



UIS Messaging Optimization

Best Practices, Techniques, and Tips for Tuning a Universal Interface Service (UIS)

Fall, 2014

Contents

- Contents** **2**
- Overview** **3**
- Hardware Considerations** **3**
 - CPU Processing Speed 3
 - Memory Volume and Speed 3
 - Hard-Disk Performance 4
 - Best Practices, Techniques, and Tips 4
- UIS Messaging Considerations** **4**
 - Message Creation 4
 - Message Processing 5
 - Response Processing 6
 - Transaction processing 6
 - Point processing 7
 - Best Practices, Techniques, and Tips 8
- Conclusion** **8**
- Appendix A: Setting Up Proper UIS Monitoring** **9**
- Appendix B: Message Flow Diagram** **12**

Overview

The ability to poll external field devices is central to the overall design and aim of CygNet SCADA. The key polling service – the Universal Interface Service (UIS) – is *the* CygNet SCADA polling workhorse.

The UIS is a current value service that performs real-time messaging. It powers the communication of device data to and from the field, and it plays a key role in processing retrieved data for broader use in a CygNet system.

A UIS can be seen as performing two high-level tasks: communicating with field devices and communicating with other CygNet services. It is paired with a Device Definition Service (DDS), which is a database of device configurations that provide the organization and structure of field data as it is consumed by CygNet SCADA. But the UIS also interfaces with internal services, like the Point Service (PNT), supplying data where required within the CygNet system.

This paper focuses on ways to tune a UIS. It attempts to help you understand how a UIS works in practical terms, and it attempts to help you balance the two high-level roles of a UIS: polling and processing.

However, there is no single formula available for instructing you how to set up or administer your UIS to satisfy your specific needs. There are many variables – forethought, planning, and experimentation are necessary.

Hardware Considerations

When implementing or administering a UIS, first consider the server hardware on which the service runs. No amount of software tuning can help you if your hardware is not suited to your UIS demands. Such a consideration must be balanced with the qualities and capabilities of a UIS, as well as your specific needs.

The most important hardware variables to consider are CPU processing speed, memory volume and speed, and hard-disk performance.

CPU Processing Speed

CPU speed affects how fast UIS messages are processed. CPU speed is a function of the total number of physical cores in your production server and the frequency of each of those cores. The more cores in a multiprocessor and the faster those cores, the more potential processing speed.

Memory Volume and Speed

Memory volume and speed also affect how fast UIS messages are processed. An ample amount of physical memory is required to handle the high volume of activity a production system creates. Your hardware will under-perform if your server operating system regularly swaps physical memory with virtual memory, creating page faults, in order to compensate.

To calculate an adequate amount of memory, see the latest "CygNet Memory and Disk Calculator" spreadsheet located in your product source.

Hard-Disk Performance

Hard-disk performance needs to be suited to the volume of read/write traffic handled by your system. Such traffic includes DDS transactions, point value updates, historical value updates, event record updates, and audit record updates. But such traffic also includes less obvious activities, like the creation and management of data persist files and change queues.

DDS and UIS configuration settings are available to help regulate the amount of hard-disk space traffic (that is, I/O) created by DDS and UIS needs. For more information about hard-disk performance, see the "CygNet and Solid State Drive" technical note (20141015).

Best Practices, Techniques, and Tips

It would be impossible to list all possible variables that affect UIS performance, but it is possible to lay out many of the known variables, along with advice.

- Set up applicable service monitoring points to monitor server machine performance.
- Do not configure queues to use the maximum number of cores (that is, 16).
- Balance the number of threads with the number and speed of cores available.
- Avoid situations where all cores are approaching 100% usage. Cores must be able to handle non-CygNet needs.
- Observe individual core history when your highest-volume polls execute to gauge how individual cores are handling maximum loads.
- Avoid situations where disk I/O causes the disk queue length to exceed 1.00 for an extended period of time.

For detailed procedures, see [Appendix A: Setting Up Proper UIS Monitoring](#).

UIS Messaging Considerations

Each UIS message is part of a cycle. The cycle starts when a message is created and completes only after a message has been fully processed.

For a high-level diagram, see [Appendix B: Message Flow Diagram](#).

Message Creation

A UIS message is a single instance of communication of data group data to and from a field device. Exactly what data is communicated on which leg of the trip depends on the data group and whether the message is a send or a get.

Usually, but not always, there is a 1:1 relationship between a message and a transaction. But a single message can be made up of multiple transactions. That is to say, a message might require several communications with a field device in order to satisfy message requirements.

A UIS message is created by a DDS. The DDS contains records that depend on a UIS to execute communication on its behalf. However, a message might be initiated in some other specialized service, like a Master Scheduling Service (MSS). And multiple messages might be chained together if they are part of a UIS command.

When a command is created by a DDS, it contains the following primary pieces of information:

- command name
- resolution criteria

The command name merely labels a logical collection of data. The resolution criteria specify the components of that collection. Put more simply, the command name describes the purpose of the command, and the components, or data groups, fulfill that purpose.

For example, you might name a command "Basic Poll" and include data groups in it that contain the data you want to make up the command.

Each data group in a command is always considered a discrete collection of properties and data. But each data group in a command is always linked to the other data groups in the command throughout a message cycle.

Message Processing

After a message is initiated, it is passed to the applicable communication device. The applicable communication device is whichever one is associated with the remote device that is associated with the message.

The message's first stop is the communication queue. Each communication device includes a pending communication queue. The pending communication queue handles messages on a first-in/first-out (FIFO) basis. But the entire message must be processed (or time out) before another queued message can enter communication processing.

The communication queue is affected by dynamic prioritization. Messages can enter the queue at a higher priority level than messages already within the queue. For example, if a manually initiated poll is executed, the corresponding message bypasses other messages in the queue if they have a lower priority, but it does not bypass other messages with the same or higher priority. Prioritization can be controlled by a setting if a message is part of a command as scheduled in an MSS.

Prioritization affects the position of other messages already in a queue. While higher priority messages jump ahead of lower priority messages, lower priority messages can fall farther and farther behind if the queue is not tuned correctly. This can cause an out-of-memory state or result in duplicate requests.

Data groups in a command are chained together until they exit the communication queue. At that point, the command is broken down into bytes and sent to the applicable field device.

Responses from a field device return to the communication device where they originated, and they are evaluated. If a response is acceptable and does not warrant a retry,

error, or some other action, the complete response for the entire command is typically sent to response processing.

To monitor the state of this queue, go to CygNet Explorer, select the applicable UIS, click **Comm Devices**, highlight the applicable communication device, right-click in the device pane, and select **View Pending Queue**.

Response Processing

After a response is evaluated and vetted, it is sent to a response processing queue selector.

In the selector, an important evaluation occurs: data groups are evaluated to see if an XML response record needs to be written to disk or not. (Sometimes the XML response record is referred to as a "DDS transaction," so its queue is called the transaction processing response queue.) If it needs to be written, messages go to the transaction processing response queue. If it does not need to be written, messages go to the point processing response queue.

These are the four cases under which XML response records are always written:

- Every time a user manually issues a read or write. For instance, from a DDS editor or CygNet Studio button.
- Every time a data group is written to a field device.
- Every time certain data groups are read from a device. This varies by EIE and data group type. For example, GMR and FMS data groups always result in transactions.
- Every time the last data group in a command includes a **forceSave** attribute set to **true**. Every data group has the **forceSave** setting, although it is not always exposed in a device template file.

The last bullet point above presents a situation you must take note of. If the case is true, then all data groups in the command go to the transaction processing response queue. If the case is false, then all data groups in the command go to the point processing response queue. This outcome can significantly affect how efficiently and quickly messages are processed.

Messages that end up in the transaction processing queue still receive point processing, if appropriate. And messages that end up in the point processing queue still receive transaction processing, if appropriate. Both processors are exactly the same. The evaluation merely decides which kind of message belongs in which queue, and therefore sends slower processing transaction messages to the transaction processing response queue and faster processing point messages to the point processing response queue.

To monitor the state of this queue, go to CygNet Explorer, select the applicable UIS, click **Comm Devices**, right-click in the device pane, and select **View Wait Queue**.

Transaction processing

If an XML response record needs to be written to disk, a message moves to the transaction response queue, then on to a processor.

Transaction processing causes an XML record to be saved to storage on a per-data-group basis. Such records are created in cases where the XML response record must be

persisted for some period of time, as when a write occurs or when dynagraph cards are collected.

Creation of the XML response record results in a relatively high processing burden for a server. The burden increases with message size – the larger the message, the more processing that occurs and the more XML that must be written. XML response records should therefore be created selectively.

Assuming you've carefully specified that only relevant XML response records are written and that your commands are created in a way that optimizes processing queue selection, you still must consider XML response record retention.

XML response record retention (that is, how long a XML response record is retained) is subject to the default DDS scrubbing rate and to settings made in data group properties in the DDS. The default DDS scrubbing rate is 30 minutes between scrubbing periods; the scrubbing period itself might vary in duration depending on the number of records to be deleted.

If any data item in the message requires point processing, that also occurs at this stage of processing, even though the message went to the transaction processing queue and not to the point processing queue.

By default, three cores are configured to be available for use by each response processor. (Each core processes a single thread at a time.) But with the release of v8.1.2 CygNet SCADA, you can configure up to 16 cores per response processor. Two specialized keywords are available to configure the number cores used per response processor. They are **TXTHREADS** and **PNTTHREADS**. If you choose to change either of these values from its default value, only do so after fine tuning the other variables discussed in this paper.

To monitor the state of this queue, go to CygNet Explorer, select the applicable UIS, click **Comm Devices**, right-click in the device pane, and select **View Response Transaction Queue**.

Point processing

If an XML response record does not need to be written to disk, a message moves to the point response queue, then on to a processor.

Every data element mapped to a PNT point requires point processing. This might include updating a PNT point value, updating alarm settings for the point, historical point processing, and so forth. Only applicable values associated with a PNT point record are updated.

Because no XML response record is written when a point is updated, point processing puts a relatively low processing burden on a server.

If any data item in the command requires XML response record processing, that also occurs at this stage of processing, even though the message went to the point processing queue and not to the transaction processing queue.

By default, three cores are configured to be available for use by each response processor. (Each core processes a single thread at a time.) But with the release of v8.1.2

CygNet SCADA, you can configure up to 16 cores per response processor. Two specialized keywords are available to configure the number cores used per response processor. They are **TXTHREADS** and **PNTTHREADS**. If you choose to change either of these values from its default value, only do so after fine tuning the other variables discussed in this paper.

To monitor the state of this queue, go to CygNet Explorer, select the applicable UIS, click **Comm Devices**, right-click in the device pane, and select **View Response Point Queue**.

Best Practices, Techniques, and Tips

It would be impossible to list all possible variables that affect UIS performance, but it is possible to lay out many of the known variables, along with advice.

- Set up applicable service monitoring points to monitor UIS performance.
- Carefully configure communication devices, considering settings like timeouts and delays.
- Be aware of the amount of data returned by a polled data group. The larger the amount of data, the slower the processing of that data.
- Ensure that the status of the last data group in a command represents how you want all data groups in the command processed.
- Schedule polls so that you poll only as often as is necessary to suit your needs.
- Balance polling. For example, do not overload your system with massive polls followed by little polling activity.
- Prioritize scheduled commands with queue performance in mind.
- Balance the number of communication threads with CPU processing speed to ensure that queues do not back up.

For detailed procedures, see [Appendix A: Setting Up Proper UIS Monitoring](#).

Conclusion

The central importance of a UIS demands that it be properly implemented and administered to ensure that a high volume of operational data is consumed quickly, efficiently, and dependably.

But there is no single formula available for instructing you how best to set up or administer your UIS to satisfy your specific needs.

You should carefully consider server hardware, message processing, and response processing. Many variables affect how optimally your UIS works.

About CygNet Software

CygNet Software provides enterprise software and industry solutions that collect, manage, and distribute essential operations data. Robust, proven, and evergreen, CygNet extends customer horizons with improved operational efficiency and increased market responsiveness.

>> Visit www.cygnets.com for more information.

Legal Notices

Copyright © 2000-2015 CygNet Software (A Weatherford company). All rights reserved.

Appendix A: Setting Up Proper UIS Monitoring

The following procedures are sets of best practices. They are not intended to provide a foolproof method for monitoring a UIS. Your needs dictate what is best in your specific production environment.

To Monitor Individual Core Usage

1. Go to your Windows **Start** menu and click **Administrative Tools**.
2. In the directory that opens, click **Resource Monitor**.
 - a. Once in Resource Monitor, click the **CPU** tab.
 - b. Execute your highest-volume poll. Then watch the CPU trends to gauge how well individual cores handle maximum loads. Overall CPU usage percentage is not a trustworthy indicator of core performance.
3. If core usage is unsatisfactory, make UIS or related settings changes and return to Resource Monitor to view the results as another high-volume poll executes.

To Monitor Memory Usage with CygNet Studio

1. Add service monitoring points to your CygNet SCADA system.
 - a. Open the **CygNet ServiceMon Administration** client.
 - b. Click **Info Items**.
 - c. In the **Site Service** field, click the Site.UIS you want to monitor.
 - d. Add the following points to the **Points to create** list:
 - **SVMPGFAULT**
 - **SVMPGFILE**
 - **SVMPRIV**
 - **SVMTHRDFLT**
 - **SVMVIRT**
 - e. Click **Create Info Item Points**.
2. Download a CygNet Studio screen to display trends of the service monitoring points above.
 - a. Open CygNet Explorer and go to your Application BLOB Service (APPS).
 - b. In the APPS service pane, locate the SCREXAMP folder, then locate the UDC_TREND.csf file in the folder.
 - c. Right-click UDC_TREND.csf, select **Download File**, and download it to a location of your choosing.
 - d. Rename the file to indicate its purpose. For example, name it SVM_MEMORY.csf.
3. Edit the CygNet Studio screen to display trends of the service monitoring points above.
 - a. Using CygNet Studio, open the .csf file you downloaded above.
 - b. In the **Site Service** list, select the applicable Service Monitoring (SVCMON) service.
 - c. In the **Facility** list, select the applicable facility. Typically, this is the UIS the SVCMON is monitoring.

- d. In the **Uniform Data Code** fields, select the SVCMON points you created above.
 - e. Make other screen edits to suit your needs.
4. Proceed to monitor the configured SVCMON points.

To Monitor Disk Usage with CygNet Studio

1. Add service monitoring points to your CygNet SCADA system.
 - a. Open the **CygNet ServiceMon Administration** client.
 - b. Click **Info Items**.
 - c. In the **Site Service** field, click the Site.UIS you want to monitor.
 - d. Add the following points to the **Points to create** list:
 - **SVMADQLT**
 - e. Click **Create Info Item Points**.
2. Download a CygNet Studio screen to display trends of the service monitoring points above.
 - a. Open CygNet Explorer and go to your Application BLOB Service (APPS).
 - b. In the APPS service pane, locate the SCREXAMP folder, then locate the UDC_TREND.csf file in the folder.
 - c. Right-click UDC_TREND.csf, select **Download File**, and download it to a location of your choosing.
 - d. Rename the file to indicate its purpose. For example, name it SVM_DISK.csf.
3. Edit the CygNet Studio screen to display trends of the service monitoring points above.
 - a. Using CygNet Studio, open the .csf file you downloaded above.
 - b. In the **Site Service** list, select the applicable Service Monitoring (SVCMON) service.
 - c. In the **Facility** list, select the applicable facility. Typically, this is the UIS the SVCMON is monitoring.
 - d. In the **Uniform Data Code** fields, select the SVCMON points you created above.
 - e. Make other screen edits to suit your needs.
4. Proceed to monitor the configured SVCMON points.

To Monitor Queue Counts with CygNet Studio

1. Add service monitoring points to your CygNet SCADA system.
 - a. Open the **CygNet ServiceMon Administration** client.
 - b. Click **Info Items**.
 - c. In the **Site Service** field, click the Site.UIS you want to monitor.
 - d. Add the following points to the **Points to create** list:
 - **SVMUISPNDC**
 - **SVMUISRPQC**
 - **SVMUISRPQP**
 - **SVMUISRPQT**
 - e. Click **Create Info Item Points**.
2. Download a CygNet Studio screen to display trends of the service monitoring points above.

- a. Open CygNet Explorer and go to your Application BLOB Service (APPS).
 - b. In the APPS service pane, locate the SCREXAMP folder, then locate the UDC_TREND.csf file in the folder.
 - c. Right-click UDC_TREND.csf, select **Download File**, and download it to a location of your choosing.
 - d. Rename the file to indicate its purpose. For example, name it SVM_QUEUE.csf.
3. Edit the CygNet Studio screen to display trends of the service monitoring points above.
 - a. Using CygNet Studio, open the .csf file you downloaded above.
 - b. In the **Site Service** list, select the applicable Service Monitoring (SVCMON) service.
 - c. In the **Facility** list, select the applicable facility. Typically, this is the UIS the SVCMON is monitoring.
 - d. In the **Uniform Data Code** fields, select the SVCMON points you created above.
 - e. Make other screen edits to suit your needs.
 4. Proceed to monitor the configured SVCMON points.

To Set Up Long-Term Point Monitoring

1. Open the Utilities folder in your CygNet SCADA source and open the **Point Configuration Manager** utility (PointCfgMgr.cfg).
 - a. Select the applicable Point Service (PNT).
 - b. Select all applicable service monitoring points and click **Modify Points**. (The service monitoring points included in the procedures above are applicable.)
 - c. In the dialog box that appears, click **Select Attribute**.
 - d. In the dialog box that appears, go to the **Select Attribute by Category** drop-down box, select the **Point: History**, and select **History On Line Days**.
 - e. In the **New Value** field, type the number of days you want to retain service monitoring data. For example, type 1095, which is the number of days in three years.
 - f. Click **Update Point Records**, then click **Yes** when asked to confirm.
 - g. Click **Close**. You can now study long-term performance on the service monitoring points just edited.
2. Make any other point edits you require.

Appendix B: Message Flow Diagram

