU utlization is more than this, a faster PC must be used or the applications running on the PC should be distributed among multiple PCs.
- There may be excessive queuing delays for taglist reads. This manifests itself as a high value for MaxScanTime, while MaxReadListTime is low. Queuing delays can be caused by configuring too few APACS Communications buffers in the API (Apacs.ini) or by having too low a limit on Operations or ReadLists ($System.Apacs.Limits.OperationsActive, $System.Apacs.Limits.ReadListsActive, $System.Apacs.Resource.Limits.OperationsActive and $System.Apacs.Resource.Limits.ReadListsActive). These should be checked against the corresponding Max Queued and Max Active to identify where the bottleneck is.

5.2 Reducing Load on the Controllers

The load presented to the controllers can be reduced (with a corresponding decrease in performance) by reducing the limits discussed in the previous bulleted item. Of course, excessively reducing these limits will cause overruns.

Overruns

Under heavy load situations, or when controller or APACS communications performance is low, overruns can occur for taglist reads. An overrun indicates that a taglist could not be read within the scan time requested. There are two possible cases:

- The actual taglist read exceeded the scan time requested. In this case, the taglist will continue to be scanned, but to avoid consuming excessive resources, the next read of the taglist will be delayed by 100ms. This will allow a window for other taglists to get in and attempt their reads.
- The scan of the taglist (queuing time + read time = scan time) exceeded the scan time requested. In this case the taglist will continue to be scanned and will be scheduled to be read again as soon as possible.

The presence of overruns can be detected using the $System.Apacs.Statistics.TotalOverRuns system tag. Similarly, the $System.Apacs.Statistics.LastReadListTime and LastScanTime system tags will report the disposition of the taglist operations. The Max and Min for these counters can be used to identify how bad the overrun situation is.

6 Diagnostic Counters

In the WinCC Channel Diagnosis window, the *Channels/Connections* tab displays the current channel and connections to WinCC. Selecting a channel in the left pane displays the available counters and their value for the channel. Selecting a connection listed beneath a channel displays the available counters for the connection. The *Configuration* tab leads to functions that enable various logging mechanisms. Enabling logging can, however, degrade performance.

To display the WinCC Channel Diagnosis window:

With an APACS+ OS application running, Select
Start > Programs > WinCC > Tools > Channel Diagnosis.
The **WinCC Channel Diagnosis** Window opens with the connection name *APACS+.*

Click the plus sign in front of **APACS+** to display the connections.
The connection name displays, for example, *APACS+*

Click the channeconnection to reveal its diagnostic counters.

KEY	**Meaning**
R	Readable
W	Writeable
ML	Max Long, (2^31)-1
NA	Not applicable
R(#)	In the default column, indicates an initial value as well as what the variable is reset to via $System.ResetStatistics

Basic Channel Counters

System Tag/ Diagnostic Counter Name	**Type**	**RW**	**Range**	**Default**
$System.Version Chnl - Version	BSTR	R	NA	0.0.x
$System.TotalConnections Chnl - TotalConnections	I4	R	0-ML	0
$System.TotalVariables Chnl - TotalVariables	I4	R	0-ML	0

Basic APACS Counters

System Tag Diagnostic Counter Name	**Type**	**RW**	**Range**	**Default**

$System.Apacs.Version Apacs - Version	BSTR	R	NA	0.0.x
$System.Apacs.Heartbeat Apacs - Heartbeat	I4	R	0-ML	0
$System.Apacs.Statistics.SuccessRate Apacs - SuccessRate	I4	R	0-ML	0
$System.Apacs.NumResources Apacs - NumResources	I4	R	0-ML	0
$System.Apacs.ResourceNames Apacs - ResourceNames	BSTR	R	NA	
$System.Apacs.Statistics.TotalOperations Apacs - TotalOperations	I4	R	0-ML	0
$System.Apacs.Statistics.TotalReadLists Apacs - TotalReadLists	I4	R	0-ML	0
$System.Apacs.Statistics.TotalWrites Apacs - TotalWrites	I4	R	0-ML	0
$System.Apacs.Statistics.TotalReads Apacs - TotalReads	I4	R	0-ML	0
$System.Apacs.Statistics.TotalFailedOperations Apacs - TotalFailedOperations	I4	R	0-ML	0
$System.Apacs.Statistics.TotalFailedReadLists Apacs - TotalFailedReadLists	I4	R	0-ML	0
$System.Apacs.Statistics.TotalFailedWrites Apacs - TotalFailedWrites	I4	R	0-ML	0
$System.Apacs.Statistics.TotalFailedReads Apacs - TotalFailedReads	I4	R	0-ML	0
$System.Apacs.Statistics.TotalOverRuns Apacs - TotalOverRuns	I4	R	0-ML	0
$System.Apacs.Statistics.TotalReadLists Apacs - MaxReadListTime	I4	R	0-ML	0
$System.Apacs.Statistics.MaxScanTime Apacs - MaxScanTime	I4	R	0-ML	0
$System.Apacs.Statistics.MaxWriteTime Apacs - MaxWriteTime	I4	R	0-ML	0
$System.Apacs.Statistics.MaxPollLoopTime Apacs - MaxPollLoopTime	I4	R	0-ML	0

$System.Apacs.Statistics.TotalLongPollLoops Apacs - TotalLongPollLoops	I4	R	0-ML	0
$System.Apacs.Statistics.TotalLongWrites Apacs - TotalLongWrites	I4	R	0-ML	0
$System.Apacs.Statistics.NumTagLists Apacs - NumTagLists	I4	R	0-ML	0
$System.Apacs.Statistics.NumReadsActive Apacs - NumReadsActive	I4	R	0-ML	0
$System.Apacs.Statistics.NumReadListsActive Apacs - NumReadListsActive	I4	R	0-ML	0
$System.Apacs.Statistics.NumWritesActive Apacs - NumWritesActive	I4	R	0-ML	0
$System.Apacs.Statistics.NumOperationsActive Apacs - NumOperationsActive	I4	R	0-ML	0
$System.Apacs.Statistics.NumReadsQueued Apacs - NumReadsQueued	I4	R	0-ML	0
$System.Apacs.Statistics.NumReadListsQueued Apacs - NumReadListsQueued	I4	R	0-ML	0
$System.Apacs.Statistics.NumWritesQueued Apacs - NumWritesQueued	I4	R	0-ML	0
$System.Apacs.Statistics.NumOperationsQueued Apacs - NumOperationsQueued	I4	R	0-ML	0
$System.Apacs.Statistics.LastPollLoopTime Apacs - LastPollLoopTime	I4	R	0-ML	0
$System.Apacs.Statistics.LastReadListTime Apacs - LastReadListTime	I4	R	0-ML	0
$System.Apacs.Statistics.LastScanTime Apacs - LastScanTime	I4	R	0-ML	0
$System.Apacs.Statistics.LastWriteTime Apacs - LastWriteTime	I4	R	0-ML	0
$System.Apacs.Cache.TotalLookups Apacs - CacheTotalLookups	I4	R	0-ML	0
$System.Apacs.Cache.TotalHits Apacs - CacheTotalHits	I4	R	0-ML	0

$System.Apacs.Cache. NumTags Apacs - CacheNumTags	I4	R	0-ML	0
$System.Apacs.Cache. NumActiveTags Apacs - CacheNumActiveTags	I4	R	0-ML	0
$System.Apacs.Cache. NumRecentInactiveTags Apacs - CacheNumRecentInactiveTags	I4	R	0-ML	0
$System.Apacs.Cache. NumLongInactiveTags Apacs - CacheNumLongInactiveTags	I4	R	0-ML	0

Basic Channel Counters (Per-Connection)

System Tag Diagnostic Counter Name (per connection)	**Type**	**RW**	**Range**	**Default**
$System.CONNECTION. Statistics.NumReadThreadsActive Active Read Threads.	I4	R	0-ML	0
$System.CONNECTION. Statistics.TotalOperations Total Tag Operations	I4	R	0-ML	0
$System.CONNECTION. Statistics.TotalReads Total tag reads	I4	R	0-ML	0
$System.CONNECTION. Statistics. TotalWrites Total tag writes	I4	R	0-ML	0
$System.CONNECTION. Statistics. TotalFailedOperations Total tag failed operations	I4	R	0-ML	0
$System.CONNECTION. Statistics. TotalFailedReads Total tag failed reads	I4	R	0-ML	0
$System.CONNECTION. Statistics. TotalFailedWrites Total tag failed writes	I4	R	0-ML	0
$System.CONNECTION. Statistics. MaxReadThreadsActive Max active read threads	I4	R	0-ML	0
$System.CONNECTION. Statistics. MaxOperationsActive Max active tag operations	I4	R	0-ML	0
$System.CONNECTION. Statistics. MaxReadsActive Max active tag reads	I4	R	0-ML	0

$System.CONNECTION. Statistics. MaxWritesActive Max active tag writes	I4	R	0-ML	0
$System.CONNECTION. Statistics. MaxAsynchOperationsActive Max active async operations	I4	R	0-ML	0
$System.CONNECTION. Statistics. MaxAsynchReadsActive Max active async reads	I4	R	0-ML	0
$System.CONNECTION. Statistics. MaxAsynchWritesActive Max active async writes	I4	R	0-ML	0
$System.CONNECTION. Statistics. NumOperationsActive Num active tag operations	I4	R	0-ML	0
$System.CONNECTION. Statistics. NumReadsActive Num active tag reads	I4	R	0-ML	0
$System.CONNECTION. Statistics. NumWritesActive Num active tag writes	I4	R	0-ML	0
$System.CONNECTION. Statistics. NumAsynchOperationsActive Num active async operations	I4	R	0-ML	0
$System.CONNECTION. Statistics. NumAsynchReadsActive Num active async reads	I4	R	0-ML	0
$System.CONNECTION. Statistics. NumAsynchWritesActive Num active async writes	I4	R	0-ML	0

7 Channel System Tags Reference

System tags control and monitor application loading, performance and tuning. Internally, only the system tags actively being scanned by clients are scanned for changes. All system tags are scanned at the fastest update rate requested for any system tag. To this end it is recommended that clients scan system tags at modest rates (1-10 second update rates) and limit the number of system tags being scanned at any one time (10-200 is reasonable).

System tags all start with `$System.` as the first component of their name. System tags follow a naming convention that indicate whether the tag is a tunable parameter.

Any system tag which includes the ".Limits." in its name is an tunable tag.

Any system tag which includes the ".Statistics." component in its name is a statistical counter tag updated by the software and reset by the user. Performance and use information is generally maintained in the tags with "Statistics" in their name.

Most system tags are both read and write, but some are read-only. Writing to statistical tags resets the statistic or sets a threshold so that the activity over a specific time interval can be observed.

System tags are targeted for use by knowledgeable process engineering staff. The tags are not part of normal process control procedures, and should not generally be used by an operator at an HMI.

Some system tags (such as APACS per controller tags) are dynamically created. If you attempt to monitor these tags before they are created, they will have the Bad Quality attribute until they are created.

7.1 System Tag Naming Convention

A consistent naming convention describes System tags:

- All System tags will start with `$System`
- The prefix "`Num`" in a tagname implies a dynamically updating statistic tracking the present number of something (channels, tags, etc). This value is not writable by the Channel Subsystems.
- The prefix "`Max`" in a tagname implies a dynamically updating statistic tracking the maximum value observed for the corresponding statistic (typically a `Num`). Channel subsystems can write to these tags. Users may wish to clear the value so that they can observe a peak value over some time interval. In general all Num statistics have corresponding Max stastics (implemented using stag::CountVariable)
- The prefix "`Min`" in a tagname works the same as `Max`, but tracks the minimum value observed for the corresponding statistic. Only selected `Num` statistics will have a corresponding `Min` (implemented using stag::MinCountVariable). It is possible to write to these tags so the value can be "cleared." In this case, *clearing* would mean assigning a high value (such as MAX_INT) so that the first update to the statistic will decrease the value to reflect the true minimum over an interval.

- The prefix "`Total`" in a tagname implies an accumulating statistics. These values are distinct from `Max`, because the tag value constantly grows.These are writable by the channel subsystems . The intent is that a client may wish to clear the value so that it can observed accumulating over some time interval.
- To support more than one subsystem, system tag names for subsystem information will include the Subsystem as part of the tagname. For example: `$System.Apacs.NumResources`
- When a given subsystem has a significant number of system tags, they will be grouped into 2 main sub-groups:
- `Limits` – tuning parameters to adjust the performance of the sub-system. These will be writable (with a limited range of valid values)
- `Statistics` – statistics reported by the sub-system, many of which will be `Num`, `Max`, `Min` and `Total` style names.

7.2 Resetting Statistics

For System tags which record statistics, and can be cleared, the following approaches are supported:

- Individual specific tags are writable. Write 0 or some high value (for Min) to reset the statistic.
- A single boolean tag, `$System.ResetStatistics,` resets all tags to a reasonable value. When pulsed true, all statistics are reset. Writing false to the tag has no effect, and it will always read false. The tags reset by this method are documented in the various System Tag tables with R(#) in the Default column. In some cases, such as Max and Min tags, R(`name`) may be used where `name` is the name of the related tag whose value will be used during a reset.

7.3 Basic System Tags

The following system tags report status information about the overall product and the registry system tag interface:

KEY	**Meaning**
R	Readable
W	Writeable
ML	Max Long, (2^31)-1
NA	Not applicable
R(#)	In the default column, indicates an initial value as well as what the variable is reset to via $System.ResetStatistics

System Tag Name	Type	RW	Range	Default
$System.Version Version number of the product, major.minor.service_pack.build	BSTR	R	NA	4.0.0.x
$System.License True if Invalid License	BOOL	R	TRUE-FALSE	FALSE
$System.TotalConnections Count of connections to the channel	I4	R	0-ML	0
$System.TotalVariables Count of all variables	I4	R	0-ML	0
The APACS+ system tags provide current operational counters, limits, and statistics about the APACS+ subsystem.				
$System.Apacs.Heartbeat This tag is incremented (at approximately the PollDelay rate) to reflect that the channel is still operational and that the APACS+ API is not hung. Clients can monitor this tag is changing to determine that the channel is still working	I4	R	0-ML	NA
$System.Apacs.Limits. TagListLength Sets the maximum number of tags per APACS+ API Taglist.	I4	RW	1000	50
$System.Apacs.Limits. ForceTagListCompare The number of scans of a taglist after which a full compare of all tags on the taglist is forced (i.e., change list is ignored). If set to 0, all scans do a full compare.	I4	RW	0-ML	10
$System.Apacs.Limits. CombineRead Minimum number of single tag reads outstanding at once to cause a taglist to be used to read them, as opposed to doing single tag reads. Setting to 0 disables the use of taglists for single tag reads. Setting to 1 will cause single tag reads to always use taglists. Note the present API implementation requires 1 message per single tag read, regardless of whether it is resolving the tag or not. The API uses 1 message for a taglist read which is already resolved and 2 messages for a taglist read which is not yet resolved (actually 1 message per 3 tags on the taglist which need resolution). However, the resolution information for a single tag read is not retained to facilitate subsequent taglist reads of the tag.	I4	RW	0-1000	

$System.Apacs.Limits. MaxCombineRead Maximum number of single tag reads to combine into a single taglist. If CombineRead is 0, this is ignored. This value should be >= Limits.CombineRead and <= Limits.TagListLength. If it is outside these bounds, the corresponding boundary value will be used. If this value is set too large, a large number of API buffers could be used for resolution, possibly affecting the normal scanning of taglists on other resources. Conversely, in an environment where single tag reads are often used by clients, a larger value can improve performance of those reads.	I4	RW	1000	15
$System.Apacs.Limits. PollLoopTime Maximum time in milliseconds expected for a good Exec Loop. When a loop exceeds this limit, TotalLongPollLoops is incremented.	I4	RW	100-10000	500
$System.Apacs.Limits. PollDelay Number of milliseconds to delay between API polling loops	I4	RW	10-100	10
$System.Apacs.Limits. OperationsActive Maximum total of Reads, ReadLists and Writes to initiate within the APACS+ API at once.	I4	RW	1000	135
$System.Apacs.Limits. ReadsActive Maximum total single tag reads to initiate within the APACS+ API at once.	I4	RW	1000	10
$System.Apacs.Limits. ReadListsActive Maximum total taglist reads to initiate within the APACS+ API at once.	I4	RW	1000	100
$System.Apacs.Limits. WritesActive Maximum total single tag writes to initiate within the APACS+ API at once	I4	RW	1000	25
$System.Apacs.Resource.Limits. OperationsActive Maximum total of Reads, ReadLists and Writes to initiate against any one APACS+ resource at once. If this is 1, only a single operation will be done against the resource at once. This prevents write's from overlapping with reads and prevents single tag operations from overlapping with taglist operations, however overall throughput may be reduced.	I4	RW	1000	38
$System.Apacs.Resource.Limits. ReadsActive Maximum total single tag reads to initiate against any one APACS+ resource at once.	I4	RW	1000	3

$System.Apacs.Resource.Limits.ReadListsActive Maximum total taglist reads to initiate against any one APACS+ resource at once. If this is 1, only a single taglist will be read at once against a resource, reducing load on the resource and causing "taglist staggering", however overall throughput may be reduced	I4	RW	1000	10
$System.Apacs.Resource.Limits.WritesActive Maximum total single tag writes to initiate against any one APACS+ resource at once.	I4	RW	1000	25
$System.Apacs.ResourceNames A comma separated list of all the APACS+ Resources accessed. (this includes resources requested by clients which failed to be resolved). Note: even once all tags on a given resource are made inactive, the resource still remains in this list. This is necessary since this list also reflects the resource names available for resource level tuning and statistics. Hence tuning parameters set when a resource is first accessed will not be lost as tags go inactive and are re-activated for the resource.	STR	R	NA	NA
$System.Apacs.NumResources Number of resources in ResourcesNames	I4	R	0-ML	NA
$System.Apacs.MaxResources Max NumResources	I4	RW	0-ML	R(Num Resources)
$System.Apacs.Statistics.SuccessRate Out of the last 100 operations, how many were successful (if < 100 tag accesses has been performed since startup, shows 100-number of failures to date)	I4	R	0-100	R(100)
$System.Apacs.Statistics.TotalOperations Cumulative total tag reads, writes and taglist reads completed (success and failure)	I4	RW	0-ML	R(0)
$System.Apacs.Statistics.TotalReads Cumulative total single tag reads completed (success and failure)	I4	RW	0-ML	R(0)
$System.Apacs.Statistics.TotalReadLists Cumulative total taglist reads completed (success and failure)	I4	RW	0-ML	R(0)
$System.Apacs.Statistics.TotalWrites Cumulative total single tag writes completed (success and failure)	I4	RW	0-ML	R(0)
$System.Apacs.Statistics.TotalFailedOperations Cumulative total tag reads, writes and taglist reads completed with failure	I4	RW	0-ML	R(0)

$System.Apacs.Statistics. TotalFailedReads Cumulative total single tag reads completed with failure	I4	RW	0-ML	R(0)
$System.Apacs.Statistics. TotalFailedReadLists Cumulative total taglist reads completed with failure	I4	RW	0-ML	R(0)
$System.Apacs.Statistics. TotalFailedWrites Cumulative total single tag writes completed with failure	I4	RW	0-ML	R(0)
$System.Apacs.Statistics. NumOperationsActive Total Number of reads/readlists and writes presently in progress	I4	R	0-ML	NA
$System.Apacs.Statistics. NumReadsActive Number of Single Tag reads presently in progress	I4	R	0-ML	NA
$System.Apacs.Statistics. NumReadListsActive Number of Taglist reads presently in progress	I4	R	0-ML	NA
$System.Apacs.Statistics. NumWritesActive Number of Single Tag writes presently in progress	I4	R	0-ML	NA
$System.Apacs.Statistics. MaxOperationsActive Max NumOperationsActive	I4	RW	0-ML	R(Num Operations Active)
$System.Apacs.Statistics. MaxReadsActive Max NumReadsActive	I4	RW	0-ML	R(Num Reads Active)
$System.Apacs.Statistics. MaxReadListsActive Max NumReadListsActive	I4	RW	0-ML	R(Num Read Lists Active)
$System.Apacs.Statistics. MaxWritesActive Max NumWritesActive	I4	RW	0-ML	R(Num Writes Active)
$System.Apacs.Statistics. NumOperationsQueued Total Number of resources presently queued to read, readlist or write. (Note this count represents resources needing to do work, not number of tags/taglists needing to be read/written).	I4	R	0-ML	NA
$System.Apacs.Statistics. NumReadsQueued Total Number of resources presently queued to do single tag reads. (Note this count represents resources needing to do work, not number of tags needing to be read).	I4	R	0-ML	NA

$System.Apacs.Statistics. NumReadListsQueued Total Number of resources presently queued to do taglist reads. (Note this count represents resources needing to do work, not number of taglists needing to be read).	I4	R	0-ML	NA
$System.Apacs.Statistics. NumWritesQueued Total Number of resources presently queued to do single tag writes. (Note this count represents resources needing to do work, not number of tags needing to be written).	I4	R	0-ML	NA
$System.Apacs.Statistics. MaxOperationsQueued Max NumOperationsQueued	I4	RW	0-ML	R(Num Operations Queued)
$System.Apacs.Statistics. MaxReadsQueued Max NumReadsQueued	I4	RW	0-ML	R(Num Reads Queued)
Max NumReadListsQueued	I4	RW	0-ML	R(Num Read Lists Queued)
$System.Apacs.Statistics. MaxWritesQueued Max NumWritesQueued	I4	RW	0-ML	R(Num Writes Queued)
$System.Apacs.Statistics. TotalPollLoops Count of number of exec thread loops through polling all active APACS+ operations.	I4	RW	0-ML	R(0)
$System.Apacs.Statistics. LastPollLoopTime Most recent Loop time observed for exec thread loop, in milliseconds	I4	R	0-ML	NA
$System.Apacs.Statistics. MaxPollLoopTime Max LastPollLoopTime	I4	RW	0-ML	R(Last Poll Loop Time)
$System.Apacs.Statistics. MinPollLoopTime Min LastPollLoopTime	I4	RW	0-ML	R(Min Poll Loop Time)
$System.Apacs.Statistics. TotalLongPollLoops Count of number of loops which exceeded Limits.PollLoopTime to execute, indicating excessive API delays, API problems or unexpected resolving of tags synchronously in Async calls.	I4	RW	0-ML	R(0)
$System.Apacs.Statistics. NumTags Number of APACS+ API Data Tags allocated	I4	R	0-ML	NA

$System.Apacs.Statistics.MaxTags Max NumTags	I4	RW	0-ML	R(Num Items)
$System.Apacs.Statistics.NumTagLists Number of Apacs API Data Tag lists allocated including the 3 used for the tag cache (includes temporary taglists created to optimize groups of reads)	I4	R	0-ML	NA
$System.Apacs.Statistics.MaxTagLists Max NumTagLists	I4	RW	0-ML	R(Num Tag Lists)
$System.Apacs.Statistics.TotalOverRuns Number of taglists which have exceeded their scan rate (could partially be due to delays in starting the scan due to limits on active operations)	I4	RW	0-ML	R(0)
$System.Apacs.Statistics.LastScanTime Most recent Scan time observed for a taglist, in milliseconds. Scantime is the total from the scheduled start of the taglist read to the completion of the read	I4	R	0-ML	0
$System.Apacs.Statistics.MaxScanTime Max LastScanTime	I4	RW	0-ML	R(Last Scan Time)
$System.Apacs.Statistics.MinScanTime Min LastScanTime	I4	RW	0-ML	R(Last Scan Time)
$System.Apacs.Statistics.LastReadListTime Most recent read time observed for a taglist in milliseconds. Read time is the time from the actual start of the taglist read to the completion of the read	I4	R	0-ML	0
$System.Apacs.Statistics.MaxReadListTime Max LastReadListTime	I4	RW	0-ML	R(Last Read ListTime)
$System.Apacs.Statistics.MinReadListTime Min LastReadListTime	I4	RW	0-ML	R(Last Read ListTime)
$System.Apacs.Statistics.LastReadTime Most recent read time observed for a tag in milliseconds. Read time is the time from the actual start of the tag read to the completion of the read	I4	R	0-ML	0
$System.Apacs.Statistics.MaxReadTime Max LastReadTime	I4	RW	0-ML	R(Last Read Time)
$System.Apacs.Statistics.MinReadTime Min LastReadTime	I4	RW	0-ML	R(Last Read Time)

`$System.Apacs.Statistics.LastWriteTime` Most recent read time observed for a write in milliseconds. Write time is the time from the actual start of the write to the completion of the write	I4	R	0-ML	0
`$System.Apacs.Statistics.MaxWriteTime` Max LastWriteTime	I4	RW	0-ML	R(Last Write Time)
`$System.Apacs.Statistics.MinWriteTime` Min LastWriteTime	I4	RW	0-ML	R(Last Write Time)
`$System.Apacs.Cache.AgingRate` Rate at which tags should be aged, in milliseconds. The operation of cache aging is a 2 step process. Hence, tags will stay in the cache for AgingRate to 2*AgingRate once they are not in use anymore. Any changes to the aging rate will take effect within 1 second of the change, even if the change is a large increase or decrease in the aging rate. This is useful, especially if the aging rate is being decreased significantly from the default of 12 hours.	I4	RW	2000-ML	12*60*60* 1000 (12 hours)
`$System.Apacs.Cache.NumTags` Total of NumActiveTags, NumRecentInactiveTags and NumLongInactiveTags	I4	R	0-ML	NA
`$System.Apacs.Cache.MaxTags` Max NumTags	I4	RW	0-ML	R(Num Tags)
`$System.Apacs.Cache.NumActiveTags` Number of tags presently active (being scanned or having a single tag read/write done, i.e., associated with an ActiveItem).	I4	R	0-ML	NA
`$System.Apacs.Cache.MaxActiveTags` NumActiveTags	I4	RW	0-ML	R(Num Active Tags)
`$System.Apacs.Cache.NumRecentInactiveTags` The number of tags which have recently (<m_AgingRate) gone inactive. Tags in this category are subject to move to the LongInactive status on the next AgingRate loop of the cache aging thread.	I4	R	0-ML	NA

`$System.Apacs.Cache.MaxRecentInactiveTags`
R(NumRecentInactiveTags)
	I4	RW	0-ML	R(Num Recent Inactive Tags)

`$System.Apacs.Cache.NumLongInactiveTags` The number of tags which have been inactive for a long time (0 to m_AgingRate. Tags in this category are subject to be removed from the cache on the next AgingRate loop of the cache aging thread.	I4	R	0-ML	NA
`$System.Apacs.Cache.MaxLongInactiveTags` NumLongInactiveTags	I4	RW	0-ML	R(Num Long Inactive Tags)

$System.Apacs.Cache.TotalLookups Running total of lookups performed against the cache	I4	RW	0-ML	R(0)
$System.Apacs.Cache.TotalHits Running total of lookups which found the desired tag in the cache	I4	RW	0-ML	R(0)
$System.Apacs.Version Version string of the APACS+ API exactly as reported by the API. Only versions after 4.35 will be supported.	STR	R	NA	NA

7.4 APACS+ System Tags (Per-Controller)

The controller-specific APACS+ system tags are a set of tags that only exist when an APACS+ controller is presently being scanned. One complete set of the controller-specific system tags exists for every controller presently being scanned. The tag name syntax for each set of controller-specific system tags is

`$System.Apacs.RESOURCE.X,` where *RESOURCE* is a specific APACS+ controller name

System Tag Name	Type	RW	Range	Default
$System.Apacs.RESOURCE.Statistics.SuccessRate Out of the last 100 operations, how many were successful (if < 100 tag accesses has been performed since Channel startup, shows 100-number of failures to date).	I4	R	0-100	R(100)
$System.Apacs.RESOURCE.Statistics.TotalOperations Cumulative total tag reads, writes and taglist reads completed (success and failure) against the resource	I4	RW	0-ML	R(0)
$System.Apacs.RESOURCE.Statistics.TotalReads Cumulative total single tag reads completed (success and failure) against the resource	I4	RW	0-ML	R(0)
$System.Apacs.RESOURCE.Statistics.TotalReadLists Cumulative total taglist reads completed (success and failure) against the resource	I4	RW	0-ML	R(0)
$System.Apacs.RESOURCE.Statistics.TotalWrites Cumulative total single tag writes completed (success and failure) against the resource	I4	RW	0-ML	R(0)

`$System.Apacs.RESOURCE.Statistics.TotalFailedOperations` Cumulative total tag reads, writes and taglist reads completed with failure against the resource	I4	RW	0-ML	R(0)
`$System.Apacs.RESOURCE.Statistics.TotalFailedReads` Cumulative total single tag reads completed with failure against the resource	I4	RW	0-ML	R(0)
`$System.Apacs.RESOURCE.Statistics.TotalFailedReadLists` Cumulative total taglist reads completed with failure against the resource	I4	RW	0-ML	R(0)
`$System.Apacs.RESOURCE.Statistics.TotalFailedWrites` Cumulative total single tag writes completed with failure against the resource	I4	RW	0-ML	R(0)
`$System.Apacs.RESOURCE.Statistics.NumOperationsActive`	I4	R	0-ML	NA
`$System.Apacs.RESOURCE.Statistics.NumReadsActive` Total Number of reads/readlists and writes presently in progress against the resource	I4	R	0-ML	NA
`$System.Apacs.RESOURCE.Statistics.NumReadListsActive` Number of Taglist reads presently in progress against the resource	I4	R	0-ML	NA
`$System.Apacs.RESOURCE.Statistics.NumWritesActive` Number of Single Tag writes presently in progress against the resource	I4	R	0-ML	NA
`$System.Apacs.RESOURCE.Statistics.MaxOperationsActive` Max NumOperationsActive	I4	RW	0-ML	R (Num Operations Active)
`$System.Apacs.RESOURCE.Statistics.MaxReadsActive` Max NumReadsActive	I4	RW	0-ML	R (Num Reads Active)
`$System.Apacs.RESOURCE.Statistics.MaxReadListsActive` Max NumReadListsActive	I4	RW	0-ML	R (Num Read ListsActive)
`$System.Apacs.RESOURCE.Statistics.MaxWritesActive` Max NumWritesActive	I4	RW	0-ML	R (Num WritesActive)

$System.Apacs.RESOURCE.Statistics.NumOperationsQueued Total Number of reads, readlists or writes presently queued against the resource	I4	R	0-ML	NA
$System.Apacs.RESOURCE.Statistics.NumReadsQueued Total Number of single tag reads presently queued against the resource	I4	R	0-ML	NA
$System.Apacs.RESOURCE.Statistics.NumReadListsQueued Total Number of taglist reads presently queued against the resource	I4	R	0-ML	NA
$System.Apacs.RESOURCE.Statistics.NumWritesQueued Total Number of single tag writes presently queued against the resource	I4	R	0-ML	NA
$System.Apacs.RESOURCE.Statistics.MaxOperationsQueued Max NumOperationsQueued	I4	RW	0-ML	R (Num Operations Queued)
$System.Apacs.RESOURCE.Statistics.MaxReadsQueued Max NumReadsQueued	I4	RW	0-ML	R (Num Reads Queued)
$System.Apacs.RESOURCE.Statistics.MaxReadListsQueued MaxNumReadListsQueued	I4	RW	0-ML	R (Num ReadLists Queued)
$System.Apacs.RESOURCE.Statistics.MaxWritesQueued Max NumWritesQueued	I4	RW	0-ML	R (NumWrites Queued)
$System.Apacs.RESOURCE.Statistics.NumItems Number of items active in the resource.	I4	R	0-ML	NA
$System.Apacs.RESOURCE.Statistics.MaxItems Max NumItems	I4	RW	0-ML	R (Num Items)
$System.Apacs.RESOURCE.Statistics.NumTagLists Number of taglists active in the resource (includes temporary taglists created to optimize groups of reads)	I4	R	0-ML	NA
$System.Apacs.RESOURCE.Statistics.MaxTagLists Max NumTagLists	I4	RW	0-ML	R (Num TagLists)
$System.Apacs.RESOURCE.Statistics.TotalOverRuns Number of taglists which have exceeded their scan rate (could partially be due to delays in starting the scan due to limits on active operations)	I4	RW	0-ML	R(0)

$System.Apacs.RESOURCE.Statistics.LastScanTime Most recent Scan time observed for a taglist, in milliseconds. Scantime is the total from the scheduled start of the taglist read to the completion of the read	I4	R	0-ML	0
$System.Apacs.RESOURCE.Statistics.MaxScanTime Max LastScanTime	I4	RW	0-ML	R (Last ScanTime)
$System.Apacs.RESOURCE.Statistics.MinScanTime Min LastScanTime	I4	RW	0-ML	R (Last ScanTime)
$System.Apacs.RESOURCE.Statistics.LastReadListTime Most recent read time observed for a taglist in milliseconds. Read time is the time from the actual start of the taglist read to the completion of the read	I4	R	0-ML	0
$System.Apacs.RESOURCE.Statistics.MaxReadListTime Max LastReadListTime	I4	RW	0-ML	R (LastRead ListTime)
$System.Apacs.RESOURCE.Statistics.MinReadListTime Min LastReadListTime	I4	RW	0-ML	R (LastRead ListTime)

7.5 Channel System Tags (Per-Connection)

The connection-specific system tags are a set of tags that only exist when an APACS+ channel has a connection from the data manager.One complete set of connection-specific system tags exists for every connection. The tag name syntax for each set of connection-specific system tags is:

$System.CONNECTION.X, where *CONNECTION* is a specific connection name

System Tag Name	Type	RW	Range	Default
$System.Apacs.CONNECTION.Statistics.NumReadThreadsActive Active Read Threads.	I4	R	0-ML	R(0)
$System.Apacs.CONNECTION.Statistics.TotalOperations Total Tag Operations	I4	R	0-ML	R(0)
$System.Apacs.CONNECTION.Statistics.TotalReads Total tag reads	I4	R	0-ML	R(0)
$System.Apacs.CONNECTION.Statistics.TotalWrites Total tag writes	I4	R	0-ML	R(0)

`$System.Apacs.CONNECTION.Statistics.TotalFailedOperations` Total tag failed operations	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.TotalFailedReads` Total tag failed reads	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.TotalFailedWrites` Total tag failed write	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.MaxReadThreadsActive` Max active read threads	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.MaxOperationsActive` Max active tag operations	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.MaxReadsActive` Max active tag reads	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.MaxWritesActive` Max active tag writes	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.MaxAsynchOperationsActive` Max active async operations	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.MaxAsynchReadsActive`	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.MaxAsynchWritesActive` Max active async writes	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.NumOperationsActive` Num active tag operations	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.NumReadsActive` Num active tag reads	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.NumWritesActive` Num active tag writes	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.NumAsynchOperationsActive` Num active async operations	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.NumAsynchReadsActive` Num active async reads	I4	R	0-ML	R(0)
`$System.Apacs.CONNECTION.Statistics.NumAsynchWritesActive` Num active async writes	I4	R	0-ML	R(0)

7.6 APACS Tag Cache

A tag cache allows previously resolved tags to stay in memory for a period of time. Hence new requests to access tags which have been recently used (but are not presently used) can be satisfied without the need to re-resolve the tag.

When working with a significant number of tags, this can greatly improve performance.

To understand the benefit of this feature, it is important to understand the nature of the APACS communications protocol. Within APACS, the performance of a given operation is most affected by the number of APACS messages needed to accomplish the operation. The most efficient operation is a taglist read. This operation can fetch the value of approximately 50 tags (the actual number depends on the data type of the tags, strings can fetch much less), in a single APACS message. In comparison, a single tag read reads 1 tag in a single message. Due to communications overheads and waits for the controller to reach the point where it processes APACS communications, the actual processing time for the message is not significant in these cases.

In comparison, the resolution message resolves between 1 and 3 tags per message (the combinereads feature (see System tags) can be used to allow the APACS API Interface to take advantage of the ability to resolve up to 3 tags at a time). In the case of resolution, the controller must do a search of its tag database for the tag with the specified name. If the controller database is large (more than 500 tags), this search could take a significant amount of time, hence reducing the opportunity for the controller to process other APACS communications messages during the present controller poll loop. Compounding the matter is the fact that the tag resolution does not actually read the tag, it merely provides the information necessary for the read message to be sent. As a result, the resolution of 90 tags will take at least 30 resolution messages and 2 taglist read messages. In actual use, more than 2 taglist read messages may be issued (due to APACS/API communications inefficiencies). In comparison, those same 90 tags could be read (if already resolved), in 2 messages. That is a savings of 30 messages, which when comparing 32 to 2 messages is a 16 times performance boost.

Given the above discussion, it is clear why tag resolution caching can be important. Within the APACS API Interface, DATA_TAGs are retained after their initial resolution. These same DATA_TAGs can be placed on DATA_TAGLISTs while being scanned. In addition, they are kept on one of 3 unscanned DATA_TAGLISTS within the tag cache.

The tag cache uses the following three taglists:

Active Tags – those which are presently being scanned

Recent Inactive Tags – tags which are not being scanned, but were read/written or scanned within the last AgingRate milliseconds (default AgingRate is 12 hours, which equals 43,200,000 milliseconds)

Long Inactive Tags – tags which are not being scanned and have not been read/written or scanned for AgingRate or longer.

Every AgingRate, the cache frees all the tags in the long inactive list and moves all the tags from the recent Inactive List to the long inactive list. Once tags are freed, their next usage (read, write, scan, AddItems, ValidateItem, etc) will require a resolution operation.

It is important to note that tags which are being read via synchronous or asynchronous read/write, and are not actively being scanned, will spend most of their time on the recent Inactive list. Each time the tag is read/written or scanned, it is moved on the active list. In the case of a single synchronous read/write, once the operation completes, the tag returns to the recent inactive list. Hence it is important to account for this when selecting an AgingRate. Picking too low an AgingRate could require a re-resolution for those single tag operations.

In contrast, tags which are actively being scanned (as opposed to being explicitly read/written) will spend most of their time on the Active list. Such tags will not be aged and will not incur resolution costs.