Next Page →

Welcome to the JReport Server Monitor User's Guide

This User's Guide describes JReport Server Monitor. JReport Server Monitor, which is a standalone web-based application used to monitor the overall performance of JReport Server, allows you to inspect the status of JReport Server, including the status of the servers in the cluster, the status of different reports, the status of online users and so on. JReport Server Monitor is able to generate and display JReport Server performance statistics via line charts and text. You can also use JReport Server Monitor to maintain different functions of JReport Server, such as shutting down servers, stopping, pausing, running tasks, logging out a valid UserSession, and so on. By creating profiling reports using JReport Server Monitor, you can inspect server performance during different periods of time.

Other JReport documentation

This guide is one in the complete JReport documentation set. The documentation set also includes the following:

- Getting Started with JReport
- JReport Tutorial
- JReport Server User's Guide

- JReport Designer User's Guide
- JReport API Javadoc

Next Page →

Introduction to JReport Server Monitor

JReport Server Monitor is a standalone web-based application used for monitoring the overall performance of JReport Server. This chapter provides an overall introduction to JReport Server Monitor, including the characteristics and functions of the product.

The following are the main features of JReport Server Monitor:

- **Inspects the status of JReport Server**

  JReport Server Monitor allows you to view the status of servers in a cluster. It also allows you to view the status of different reports and the status of online users and so on.

- **Shows server performance statistics in Graph/Text mode**

  JReport Server Monitor is able to generate and display a performance chart of JReport Server according to its statistics. In this way, you can view the performance of JReport Server in the form of Line chart Graph and Text.

- **Maintains JReport Server**

  You can perform maintenance tasks from JReport Server Monitor, such as shutting down servers, stopping waiting/running tasks, and logging out a valid UserSession and so on.

- **Creates profiling reports: performance reports and statistic reports**

  By creating profiling reports using JReport Server Monitor, you can inspect server

performance in a certain period of time.

- **Provides JMX remote/local monitoring feature**

  JReport has constructed JMX MBeans which correspond to all the monitored objects (including Cluster, JReport Server, Report Task, UserSession and the Database Connection Pool). You can use MBeanServer to find these registered MBeans and perform any operation on them. These JMX MBeans provide the same monitoring functions as with our normal Server Monitor's jsp pages.

← Previous Page Next Page →

Installing and Uninstalling JReport Server Monitor

JReport Server Monitor is an application for the administrator which does not need to go on the same JReport Server as the report system. So it is recommended that Monitor be installed on a separate system or on systems for system administrators.

To install JReport Server Monitor on Windows:

1. Download the JReport Server Monitor installation file from the Jinfonet download center.
2. Execute the installation file.
3. Once the Installation Wizard has been successfully loaded, the Welcome screen appears. Click **Next** to continue.
4. In the License Agreement screen, read the license agreement carefully. If you agree with all the terms, select **I accept the terms of the License Agreement**, and then click **Next**. The installation process will not continue unless you accept the terms stated.
5. In the Choose Installation Folder screen, provide the absolute path for where you want to install JReport Server Monitor. Click **Browse** to navigate to the destination folder. Click **Next** to continue.

6. In the Configuration screen, specify the administration port and the active host address. Click **Next** to continue.

7. In the Select JDK screen, select the JDK version with which JReport Server Monitor will be installed. The selected JDK will be used to execute JReport Server Monitor. The Installation Wizard collects and lists all the available JDK versions installed on your machine, you can then select one of the JDK versions from the list. Click **Next** to continue.

   Note: The Installation Wizard doesn't list JRE versions found on your machine. The lowest JDK version JReport supports is 1.8.0. JDK 9 is currently not supported.

8. In the Add Class Path screen, a class path is necessary when using a JDBC driver or user-defined object. A class path is composed of a file path plus a zip file, jar file, or directory path. For example, `C:\jdk1.8.0_51\lib\tools.jar`.

   Click the **Add** button to add the selected class path to the class path list. Click **Delete** to delete the selected class path from the class path list. Then click **Next** to continue.

   Or, you can skip this screen by directly clicking Install. Later you can manually edit the batch file or the command line starting JReport Server Monitor to add class paths.

9. In the Install Summary screen, the product name, location, and disk space information are shown. Click **Install**, and the Installation Wizard will start to install JReport Server Monitor.

10. In the Installing screen, the installing process and status are shown.

11. After installation, the Read Me screen displays. Read the information and click **Done** to close the Installation Wizard.

To remove JReport Server Monitor on Windows:

- Open **Control Panel** > **Programs and Features** to remove it.
- Run uninstaller.exe (uninstaller on Unix) in `<install_root>\_uninst`.

Note: The uninstaller will remove all the files generated by the installer, while the files that are created later by the program will be retained. You can remove them manually.

← Previous Page Next Page →

Launching and Accessing JReport Server Monitor

This chapter explains how to launch JReport Server Monitor and access it from a web browser.

Starting JReport Server Monitor

After you have installed JReport Server Monitor, a batch file named MonitorServer.bat is generated automatically in `<install_root>\bin`. You can start JReport Server Monitor by running this batch file.

To successfully start JReport Server Monitor, these requirements should be met:

- Copy rmi.auth from `<adminserver_install_root>\bin` of the admin server to `<monitor_install_root>\bin`, or remove rmi.auth from `<adminserver_install_root>\bin`, or add `-Djrs.rmi.auth_file=%authFileName%` to MonitorServer.bat to specify the authentication file.

- Make sure that the server.rmimonitor.enable property in the server.properties file in the `<adminserver_install_root>\bin` directory is set to true. The default value is true.

- If the admin server and Monitor are installed in different computers, you need specify the right host and port of the admin server in Monitor side via the two properties in the server.properties file in the `<monitor_install_root>\bin` directory before starting JReport Server Monitor:

  admin.server.host=The RMI host name or IP address of the admin server.

  admin.server.port=The RMI port number of the admin server.

  The default values of these two properties are for the admin server which is installed in the same computer.

  Note: The server.properties file is generated when you run MonitorServer.bat for the first time after installation. The step does not require that the Monitor is started and connected to the admin server successfully.

- The admin server is started successfully. It does not matter whether you start the admin server first or the Monitor first.

Accessing JReport Server Monitor services

After JReport Server Monitor is started successfully, you can take either of the following ways

to access JReport Server Monitor via browsers:

By URL

Use the service port number specified in the server.properties file to access the services of JReport Server Monitor. The default format of the accessing URL is: http://MonitorHost:MonitorServicerPort/monitor/index.jsp. For example, if JReport Server Monitor and JReport Server are installed on one computer, you can use `http://localhost:8848.`

From JReport Server UI

You can also access JReport Server Monitor by clicking the Monitor link on the JReport Server console > Administration > Other drop-down menu. To do this, you need make sure:

- The web.monitor.link.enable property in the server.properties file in `<server_install_root>\bin` is set to true. The default value is true.
- You are an administrator or have the privilege to access the Administration page of the server console.

Properties in the server.properties file

The following details the properties in the server.properties file which is located in the `<monitor_install_root>\bin` directory:

Property	Default Value	Description
admin.server.host	localhost	The RMI host name or IP address of the admin server. JReport Server Monitor will check this property value to find and connect to the admin server. If the admin server resides in a different computer from the monitor, you need modify the value manually.
admin.server.port	1129	The RMI port number of the admin server. JReport Server Monitor will check this property value to connect to the admin server.
log.config.update	false	Specifies to enable auto-update of logging configuration changes (by manually modifying the configuration file) at runtime. If set to true, any changes to the logging configuration file at runtime will automatically take effect after the specified update interval. Otherwise, any changes to the configuration file will not take effect until you restart the Server.
log.config.update.interval	60000	Specifies the interval value (in milliseconds) for when logging configuration changes will be auto-updated. This property will only function after the log.config.update property has been set to true.

monitor.homepage	-	Specifies the web entry of JReport Server Monitor. The valid format is `http://<host>:<port>/<context_path>/index.jsp`. In a standalone environment, if there is no value set to this property, JReport Server Monitor uses `http://localhost:8848/monitor/index.jsp` by default, and will prompt warning messages to the Console and logging destination. In the integrated environment, if this property has not been set to an explicit value, JReport Server Monitor will not be able to construct the default value.
monitor.jmx.htmladaptor.port	8849	The port that the JMX HTMLAdaptor listens to. This property depends on the monitor.jmx.htmladaptor.startup property.
monitor.jmx.htmladaptor.startup	true	Specifies whether to start up JMX HTMLAdaptor as the default MBean viewer internally. It depends on the monitor.jmx.startup property.
monitor.jmx.startup	false	Specifies whether to start up the JMX monitoring function and construct the Monitoring MBean when launching JReport Server Monitor.

monitor.max.report.save.number	50	The number of finished reports to be shown on the Finished Reports list.
monitor.max.report.save.time	5	The time span that a finished set should be shown on the Finished Reports list.
monitor.refresh.interval	2000	The time interval for automatically updating the statistics of the current program status is automatically updated.
monitor.rmi.port	1248	The RMI port number of JReport Server Monitor.
monitor.service.port	8848	The service port number of JReport Server Monitor. You can use this port to access Server Monitor.

← Previous Page Next Page →

Using JReport Server Monitor

JReport Server Monitor is a standalone web-based application used for monitoring the overall performance of JReport Server. It should be used together with JReport Server.

The following are the main features of JReport Server Monitor:

- Inspects the status of JReport Server.
- Shows server performance statistics in Graph/Text mode.
- Maintains JReport Server.
- Creates profiling reports, performance reports and statistic reports.

The following topics show you how these features can be used:

- Configuring JReport Server Monitor
- Monitoring the Status of JReport Server
- Monitoring the Performance of JReport Server
- Maintaining JReport Server
- Creating Profiling Reports

Configuring JReport Server Monitor

The Status and Setting tabs are displayed as the default page after you logged onto JReport Server Monitor. By using the Setting tab, you are able to configure some of your preferences. When you are at another page and want to access the Setting tab, click the root node in the left panel of the JReport Monitor page.

The following are descriptions of the options on the Setting tab:

Auto-refresh every _ seconds

Specifies the time interval at which the status of JReport Server Monitor will get updated automatically.

Show reports finished in the past _ minutes

Specifies the time span when the reports finish running that will be shown on the Finished Report list.

Maximum number of reports

Specifies the maximum number of reports that can be shown on the Finished Report list.

Display the Last Login Time

Specifies whether to display the last login time of a user on the top banner of JReport Server Monitor. If checked, the login time will be recorded in the login.properties file in `<install_root>\bin` after JReport Server Monitor shuts down.

Submit

Applies your changes.

Reset

Discards your modifications and restores the tab to its default status.

← Previous Page Next Page →

Monitoring the Status of JReport Server

JReport Server Monitor allows you to inspect the status of JReport Server. You can list the servers and their status from a cluster. You can also list the reports by drilling down into the servers. For example, running reports, waiting reports, finished reports, and failed reports. Tracking further down, you can even view the status of these reports. Note that JReport Server Monitor is not able to monitor the information of a report that is running in Page Report Studio.

Showing the status of servers in a cluster

By accessing the home page of JReport Server Monitor, you can see the status of each server in a cluster, including the cluster member ID, host IP, port, and its status.

The Status table of the servers includes:

Column	Description
Cluster Member ID	The ID of the server as a cluster member.
Host	The host IP address of the server.
Port	The RMI port number of the server.
Status	The status of the server. Can be either Active or Inactive. • **Active** - The server has been started and is ready for service. • **Inactive** - The server is inactive and cannot be available for service.

Monitoring the status of reports

Expand any server node in the left panel of the JReport Server Monitor home page, and then click **Reports**, you can see the status of the page reports, web reports and dashboards on the server.

There are five types of report status: all reports, running reports, waiting reports, finished reports, and failed reports. You can select to view the status of different reports from the drop-down list.

Status of all reports

The status table of all reports includes:

Column	Description
Report	The full path name of the report.
User ID	The ID of the user who opened the report.
Submit Time	The time when the report was last opened.
Pages	The total number of pages the last opened report has. Not available for web reports and dashboards.
Number of Runs	The total number of report runs since it is first published to JReport Server.
Status	The status of the opened report.

In addition, when you click the full path name of a report, the Task Statistics dialog will pop up, showing you the detailed task statistics, which includes the following:

- **Task ID**

  The exact date and time when the report was opened.

- **Report**

  The path and name of the opened report.

- **Catalog**

  The catalog where the opened report lists.

- **Total Number of Times Run**

  The total number of times the opened report has run ever since a specific time.

- **Average Number of Times per Day**

  The average number of times the opened report has run per day.

- **Last Submit Time**

  The time when the report was last opened.

Status of the running reports

The status table of the running reports includes:

Column	Description
Action	Stops the report from running and makes it a failed report.
Task ID	The internal ID for this report.
Report	The full path name of the report.
User ID	The ID of the user who opened the report.
Task Status	For running reports, can be one of the following: • **Running** - The task is currently being processed. • **Initializing Engine** - The task is currently in the initializing engine stage. • **Loading Report** - The task is currently in the loading report stage.

- **Exporting** - The task is currently in the exporting stage.
- **Exiting Engine** - The task is currently in the exiting engine stage.

Task Type	The task type. Can be one of the following: Schedule, On-Demand, Page Report Studio, Web Report Studio, Dashboard, and Bursting.
Start Time	The time when the task was started.
Elapsed Time	The elapsed time since the task was started.
Submit Time	The time when the report was opened.
Run Host	The name of the host on which the task is performed.
Run Port	The port number of the host on which the task is performed.
Catalog	The catalog that the report belongs to.

In addition, when you click the task ID of a report, the Task Information dialog will pop up, showing you the detailed task information, which includes the following:

- **Task ID**

  The exact date and time when the report was opened.

- **Task Type**

The task type. There are six task types: Schedule, On-Demand, Page Report Studio, Web Report Studio, Dashboard and Bursting.

- **Report**

  The path and name of the opened report.

- **Catalog**

  The catalog where the opened report lists.

- **Report Pages**

  The total page number of the report. Not available when task type is Web Report Studio or Dashboard.

- **Submit Time**

  The time when the report was opened.

- **Start Run Time**

  The time when the report was started to run.

- **Completed Time**

  The time when the report was closed.

- **Parameters**

  The parameters that user uses to run the report. If the report runs with no parameters, this column will be left empty.

Status of the waiting reports

The status table of the waiting reports includes:

Column	Description
Action	Stops the report from running and makes it a failed report.
Task ID	The internal ID for this report. If you click the ID, the Task Information dialog will pop up showing you the detailed task information.
Report	The full path name of the report.
User ID	The ID of the user who opened the report.
Task Status	For waiting reports, can be one of the following: • **Submitted** - The task has been submitted successfully. • **Unlaunch** - The task is currently in the unlaunch queue waiting to be processed. • **Task Queue** - The task is currently in the task thread queue waiting to be processed.
Task Type	The task type. Can be one of the following: Schedule, On-Demand, Page Report Studio, Web Report Studio, Dashboard, and Bursting.
Submit Time	The time when the report was opened.

Catalog	The catalog that the report belongs to.

Status of the finished reports

The Status table of the finished reports includes:

Column	Description
Task ID	The internal ID for this report. If you click the ID, the Task Information dialog will pop up showing you the detailed task information.
Report	The full path name of the report.
User ID	The ID of the user who opened the report.
Task Status	Completed. The task has been processed successfully, and has accomplished all the requirements.
Task Type	The task type. Can be one of the following: Schedule, On-Demand, Page Report Studio, Web Report Studio, Dashboard, and Bursting.
Run Host	The name of the host on which the task is performed.
Run Port	The port number of the host on which the task is performed.
Catalog	The catalog that the report belongs to.
Result Files	The path of the result files.

Report Pages	The total page number of the report. Not available when task type is Web Report Studio or Dashboard.
Reason	The reason why the task failed. Can be an exception or a meaningful description.
Submit Time	The time when the report was opened.
Start Run Time	The time when the report was started to run.
Completed Time	The time when the report was closed.

Status of the failed reports

The Status table of the failed reports includes:

Column	Description
Task ID	The internal ID for this report. If you click the ID, the Task Information dialog will pop up showing you the detailed task information.
Report	The full path name of the report.
User ID	The ID of the user who opened the report.
Task Status	Failed. The task has encountered errors, and has failed to accomplish all the requirements.

Task Type	The task type. Can be one of the following: Schedule, On-Demand, Page Report Studio, Web Report Studio, Dashboard, and Bursting.
Run Host	The name of the host on which the task is performed.
Run Port	The port number of the host on which the task is performed.
Catalog	The catalog that the report belongs to.
Result Files	The path of the result files.
Report Pages	The total page number of the report. Not available when task type is Web Report Studio or Dashboard.
Reason	The reason why the task fails. Can be an exception or a meaningful description.
Failed Info	The information about the report's failure.
Submit Time	The time when the report was opened.
Start Run Time	The time when the report was started to run.
Completed Time	The time when the report was closed.

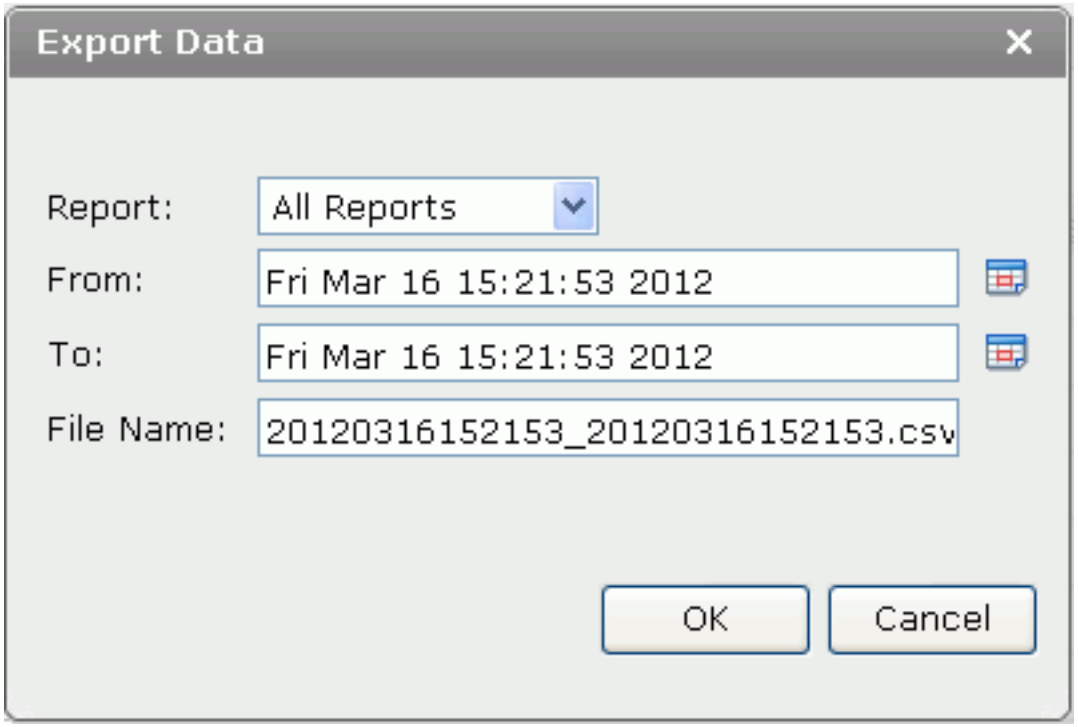
Exporting the monitoring data

You can export the monitoring data of the reports' status to a CSV file. However, before doing this, you need to make sure the profiling DB on the server node where the monitoring data will

be saved has already been configured. For details about how to configure the profiling DB, see the topic Configuring the server database in a standalone environment in the *JReport Server User's Guide*.

To export the monitoring data of the reports' status:

1. Click the **Export Data** link on the reports status panel. The Export Data dialog is displayed.



2. Specify the report type, the time period of the monitoring data and the name of the exported file respectively.

3. Click **OK** to export the data.

The following are details about options in the Export Data dialog:

Report

The type of reports you want to export the monitoring data.

From

The date and time from which the monitoring data starts. You can type it in the text field or click the calendar button to select the date and time.

To

The date and time to which the monitoring data ends. You can type it in the text field or click the calendar button to select the date and time.

File Name

The name of the exported file.

OK

Exports the data to a CSV file and opens the File Download panel. You can choose to open the file or save it elsewhere.

Cancel

Does not retain any changes and closes the dialog.

Note: The exported monitoring data in the CSV file are encoded with UTF-8. If you open an exported CSV file which contains Chinese strings by a double-click, the Chinese strings would be

displayed as random code in the file. To solve this problem, you need to open the file in the following way:

1. Start Microsoft Excel first.
2. Click **From Text** in the Data tab, then select the exported CSV file and click **Import**.
3. Select **65001: Unicode (UTF-8)** in the File Origin selector, then keep clicking **Next** and at last click **Finish** and **OK**.

Monitoring the status of online users

Expand any server node in the left panel of the JReport Server Monitor home page, and then click **Users**, you can see the status of the online users on the server. You can also select and remove specific users.

The Status table of the online users includes:

Column	Description
Session ID	The internal ID of the user session.
User ID	The ID of the user logged onto the server.
Create Time	The time when the user session was created.
Last Access Time	The time when the user last accessed the server.
HTTP Session ID	The session ID in the HTTP service.

Authentication	The authentication type. It can be Internal or External.

Monitoring the status of the database connection pool

Expand any server node in the left panel of the JReport Server Monitor home page, and then click **Databases**, you can see the status of the server's database connection pool. You can also select and remove specific connections.

The Status table of the connection pool includes:

Column	Value	Description
User	String	The user who is currently using the connection.
URL	A URL connecting to a database.	It specifies the connections that are based on a URL which will be caught in the pool.

Expiring Time (s)	0 (default) or expiring time. The unit is second. If the value is zero then the connection will never expire.	Shows the time during which a connection can be alive. If a connection has expired, the connection pool will close it.
Idle Expiring Time(s)	1 (default) or expiring time. The unit is second. Its value must be larger than or equals to 1.	Shows the time during which a connection is kept after it starts idling. If a connection is not used, it will stay open until the idle expiry time has been reached.
Maximal Connection Count	0 (default) or a positive integer.	Shows the pool size, which limits the number of connections under a single URL.

Maximal Share Count	0 (default) or a positive integer.	Shows the number of users who can share a connection simultaneously.
Attempt	A positive integer, the default value is 1.	Shows whether to re-create a connection when the connection pool has failed to create one and the number of attempts for creating the connection. If the user sets this value to a non-positive integer, the default value (1) will be used.
Interval(ms)	0 (default) or a positive integer. The unit is millisecond (ms).	If property ‘Attempt' is larger than 1, then before the connection pool retries to create a connection it will wait for an interval time. This property defines the interval time.
Last User Connecting Time(s)	0 (default) or a positive integer. The unit is second.	It shows the time elapsed since the last user who has taken this connection.

Current Idle Time(s)	0 (default) or a positive integer. The unit is second.	Shows the time elapsed since the connection started to idle.
Current Share Count	0 (default) or a positive integer.	Shows the number of users who are currently sharing this connection.

Note: The properties numbered 2 to 8 can be set in the ConnectionPoolConfig.properties file in `<server_install_root>\bin` and the last three properties will be shown according to the real time status.

← Previous Page Next Page →

Monitoring the Performance of JReport Server

JReport Server Monitor can show performance counters in graph (Line chart and Bar chart) and text mode.

To monitor the performance of JReport Server:

1. Access the home page of JReport Server Monitor.
2. In the left panel, expand the tree to select any server node.
3. Click the **Performance** tab and the performance chart of the specified server will be displayed.
4. If you want to configure the performance chart, click the **Graph Options** button  on the toolbar. In the Graph Options dialog, set how many tick marks you need to display on the X axis in the Keep Last N Records text field. In the Y Axis Limit and Y2 Axis Limit text fields, set respectively the maximum value on the left Y axis and the right Y axis. In the Interval text field, set the time interval (in seconds) the performance chart will use to get data and refresh itself.

   Click the **Clear Display** button  if you want to clear the current display of the performance counters in the performance chart.

To stop the current display of the performance counters, click the **Stop** button  on the toolbar.

The following are the available counters:

Performance Counter	Description
Waiting Reports	The number of the currently waiting reports.
Running Reports	The number of the currently running reports.
Finished Reports	The number of the finished reports.
Finished Report Pages	The number of pages of the finished reports.
Report Average Processing Time	The average processing time of each report.
Report Average Waiting Time	The average waiting time of each report.
Valid User Sessions	The number of valid user sessions.

Average Submitted Tasks per User	The average number of tasks that each user has submitted since JReport Server started.
Database Connections	The number of database connections.

← Previous Page Next Page →

Maintaining JReport Server

You can also perform some maintenance tasks on JReport Server Monitor, such as stopping problematic reports and connections and shutting down servers in a cluster.

Stopping problematic reports and connections

To stop a problematic report from running:

1. Expand the server node in the left panel of the JReport Server Monitor home page, and then click **Reports**. You can then see the status of the reports.
2. Click to select different reports from the drop-down list. There are five types of reports - all reports, running reports, waiting reports, finished reports and failed reports.
3. Choose to view running reports or waiting reports. Click on the **Action** column, you will find a command link Kill in front of each report status row. Click **Kill** to terminate the report running process.

To disconnect a connection:

1. Expand the server node in the left panel of the JReport Server Monitor home page, and click **Databases**. You can see the status of the database connections.

2. Check the box in front of the connection that you want to disconnect, and then click the **Disconnect** link on the top.

Shutting down servers in a cluster

To shut down one or more servers in a cluster, in the left panel of the JReport Server Monitor home page, select the root node, then in the Status tab, select the servers you want to shut down and click the **Shut Down** link.

← Previous Page Next Page →

Creating Profiling Reports

JReport Server Monitor can generate a JReport Server performance statistic report for you to inspect the server performance during a certain period of time. There can be two types of profiling reports: one collects report running information and the other obtains a specified number of frequently accessed reports. The profiling report that collects report running information categories the reports into three groups: All, Page Report and Non Page Report. All page reports running in Page Report Studio and other formats, all web reports on-demand running in the PDF, Excel, HTML formats, etc will be recorded. However, the profiling report does not record web reports running in Web Report Studio.

JReport Server Monitor generates the JReport Server performance statistic report using the information that JReport Server collects and saves to its own database. Whether or not JReport Server collects profiling information is controlled by the property server.profiling.enable=true/false in the server.properties file in `<server_install_root>\bin`.

Before JReport Server Monitor can inspect server performance, you must first make sure that JReport Server has collected its report-running information. To make certain of this, make sure that the property server.profiling.enable is set to true.

To create a profiling report:

1. Select a server node in the left tree, and then click the **Profiling** tab.

2. Specify the time period for JReport Server Monitor to get related data.

   If you want to create a profiling report for the topN most frequently viewed reports, in the Top box, specify how many most frequently accessed reports are to be recorded. Make sure that the number specified here must be valid. If not, the latest used number will be applied.

3. Click **Submit** to create the report.

← Previous Page Next Page →

Managing and Monitoring Resources Using the JMX Monitoring Features

JMX is a Java standard in the aspect of management. It provides a local or remote model for managing Java application systems. Its target is to offer the instrumentation, which can be accessed by all kinds of application management systems.

A JMX-compliant Application Server makes all registered manageable resources possible to be managed, configured, and controlled easily and flexibly. All those managed resources can be dynamically modified and processed by third-party management software or other management component systems.

JReport Server's JMX Monitoring features

This section introduces the infrastructure with which the JMX Monitoring features are

implemented and the implementing methods that can be used.

MBeans building principle

JReport adopts a Standard MBean specification for building JReport Server's underlying managed beans.

In the process of creating and registering the JMX MBean, JReport designed and abstracted all meta-data of all managed resources and then saved them into their corresponding persistent entities. At runtime, the program changes all these meta-data into MBeanInfo and then adds them to the related Managed Bean.

After all these MBeans have been built, they are then registered into an internal or external MBeanServer Agent. Only when top-level management software or other integration environments can find the MBeanServer agent, can they manage and control all the underlying Managed Beans registered in the MBeanServer.

Working principle behind the JMX Monitoring function

With JReport Server Monitor, when all the managed resources objects of Report Server have been initiated and instantiated, the Managed MBeans are then registered into the MbeanServer. Then all the attributes, operations, notifications and constructors of the associated Managed resource object are exposed to the MBeanServer via the corresponding Managed Bean. The MBeanServer then exposes all of these registered Managed Beans to the top-level management system resorted to the related Protocol Adaptors or Connectors.

In Server Monitor's environment, when launching the Server Monitor, JReport firstly created ClusterRuntimeMBean, ServerRuntimeMBean for every active JReport Server in the cluster, and TaskRuntimeMBean, UserSessionRuntimeMBean, DatabaseRuntimeMBean, AdhocRuntimeMBean for every active running JReport Server. Then all these MBeans can be registered into the integrated customized MBeanServer or Standalone MBeanServer agent. At any time, when a JReport Server is launched, the TaskRuntimeMBean, UserSessionMBean, DatabaseRuntimeMBean and AdhocRuntimeMBean associated with it are created and registered into the MBeanServer agent too. Then, all of JReport Server's monitoring features can be exposed to top-level management software or other integration environment resorted to the MBeanServer agent.

So, all the Runtime MBeans of Task, User Session, Database and Ad hoc related are dynamically created in real time and registered into a local MBeanServer. The MBeanServer will then invoke all registered MBean operation or attributes to implement the JMX monitoring function.

Descriptions of MBeans in JReport Server

When starting JReport Server Monitor, the following MBeans are created:

- **ClusterRuntimeMBean**
    - **Attributes**:

      ClusterEnable (boolean)

      ClusterName (java.lang.String)

      PredefinedServerStatus(jet.server.monitor.api.ServerStatus[])

      PredefinedServers(jet.server.api.cluster.Member[])

- **Operations**:

  ServerStatus getServerStatus(String host, int port);

  stopServer(String host, int port);

- **ServerRuntimeMBean**

  - **Attributes**:

    Name (java.lang.String)

    Host (java.lang.String)

    Port (int)

    Status (java.lang.String)

    Type (java.lang.String)

    WaitingRptNums (int)

    RunningRptNums (int)

    FinishedRptNums (int)

    FinishedRptPages (int)

    AvgProcessTimeByRpt (int)

    AvgWaitTimeByRpt (int)

    ValidUserSessionNums (int)

    AvgRptNumsSubmittedByUser (int)

    CurrentConnectionNums (int)

  - **Operations**:

    void stop();

- **TaskRuntimeMBean**
  - **Attributes**:

    AllFailedTaskInfos(jet.server.monitor.api.task.TaskInfo[])

    AllFinishedTaskInfos(jet.server.monitor.api.task.TaskInfo[])

    AllRunningTaskInfos(jet.server.monitor.api.task.TaskInfo[])

    AllWaitingTaskInfos(jet.server.monitor.api.task.TaskInfo[])

  - **Operations**:

    TaskInfo[] getRunningTaskInfosByUser(String userName);

    TaskInfo[] getWaitingTaskInfosByUser(String userName);

    TaskInfo[] getFinishedTaskInfosByUser(String userName);

    TaskInfo[] getFailedTaskInfosByUser(String userName);

    void stopTask(String taskID);

- **UserSessionRuntimeMBean**
  - **Attributes**:

    AllUserSessions(jet.server.api.UserSession[])

    SessionTimeout (int)

  - **Operations**:

    UserSession[] getUserSessionsByUser(String userName);

    void removeUserSessionBySessionID (String sessionID);

    void removeUserSessionsByUserID (String userID);

void removeAllUserSessions ();

String getHttpSessionID (String usID);

- **DatabaseRuntimeMBean**
  - **Attributes**:

    ConnectionProperties(java.util.Properties[])

  - **Operations**:

    void disconnect (String connInfoID);

HtmlAdaptor

JReport provides an HtmlAdaptor which allows all registered JReport Server's MBeans to be viewed from web pages. This HtmlAdaptor is based on the default implementation of Sun's JMX tools, and can process a user's HTTP request and dispatch it to the MBeanServer Agent. Then respond to client-end web pages according to the MBeanServer's returned information.

The HtmlAdaptor is also instantiated as an MBean and registered into the MBeanServer Agent. Due to this, you can modify its listener port or other properties from the MBeanServer just like other MBeans.

By default, the HTMLAdaptor uses the port 8849 for providing web services. You can modify it and specify a distinct port in the server.properties file in `<install_root>\bin`. If you have launched Server Monitor and JMX startup has been set to true, a property item named monitor.jmx.htmladaptor.port will then be appended to the server.properties file in

`<install_root>\bin`.

After launching JReport Server and Server Monitor, and setting the monitor JMX startup property to true, all registered JReport Server's MBeans will then be available for viewing from a web page (`http://localhost:8849` by default).

Supporting tips information in the web UI (some auxiliary tools)

In the default implementation mode, all MBean information is built by MBeanServer. All the descriptions and names of attributes, operations, and parameters displayed in the web UI are adopted as the default contents by the MBeanServer Agent.

For friendly supporting tips information in the web UI, JReport has improved all the managed beans with extended methods for supporting customized descriptions and names of the attributes, operations and parameters. All these customized descriptions and names are built into the MBeanInfo of all the registered MBeans. The MBeanServer will invoke these methods to obtain our customized descriptions and names of all displayed MBeanInfo's attributes, operations and parameters in the web UI.

Using the JMX Monitoring features

JReport have created a local MBeanServer for managing all Runtime MBeans in default mode. The management program within a standalone can directly get an MBeanServer's reference via the default implementation of the MBeanServerFinder interface.

In addition, you can also specify your customized MBeanServerFinder's implementation for obtaining an external MBeanServer agent when launching the Server Monitor. You then need only to append the Java option `-Djrs.externalMBeanServerProvider=` with your fully-qualified customized class file name to the Java command line that launches the Server Monitor.

When the Server Monitor has been started and the JMX feature supporting has been set to true, it will then get the System property of rs.externalMBeanServerProvider. If this property has been set, the Server Monitor will use it to obtain an external MBeanServer Agent. Otherwise, a default standalone MBeanServerFinder implementation will be adopted to get the internal default MBeanServer Agent.

After launching JReport Server and Server Monitor, and setting monitor JMX startup property to true, all registered JReport Server's MBeans will be available for view from a web page (default `http://localhost:8849`).

JMX Monitoring in an integration environment

In an integration environment with top-level java-enabled management software or another JMX-compliant HTTP application server, that application server has probably already provided an MBeanServer Agent for managing its manageable beans. When integrating JReport Server and Server Monitor with that server, all JReport Server's MBeans should also be registered into its existing MBeanServer.

We provide an MBeanServerFinder interface to support user's customized implementation for

obtaining an external MBeanServer Agent.

If you want to integrate JReport Server and Server Monitor with other JMX-enabled application server environment, you should provide an MBeanServerFinder's implementation for obtaining the application server's MBeanServer Agent, and append their implemented fully-qualified class file name with the java option `-Djrs.externalMBeanServerProvider=` to the corresponding application server's launching java command line. Then, after launching the application server integrated with JReport Server and Server Monitor, if you specify jmx startup to be enabled, then all JReport Server's MBeans will be built and registered into the application server's MBeanServer Agent. At the same time, the default HtmlAdaptor JReport provides will also be built and registered into the external MBeanServer Agent. Then all registered JReport Server's MBeans and the application server's registered managed beans from the HtmlAdaptor or via the application server's provided interface can be viewed.

In the default implementation, JReport has also provided three general integration-related MBeanServerFinder implementations as simple samples: JBossMBeanServerFinderImpl, TomcatMBeanServerFinderImpl and WebLogicMBeanServerFinderImpl. When you integrate JReport Server and Server Monitor with JBoss, Tomcat and WebLogic complied with JMX features, they can provide their implemented external MBeanServerFinder implementation to the java command line option `-Djrs.externalMBeanServerProvider=`, or directly use our default implementation and append it to the java option.

Notes for using the JMX Monitoring function

When using the JMX Monitoring function, pay attention to the following aspects:

- In the MBeanServerFinder interface, JReport has just specified one interface method public MBeanServer getMBeanServer(); You will need to implement this interface method and provide a public Constructor method without any parameters.
- When compiling and releasing our JMX features, it is required to append two Sun's JMX standard packages jmxri.jar and jmxtools.jar. Otherwise, you will be required to use JDK V1.8.0 or later version (In J2SE V1.8.0 and later, Sun Co. has also provided a JMX standard API as a part of JDK's standard APIs).
- When compiling our JMX features with the four external integration MBeanServerFinder's implementation(JBossMBeanServerFinderImpl, TomcatMBeanServerFinderImpl, WebLogicMBeanServerFinderImpl and WebSphereMBeanServerFinderImpl), it is required to append the three integration platform-related packages to the javac class path: "`$jboss-root$\lib\jboss-jmx.jar` (for JBoss V3.2.3)", "`$tomcat-root$\server\lib\catalina.jar` (for Tomcat V4.1.30)" and "`$weblogic-root$\server\lib\weblogic.jar` (for WebLogic V810)". However, these three packages do not need to be appended to our release products.
- Since the JMX monitoring function is based on the currently existing Server Monitor and is built into Server Monitor, then only Server Monitor's release version is required for appending Sun's JMX standard packages to Server Monitor's lib directory. The JMX feature of JReport Server does not require to append anything.

Integrating with a Servlet-enabled Web Server

JReport Server Monitor should be deployed to a separate system from JReport Server or on systems for system administrators.

This chapter shows how to integrate JReport Server Monitor with other servlet-enabled web servers on Windows:

- Running with WebLogic 12c Release 1 (12.1.2)
- Running with Tomcat 8.0.15
- Running with IBM WebSphere 8.5.3.3 by WAR File

Running with WebLogic 12c Release 1 (12.1.2)

Assume that:

- WebLogic 12c Release 1 (12.1.2) has been installed in `D:\bea`.
- JReport Server Monitor has been installed in `C:\JReport\Monitor`.

Generating the WAR file

Use the tool makewar.bat/makewar.sh to build the JReport Server Monitor WAR file as defined by makewar.xml for remote integration. Both makewar.bat/makewar.sh and makewar.xml are located in `C:\JReport\Monitor\bin`. The generated WAR file monitor.war will be saved to the directory `C:\JReport\Monitor\bin\distribute`.

Deploying the WAR file

1. If you have not already created a WebLogic Domain for JReport Server you must create one before starting the integration.
2. Start WebLogic by running startWeblogic.cmd in `D:\bea\user_projects\domains\domain_name\bin`.
3. Access the WebLogic Administrative Console by using URL `http://`

hostname:7001/console/, where the hostname is host name or IP address, and 7001 is the port number.

4. After your successful login, in the Domain Structure panel on the left, click **Deployments** node.

5. In the Summary of Deployments panel, click **Install**.

6. In the Install Application Assistant panel, click the **upload your file(s)** link.

7. In the Deployment Archive section, click **Browse** to select your monitor.war file, and then click **Next**.

8. Keep clicking **Next** until the Finish button is enabled, and then click **Finish**.

9. In D:\bea\user_projects\domains\domain_name\bin, edit the file startWebLogic.cmd by adding a line: set JAVA_OPTIONS="-Dmonitor.home=C:\JReport\Monitor".

10. Copy rmi.auth from <server_intall_root>\bin to C:\JReport\Monitor\bin. Enable RMI service for remote connection by setting the property server.rmiserver.enable=true in server.properties in <server_intall_root>\bin. Then start JReport Server.

11. Access the Server Monitor from a web browser (the default port of WebLogic is 7001) using the URL http://hostname:7001/monitor or http://hostname:7001/monitor/monitor/index.jsp.

Running with Tomcat 8.0.15

Assume that:

- JReport Server Monitor has been installed to `C:\JReport\Monitor.`
- Tomcat has been installed to `C:\tomcat`.

Configuring Tomcat

1. Build the JReport Server Monitor WAR file using the makewar.bat/makewar.sh utility in the `C:\JReport\Monitor\bin` directory. The generated WAR file monitor.war will be saved to the directory `C:\JReport\Monitor\bin\distribute`.
2. Set JAVA_HOME in the batch file catalina.bat in `C:\tomcat\bin`. For example, we use `C:\jdk1.8.0`.

```
rem $Id: catalina.bat,v 1.29 2002/04/01 19:51:31 patrickl Exp $
rem
-----------------------------------------------------------------------------
set JAVA_HOME=C:\jdk1.8.0
rem Guess CATALINA_HOME if not defined
if not "%CATALINA_HOME%" == "" goto gotHome
set CATALINA_HOME=.
```

3. Modify catalina.bat again to set the Java system variable. For example,

```
if not "%SECURITY_POLICY_FILE%" == "" goto doSecurity
%_EXECJAVA% %JAVA_OPTS% %CATALINA_OPTS% %DEBUG_OPTS%
-Djava.endorsed.dirs="%JAVA_ENDORSED_DIRS%"
-classpath "%CLASSPATH%"
-Dcatalina.base="%CATALINA_BASE%"
-Dmonitor.home="C:\JReport\Monitor"
```

4. Copy the WAR file monitor.war to `C:\tomcat\webapps`.
5. Copy rmi.auth from <server_install_root>\bin of the admin server (JReport Server you want to monitor) to C:\JReport\Monitor\bin. Enable RMI service for remote connection by setting the property server.rmiserver.enable to true in server.properties in <server_intall_root>\bin. And then start the admin server.

Launching Tomcat

1. Start Tomcat using `C:\tomcat\bin\startup`.
2. Load the default page of Server Monitor using `http://hostname:8080/monitor` or `http://hostname:8080/monitor/monitor/index.jsp`.
3. If the admin server and JReport Server Monitor are installed in different computers, you need to specify the right host and port of the admin server in JReport Server Monitor side via the following two properties in server.properties in C:\JReport\Monitor\bin before starting Tomcat:

   admin.server.host = The RMI host name or IP address of the admin server.

   admin.server.port = The RMI port number of the admin server.

← Previous Page Next Page →

Previous Page

Running with IBM WebSphere 8.5.3.3 by WAR File

Assume that:

- WebSphere 8.5.3.3 has been installed to `C:\WebSphere`.
- JReport Server Monitor has been installed to `C:\JReport\Monitor`.

Generating the WAR file

Use the tool makewar.bat/makewar.sh to build the JReport Server Monitor WAR file as defined by makewar.xml for remote integration. Both makewar.bat/makewar.sh and makewar.xml are located in `C:\JReport\Monitor\bin`. The generated WAR file monitor.war will be saved to the directory `C:\JReport\Monitor\bin\distribute`.

Deploying the WAR file

To integrate JReport Server Monitor with IBM WebSphere by a WAR file, take the following steps:

1. Start IBM WebSphere.
2. Open the **Administrative Console**. You can open the Administrative Console by using the

Start menu, or by using the URL `http://hostname:9080/ibm/console`, where the hostname is host name or IP address, and 9080 is the port number. Note that the user ID must contain only letters and numbers. You can use any of them as your user ID.

3. After successfully logging in, expand the **Applications** node in the left tree, and then click **New Application**.
4. Click **New Enterprise Application** on the right panel to install a new application.
5. Select **Local file system** , and then use **Browse** to select your monitor.war file. Then click **Next**.
6. Click **Next**.
7. In Step1, type **JReportMonitor** in the Application Name field, and then click **Next**.
8. Click **Next** for Step 2 and 3.
9. In Step 4, in the Context Root section, type **/monitor/**, and then click **Next**.
10. In Step 5, click **Finish**. The installing process may take several minutes. Please wait until the process is complete.
11. After the installation process has been successfully completed, click **Save**.
12. On the left resource tree, expand **Servers**, go through **Server Types > WebSphere application servers** > **Server1** > **Process Definition** (in the Java and Process Management node of the Server Infrastructure section) > **Java Virtual Machine** (in the Additional Properties section), and type **-Dmonitor.home=C:\JReport\Monitor** (JReport

Server Monitor installation folder) in Generic JVM arguments field in the Configuration tab.

13. Click **OK**, then click **Save** in the Messages box. Then stop Websphere.

14. Copy rmi.auth from `<server_intall_root>\bin` to `C:\JReport\Monitor\bin`. Enable RMI service for remote connection by setting the property `server.rmiserver.enable=true` in server.properties in `<server_intall_root>\bin`. Then start JReport Server.

15. Start WebSphere.

16. Access JReport Server Monitor using the URL `http://hostname:9080/monitor` or `http://hostname:9080/monitor/monitor/index.jsp`.

← Previous Page