CHAPTER 14

Troubleshooting

Overview

When issues arise during TES operations, the operator must gather as much information as possible to troubleshoot the problem. Each release of TES undergoes rigorous testing before it becomes available to customers. Problems that are discovered during in-house testing can be resolved but the large variety of system configurations that exist outside our Quality Assurance labs mean that unknown issues inevitably arise as TES enters the field.

Some difficulties commonly reported by our customers are noted later in this troubleshooting section but most of the issues that users encounter are unknown until they are discovered in a customer's individual system environment.

Our Technical Services team works with our developers to diagnose and fix issues as they are reported. A vital part of this process is collecting diagnostic information on the processes occurring in TES at the time of the problem. While any information that the operator can recall during the time is useful, the best method is to turn on diagnostic logging and record in detail information on system processes when the problem occurs.

If a problem occurs during operation of TES, Support will ask you to enable the diagnostic logging for the system component that seems most likely as the source of the problem. To be proactive and to reduce our customers downtime as much as possible, we recommend that you follow the procedures provided below for turning on diagnostic logging. Try to reproduce the problem so the state of the system can be recorded when the problem occurs.

Collecting diagnostic information before you contact Technical Services provides a head start to resolving the problem. When you are experiencing problems with TES as indicated by error messages, turning on diagnostic logging for that system component collects information that can be used to pinpoint the problem.

Note If an anti-virus program will be scanning folders where TES components reside, configure the program to avoid scanning the log and diagnostic files of the master and client components. The constant updating that these files undergo will consume a large amount of processing resources.

Collecting Diagnostic Information

The **Logs** pane in TES shows basis auditing information and some error gathering from normal operation. This is enough to tell us that there is a problem, but it is usually not enough to explain *why* there is a problem.

The term, logging, generally refers to an auditing level of information that is always available without special configuration. Diagnostic logging refers to information that is gathered only for troubleshooting purposes and is not normally turned on.

When trying to troubleshoot an issue with the master, client or agent components of Tidal TES the diagnostic and logging tools that come standard with the application can narrow down and pinpoint where the existing problem might be.

Other systems that TES is working with such as SAP, PeopleSoft, Oracle Applications and various operating systems provide their own diagnostic and logging information. This diagnostic information may be needed to get to the root of any given problem, so working with the appropriate system administrators is crucial to troubleshooting success. Anything from a database trace or a network sniffer to the operating system's own internal logging can be helpful.

Diagnostic Logging

When a problem occurs that prevents TES from running correctly, diagnostic logging collects information about the various processes that are running on the system as the problem occurs. You collect information to find clues in the various system messages about the cause of the problem.

Often the error messages displayed in the client console that are generated by a problem, provide an indication of the source of the problem revealing which component should be monitored. If no source is indicated in the error message, check the messages in the system log for clues to the source of the problem. Once the source of the problem is identified, you can enable diagnostic logging for that system component. Select the level of logging detail for the system components being monitored from the **Logging** tab of the **System Configuration** dialog.

Be careful when using the diagnostic logging function as logging messages can consume a large amount of disk space very quickly. Carefully monitor the size of the log file being created as excessive logging consumes system resources as well as disk space. Try to repeat the scenario that produced the problem, so system activity can be recorded for Technical Services to interpret. Once the problem occurs again, contact Technical Services and send them the logging file that is generated.

Note It is recommended that anti-virus software either be disabled during diagnostic logging or configured to not check the diagnostic files that are created during diagnostic logging. The constant writing of diagnostic information to these files will consume too much attention from the anti-virus software and consume an extensive amount of system resources. The default location for diagnostic logs on the master machine is C:\Program Files\Tidal\Scheduler\master\logs.

Troubleshooting the Master

Verifying the Master's Version Number

Technical Services need to know which version of the TES master is being used. This information is available in the **Master Status** window.

To verify the master's version number:

Step 1 In the **Navigator** pane of the client, select **Operations>Master Status** to display the **Master Status** pane.

Step 2 The version of the master being used is noted at the top of the **Master Status** pane.

Diagnostic Logging for the Master

If you are enabling diagnostics before contacting Technical Services, start at the **Info** level and Technical Services will provide additional instructions after receiving the diagnostic files you have recorded.

The following components are monitored:

- **Scheduler Log** – Records system level messages regarding the master.
- **Client Manager Log** – Records messages about client activity.
- **Agent Manager Log** – Records messages from agents running jobs.
- **Compiler Log** – Records messages about the status of production schedules being compiled.
- **Job Manager Log** – Records messages about the status of jobs.
- **Event Manager Log** – Records messages about events defined in TES.
- **Queue Manager Log** – Records messages about queue activity.
- **Database Log** – Records messages relating to the state of the database.
- **Communications Log** – Records messages concerning all defined connections and sockets. Be aware that setting this component to a high level of logging results in a large amount of information that consumes large amounts of disk space.

Each component has a drop-down list with seven levels of progressively more detailed logging. Each level includes the messages of the previous levels of logging. The levels of logging are:

- **None** – No logging for the component.
- **Severe** – Logs only serious problems for that component (default).
- **Warning** – Logs potential problems for the component as well as messages from the **Severe** logging level.
- **Info** – Logs status messages about the normal operation as well as messages from lower logging levels.
- **Low Debug** – Logs important debugging messages as well as messages from lower logging levels.
- **Medium Debug** – Logs an increasing amount of debugging information as well as messages from lower logging levels.
- **High Debug** – Logs the largest amount of debugging information as well as messages from lower logging levels.

Diagnostic data is recorded to the log file until it reaches one megabyte in size. Once the log file reaches a megabyte in size, it is saved, named and another log file is started. The diagnostic log files are located in the **Log** directory of the master. The current log file is named *Master-YYYYMMDDHHMMSS.xmllog* where the name provides the year, month, day, hour (24 hour clock), minute and second of the log file, e.g., *Master-20020814042231.xml log*.

There are two reasons why a log file may be less than one megabyte in size.

- The master shut down (whether by design or not) before the log file could reach one megabyte.
- It is the current log file that the master is still creating.

Note Do not delete the current log file, which is always the log file with the latest timestamp. Even if the file does not exist, the master will continue to relay diagnostic information to the log file until it has relayed 1 MB of information. At that point, the master starts a new log file but any diagnostic information from the time between the deletion of the current log file and the creation of a new log file is lost.

Adjusting the Log File Retention

Each time the log file for the master reaches one megabyte in size, it is saved and another log file started. By default, the master is configured to store the 25 most recent log files. When the log files exceed the maximum number, the oldest log file is automatically deleted to make room for a newer log file.

You may need to adjust the log file retention number whenever you are running diagnostic logging. A higher level of diagnostic detail will create log files that add up quickly. Increase the log file retention number whenever you run diagnostic logging to compensate or you will start deleting the earliest log files before you have a chance to examine them.

To adjust the log file retention:

Step 1 From the command line of the master machine, stop the master:

tesm stop

Step 2 Locate the *Master.Props* file in the **config** directory of the master machine and open it.

Step 3 Add the following line to the *Master.Props* file:

MaxLogFiles=<number of log files>

where **<number of log files>** is the maximum number of log files to retain before deleting the oldest file.

The minimum number of log files to be retained is three. There is no maximum number.

Step 4 From the command line, restart the master:

tesm start

Purging Old Job Data

On occasion, the Admiral database and Tidal schema will not stop growing in size and very old data builds up in the database. This old data should be purged on a regular basis or the database will continue to bloat.

To purge old job data:

Step 1 Select the **System Configuration** option from the **Activities** main menu to check the data retention value in the **System Configuration** dialog.

Step 2 On the **Master** tab, verify that the **Automatic Daily History Cleanup** option is selected and select it if it is not already selected. If you are using a database alias other than the default “Admiral,” this feature may not operate correctly.

Running Diagnostic Logging as a Job

If a problem is occurring at a time that is difficult for an operator to monitor, say when a certain job deep in the production schedule runs, you can run a job to enable diagnostic logging just as the suspected job runs. You can create a logging job from the command line using a Java utility that TES provides. Configure the job to run just before the problem occurs and then create another similar job to return the diagnostic logging levels to their original settings after recording the system information. This job will limit how long the diagnostic logging runs to prevent massive amounts of information from clogging your disk space.

To run diagnostic logging as a job:

Step 1 Click the **Add Job** button on the TES toolbar or right-click in the **Jobs** pane and select **Add Job** to display the **Job Definition** dialog.

Step 2 Type a name for the job.

Step 3 In the **Command** field on the **Program** tab, type `java`.

Step 4 In the **Command Parameters** field, modify the following command parameters as needed:

```
java -DTidal_HOME=/opt/Tidal/master -cp/opt/Tidal/lib/scheduler.jar
com.tidalsoft.scheduler.SetDebugLevel
-m <Master Machine Name>
-p ClientPort(defaults to correct number)
-S (Scheduler Log)SEVERE|WARNING|INFO|LOW|MEDIUM|HIGH
-CL (Client Manager Log)SEVERE|WARNING|INFO|LOW|MEDIUM|HIGH
-A (Agent Manager Log)SEVERE|WARNING|INFO|LOW|MEDIUM|HIGH
-CO (Compiler Log)SEVERE|WARNING|INFO|LOW|MEDIUM|HIGH
-J (Job Manager Log)SEVERE|WARNING|INFO|LOW|MEDIUM|HIGH
-E (Event Manager Log)SEVERE|WARNING|INFO|LOW|MEDIUM|HIGH
-Q (Queue Manager Log)SEVERE|WARNING|INFO|LOW|MEDIUM|HIGH
-D (Database Log) SEVERE|WARNING|INFO|LOW|MEDIUM|HIGH
-COM(Comununcations Log)SEVERE|WARNING|INFO|LOW|MEDIUM|HIGH
```

Do not include any of the explanatory text that is in parentheses or brackets. Replace the text in brackets with the name of the machine that the master resides upon. The section of the first line "opt/Tidal" should be the directory location where the master is installed. Select just one of the logging levels to use with only the components you wish to enable for diagnostic logging. Any component log that is not included in the parameters is not affected by the logging job and retains its original setting. Don't forget to run another job to return the diagnostics to the original setting.

Running the Windows Master in Debug Mode

Even if you have configured the master to log diagnostic information, there is still a period of time between when the master starts and when it begins logging information. If the issue you are trying to troubleshoot occurs in this time window before diagnostic information is recorded than you should configure the master to run in debug mode. This will ensure that information is collected the entire time the master runs.

To run the Windows master in **Debug** mode:

Step 1 Open a command prompt screen from the **Start** button menu by selecting **Programs>Accessories>Command Prompt**.

Step 2 Type the directory path to the master executable file and then place a debug parameter after, e.g.,

"C:\Program Files\Tidal\TES\master\samaster.exe" -debug

Turn off the debugging mode by killing the **samaster.exe** process that you started.

Running the Unix Master in Debug Mode

Even if you have configured the master to log diagnostic information, there is still a period of time between when the master starts and when it begins logging information. If the issue you are trying to troubleshoot occurs in this time window before diagnostic information is recorded than you should configure the master to run in debug mode. This will ensure that information is collected the entire time the master runs.

From the command line program on the Unix master machine, type:

tesm -debug start

Turn off the debugging mode by killing the debug process that you started.

OCSEXIT Jobs

If you find that jobs created using the **OCSEXIT** variable, and that run on Windows agents, consistently end in **Completed Abnormally** you may need to update your system path.

To update your system path:

Step 1 Click the Windows **Start** menu button and select **Settings>Control Panel**.

Step 2 Start the **System Properties** control panel.

Step 3 Click the appropriate tab and append **%systemroot%\system32** to your system's **Path** variable.

Master Error

The Windows TES master runs as a Windows service. Services need to be started and controlled through a service manager, such as the TES Service Control Manager utility. Double-clicking directly on the *samaster.exe* file or an icon shortcut associated with the **samaster** executable will result in application errors and an access violation.

To resolve this issue, always start and stop the TES master through the Service Control Manager or the Windows Services program.

Email Problems

Note If you are having trouble sending email through Scheduler, examine the Windows Event Viewer for application errors attributed to TidalSAMASTER. The information provided in these logs can help diagnose the problem.

Internet Email

For TES to use internet mail effectively, the master machine must have a continuous internet connection. If the mail system goes offline, you will miss email notifications. Verify that you have followed the email prerequisites and configured the email functions of TES.

Microsoft Exchange and Novell GroupWise

To use email in TES with Microsoft Exchange and Novell GroupWise, the master service must run as a user with access to the mail system and the advanced user right **logon as a service**. If the user account does not have administrative capabilities, an Administrator must run the **dcomcnfg** utility on the TES master machine to add the user to the access and launch permissions for the **idsMail** server. If this is not done, TES and the Windows Event Viewer will log an error attributed to **idsMail32** and email will not function properly.

Making the Master Service Run as a User

To make the master service run as a user:

Step 1 Click the Windows **Start** menu button, then select **Programs>Tidal Software>Scheduler>Master>Service Control Manager.**

Step 2 Select the **Scheduler Master** service from the **Service** field drop-down menu.

Step 3 Click **Configure** to display the **Service Configuration** dialog.

Step 4 In the **Logon Information** section, select the **Other Account** option.

Step 5 Type the complete user account (include domain) and password information.

Step 6 Click **OK**, then exit the **Service Control Manager**.

Adding Access Permissions to an Account with dcomcnfg

Note This procedure must be run as Windows Administrator.

To add access permissions to an account:

Step 1 Click the Windows **Start** button, then select **Run** and type `dcomcnfg` in the **Open** field.

Step 2 Click the **Applications** tab in the **DCOM** dialog.

Step 3 Select the idsMail server from the **Applications** list (listed as **This server object provides a universal Email send/receive...**) and click **Properties**.

Step 4 When the **idsMail** dialog displays, click the **Security** tab.

Step 5 In the first section on the tab, select the **Use custom access permissions** option.

Step 6 Click the **Edit** button for access permissions to display the **Registry Value Permissions** dialog.

Step 7 Click **Add** to display the **Add Users and Groups** dialog.

Step 8 Select and add the TES master user account from the **Add Users and Groups** dialog.

Step 9 Click **OK** and click **OK** again in the **Registry Value Permissions** dialog to return to the **Security** tab.

Step 10 In the second section on the tab, select the **Use custom launch permissions** option.

Step 11 Click the **Edit** button to display the **Registry Value Permissions** dialog.

Step 12 Click **Add** and select the TES master user account from the **Add Users and Groups** dialog.

Step 13 Click **OK** and click **OK** again to close the **Registry Value Permissions** dialog.

Step 14 Click **OK** to exit the **idsmail** dialog.

Step 15 Click **OK** again at the **DCOM** dialog to save your changes and exit.

Step 16 Reboot the master machine.

Troubleshooting the Client Manager

Client Manager Cache Database

As of TES 6.0.3, the option of deploying a stand-alone TES 6 cache database as oppose to using the default embedded Derby cache database is supported. A stand-alone cache database allows for a faster synchronization time upon Client Manager startup. Additionally, a stand-alone cache database improves the overall UI experience by offering faster filtering and scrolling response times.

Suported Databases

Supported TES stand-alone databases are MSSQL 2005 & 2008, Oracle 10g and 11i. These are the same database versions currently being supported by the TES Master.

- Client Manager Operating System
  - Windows
    - MSSQL 2005
    - MSSQL 2008
    - ORACLE 10g
    - ORACLE 11i
  - Linux
    - ORACLE 10g
    - ORACLE 11i

Database Sizing

The Client Manager cache database maintains the same data in a slightly different format compared to the Master database. The sizing of the cache database should be identical to that of the Master database.

Database Location

For ideal performance, it is recommended that the stand-alone TES cache database be located exclusively on its own server and have optimal network connection between the Client Manager server and the TES cache database server.

How to clear the cache when needed

Clearing the cache involves truncating all tables in the schema except the table CACHEPROPERTY. CACHEPROPERTY holds the value of Cache.Version. Optionally, dropping all tables in the schema can also clear the cache. A SQL script will be provided to the customer for clearing cache via table drops.

To clear the cache:

Step 1 Locate the *clearcache.sql* script.

Step 2 Execute script as the user for the cache database.

Verifying the Version of the Client Manager

When consulting with Technical Services about a problem with the Client Manager, one of the most basic pieces of information they need is which version of the Client Manager is being used.

To verify the version of the Client Manager:

In the Client Manager **log** folder, open any log file. Verify at the top of the file the four digit version number. For example, 6.0.0.1192.

You can also check the *clientmgr.out* file located in the same folder.

Diagnostic Logging for the Client Manager

clientmgr.props file contains several logging categories. The following two are the main logging categories that apply to the Client Manger:

- **DspLog** – For items related to the DSP module.
- **CommonsLog** – For items related to core component logging, such as the Jetty embedded Web server.

The following is an example *clientmgr.props* file:

```
PrimaryServer=tcp://server:6215
Security.Authentication=ActiveDirectory
ActiveDirectory.Host=adhost
ActiveDirectory.Port=389
ActiveDirectory.UserSearchPrefix=DC=testdomain,DC=local
ActiveDirectory.GroupSearchPrefix=DC=testdomain,DC=local
JAVA_HOME=C:\Program Files\Java\jre6
JVMARGS=-Xms8192m -Xmx8192m -XX:PermSize=128m -XX:MaxPermSize=128m
MaxLogFiles=1000
JmxRmiPort=1100
```

```
JmxOn=Y
```

<DSP>.dsp contains the following properties for the DSP:

- **CacheLog** – Applies to writing to the cache during synchronization.
- **DspLog** – For items related to the DSP module.
- **RequestLog** – Relates to all requests of the DSP.
- **RPCLog** – Communication between the Client Manager and Master.
- **ClientLog** – Client related logging.
- **CommonsLog** – Relates to core component logging.

The following is an example *<DSP>.dsp* file:

```
BackupServer=<BackupServer>:6215
ClientType=tes-6.0.0.0
PrimaryServer=<PrimaryServer>:6215
DspLog=FINEST
CacheLog=FINER
RequestLog=FINE
RpcLog=WARNING
ClientLog=SEVERE
```

Monitoring the Client Manager

To turn JMX monitoring on:

Step 1 In *clientmgr.props*, set the JMX setting to yes, **JmxOn=Y**.

Step 2 Stop the Client Manager.

Step 3 Restart the Client Manager.

Configuring the DSP

The DSP config file, *<DSP>.dsp,* needs to know where the primary master and backup master is located.

To point the ClientManager/DSP to a different primary and/or backup master:

Modify the PrimaryServer and BackupServer settings in the *<DSP>.dsp* file. For example:

```
PrimaryServer=tcp://<hostname>:<port>
BackupServer=tcp://<hostname>:<port>
```

Note These settings are automatically set by the installer when you install the ClientManager. Modify these only if you wish to change the primary and backup master locations after install or if you did not specify them during the install.

Synchronization

Synchronizing may take a while. If a record is not synched for any reason, set the CacheLog level to **FINEST**. This will log every record synchronized from the master.

To check the status of the synchronization, see the *clientmgr.out* file in the **logs** folder.

Web Client is Slow

If the Web client is slow, change the logging levels of the following log settings in the DSP configuration file, *<DSP>.dsp*. This allows for more verbose logging details.

- DspLog
- ClientLog
- RequestLog
- CommonsLog

Working with JConsole

If adjusting the logging levels fails to fix a problem, you can point JConsole to the Client Manager JVM to monitor the internals.

To configure:

Step 1 From the **JConsole:Connect to Agent** dialog, select the **Advanced** tab.

Step 2 Input the JMX URL to the destination.

For example:

```
[hostname]:[port]
```

-or-

```
service:jmx:rmi://[hostname]/jndi/rmi://[hostname]/ClientManager
```

Step 3 Enter the username and password for the connection.

Step 4 Click **Connect**.

Monitoring Memory Usage and Thread State of CM and DSP

You can use this to diagnose situations where the CM appears to be stuck or non-responsive. Refer to the JMX Guide for further information.

Troubleshooting the Agent

Verifying the Version of the Agent

When consulting with Technical Services about a problem with an agent, one of the most basic pieces of information they need is which version of the agent is being used.

To verify the version of the agent:

Step 1 In the **Navigator** pane of the Tidal Web client, select **Administration>Connections** to display the **Connections** pane.

Step 2 In the various connections listed in the pane, locate the agent with the problem.

Step 3 Look in the **Version** column of that agent to see the version of the agent being used.

Proper Configuration of the Windows Agent

The Agent for Windows should have been configured properly at the time of installation but sometimes the configuration is unintentionally changed and causes problems.

The Agent for Windows will not operate correctly unless it is configured as follows:

- The agent service must be configured to run as a Windows user, either as local to the agent server or a domain user.
- This Windows user must be an administrator on the local machine and have the following four user rights assigned
  - Act as part of the Operating System
  - Logon as a batch
  - Logon as a service
  - Replace a process level token
- All runtime users must have the **Logon as a batch user** right on every Windows server that the runtime user will run jobs on.

Note If you have to change any of the above, it is best to restart the server, as Windows does not always make those changes effective without rebooting.

- In the Client Manager, edit each runtime user (and regular user if those are being used to run jobs) and type the Windows password on the **Passwords** tab of the user's definition. If a Windows domain is not specified, a runtime user is assumed to be local to the agent where it is running a job.
- Also in the Client Manager, from the main menu, select **Activities>System Configuration** to display the **System Configuration** dialog and select the **Use Passwords to Run Windows Jobs** option.

After confirming that the agent service has been configured as recommended, run this test job from a command prompt:

C:\WINNT\system32\cmd.exe
/c set

Look at the output for the **Username** variable to see if the runtime user's name listed. If that is working, then you should be able to run jobs wherever the Windows Authentication is accepted.

Unix Agents

The first line of every Unix agent shell script must adhere to standard Unix scripting guidelines and refer to a shell; for example, **#!/bin/sh**. For more information, refer to your Unix system documentation or consult your System Administrator.

Diagnostics for the Agent for Unix

When the Agent for Unix generates errors and doesn't operate properly, you need to contact Technical Services to help you resolve the technical issues. However, Technical Services requires specific information on how the Agent for Unix is operating before they can track down the source of the problem. Before contacting Technical Services about an agent issue, you should turn on diagnostic logging to collect information about the way the agent is functioning. This is the first step that Technical Services will have you do, if you have not done it before contacting them.

To turn on diagnostic logging:

Step 1 On the agent machine type the following command to stop the agent:

./tagent <agent name> stop

Step 2 Go to the **/bin** directory and locate the *tagent.ini* file for the desired agent.

Step 3 Inside the *tagent.ini* file, under the port setting, type the following:

ovb=Tidaldebug

Step 4 Save the file and its changes.

Step 5 Start the agent:

./tagent <agent name> start

Ideally, you want to reproduce the situation that caused the issue so the diagnostics can log what occurred in the system at that time. As soon as the problem reoccurs, contact Technical Services.

Step 6 Once the problem repeats itself and the diagnostic information is recorded, turn off the agent diagnostics by commenting out the debugging parameter:

#ovb=Tidaldebug

Step 7 Go to the **Log** directory to get the diagnostic file to send to Technical Services:

cd <agent directory>/<agent name>/logs

Each agent instance has its own directory. The diagnostic files are named *<agent name>.log* and *<master server>.log*.

Restarting the agent does not override the recorded information. Though only a small amount of information is normally recorded without the debug parameter, the file will continue to grow in size. You should delete or rename the file after you finish debugging the agent.

Note Whenever diagnostic logging is being used, you must carefully monitor the amount of disk and database space being consumed. Diagnostic logging can generate large amounts of data and affect system performance.

Diagnostics for the Windows Agent

To run diagnostics for the Windows agent:

Step 1 Login to the agent console as the same user that the agent service is configured to run as.

Step 2 Create a shortcut by right-clicking the Windows Desktop and selecting **New>Shortcut** from the context menu to display the **Create Shortcut** dialog.

Step 3 In the **Type the location of the item** field, type the following command path:

```
"C:\Program Files\Tidal\tsagent\bin\Agent.exe" Agent=TidalSAAgent_0 Port=5912
Verbosity=Tidaldebug Debug=Yes sysadmiral=yes
```

Note Your installation path and port number may be different.

Step 4 Using the Service Control Manager, stop the agent.

Step 5 Double-click the agent shortcut you created to start the agent in debug mode.

A window named after the agent displays scrolling messages as the agent does its work. The same data displayed in the window is also logged to the Application Log of the Windows Event Viewer. The system must remain logged on and the agent must continue to run in this Application Mode to gather diagnostics.

It is important to modify the properties for the Application Log so it saves an appropriate amount of data.

To modify the properties:

Step 1 In the **Event View** tree, select the **Application Log**.

Step 2 Right-click and select the **Properties** option to display the **Application Log Properties** dialog.

Step 3 In the **Log size** section, adjust the value in the **Maximum log size** field.

Step 4 The value will vary significantly from agent to agent, depending on the environment and usage. Start with 5-10 megabytes of data and increase as needed. You can configure the Event Viewer to either overwrite the previously written data or stop logging when the log size reaches the maximum size.

To stop diagnostics, close the agent application window and restart the agent from the Service Control Manager.

Other

Unable to scroll using scroll buttons, Runtime User - FireFox 3.6.x

If you can use a slider to drag down the list, but not a down button of a scroll bar, this is happening because of the following bug in Firefox.

https://bugzilla.mozilla.org/show_bug.cgi?id=511075

This Firefox bug is already fixed by Firefox and is part of Firefox 3.6.4 Beta release.

http://www.mozilla.com/en-US/firefox/all-beta.html.