

Rational Performance Tester

Load Generation Agent **Architecture and Support**

RPT Version 8.3.0

Kevin Mooney (kmooney@us.ibm.com)

Richard Ptasnik (ptasnik@us.ibm.com)

Table of Contents

1. Load Generation Agent Introduction	3
2. Agent Controller	4
3. Architecture	4
4. Installation	5
4.1 Workbench Local Execution	5
4.2 Installation Features	5
4.3 Load Generation Agent Configuration	6
5. Workbench Configuration	6
5.1 Server Port	6
5.2 Secure Communication using TLS/SSL	7
5.3 Delete Deployment Directory	7
6. Configuration Modification	7
6.1 Add Support for Another Workstation to an Agent	7
6.2 Load Generation Agent Configuration File	8
7. Agent Status	9
7.1 Share Agent with New Workbench	9
7.2 Disconnect Agent from this Workbench	9
8. Troubleshooting	9
8.1 Port In Use	9
8.2 Check Agents Failed	10
8.3 Process on driver terminated	11
9. Tips, Tricks and Miscellaneous	13
9.1 Privileges	13
9.2 Multiple Engines On One Agent Machine	13
9.3 Locations	13
9.4 Ports with multiple workbenches	13
9.5 Max heap	13
9.6 Majordomo	14
9.7 Majordomo debug flag	14

1. Load Generation Agent Introduction

A new Load Generation Agent is a vital part of Rational Performance Tester (RPT) Version 8.3.0. As a replacement for the Agent Controller the Load Generation Agent offers the following features:

- Very strong execution process launch reliability for agents. The Load Generation Agent is designed specifically to work with the Rational Performance Tester workbench in the launch and control of load tests.
- Agents poll supported workbenches looking for schedule execution work instead of the workbench having to establish connections with agents. This scheme provides instant assessment of agent readiness and rapid delivery of launch commands.
- With the Load Generation Agent the workbench is capable of launching agents in parallel thus greatly reducing overall launch time for large load testing operations involving many agents.
- The Load Generation Agent supports intelligent test asset deployment only pulling assets from the workbench that it does not have or that have changed. This feature results in faster launch times for subsequent executions.
- The Load Generation Agent uses an HTTP server and the HTTP protocol between the workbench and agents. Loss of a connection simply means another is established with no harm to the overall execution providing robustness during multi-day executions.
- The Load Generation Agent is more firewall friendly, reversing the contact initiation such that agents connect to the workbench instead of vice versa. The use of an HTTP server within the workbench means the customer has only one port to configure with the firewall instead of the many ports on both ends that the Agent Controller required.
- Secure communications are much easier with the Load Generation Agent. The preference to secure communication between workbench and all agents can be done from the workbench instead of having to configure each individual agent.

2. Agent Controller

The Agent Controller is still provided with version 8.3.0. As an optional part of the installation the Agent Controller can be installed in order to support these features:

- Response Time Breakdown (RTB)
- Service-oriented Architecture (SOA) Stubs

While not a typical customer configuration installing both the Load Generation Agent and the Agent Controller is a supported configuration.

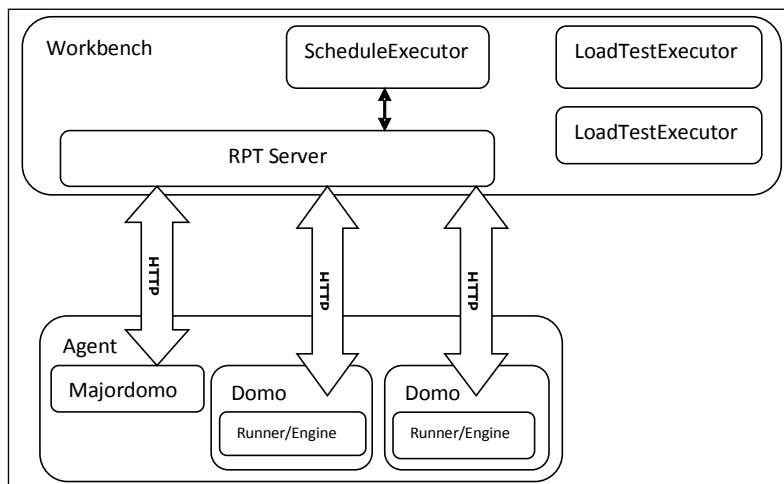
The Agent Controller cannot be used for load generation with the RPT 8.3.0 workbench.

3. Architecture

This section describes the architecture of the Load Generation Agent. Unlike the Agent Controller architecture the workbench does not make connections to the agents when a schedule is to be executed. The ports that had to be opened by firewalls for those connections and the connections the agents made back to the workbench on ephemeral ports are no longer necessary.

With the Load Generation Agent the agent machines poll for work sending HTTP requests to the Rational Performance Tester Server running with the workbench. When a customer presses the Run button to start schedule execution the next time a participating agent contacts the workbench it responds to the agent indicating its participation is needed.

The agent then launches the execution engine as the Agent Controller would have. The execution engine then contacts the RPT Server and the usual exchange of commands to begin schedule execution commences.



In the figure above the Workbench box represents the Eclipse user interface of Rational Performance Tester which is a single process and one Java Virtual Machine (JVM). Rational Performance Tester 8.3.0 also includes the jetty web server to support communication between the workbench and agents. Within the workbench each agent has a LoadTestExecutor for command, control and state management of each agent.

On a single agent there exists the Majordomo service which is one Java Virtual Machine that polls supported workbenches for schedule execution work. Each Domo box in the figure is a single Java Virtual Machine and includes the execution engine. Typically there will be only one Java process running on the agent during schedule execution. That single Java process is the Domo/Engine process. The figure above shows that it is possible for one agent machine to support multiple execution engines. Customers that have a powerful 64-bit agent with a lot of memory may find it overall more efficient to run with multiple engine processes of modest heap size than one single process with enormous heap size due to garbage collection considerations.

The Load Generation Agent is designed to be simple and compact. The Domo component downloads from the workbench the assets needed for schedule execution, including the execution engine. On subsequent executions Domo only downloads assets that have changed since the previous execution.

4. Installation

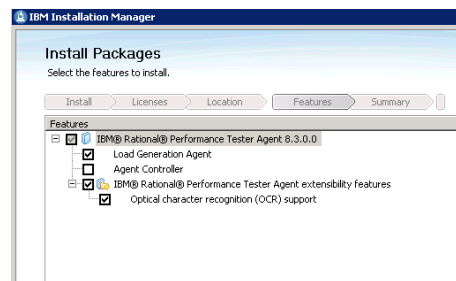
The installation of the Load Generation Agent is similar to the installation of the Agent Controller. In fact, both features are available when you install the Rational Performance Tester Agent product for version 8.3.0.

4.1 Workbench Local Execution

While installing the Rational Performance Tester Agent product with the workbench product is a supported configuration it is not required. As with previous releases it is possible to execute schedules locally on the same machine that the workbench is running on without having to install the Rational Performance Tester Agent product. The workbench will automatically spawn an instantiation of the Load Generation Agent to support local execution during the life of the execution of the schedule.

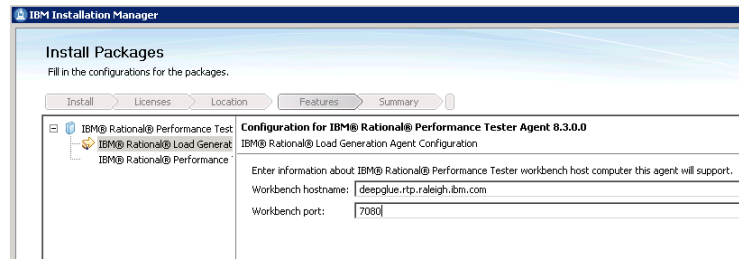
4.2 Installation Features

By default a Rational Performance Tester Agent installation will install the Load Generation Agent and not install the Agent Controller.



4.3 Load Generation Agent Configuration

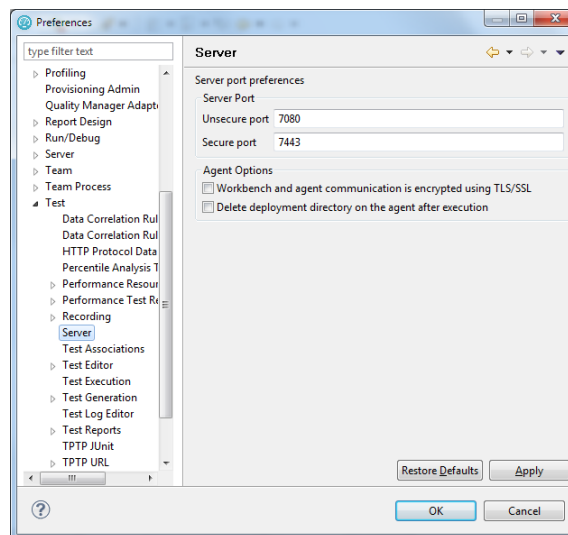
The Load Generation Agent has two configuration parameters that need to be entered during installation.



- Workbench hostname. Enter one and only one hostname that this agent will poll for work.
- Workbench port. Enter the port number supported by the workbench. The default port number is 7080.

5. Workbench Configuration

The Load Generation Agent has several options that can be configured from the workbench. These options do not require changes on the agent. Configure Load Generation Agent options via Window > Preferences > Test > Server.



5.1 Server Port

Select the unsecure and secure ports the server will use for communication with agents. By default communication is unsecure and so the unsecure port is the only one that may need to be changed. The port does not have to be changed unless it is already in use by another product running on the workbench. If you change the workbench port the agent configuration must also be changed to use the same port.

The secure port is only used when running with the option to encrypt messages between the workbench and an agent. The configuration of the secure port number is only done on the workbench. There is no configuration change required on the agent related to the secure port.

5.2 Secure Communication using TLS/SSL

Check **Workbench and agent communication is encrypted using TLS/SSL** to have all messages and execution data exchanged between the workbench and agents encrypted. All schedules executed from this workbench will use SSL. The workbench will tell the agent what port to use for secure communication via the unsecure port prior to starting schedule execution.

5.3 Delete Deployment Directory

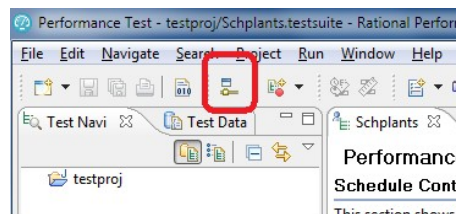
Check **Delete deployment directory on the agent after execution** to have the contents of the deployment directory removed from the agent after schedule execution completes. The deployment directory contains all the test assets the agent requires in order to participate in schedule execution. Selecting this option defeats the cached test assets launch time improvement feature making subsequent schedule launch times no shorter than the first schedule launch. The feature would be of interest to customers concerned about disk space accumulating on the agent from many different schedule executions.

6. Configuration Modification

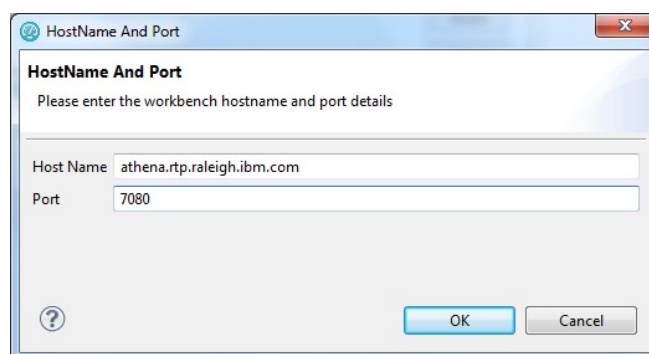
Once a workbench and many agents are installed it may become necessary to make configuration changes. This section describes how to modify an agent configuration after installation.

6.1 Add Support for Another Workstation to an Agent

If you recall from the Installation section it was only possible to specify the hostname of one agent when configuring the Load Generation Agent. So what happens if you add another workstation to the lab and want an agent to support it? The recommended approach is to use the Agent Status button on an already supported workbench to add the new workbench to the agent's configuration.



When you press the Agent Status button a window appears showing the status of agents in contact with this workbench. To have an agent support another workbench in the lab check the box beside the agent and press the button labeled **Share Agent with New Workbench**.



You will then be prompted to enter the hostname and port number for the workbench to add to the agent's configuration. After pressing the OK button the request to add the new workbench will be delivered to the agent. Within a short time the agent should begin polling the new workbench for schedule execution work.

6.2 Load Generation Agent Configuration File

After installation of the Rational Performance Tester Agent with the Load Generation Agent feature there will be an agent configuration file stored on the agent. By editing, copying or replacing the configuration file you can do the following:

- Change the hostname of a workbench the agent polls for work
- Change the port the agent will use when contacting the workbench
- Add one or more additional workbenches for the agent to support
- Replace the entire configuration en masse by copying over the existing configuration file
- Push a new configuration out to many agents using remote copy commands

The Load Generation Agent will refresh its configuration based on the contents of the file in about ten seconds after the change is made. There is no need to stop and restart the agent service.

The contents of the configuration file are XML and the format is intuitive. Below is an example configuration file for an agent that is supporting two workbenches.

```
<MajordomoConfig xmlns="http://www.example.org/MajordomoConfiguration">
  <debug>false</debug>
  <workbenches>
    <hostname>deepglue.rtp.raleigh.ibm.com</hostname>
    <port>7080</port>
  </workbenches>
  <workbenches>
    <hostname>athena.rtp.raleigh.ibm.com</hostname>
    <port>7080</port>
  </workbenches>
</MajordomoConfig>
```

The Load Generation Agent configuration file is stored typically in one of these locations:

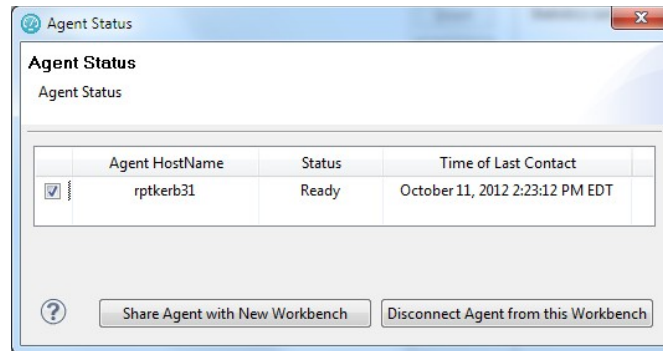
C:\Program Files\IBM\SDP\Majordomo\majordomo.config (Windows 64-bit)

C:\Program Files (x86)\IBM\SDP\Majordomo\majordomo.config (Windows 32-bit)

/var/opt/IBM/SDP/Majordomo/majordomo.config (Unix)

7. Agent Status

As mentioned above, the workbench contains an Agent Status button. When selected it is possible to see what agents are polling the workbench for work and if they are ready to participate in schedule execution.



The states an agent may display are:

- Ready. The agent is ready to execute a schedule.
- Busy. The agent is ready to execute a schedule. The agent is currently executing a schedule. It is possible for an agent to be executing more than one schedule at a time.
- Lost Contact. The agent is no longer polling this workbench.

7.1 Share Agent with New Workbench

If you select one or more agents and press **Share Agent with New Workbench** you can specify the hostname and port of a new workbench that you want the agent to poll for work in addition to the this workbench.

7.2 Disconnect Agent from this Workbench

If you select one or more agents and press **Disconnect Agent from this Workbench** the hostname for this workbench will be removed from the agents' configuration files and they will no longer poll this workbench for work.

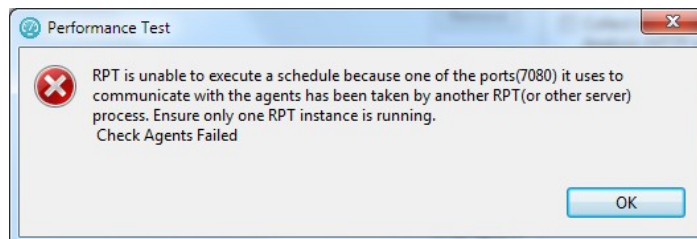
8. Troubleshooting

This section describes some common problem scenarios and provides suggestions for finding a solution.

8.1 Port In Use

Symptom

When you try to run a schedule the following error message dialog appears:



Meaning

Some product running on the workbench is already using port 7080, the default unsecure port RPT uses for communication with agents.

Solution #1

Using Window > Preferences > Test > Server choose a different port such as 7081. All agent's configuration files must also specify port 7081. There is no need to restart the workbench for this change to take effect.

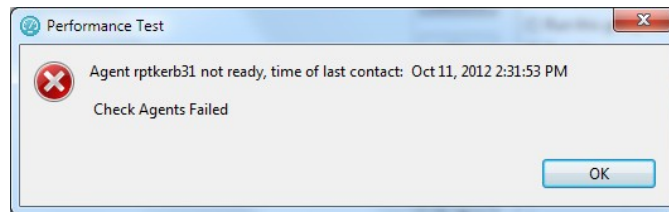
Solution #2

It is possible the product using port 7080 is no longer needed on this machine. If so, stop or uninstall the product using port 7080.

8.2 Check Agents Failed

Symptom

When you try to run a schedule the following error message dialog appears:



Meaning

One or more of the agents listed to participate in executing this schedule are not in active contact with this workbench. The Agent Status button can be used to view agent status prior to schedule execution to determine the likelihood of encountering the Check Agents Failed message.

Solution #1

The RPT Agent is not installed. On an agent you can use the IBM Installation Manager View Installed Packages feature to ensure that the Rational Performance Tester Agent is installed.

Solution #2

The Load Generation Agent is installed but the agent service, Majordomo, is not running.

<i>Windows</i>	Look for the MajordomoService process. If not running: cd "Program Files/SDP/IBM/Majordomo" NGAStart
<i>Unix</i>	Look for the Majordomo process. If not running: cd /opt/IBM/SDP/Majordomo ./MDStart.sh

Enter the commands above using command prompt or a shell. Substitute appropriate paths based on installation. Use Task Manager (Windows) or the ps command (Unix) to verify the agent is running. Look for Majordomo on Unix systems. On Windows systems look for MajordomoService.exe.

Solution #3

Majordomo is running on the agent but it is not polling the workbench for work. Verify that the agent can DNS resolve the hostname as specified in the configuration file. Use the **ping** command specifying the workbench hostname exactly as entered in the majordomo.config file. If the ping command fails using the hostname but you can successfully ping the workbench using its IP address a workaround could be to specify the workbench IP

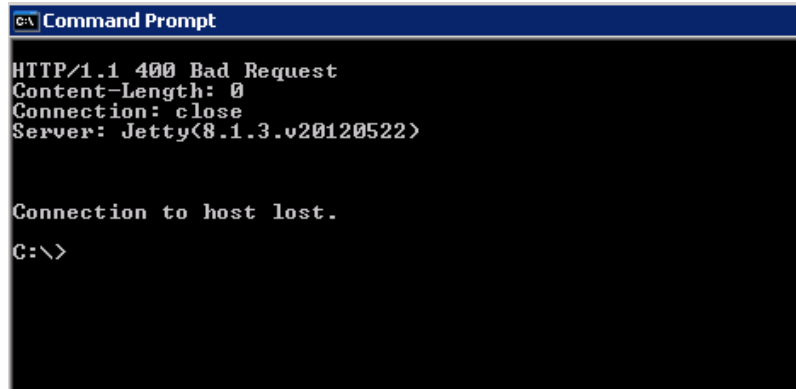
address in majordomo.config instead of the host name.

Solution #4

If Majordomo is running and the workbench can be pinged from the agent then verify there is not a firewall blocking access. From a command prompt or shell enter the following:

telnet WORKBENCH_HOST_NAME 7080

Type a few characters and press Enter. You should see the following:



```
C:\>telnet WORKBENCH_HOST_NAME 7080
HTTP/1.1 400 Bad Request
Content-Length: 0
Connection: close
Server: Jetty(8.1.3.v20120522)

Connection to host lost.
C:\>
```

If unable to connect or you do not receive a response like the one above perhaps try temporarily disabling firewalls to confirm that is the source of the problem.

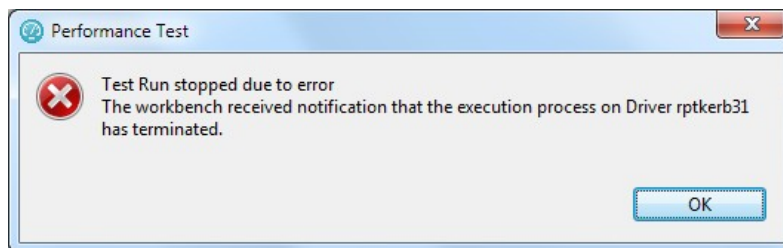
Solution #5

It is also possible to receive the Check Agents Failed if the agent configuration contains workbenches that are not running, had RPT uninstalled or are shutdown. An agent encountering communication problems may get delayed in its polling work long enough that a supported workbench may conclude that agent is out of contact. If this case is suspected remove any stale workbench information from each agent's configuration file.

8.3 Process on driver terminated

Symptom

When you try to run a schedule the following error dialog appears:

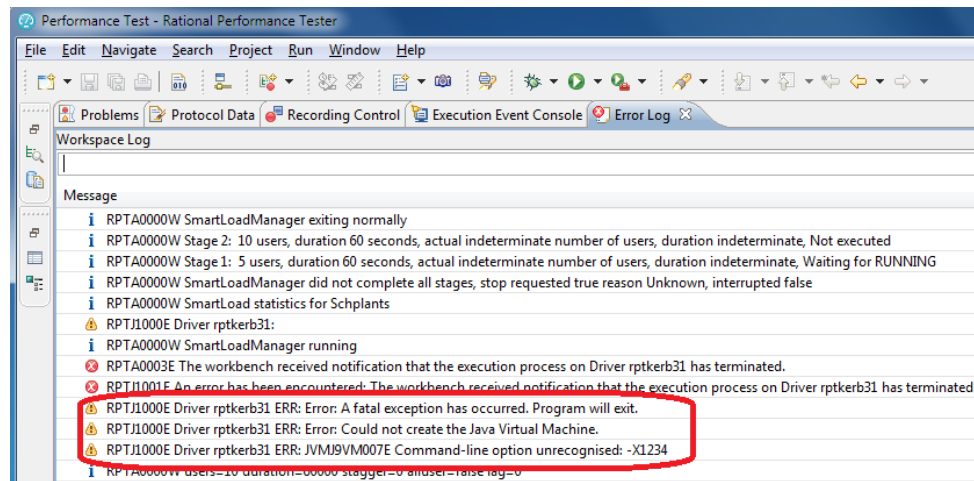


Meaning

Shortly after Majordomo launched the engine execution process on the agent that process terminated unexpectedly. There may be messages explaining the reason for the failure in the workbench Error Log. This log can be opened using preferences via Window > Show View > Error Log. At the default log level of WARNING all messages written to stderr by the native java program on the agent will be sent to the workbench and stored in the Error Log.

Solution #1

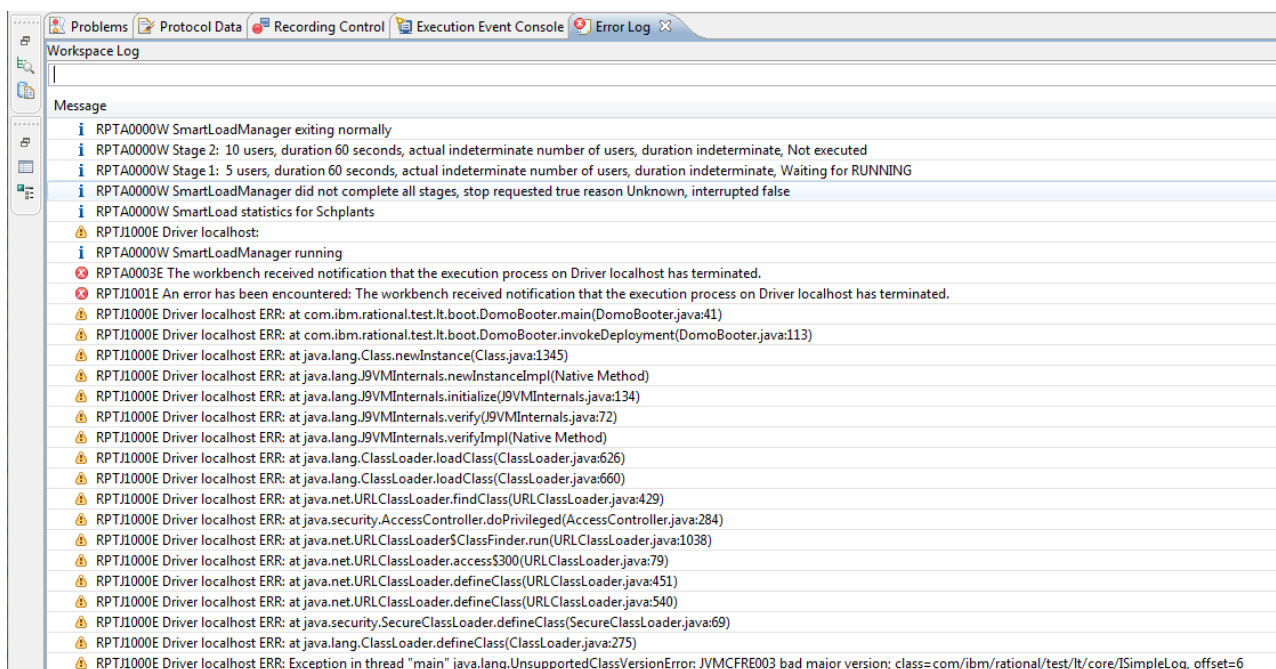
The execution process will terminate at startup if bad arguments are passed to the JVM. Here is an example of the contents of the Error Log if a bad argument is passed to the JVM:



Note the JVM complaining about an unrecognized command line option. In this case examine any arguments specified in the RPT_VMARGS general property for the location and ensure there are no typos or bogus command line options.

Solution #2

Another example of a problem that produces the process terminated message is if somehow the wrong JVM was used when launching the engine execution process. Here is an example of what happens if the agent engine execution is launched with Java 1.6 instead of the desired Java 1.7 JVM:



The appearance of **UnsupportedClassVersionError** is a clue that the JVM on the agent is not the required IBM Java 1.7 JVM. This problem could occur if the RPT Agent was installed in a non-default manner or if the installation path contained some unexpected characters. Here are some suggested workarounds:

1. Uninstall and re-install the product. Take default values for where to install the agent.
2. Use the environment variable RPT_JAVA to explicitly specify the Java executable the agent should launch for the execution engine. For example:

RPT_JAVA=C:\Program Files\IBM\SDP\jdk\bin\java.exe

The java.exe to use should be the IBM Java 1.7 java binary installed with the RPT Agent. After you set this environment variable stop and restart the Majordomo service. From a command prompt or shell enter the commands below:

<i>Windows</i>	cd "Program Files\IBM\SDP\Majordomo" NGAStop NGAStart
<i>Unix:</i>	cd /opt/IBM/SDP/Majordomo ./MDStop.sh ./MDStart.sh

9. Tips, Tricks and Miscellaneous

9.1 Privileges

The Rational Performance Tester Agent product must be installed by a user with Administrator privileges on Windows and as root on Unix.

9.2 Multiple Engines On One Agent Machine

Previous versions of Rational Performance Tester allowed multiple execution engines on one agent machine. The only restriction is that the location hostnames and deployment directories must be unique. This trick also works with the new Load Generation Agent. Rational Performance Tester does not limit the number of engines executing on one agent.

9.3 Locations

There are no changes related to locations and the Load Generation Agent. If running using a remote agent a location must be created and added to the schedule as in previous releases of Rational Performance Tester.

9.4 Ports with multiple workbenches

Note that if one agent is supporting multiple workbenches it is okay for the port number to be the same for every workbench listed in the configuration file.

9.5 Max heap

Previous releases of Rational Performance Tester required one successful execution of a schedule in order to set the RPT_DEFAULT_MEMORY_SIZE property that specifies the maximum Java heap for a location. This requirement

holds for the new Load Generation Agent. On the very first execution the max heap will be **256m**, a heap amount that will ensure disaster for a large load test. A customer trying to do a large load test straight out of the box will have agents that experience OutOfMemoryError and the processes on those agent may terminate early.

9.6 Majordomo

The heart of the Load Generation Agent is called Majordomo. On Windows it runs as a service called MajordomoService.exe. On Unix it runs as Majordomo. The Unix systems require customer intervention if they want Majordomo running after a reboot. Because it is a service, MajordomoService.exe should be running automatically after reboot on Windows.

On Windows systems MajordomoService will log informational messages at startup in the system temp directory which is usually C:\Windows\Temp\majordomoservice.log. If there is a problem, perhaps related to finding the IBM JVM, there may be messages in that log that could help determine the cause of the problem.

9.7 Majordomo debug flag

You may have noticed there is a debug flag in the majordomo.config file. If set to true, the Majordomo process and any engine processes launched create log files that can be useful for debugging Load Generation Agent launch and problems that occur early in schedule execution. Examples of these files are:

C:\Windows\Temp\SERVICE\majordomo.log

C:\Windows\Temp\SERVICE\engine_HOSTNAME.log

Similar files can be found on Unix systems stored under %TMP%/USERNAME.