Application Monitors

up.time Agent

The up.time Agent monitor determines whether or not an agent is running on a system that you are monitoring.

Configuring up.time Agent Monitors

To configure up.time Agent monitors, do the following:

1. In the up.time Agent monitor template, complete the monitor information fields.
   To learn how to configure monitor information fields, see Monitor Identification.
2. Complete the following options by clicking the checkbox beside each option, then specifying a warning and critical threshold.
   If the thresholds that you set are exceeded, then *up.time* generates an alert. For more information, see Configuring Warning and Critical Thresholds.
   - Major
     The major version number of the agent. For more information, see Understanding Major and Minor Versions.
   - Platform
     The operating system on which the agent is installed and running.
   - Response Time
     Enter the Warning and Critical Response Time thresholds for the length of time a service check takes to complete. For more information, see Configuring Warning and Critical Thresholds.
3. To save the data from the thresholds for graphing or reporting, click the *Save for Graphing* checkbox beside each of the metrics that you selected in step 3.
4. Complete the following settings:
   - Timing Settings (see Adding Monitor Timing Settings Information for more information)
   - Alert Settings (see Monitor Alert Settings for more information)
   - Monitoring Period settings (see Monitor Timing Settings for more information)
   - Alert Profile settings (see Alert Profiles for more information)
   - Action Profile settings (see Action Profiles for more information)
5. Click *Finish*.

Exchange

The Exchange 2003 and Exchange monitors identify when certain performance counters for Microsoft Exchange servers have exceeded user-defined thresholds. These thresholds can be, for example, an inordinately high number of inbound connections or a rapidly-growing message queue. Whenever a threshold exceeds a warning or critical amount, *up.time* generates an alert.

Use *up.time* 's Exchange 2003 monitor if you are using and monitoring Microsoft Exchange 2000 or 2003; use the Exchange monitor for later versions (e. g., Microsoft Exchange 2007 and 2010).

Configuring Exchange 2003 Monitors

To configure an Exchange 2003 monitor for your Microsoft Exchange 2000 or 2003 server, do the following:

1. Complete the monitor information fields.
   To learn how to configure monitor information fields, see Monitor Identification.
2. Complete the following settings by clicking the checkbox beside each option, and then specifying a warning and critical threshold.
   If the thresholds that you set are exceeded, then *up.time* generates an alert. For more information, see Configuring Warning and Critical Thresholds.
   - Web Mail Sends Per Second
     The maximum number of messages that can be sent from the Exchange server each second.
   - Web Mail Auths Per Second
     The maximum number of authorization requests that can be sent to the Exchange server each second.
   - SMTP Bytes Sent Per Second
     The total number of bytes sent per second by the Exchange SMTP server.
   - SMTP Bytes Received Per Second
     The total number of bytes received per second by the Exchange SMTP server.
   - SMTP Bytes Total Per Second
     The total number of bytes of information passing through the Exchange SMTP server each second.
   - SMTP Local Queue Length
     The number of messages in the SMTP queue that are scheduled for local delivery.
   - SMTP Messages Per Second
     The maximum number of messages per second that are allowed by the SMTP server.
   - SMTP Inbound Connections
     The number of incoming connections that the SMTP server allows.
   - SMTP Outbound Connections
     The number of outbound connections that the server allows to all remote domains.
   - SMTP Connection Errors Per Second
     The number of number of connection errors that occur per second.
   - Response Time
     Enter the Warning and Critical Response Time thresholds. For more information, see Configuring Warning and Critical Thresholds.
3. To save the data from the thresholds for graphing or reporting, click the *Save for Graphing* checkbox beside each of the metrics that you selected in step 2.
4. Complete the following settings:
   - Timing Settings (see Adding Monitor Timing Settings Information for more information)

- Alert Settings (see Monitor Alert Settings for more information)
- Monitoring Period settings (see Monitor Timing Settings for more information)
- Alert Profile settings (see Alert Profiles for more information)
- Action Profile settings (see Action Profiles for more information)

5. Click *Finish*.

Configuring Exchange Monitors

To configure an Exchange monitor for your Micorsoft Exchange 2007 or 2010 server, do the following:

1. Complete the monitor information fields.
2. To learn how to configure monitor information fields, see Monitor Identification.
3. Complete the following settings by clicking the checkbox beside each option, and then specifying a warning and critical threshold. If the thresholds that you set are exceeded, then *up.time* generates an alert. For more information, see Configuring Warning and Critical Thresholds.
    - SMTP Bytes Sent Per Second
      The total number of bytes sent per second by the Exchange SMTP server.
    - SMTP Bytes Received Per Second
      The total number of bytes received per second by the Exchange SMTP server.
    - SMTP Messages Sent Per Second
      The maximum number of messages sent per second allowed by the SMTP server.
    - SMTP Messages Received Per Second
      The maximum number of messages received per second allowed by the SMTP server.
    - SMTP Average Bytes Per Message
      The average number of message bytes per inbound message received, indicating the size of messages received through an SMTP receive connector.
    - SMTP Inbound Connections
      The number of incoming connections that the SMTP server allows.
    - SMTP Outbound Connections
      The number of outbound connections that the server allows to all remote domains.
    - Average Delivery Time
      The average time, in milliseconds, between an Exchange server receiving a message from the client, and an Exchange server deliverying the message to an Inbox.
    - Active Connections
      The number of connections to the Exchange store that have shown activity in the last 10 minutes.
    - Active Client Logons
      The number of clients that performed any action within the last 10-minute time interval.
    - Active User Count
      The number of unique user connections that have logged on to the server and shown activity in the last 10-minute time interval.
    - Current Webmail Users
      The number of unique users currently logged in to Outlook Web Access. This counter decreases when users manually log out or their sessions time out.
    - Webmail User Logons Per Second
      The number of Outlook Web Access logins or login attempts per second.
    - RPC Averaged Latency
      The average time, in milliseconds, it takes for the last 1,024 packets to be processed.
    - RPC Operations Per Second
      The rate that RPC operations occur, and implicitly, how how many RPC requests are outstanding.
    - RPC Requests
      The number of client requests that are currently being processed by the Exchange store.
    - Response Time
      Enter the Warning and Critical Response Time thresholds. For more information, see Configuring Warning and Critical Thresholds.
4. To save the data from the thresholds for graphing or reporting, click the *Save for Graphing* checkbox beside each of the metrics that you selected in step 2.
5. Complete the following settings:
    - Timing Settings (see Adding Monitor Timing Settings Information for more information)
    - Alert Settings (see Monitor Alert Settings for more information)
    - Monitoring Period settings (see Monitor Timing Settings for more information)
    - Alert Profile settings (see Alert Profiles for more information)
    - Action Profile settings (see Action Profiles for more information)
6. Click *Finish*.

IIS

The IIS (Internet Information Server) service monitor checks the performance of an IIS Web server, based on thresholds that you set against common IIS performance counters. You can use this monitor to determine whether or not IIS is running on a defined port, and according to the thresholds you have set on common performance counters.

Configuring IIS Monitors

To configure IIS monitors, do the following:

1. In the IIS monitor template, complete the monitor information fields.
   To learn how to configure monitor information fields, see Monitor Identification.
2. Complete the following settings by clicking the checkbox beside each option, and then specifying a warning and critical threshold. If the thresholds that you set are exceeded, then *up.time* generates an alert. For more information, see Configuring Warning and Critical Thresholds.

- Bytes Sent / Sec.
  The number of bytes that are sent by the server each second.
- Bytes Received / Sec.
  The number of bytes that are received by the server each second.
- Anonymous Users / Sec.
  The rate, in seconds, at which users have made anonymous requests to the IIS server.
- Non-anonymous Users / Sec.
  The rate, in seconds, at which registered users have made non anonymous requests to the IIS service.

  > *Note - IIS 6.0+ treats both an anonymous and a non-anonymous user request as a new user.*

- Current Connections
  The number of active connections to the IIS server.
- Connection Attempts / Sec.
  The number of connection attempts that have been made, per second, since the IIS server was started.
- Logon Attempts / Sec.
  The number of attempts, per second, that are being made to log on to the server.
- Get Requests / Sec.
  The rate, in seconds, at which HTTP requests using the *GET* method have been made to the server.
- Post Requests / Sec.
  The rate, in seconds, at which HTTP requests using the *POST* method have been made to the server.
- CGI Requests / Sec.
  The rate, in seconds, at which the server is processing simultaneous CGI (Common Gateway Interface) requests.
- ISAPI Requests / Sec.
  The rate, in seconds, at which the server is processing ISAPI extension requests.
  ISAPI enables programmers to develop Web applications that are tightly integrated with IIS. ISAPI can also provide security functions to Windows servers and database connections through IIS.
- Not Found Errors / Sec.
  The maximum number of *404 file not found* errors - indicating that the requested document cannot be found on the server - that can occur each second.
- Response Time
  Enter the Warning and Critical Response Time thresholds for the length of time a service check takes to complete. For more information, see Configuring Warning and Critical Thresholds.

3. To save the data from the thresholds for graphing or reporting, click the *Save for Graphing* checkbox beside each of the metrics that you selected in step 3.
4. Complete the following settings:
   - Timing Settings (see Adding Monitor Timing Settings Information for more information)
   - Alert Settings (see Monitor Alert Settings for more information)
   - Monitoring Period settings (see Monitor Timing Settings for more information)
   - Alert Profile settings (see Alert Profiles for more information)
   - Action Profile settings (see Action Profiles for more information)
5. Click *Finish*.

WebLogic

The WebLogic monitor collect data that enables you to determine whether or not there is a performance problem or a failure on a WebLogic application server. Using the data that the WebLogic monitor collects, you can determine the root cause of the issue by generating a report (see Reports for J2EE Applications for more information).

The WebLogic monitors collect the following metrics from a WebLogic server:

Variables	Metrics
Connection Pools	FailuresToReconnectCount The number of times that the connection pool failed to reconnect to a data store. ConnectionDelayTime The average time that was required to connect to a connection pool. ActiveConnectionsCurrentCount The current number of active connections in a JDBC connection pool. ActiveConnectionsHighCount The highest number of active connections in a JDBC connection pool. LeakedConnectionsCount The total number of connections that have been checked out of, but not returned to, the connection pool. CurrCapacity The current number of database connections in the JDBC connection pool.

	NumAvailable The number of available sessions in the session pool that are not currently being used. WaitingForConnectionCurrentCount The current number of requests that are waiting for a connection to the connection pool.
Per EJB	AccessTotalCount The total number of times an attempt was made to get an EJB instance from the free pool. BeansInCurrentUseCount The number of EJB instances in the free pool which are currently in use. CachedBeansCurrentCount The total number of EJBs that are in the execution cache. ActivationCount The number of EJBs that have been activated.
Other	HeapSizeCurrent The amount of memory, in bytes, that is in the WebLogic server's JVM heap. HeapFreeCurrent The current amount of free memory, in bytes, that is in the WebLogic server's JVM heap. OpenSocketsCurrentCount The current number sockets on the server that are open and receiving requests. AcceptBacklog The number of requests that are waiting for a TCP connection. ExecuteThreadCurrentIdleCount The number of threads in the server's execution queue that are idle or which are not being used to process data. PendingRequestCurrentCount The number of pending requests that are in the server's execution queue. TransactionCommittedTotalCount The total number of transactions that have been processed by the WebLogic server. TransactionRolledBackTotalCount The total number of transactions that have been rolled back. InvocationTotalCount The total number of times that a servlet running on the WebLogic server was invoked.

Before you can use the WebLogic monitors, you must perform additional steps outside of *up.time* . The steps performed depend on the version of your WebLogic server: WebLogic monitoring requires that you enable the Internet Inter-Orb Protocol (IIOP) on your WebLogic server.

Monitoring WebLogic

In order for *up.time* to collect information from a WebLogic server, the the Internet Inter-Orb Protocol (IIOP) must be enabled on your WebLogic server.

To enable prepare your WebLogic server for monitoring, do the following:

1. Enable IIOP on your WebLogic server.
   For example, on WebLogic 11, select the *Protocols* tab when configuring server settings, then select the *Enable IIOP* checkbox.
2. Enter an IIOP user name.
3. Enter an IIOP user password.
4. If possible, restart the WebLogic server.
   The user name and password created here are used when configuring a WebLogic 11 monitor in *up.time* .

Configuring WebLogic Monitors

To configure monitors for WebLogic, do the following:

1. In the WebLogic monitor template, complete the monitor information fields.
   To learn how to configure monitor information fields, see Monitor Identification.
2. Complete the following fields:
   - Username
     The IIOP user name you created when you first enabled IIOP on the WebLogic server.
   - Password
     The IIOP password you created when you first enabled IIOP on the WebLogic server.
   - WebLogic Port
     The number of the port number on which the WebLogic server is listening. The default is 7001.
3. Limit the returned results of a specific resource type by completing some of the following fields:
   - Number of Results
     A limit on the number of matching application resources, whose metrics are collected.
   - EJB Name Regex Filter
     A regular expression used to limit metrics collection to a specific EJB or set of EJBs.
   - Servlet Name Regex Filter
     A regular expression used to limit metrics collection to a specific servlet.
   - JDBC Resource Name Regex Filter
     A regular expression used to limit metrics collection to a specific JDBC resource.
4. Specify a warning and critical threshold for the following:
   - the appropriate WebLogic metrics
     For more information about each metric, see Connection Pools.
   - Response Time
     This is the length of time a service check takes to complete.

   For more information on using thresholds to set alerts, see Configuring Warning and Critical Thresholds.
5. To save the data from the thresholds for graphing or reporting, click the *Save for Graphing* checkbox beside each of the metrics that you selected in the previous step.
6. Complete the following settings:
   - Timing Settings (see Adding Monitor Timing Settings Information for more information)
   - Alert Settings (see Monitor Alert Settings for more information)
   - Monitoring Period settings (see Monitor Timing Settings for more information)
   - Alert Profile settings (see Alert Profiles for more information)
   - Action Profile settings (see Action Profiles for more information)
7. Click *Finish*.

WebSphere

WebSphere is a software platform that provides firms with an environment for developing and deploying Web services and E-Commerce applications. Since WebSphere large and complex, it can be difficult to pinpoint the source of a problem, especially when that problem is intermittent.

The *up.time* WebSphere monitor collects data that you can use to generate a report, which will give you a historical view of problems that occur on a WebSphere server. See See WebSphere Report. for more information.

The WebSphere monitor enables you to collect data so that you can:

- determine whether or not the server can cope with its load
- determine the cause of problems with the server
- collect and retain data for later graphing and reporting

The following table lists the counters the WebSphere monitor collects from a WebSphere Application Server.

Variable	Counters
Connection pools	PoolSize The size of the connection pool to the data source. FreePoolSize The number of free connections in the pool. PercentUsed The percentage of the connection pool that is currently in use. WaitTime The average time, in milliseconds, that a connection is used. The average time is the difference between the time at which the connection is allocated and the time at which it is returned.
	CreateCount The total number of connections that were created. CloseCount The total number of connections that were closed. WaitingThreadCount

	The number of threads that are currently waiting for a connection. UseTime The average time, in milliseconds, that a connection is used. The average use time is the difference between the time at which the connection is allocated and that time at which it is returned.
Per EJB	CreateCount The number of times that the Enterprise JavaBeans that are running on the server were created. RemoveCount The number of times that the EJBs were removed. PassivateCount The number of times that EJBs were removed from the cache. Note that passivation preserves the state of the EJBs on the disk MethodCallCount The total number of method calls that were made to the EJBs. MethodResponseTime The average response time, in milliseconds, on the bean methods.
Java Virtual Machine	cpuUsage The percent of CPU resources that were used since the last query. HeapSize The total amount of memory that is available for the JVM. UsedMemory The amount of memory that is being used by the JVM.
Other	ActiveCount The number of global transactions which are concurrently active. CommittedCount The total number of global transactions that have been committed. RolledBackCount The total number of global transactions that have been rolled back. LiveCount The number of servlet sessions that are currently cached in memory. PoolSize The average number of threads in the servlet connection thread pool. TimeSinceLastActivated The difference, in milliseconds, between the previous and current access time stamps of a servlet session. This counter does not include session time out values.

Before *up.time* can start collecting performance data from a WebSphere server, you must deploy the WebSphere performance servlet.

Deploying the WebSphere Performance Servlet

The WebSphere performance servlet uses WebSphere's Performance Monitor Interface (PMI) infrastructure to retrieve performance information from a WebSphere Application Server. The information that the servlet collects is saved to an XML file.

By default, the PMI is enabled on the WebSphere server and is set to collect the performance metrics that *up.time* supports. Before *up.time* can begin collecting information from a WebSphere server, you must deploy the performance servlet in the WebSphere directory that contains your Web application.

Note - The following steps must be completed for each Web application server that you want to monitor with up.time.

To deploy the performance servlet do the following:

1. On the WebSphere server, locate the following file:
   install_root/perfServletApp.ear
   Where *install_root* is the directory under which WebSphere is installed.
2. Copy the file *perfServletApp.ear* to the directory in which your Web application is installed. For example:
   install_root/installedApps/<cell_name>/DefaultApplication.ear/DefaultApplication.war/WEB-INF/classes
   Where:
   - *install_root* is the directory under which WebSphere is installed.
   - *<cell_name>* is the name of the WebSphere node under which your Web application is installed.

Deploying the Performance Servlet on WebSphere

If you are using WebSphere Application Server, you will need to change two settings in the WebSphere management console to avoid an Access Denied error when *up.time* attempts to connect to the performance servlet to collect metrics.

To make the changes, do the following:

1. In the WebSphere management console, modify the following settings:
2. Under Security - Secure administration, applications, and infrastructure, enable the *Application Security* option.
3. Under Enterprise Applications - perfServletApp - Security role to user/group mapping, disable the *Everyone* option, and enable the *All authenticated* option.
4. Restart the server. *up.time* should now be able to connect to the servlet and gather performance metrics.

Configuring WebSphere Monitors

To configure a WebSphere monitor, do the following:

1. On the WebSphere monitor template, complete the monitor information fields.
   To learn how to configure monitor information fields, see Monitor Identification.
2. Complete the following fields:
   - WebSphere Port
     The number of the port number on which WebSphere is listening. The default is 9080.
   - Response Time
     Enter the Warning and Critical Response Time thresholds. For more information, see Configuring Warning and Critical Thresholds.
3. Optionally, click the *Save for Graphing* checkbox beside the *Response Time* option to save the data for a metric to the DataStore, which can be used to generate a report or graph.
4. Complete the following settings:
   - Timing Settings (see Adding Monitor Timing Settings Information for more information)
   - Alert Settings (see Monitor Alert Settings for more information)
   - Monitoring Period settings (see Monitor Timing Settings for more information)
   - Alert Profile settings (see Alert Profiles for more information)
   - Action Profile settings (see Action Profiles for more information)
5. Click *Finish*.

Web Application Transactions

A Web transaction is a series of Web pages that together fulfill a specific function for end users. A common Web transaction example is the checkout process on an e-commerce site, during which end users select a shipping option, pay for their items, and have their credit card verified. During this transaction, many calls are made to the application and data layers as the end-user provides, and the servers process, information.

Although the type of Web application that is monitored by *up.time* users is typically different (e.g., intranet applications), the structure of the transaction is the same: an end user steps through a sequence of Web pages that take inputted information and initiate appropriate actions with application or database servers.

The *up.time* Web Application Transaction monitor tests the speed and availability of an end-user Web transaction. Specifically, the Web Application Transaction monitor performs two roles:

- it confirms the general availability of an end-user Web transaction by executing a previously recorded script then reporting whether all pages that make up the web transaction were successfully processed
- it reports on the speed of the Web transaction both as a whole, and broken down by previously defined stages

Both the availability and speed of Web transactions can be used in reports and as triggers for alerts.

Using the Web Application Transaction Monitor

Use the Web Application Transaction monitor to record a series of URLs that together make up a transaction. This recording should be of a transaction that acts as a suitable test of your Web application delivery infrastructure.

During the recording process, declare checkpoints that demarcate significant stages in the Web transaction. Isolating the different stages in an end-user transaction allows you to view stage-specific speed tests in reports, which ultimately helps you identify where problem areas exist.

For example, if a transaction relies on processing on the application layer, makes multiple calls to the data layer, and is accessible worldwide, creating checkpoints during the recording phase helps you ascertain whether the application server, database management server, or network may be the reason behind a poorly performing transaction.

The following sample checkpoints could be created for an e-commerce transaction:

- Browse Catalog

- Add to Shopping Cart
- Checkout
- Credit Card Validation

The following sample checkpoints could be created for an internal office transaction:

- Login
- Browse Orders
- View Order Details

Configuring Web Application Transaction Monitors

You can define Web application transactions by manually stepping through one and declaring checkpoints at key stages:

1. Open a Web browser, and configure its proxy settings so that you can record a transaction:
   - Open the dialog where network and connection settings are made.
   - Configure the browser's proxy to "localhost" on port 8001.
   - Ensure these settings have also been applied to SSL or secure communications.
   - Set the proxy to bypass the Monitoring Station.
     This step will require you to select an option such as "no proxy" or "bypass proxy server," and may also require you to manually enter your Monitoring Station URL or IP address.

   Using the monitor as a proxy will allow it to intercept Web traffic as you generate it.
2. In the browser, navigate to the starting point of the Web application whose performance you will be monitoring.
3. In the *up.time* Add Service window, select the *Web Application Transaction* monitor, then click *Continue* .
   The Web Application Transaction Recorder is displayed, and the monitor is now listening on port 8001 for traffic.
4. Begin stepping through the Web transaction as an end user, providing the required data or actions.
   Every URL visited during the transaction is logged and displayed in the recorder.

   > *Note - The Web Application Transaction monitor records all data inputted during recording: this includes any login information. It is recommended that you use a test account for the Web application, otherwise any user data will be visible in the recorded script.*

5. At each major step in the Web transaction that signals a new analysis point, enter a checkpoint name in the text box at the top of the window, then click *Mark Checkpoint* .
   For example, create a checkpoint at a transaction step where the application takes user-inputted data and makes database calls.

   > *Note - You will later set Warning and Critical thresholds that apply to every segment declared in your recording. It is recommended that the divisions between your checkpoint intervals are reasonably consistent.*

6. Continue to repeat steps 4 and 5 until you have completed enough of the Web transaction to test it, then click *Next* .
7. Complete the monitor information fields.
   To learn how to configure monitor information fields, see Monitor Identification.

   > *Note - Even though the Web application performance is not directly tied to an Element's performance, making this selection is still required: the service based on this monitor needs to be associated with an Element in order to be viewed in areas such as Global Scan or My Infrastructure.*

8. Configure the Web Application Transaction Settings:
   - Script to play back
     If desired, optimize the playback script (e.g., remove extraneous URLs such as image downloads).
   - Text that must appear
     Enter a text string that can be used to confirm the script playback was successful (e.g., a phrase that appears on the final page of the application). If the monitor does not find this text, its status changes to Critical. By providing mandatory text, you can ensure an alert is triggered in cases where a Web application is malfunctioning, but checkpoint-to-checkpoint times are fast enough to fulfill response time requirements.
   - Text that must not appear
     Enter a text string that should not appear at any point during the script playback (e.g., a client- or server-error HTTP status code). If the monitor finds this text, its status changes to Critical. Use this feature, as you would use mandatory text, to ensure a malfunctioning application triggers an alert.
   - User Agent String
     Select the Web browser and version used to record the script. This selection determines the user agent string used in the HTTP requests to the application server, and should be provided in case the application blocks access by scripts.
   - Checkpoint Times
     Enter the Warning and Critical Checkpoint Time thresholds. An alert is generated with these thresholds if any of the recorded Web transaction's checkpoint times exceeds the supplied values.
   - Response Time
     Enter the Warning and Critical Response Time thresholds. An alert is generated with this threshold if the entire transaction playback time exceeds the supplied values. For more information, see See Configuring Warning and Critical Thresholds.
9. Enter Warning- and Critical-level thresholds for the overall response time of the monitor.
   Most of the monitor's Response Time is comprised of the Delivery Time and the Retrieve Time. Ensure the values provided for the Response Time thresholds roughly correspond with those provided for the other thresholds. For more information, see Configuring Warning and Critical Thresholds.
10. Complete the following settings:
    - Timing Settings (see Adding Monitor Timing Settings Information for more information)
    - Alert Settings (see Monitor Alert Settings for more information)

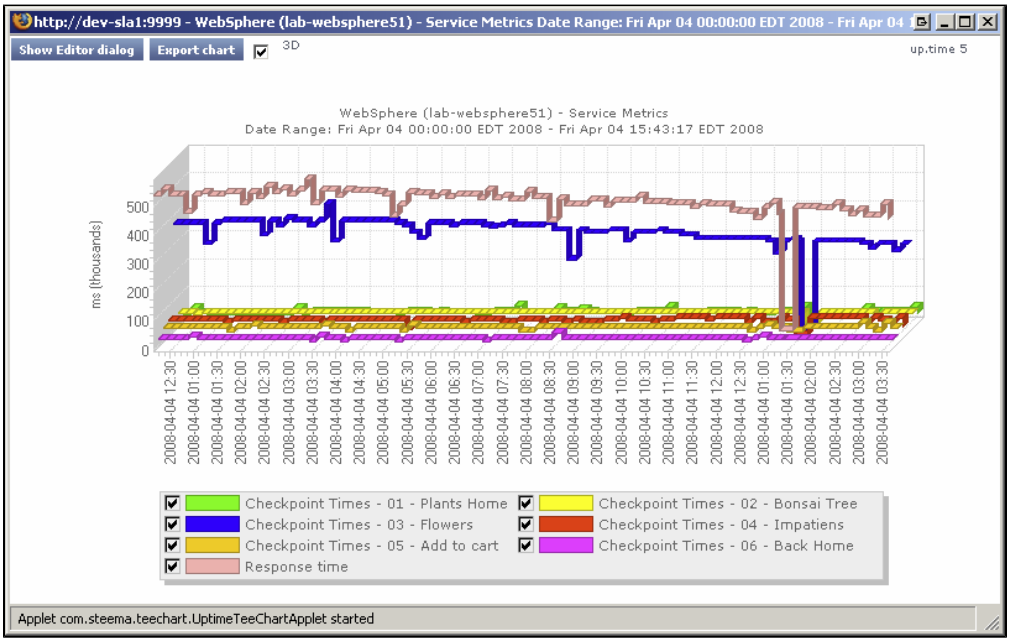
- Monitoring Period settings (see Monitor Timing Settings for more information)
- Alert Profile settings (see Alert Profiles for more information)
- Action Profile settings (see Action Profiles for more information)

11. Click *Finish*.

Viewing and Diagnosing Web Transaction Performance

To view Web transaction performance via playback, create a Service Metrics graph for the Web Application Transaction monitor's system. To generate a Service Metrics graph, either select the system to which the Web Application Transaction monitor is associated in *My Infrastructure* , or the monitor itself in the main *Services* panel. Click the *Graphics* tab, then click *Service Metrics* .

The Service Metrics graph shows how long each transaction segment took to complete during playback, and in doing so, provides an end-to-end performance snapshot of the components of your infrastructure that deliver applications to users. For example, the following metrics graph shows that the execution of the commants found in checkpoint3 took excessively long to complete:



Since other checkpoints performed well, the poor performance of a single checkpoint indicates possible issues with a particular server, and not the network infrastructure. This theory can be further investigated by looking at the performance metrics for the server in question.

Use the Web Application Transaction monitor's playback script to verify which servers are being used during a problem checkpoint. In the *Service Instances* panel, click the monitor to view the script, then locate the system that is being accessed (e.g., with GET and POST commands). Use this as an investigative starting point: although an application or Web server is often referenced in the script, the problem may be found deeper in the application stack (e.g., a database server to which the referenced Web server makes calls during the checkpoint).

Using Web Transaction Performance in SLA Reports

Your Web applications will typically call on systems on the application and database tiers, as well as make use of internal- and external-facing network devices. Since the Web Application Transaction monitor directly reports on the performance of a Web transaction, it in effect indirectly reports on the health of your IT infrastructure as a whole.

This broad reporting coverage makes the Web Application Transaction monitor an ideal monitor to include in service level agreement reports.

For more information on SLA reports, see Reports for Service Level Agreements.

Email Delivery Monitor

Although specific *up.time* monitors are available for your POP, IMAP, and SMTP servers, their monitoring duties focus on availability and response time. To test your IT infrastructure's ability to send or receive emails within a reasonable amount of time, use the Email Delivery monitor.

Typically, email delivery tests include a server that is part of your IT infrastructure and monitored by *up.time* . In these cases, you will test either incoming mail delivery times by supplying information about a monitored POP3 or IMAP server, or test outgoing mail delivery times by supplying information about a monitored SMTP server.

The Email Delivery executes several steps in order to calculate mail delivery and retrieval time:

- the monitor requests an internal or external SMTP server to send a generated test mail (when the monitor asks the SMTP server to send the mail, the monitor records the delivery time)
- the monitor waits for five seconds, then logs in to and checks an internal or external POP3 or IMAP mail server to verify the mail was received
- if the test mail is not found, the monitor waits another five seconds and checks again (and continues to check until the process has either timed out or the mail is found)
- the monitor confirms the mail was received and reports both the delivery and retrieval times

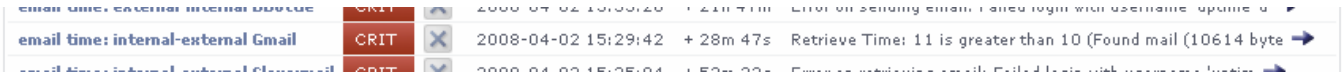
Configuring Email Delivery Monitors

Define the Email Delivery monitor by providing information about the outgoing and incoming mail servers:

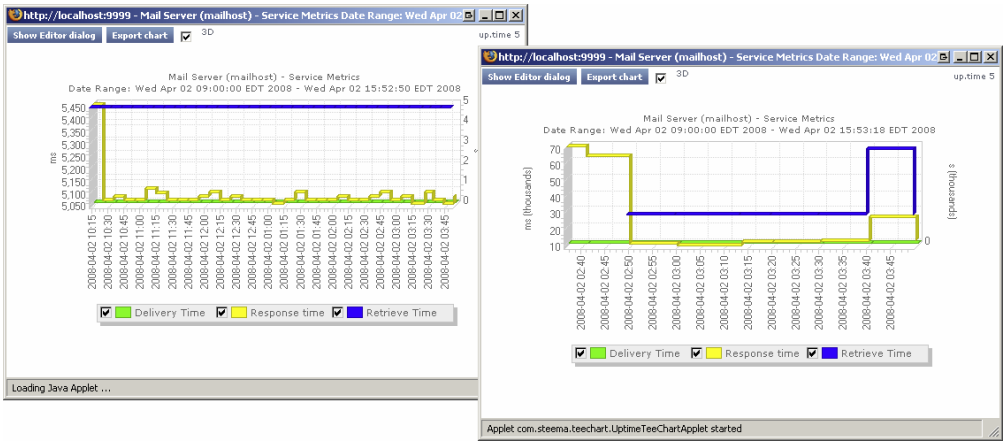
1. Complete the monitor information fields.
   To learn how to configure monitor information fields, see Monitor Identification.
   Note - Once created, the Email Delivery monitor service can be included with status reports for the system or group you select. If this monitor is reporting outgoing mail delivery times, the system should be a monitored SMTP server; if incoming mail delivery times are being measured, the system should be a monitored POP3/IMAP mail server.
2. Complete the Outgoing Email Settings:
   - SMTP Hostname
     Provide the name or IP address of the SMTP server.
   - SMTP Port
     Provide the port used to communicate with the SMTP server. Leave this field blank to use the default SMTP port (25).
   - SMTP Username
     Provide the authenticated SMTP user name.
   - SMTP Password
     Provide the authenticated SMTP user password.
   - SMTP Uses SSL
     Specify whether the SMTP server sends and receives encrypted communication using SSL.
   - Destination Email Address
     Enter the test email address used by the monitor. The monitor sends an email to this address, and this address is checked for receipt of the test email.
     Note - Although the Email Delivery monitor attempts to promptly find and delete test emails, network issues may prevent timely cleanups. To avoid potential Inbox clutter, it is recommended that you create a dedicated test email account as the destination address.
   - Delivery Time
     Enter the Warning and Critical Delivery Time thresholds. The smallest unit of time used for these thresholds is seconds. Given the speed at which SMTP servers should finish processing an outgoing email, is it recommended that you set the Warning threshold to one second.
3. Complete the Incoming Email Settings:
   - POP3/IMAP Hostname
     Provide the name or IP address of the mail server.
   - POP3/IMAP Port
     Provide the port used to communicate with the mail server. Leave this field blank to use the default POP3 or IMAP port (110 and 143, respectively).
   - POP3/IMAP Username
     Provide the login name for the destination email account.
   - POP3/IMAP Password
     Provide the password for the destination email account.
   - POP3/IMAP Uses SSL
     Specify whether the mail server sends and receives encrypted communication using SSL.
   - Retrieve Time
     Enter the Warning and Critical retrieval time thresholds. The smallest unit of time used for these thresholds is seconds, and the monitor checks for receipt of the test mail in five-second intervals. Enter values in multiples of five.
4. Enter Warning- and Critical-level thresholds for the overall response time of the monitor.
   Enter the Warning and Critical Response Time thresholds. An alert is generated with this threshold if the combined email delivery and response time exceeds the supplied values. For more information, see Configuring Warning and Critical Thresholds.
5. Complete the following settings:
   - Timing Settings (see Adding Monitor Timing Settings Information for more information)
   - Alert Settings (see Monitor Alert Settings for more information)
   - Monitoring Period settings (see Monitor Timing Settings for more information)
   - Alert Profile settings (see Alert Profiles for more information)
   - Action Profile settings (see Action Profiles for more information)
6. Click *Finish*.

Diagnosing and Reporting Email Delivery Problems

If the Email Delivery monitor reaches a Critical state, the first investigation step is to review the message produced by *up.time* . In the *System Status* panel, view the message belonging to the system to which the monitor is attached, which should point you in the right direction. For example, the status message below indicates the monitor reached a critical state because the retrieval time from an external POP3 server exceeded the defined threshold; your SMTP server is most likely not responsible for the delay:



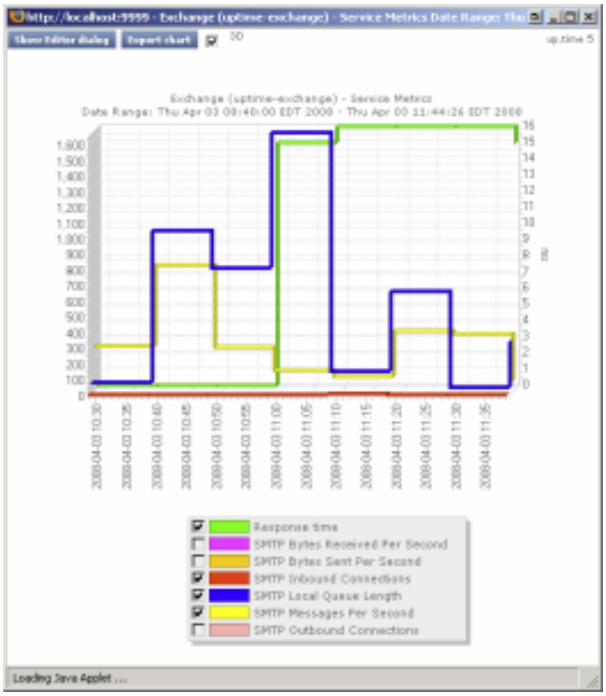
Speculation based on the status message can be confirmed using a Service Metrics graph for the Email Delivery monitor's system. This graph indicates whether the delivery and retrieval time are within acceptable limits (below left), or if one or both are unusually long (below right):



To generate a Service Metrics graph, either select the system to which the Email Delivery monitors are associated in *My Infrastructure* , or the monitor itself in the main *Services* panel. Click the *Graphics* tab, then click *Service Metrics* .

Even if the Service Metrics graph indicates delivery and retrieval times are not exceeding defined thresholds (and *up.time* is not sending out critical alerts), it is still an ideal investigative starting point if you are getting critical feedback from your users about email delivery times.

If the Email Delivery monitor's Service Metrics graph confirms that there are delays somewhere within your network infrastructure, you can investigate further by using the service monitor you created for your mail server. Co-ordinate your Email Delivery monitor's metrics graphs or reports with those from a service monitor you have assigned to your mail server (e.g. Exchange) while focusing on metrics that may be related outgoing or incoming mail time delays. For example, in the Exchange service monitor metrics graph below, the mail server experienced a high SMTP Local Queue Length that did not always coincide with the SMTP Messages Per Second count:



Splunk Query

Splunk is a third-party search engine that indexes log files and data from the devices, servers, and applications in your network. Using Splunk, you can quickly analyze your logs to pinpoint problems on a server or in a network, or ensure that you are in compliance with a regulatory mandate or Service Level Agreements. You install Splunk on a server in your data center.

When you click the Splunk icon beside the names of services that are in WARN or CRIT states in the *My Portal* panel, you will be taken to your Splunk search page.

You can use the Splunk Query monitor to perform Splunk queries on log files to pinpoint an error condition.

> *Note - Before you can use a Splunk Query monitor, you must add some settings specific to Splunk to the file uptime.conf. See Splunk Settings fo r more information.*

Configuring Splunk Query Monitors

To configure a Splunk Query monitor, do the following:

1. Complete the monitor information fields.
   To learn how to configure monitor information fields, see Monitor Identification.
2. Complete the following fields:
   - Splunk query
     The Splunk query string that you want to use to search log file for an error condition. For example, entering the following query string:
     host::mailServer sendmail error hoursago::2
     Will search log files, that were generated for the system named mailServer, for the word *sendmail* and *error* that were logged within the last two hours.
     You can enter any Splunk query string in this field. For more information on the syntax of Splunk queries, see the Splunk user manual.
   - Result count of splunk query
     Enables *up.time* to alert you when the number of results that match your Splunk query exceeds the defined warning and critical thresholds.
     For example, you can configure the monitor to issue a Warning alert when five or more Splunk queries matching your query are returned, and a Critical alert when 10 or more results for your query are returned:
   - Response Time
     Enter the Warning and Critical Response Time thresholds. For more information, see Configuring Warning and Critical Thresholds.
3. To save the data from the thresholds for graphing or reporting, click the *Save for Graphing* checkbox beside any of the options listed in step 2.
4. Complete the following settings:
   - Timing Settings (see Adding Monitor Timing Settings Information for more information)
   - Alert Settings (see Monitor Alert Settings for more information)
   - Monitoring Period settings (see Monitor Timing Settings for more information)
   - Alert Profile settings (see Alert Profiles for more information)
   - Action Profile settings (see Action Profiles for more information)
5. Click *Finish.*

Live Splunk Listener

Live Splunks are scheduled searches of Splunk queries that are saved on the Splunk server. A Live Splunk automatically runs a search, can initiate an alert, and can perform actions based on that alert. You can, for example, set up a Live Splunk to search for all critical error conditions.

The Live Splunk Listener monitor enables you to capture the information generated by a Live Splunk (from Splunk 4.x only). This monitor is very similar to the External Check monitor, and uses scripts that are bundled with *up.time* (found in the */scripts* subdirectory) to return Live Splunk information to the Monitoring Station.

To use this monitor, you must first modify the two Splunk scripts that are incpuded with *up.time* :

- *alertUptimeStatusHandler.sh*
- *alertUptime.py*

This pair of scripts take the following options:

- *--message*
  A message that will be returned to the *up.time* Monitoring Station. For example, if the Live Splunk is configured to search for warning conditions, you can enter the message *"Changed to WARN"* .
- *--status*
  The script can return the following status codes:
  - *0* - OK
    The services are functioning properly.
  - *1* - Warning
    There is a potential problem with one of more of the services being monitored.
  - *2* - Critical
    There is a critical problem with one or more of the services being monitored.
  - *3* - Unknown
    There is an error in the configuration of the monitor itself, or *up.time* cannot execute the service check.
- *--monitor*
  The name of the *up.time* monitor to which the information from the Live Splunk will be directed.

The following is an example of the script with all of its options specified:

```
alertUptimeStatusHandler.sh --message="sendmail has some traffic going through new command!"
                            --status=2 --monitorName="Live Splunk"
```

up.time captures the output from the script, which appears in the service status section of the *Global Scan* panel (see Understanding the Status of Services). The *up.time* monitoring framework picks up any error codes and triggers the appropriate monitoring action.

Before You Begin

Before you can monitor Live Splunks generated on a Splunk server, you must do the following:

1. Edit the *alertUptime.py* script to point to the *up.time* Monitoring Station:
   - Navigate to the */scripts* directory on the Monitoring Station.
   - Open the file *alertUptime.py* in a text editor.
   - Find the following entry in the file:

     ```
     host = "uptime-host"
     port = "9996"
     ```

    - Change the values for *host* and *port* to the host name and port of the Monitoring Station.
    - Save and close the file.
2. Edit the *alertUptimeStatusHandler.sh* script to configure how the Live Splunk is reported on the Monitoring Station:
    - Open *alertUptimeStatusHandler.sh* in a text editor (found in the */scripts* directory on the Monitoring Station).
    - For the *message* option, enter a diagnostic message that accompanies a Live Splunk captured by the *up.time* service monitor.
    - For the *status* option, enter the status of the service being monitored.
    - For the *monitorName* option, enter the name of the service monitor that is listening to the Live Splunk.
    - Save and close the file.
3. Copy the *alertUptimeStatusHandler.sh* and *alertUptime.py* scripts from the Monitoring Station's */scripts* directory to the */data/splunk/bin/scripts* directory on the Splunk server.
4. Configure a Live Splunk. For information on configuring Live Splunks, see the Splunk user manual.
   When setting up your Live Splunk, select the *Run the shell script option* on the configuration page. Then, enter the path to *alertUptimeStatusHandler.sh* in the field.

Configuring the Live Splunk Listener Monitor

To configure a Live Splunk Listener monitor, do the following:

1. Complete the monitor information fields.
   To learn how to configure monitor information fields, see Monitor Identification.
2. Complete the following settings:
    - Timing Settings (see Adding Monitor Timing Settings Information for more information)
    - Alert Settings (see Monitor Alert Settings for more information)
    - Monitoring Period settings (see Monitor Timing Settings for more information)
    - Alert Profile settings (see Alert Profiles for more information)
    - Action Profile settings (see Action Profiles for more information)
3. Click *Finish*.