

Configuring and Managing up.time

- Overview
 - Modifying up.time Config Panel Settings
 - Modifying uptime.conf File Settings
 - Stopping and Restarting up.time Services
- Interfacing with up.time
 - Database Settings
 - Monitoring Station Web Server
 - SMTP Server
 - Configuring Global Data Collection Methods
 - Global SNMP Configuration Settings
 - RSS Feed Settings
 - VMware vCenter Orchestrator Integration
 - Web Application Monitor Proxy Settings
 - Remote Reporting Settings
 - User Interface Instance Settings
 - Scrutinizer Settings
 - Splunk Settings
- Archiving the DataStore
 - Archive Categories
 - Configuring an Archive Policy
 - Restoring Archived Data
 - Exporting and Importing the DataStore
- up.time Diagnosis
 - Logs
 - Problem Reports
- up.time Measurement Tuning
 - Service Monitor Thread Counts
 - Java Heap Size
 - Status Thresholds
 - Platform Performance Gatherer Check Intervals
- Report Storage Options
 - Changing the Number of Days Reports Are Cached
 - Changing the Published Report Location
- Resource Usage Report Generation
- Monitoring Station Interface Changes
 - Status Alert Acknowledgement
 - 3D Graphs
 - Custom Dashboard Tabs
- License Information

Overview

up.time includes user-definable parameters that can control some aspects of its behavior including the following:

- Database Settings
- Mail Server Settings
- Global Scan threshold settings
- Resource Scan threshold settings
- Proxy settings
- Remote reporting settings
- RSS feed settings
- Splunk integration settings
- Web monitor settings

From a configuration perspective, there are two types of parameters:

- parameters whose modification does not require a restart of the Core service (also known as the *up.time* Data Collector service); these parameters can be modified in *up.time* , on the *Config* panel
- parameters whose modification requires a restart of the Core service; these parameters are found in the *uptime.conf* file

Modifying up.time Config Panel Settings

Configuration parameters that are not directly tied to, thus do not require a restart of, the *up.time* Core service can be modified directly in the *up.time* GUI (shown below):

In general, to edit these configuration settings in the *up.time* interface, do the following:

On the *up.time* tool bar, click *Config* .

1. In the *Tree* panel, click *up.time Configuration* .
2. Enter the configuration variable and new value.
3. Click *Update* to save your changes.

 Only the variables whose default values have been modified appear in `up.time` Configuration.

Modifying `uptime.conf` File Settings

Configuration parameters that are directly tied to the `up.time` Core service are found in the `uptime.conf` file. `uptime.conf` is a text file that you can modify in any text editor, and can be found in the root `up.time` installation directory.

In addition to the `up.time` database, `uptime.conf` parameters affect a variety of `up.time` behavior.

 Not all of the settings listed in this section will necessarily be found in your particular `uptime.conf` file.

Stopping and Restarting `up.time` Services

In addition to the Web interface, the `up.time` Monitoring Station consists of the following services:

- DataStore
- Web server
- Data Collector (also called the Core)

These services run in the background and start automatically after the operating system on the server hosting `up.time` starts. However, system administrators may need to stop the `up.time` services - for example, before making configuration changes to the `uptime.conf` file, performing an upgrade, or archiving the DataStore.

Stopping the `up.time` Services

To stop the `up.time` services in Windows, do the following:

1. Select *Start > Control Panel*.
2. Double click *Administrative Tools*, and then double click *Services*.
3. In the *Services* window, find the following entries and click *Stop the service*:
 - `up.time` Web Server
 - `up.time` Data Collector
 - `up.time` Data Store

To stop the `up.time` services on Solaris or Linux, do the following:

1. Log into the Monitoring Station as user `root`.
2. Type the following command to stop the Web server:
`/etc/init.d/uptime_httpd stop`
3. Type the following command to stop the Data Collector:
`/etc/init.d/uptime_core stop`
4. Type the following command to stop the database:
`/etc/init.d/uptime_datastore stop`

Starting the `up.time` Services

To restart the `up.time` services in Windows, do the following:

1. Select *Start > Control Panel*.
2. Double click *Administrative Tools*, and then double click *Services*.
3. In the *Services* window, find the following entries and click *Start the service*:
 - `up.time` Data Store
 - `up.time` Data Collector
 - `up.time` Web Server

To restart the `up.time` services on Solaris or Linux, do the following:

1. At the command line, log into the Monitoring Station as user `root`.
2. Type the following command to start the database:
`/etc/init.d/uptime_datastore start`
3. Type the following command to start the Data Collector:
`/etc/init.d/uptime_core start`
4. Type the following command to start the Web server:
`/etc/init.d/uptime_httpd start`

Interfacing with `up.time`

Some of the Monitoring Station's features require integration with other elements that make up your infrastructure. In some cases configuration is mandatory (e.g., an SMTP server will need to have been set at the time of installation), while in others it is required only when particular `up.time` features are used (e.g., using the Web Application Transaction monitor requires you to provide `up.time` with your proxy server settings). The following sections outline how to configure `up.time` to communicate with servers and databases.

Database Settings

The database settings are used to determine how up.time communicates with the DataStore, and how it will perform a database health check. The following are the database-related parameters in the `uptime.conf` file.

Connection	
<code>dbType</code>	<p>The type of database that is being used to store data from up.time. The default value is <code>mysql</code>. You can also specify <code>mssql</code> and <code>oracle</code> to use SQL Server and Oracle, respectively.</p> <p>By default, up.time uses a JDBC (Java Database Connectivity) driver, and the driver used to connect to the DataStore corresponds to the database selected:</p> <ul style="list-style-type: none">◦ <code>com.mysql.jdbc.Driver</code> (for MySQL)◦ <code>net.sourceforge.jtds.jdbc.Driver</code> (for Microsoft SQL Server)◦ <code>oracle.jdbc.OracleDriver</code> (for Oracle)
<code>dbHostname</code>	The name of the system on which the database is running. The default is <code>localhost</code> .
<code>dbPort</code>	The port on which the database is listening. The default is <code>3308</code> .
<code>dbName</code>	The name of the database. The default is <code>uptime</code> .
<code>dbUsername</code>	The name of the default database user. The default is <code>uptime</code> .
<code>dbPassword</code>	The password for the default database user. The default is <code>uptime</code> .
<code>dbJdbcProperty</code>	<p>Optional property-and-value pairs to append to the JDBC database URL. Note that only MySQL and Microsoft SQL Server supports URL properties, so this setting will do nothing if you are using Oracle. The value of the <code>dbJdbcProperty</code> parameter in <code>uptime.conf</code> should be verbatim as that which would be manually added to the URL. The exact format depends on the database type. Consider the following examples:</p> <ul style="list-style-type: none">◦ <code>dbJdbcProperty=instance=sqlserver;ssl=request</code> (for MS SQL Server)◦ <code>dbJdbcProperty=instance=mysql&useSSL=true</code> (for MySQL)
Health	
<code>datastoreHealthCheckInterval</code>	When this parameter is enabled with a non-zero value, up.time performs a database health check. The value provided is the frequency of the check, in seconds. The default is <code>5</code> .
<code>datastoreHealthCheckTimeLimit</code>	When the health check time limit has been reached (the value unit is seconds, and the default is <code>300</code>), up.time considers the database down. The Data Collector service is stopped, and administrators that are members of the SysAdmin user group are sent an email.
Performance	
<code>connectionPoolMaximum</code>	The maximum number of connections that are allowed to the DataStore. Setting this option to a lower number will help increase the performance of up.time.
<code>connectionPoolMaxIdleTime</code>	(c3p0 library) Sets the amount of time a connection can be idle before it is closed. This parameter should only be modified with the assistance of uptime software Customer Support.
<code>connectionPoolNumHelperThreads</code>	(c3p0 library) Sets the number of helper threads that can improve the performance of slow JDBC operations. This parameter should only be modified with the assistance of uptime software Customer Support.

Changing the DataStore Database

The up.time DataStore is first linked to a database during the installation process, and contains important historical performance data that has since been collected. Linking the DataStore to a new database will result in lost data unless you properly migrate your data to the new database. As such, changing the DataStore's database should be done only after some consideration and planning.

In cases where you would like to migrate the database (e.g., from the default up.time MySQL implementation to Oracle) or move the DataStore to a different system from the Monitoring Station, you will modify the aforementioned database values in the `uptime.conf` file. Note that the modification of these values is one of a series of steps. Refer to the uptime software [Knowledge Base](#) for more information on migrating your DataStore.

Monitoring Station Web Server

Monitoring Stations include a Web server component that drives the user interface. Any Monitoring Station that is accessed by users or administrators requires a URL. The Web address used to access the Monitoring Station is configured through the following `uptime.conf` parameter:

```
httpContext = http://<hostname>:<port>
```

- `<hostname>` is the host name of the server on which up.time is running (e.g., `localhost`)

- `<port>` is the port on which the *up.time* Web server is listening for requests (e.g., 9999); you can optionally omit the port number

If the *up.time* interface is being accessed via SSL, the value for this parameter should be stated as *https* instead of *http*.

SMTP Server

up.time uses a mail server to send alerts and reports to its users. After installing *up.time* for the first time, the administrator was asked to enter SMTP server information. These initial values can be modified in the *Mail Servers* configuration panel.

Modifying the SMTP Server Used by *up.time*

To configure *up.time*'s mail server, do the following:

1. On the *up.time* tool bar, click *Config*.
2. In the *Tree* panel, click *Mail Servers*.
3. In the sub panel, click *Edit Configuration*.
4. Type the name of the mail server in the *SMTP Server* field.
This value was set the first time the *up.time* administrator logged in after installation; the default value is the name of the host on which the Monitoring Station was installed at that time.
The name of the server could follow the "*smtp.<domain_name>*" convention, or could be its host name or IP address.
5. Optionally, enter the port used by the mail server in the *SMTP Port* field.
6. In the *SMTP Sender* field, enter the email address that *up.time* uses to send alert notifications and reports.
This value was set the first time the *up.time* administrator logged in after installation, and should be set to your domain (e.g., *admin@mail.uptimesoftware.com*).
A sender's name can be encapsulated with double quotes, in which case, the email address is encapsulated with angled brackets: "*uptime administrator*" `<admin@uptimesoftware.com>`
7. In the *SMTP HELO String* field, enter the string that identifies the domain from which a message is being sent.
For example, *uptimesoftware.com*.
8. In the *SMTP User* field, enter the user name that is used to authenticate connections with the SMTP server.
9. In the *SMTP Password* field, enter the password that is used to authenticate connections.
10. Click *Save*.
The edit window closes, and you are returned to the *Mail Server Configuration* panel.
11. To test the mail server configuration, click the *Test Configuration* button.
The Monitoring Station will try to send an email message containing the configuration information to the email address of the *up.time* administrator. If an error message appears in the subpanel, edit and then re-test the configuration.

Configuring Global Data Collection Methods

A Windows-based Element can retrieve metric data either through the *up.time* Agent, or via WMI (see [Agentless WMI Systems](#) for more information). You can configure details for either method at a global level, in the form of agent connection information or WMI access credentials. Having global details defined simplifies individual Element configuration, and also allows you to switch the data collection method for multiple Windows Elements, at once, as a group.

When a system is part of the *up.time* inventory, its data collection method is either configured to be based on an *Agent*, or a *WMI Agentless* method. This configuration option is set when the system is first added as an Element. If agent and WMI details have been globally defined, when adding the Element, you will be able to *Use the up.time Agent Global Configuration*, or use *WMI Global Credentials* to skip configuration steps.

Once configured, this data collection method can later be switched from an agent-based, to agentless method, or vice versa. Although this change can be made on a per-Element basis, multiple Elements can also be switched in a single batch. In the latter case, the data collection method must be globally defined.

To configure data collection methods globally, you can provide information for either the *up.time* Agent, or your organization's WMI credentials, or both. Note that batches of Elements can only be converted to a particular data collection source when that method has been globally configured in the *Global Element Settings* panel.

Configuring Global WMI Credentials

To provide WMI credentials that can be used to switch Windows Elements from agent-based data collection:

1. On the *up.time* tool bar, click *Config*.
2. In the *Tree* panel, click *Global Element Settings*.
3. In the *WMI Agentless Global Credentials* sub panel, click *Edit Configuration*.
4. Enter the *Windows Domain* in which WMI has been implemented.
5. In the *Username* field, enter the user ID that has administrative access to WMI on the Windows domain.
6. In the *Password* field, enter the password for the WMI account.
7. Click *Save* to retain your changes.
8. Click *Test Configuration* to ensure the credentials provided are correct.

Configuring a Global *up.time* Agent Configuration

To provide *up.time* Agent information that can be used to switch Windows Elements from agentless, WMI-based data collection, do the following:

1. On the *up.time* tool bar, click *Config*.
2. In the *Tree* panel, click *Global Element Settings*.
3. In the *up.time Agent Global Configuration* sub panel, click *Edit Configuration*.
4. Enter the *Agent Port Number*, indicating the port the *up.time* Agents use to communicate with the *up.time* Monitoring Station.



The port number entered reflects what the *up.time* Agents are configured to use; this setting does not modify the agent-side configuration.

5. Select the Use SSL check box if the agents securely communicate with the Monitoring Station using SSL.
6. Click *Save* to retain your changes.
7. Click *Test Configuration* to ensure the information provided is correct.

Global SNMP Configuration Settings

When you add a network device to *up.time*, as part of the configuration process, you must provide details about how SNMP has been configured to communicate with and manage other devices on the network. These details describe, among other things, the SNMP protocol being used, and encryption methods.

By default, SNMP-specific settings are inputted for each network-type device, as they are added to *up.time*. To facilitate this process, your network's SNMP settings can be defined globally in the *Global Element Settings* panel.

Global SNMP Settings

The following SNMP settings are used to configure network-related Elements, and can be defined globally.

SNMP Version	The SNMP version the network device and your network are using.
SNMP Port	The port on which network devices have been configured to listen for SNMP messages.
Read Community	A string that acts like a user ID or password, giving you access to the network device instance. Common read communities are "public", enabling you to retrieve read-only information from the device, and "private", enabling you to access all information on the device.
Username	The name that is required to connect to the network device.
Authentication Password	The password that is required to connect to the network device.
Authentication Method	This option determines how encrypted information travelling between the network device and up.time will be authenticated: <ul style="list-style-type: none"> • MD5: A widely-used method for creating digital signatures used to authenticate and verify the integrity of data. • SHA: A secure method of creating digital signatures. SHA is considered the successor of MD5 and is widely used with network and Internet data transfer protocols.
Privacy Password	The password that will be used to encrypt information travelling between the network device and up.time.
Privacy Type	From the list, select an option that will determine how information travelling between the network device and up.time will be encrypted: <ul style="list-style-type: none"> • DES: An older method used to encrypt information. • AES: The successor to DES, which is used with a variety of software that require encryption including SSL servers.
Pingable Node	This specifies whether up.time can contact the network device using the ping utility. There are scenarios in which you might not want the network device to be pingable (e.g., you have a firewall in place). Before enabling this option, you should try to contact it using the ping utility. If you cannot ping it, ensure this check box is left cleared. Then, change the default host check for the network device. See Changing Host Checks for more information.
Exports Data to Scrutinizer	If Scrutinizer has been integrated with up.time, and is also receiving NetFlow data from the network device, use this option. You will then be able to call a Scrutinizer instance directly from the network device's Graphing tab in up.time.

Globally Defining SNMP Version 2 Settings

To globally define the SNMP version 2 settings used to communicate with network devices on your network, do the following:

1. On the *up.time* tool bar, click *Config*.
2. In the *Tree* panel, click *Global Element Settings*.
3. In the *SNMP Global Credentials* sub panel, click *Edit Configuration*.
4. Enter the *SNMP Port* on which your network devices are listening.
5. Enter the *Read Community* used to access network devices (e.g., "public" or "private").
6. Indicate whether *up.time* can contact the network device with the ping utility by selecting or clearing *Is Node Pingable?* checkbox.
7. Click *Save* to retain your changes.
8. Click *Test Configuration* to ensure the information provided is correct.

Globally Defining SNMP Version 3 Settings

To globally define the SNMP version 2 settings used to communicate with network devices on your network, do the following:

1. On the *up.time* tool bar, click *Config* .
2. In the *Tree* panel, click *Global Element Settings* .
3. In the *SNMP Global Credentials* sub panel, click *Edit Configuration* .
4. For the *SNMP Version* , select *v3* .
The configuration fields relevant to that SNMP version will appear.
5. Enter the *SNMP Port* on which your network devices are listening.
6. Enter the *Username* used to connect to network devices.
7. Optionally enter the *Authentication Password* required to connect to network devices.
8. Indicate the Authentication Method used for encryption.
If no password is provided, the authentication method is ignored.
9. Optionally enter the *Privacy Password* used to encrypt communication between network devices and *up.time* .
10. Indicate the *Privacy Type* used for encryption.
If no password is provided, the authentication method is ignored.



You can set both the authentication and password types, only one of them, or neither.

11. Indicate whether *up.time* can contact the network device with the ping utility by selecting or clearing *Is Node Pingable?* checkbox.
12. Click *Save* to retain your changes.
13. Click *Test Configuration* to ensure the information provided is correct.

RSS Feed Settings

up.time displays a list of recent knowledge base articles in the *My Portal* panel. This list is fed to the *My Portal* panel via RSS (Really Simple Syndication, a method for delivering summaries of and links to Web content). Clicking the title of an article opens it in your Web browser.

By default, RSS feeds are drawn directly from the uptime software Support Portal without the use of proxy server information. If your Monitoring Station accesses the Internet through one, feeds will most likely not be available, and the following message will appear in the *My Portal* panel:

You can change the RSS feed settings to point to the proxy server rather than directly to the uptime software Web site by manually inputting settings in the *up.time Configuration* panel, as outlined in [Modifying up.time Config Panel Settings](#)
[Changing Proxy Server Information for RSS Feeds](#)

You can manually configure the settings for RSS feeds through the following parameters (default values, if applicable, are shown):

- *rssFeedUrl=http://support.uptimesoftware.com/rss/kb.xml*

The URL of the RSS feed.

- *httpProxyHost*

The host name of the proxy server that the Monitoring Station uses to access the Internet.

- *httpProxyPort*

The port through which the Monitoring Station communicates with the proxy server.

- *httpProxyUsername*

The user name required to use the proxy server.

- *httpProxyPassword*

The password required to use the proxy server.

VMware vCenter Orchestrator Integration

Administrators can configure Action Profiles to automatically carry out tasks in the event of an *up.time* alert. One such task is the initiation of contact with VMware vCenter Orchestrator, and the execution of a workflow. To have access to this functionality, *up.time* needs to know how to communicate with Orchestrator.

For information about Action Profiles and VMware vCenter Orchestrator, see [Action Profiles](#)

[Integrating up.time with VMware vCenter Orchestrator](#)

To configure *up.time* integration with Orchestrator to execute workflows, do the following:

1. On the *up.time* tool bar, click *Config* .
2. In the *Tree* panel, click *VMware vCenter Orchestrator* .
3. In the sub panel, click *Edit Configuration* .
4. Ensure the *VMware Orchestrator Enabled* check box is selected.
5. In the *VMware Orchestrator Server* field, enter the host name of, or IP address assigned to the Orchestrator server when it was configured.
6. In the *VMware Orchestrator Port* field, enter the port the Orchestrator server was configured to use in order to communicate with other systems.
7. Optionally select the *Use SSL* check box if Orchestrator was configured to use an SSL certificate.
8. Enter the *Username* and *Password* of an appropriate user account on the Orchestrator server.
For proper integration, an Orchestrator account with View and Execute permissions is required.
9. Click *Save* .
The configuration window closes, and you are returned to the *VMware vCenter Orchestrator Configuration* panel.

- To ensure the settings you provided are correct, click the *Test Configuration* button. The Monitoring Station will try to communicate with the VMware vCenter Orchestrator server. If an error message appears in the subpanel, edit and then re-test the configuration.

Web Application Monitor Proxy Settings

When the Web Application Transaction monitor is recording a user session on an external site, it is intercepting URLs by acting as your browser's proxy. For the monitor to do this, you must replace your organization's proxy server information with the Web Application Transaction monitor in your browser settings. In order for the monitor to access the Internet, you must provide your proxy settings in *up.time*.

This monitor-specific proxy information is used during transaction recording; during session playback, the proxy normally used by *up.time* (defined by the *ht tpProxy** settings) is used.

For more information about the Web Application Transaction monitor, see [Web Application Transactions](#).

You can change *up.time*'s proxy server configuration by manually inputting settings in the *up.time Configuration* panel, as outlined in [Modifying up.time Config Panel Settings](#)

Changing Proxy Server Information for *up.time*

You can configure the proxy server settings used by *up.time* when running the Web Application Transaction monitor with the following parameters:

- webmonitor.proxyHost*

The host name of the proxy server that the Web Application Transaction monitor uses to access the Internet during transaction recording.

- webmonitor.proxyPort*

The port through which the Web Application Transaction monitor communicates with the proxy server during transaction recording.

Remote Reporting Settings

If you are using a reporting instance (an *up.time* instance that only generates and serves reports), the remote reporting settings enable you to specify the location of the reporting instance, and the port on which it is listening.

Modifying the Remote Reporting Server Settings

To configure the remote reporting instance used by *up.time*, do the following:

- On the *up.time* tool bar, click *Config*.
- In the *Tree* panel, click *Remote Reporting*.
- In the sub panel, click *Edit Configuration*.
- Ensure the *Reporting Instance Enabled* check box has been selected.
- In the *Remote Reporting Server* field, enter the host name or IP address of the server on which the remote reporting instance is found.
- Enter the port used to communicate with the server.
- Click *Save*.

The edit window closes, and you are returned to the *Remote Reporting Instance Configuration* panel.

- To test the remote reporting server configuration, click *Test Configuration*.

A pop-up window appears, indicating whether *up.time* was able to connect to the remote reporting instance. If an error message is displayed, correct your configuration and re-test it.

Note that the modification of these values is one of a series of steps performed to correctly set up a remote reporting instance. Refer to the Knowledge Base article entitled "Setting up a reporting instance" for more information.

User Interface Instance Settings

A UI instance is an *up.time* installation that does not perform any data collection tasks, and is primarily used for real-time monitoring and report generation. When there are many *up.time* users who do not need to perform full administrative tasks, UI instances can divert traffic from a core Monitoring Station implementation, improving data-collection performance and UI responsiveness.

You can manually configure UI instance settings with the following *uptime.conf* parameters:

Parameter	Description
<i>uiOnlyInstance</i>	enables the Monitoring Station as a user interface instance
<i>uiOnlyInstance.monitoringStationHost</i>	the host name or IP address of the <i>up.time</i> Monitoring Station that is performing data collection, and to which this UI instance will connect
<i>uiOnlyInstance.monitoringStationCommandPort</i>	the port through which the UI instance can communicate with the core data-collecting Monitoring Station; in most cases, this port should be 9996, otherwise the UI instance will not communicate properly with the core Monitoring Station

Creating an *up.time* UI Instance

To create a UI instance, do the following:

- Install a new instance of *up.time* on another server. You can use the standard *up.time* installer; when prompted to provide DataStore configuration information, ensure the UI instance has the same database settings as the core Monitoring Station (see [Database Settings](#) for more information).

2. Enable the newly installed `up.time` as a UI instance by editing the following `uptime.conf` parameters:
 - `uiOnlyInstance=true`
 - `uiOnlyInstance.monitoringStationHost=<hostname>`
`hostname` is the hostname or IP address of the core, data-collecting Monitoring Station, with which this UI instance will communicate
 - `uiOnlyInstance.monitoringStationCommandPort=9996`
the port through which the UI instance can communicate with the core Monitoring Station



Unless your core Monitoring Station has been customized, it is configured to use port 9996 to communicate with a UI instance. If you wish to use a different port, you must ensure matching `uptime.conf` values exist on both the UI instance and core Monitoring Station. Contact uptime software Support for more information.

3. Shut down the Data Collector service on the UI instance.
4. Ensure you are logged in with a domain account, and that this account has access to the `<installDirectory>/gadgets` directory on the Monitoring Station.
5. To accommodate sharing user-created gadgets, on the core Monitoring Station system, make the `<installDirectory>/gadgets` directory accessible by the UI instance system.
How you make the `/gadgets` directory accessible depends on the Monitoring Station platform:
 - On Linux, you can use NFS to share the directory on the core Monitoring Station, then mount it on the UI instance
 - On Windows, you can use the `mklink` command to create a symbolic link on the UI instance that points to the `/gadgets` directory on the core Monitoring Station, such as in the following example:

```
mklink /D "C:\Program Files\uptime software\uptime\gadgets" "\\host\gadgets"
```

You will most likely need to modify sharing and security permissions for the directory.
6. Restart the Data Collector service on the UI instance.

Scrutinizer Settings

Scrutinizer is a NetFlow analyzer that can be installed to monitor network traffic managed by compatible switches and routers. Scrutinizer can be integrated with the **Global Scan** dashboard, as well as `up.time`'s graph generation for node-type Elements, and other hosts that are also monitored with Scrutinizer.

In order to access Scrutinizer, `up.time` needs to be pointed to your installation.

[Modifying the Scrutinizer Settings](#)

You can configure Scrutinizer's integration with `up.time` through the following parameters:

Parameter	Description
<code>netflow.enabled</code>	determines whether Scrutinizer is integrated with the Monitoring Station
<code>netflow.hostname</code>	the host name or IP address of your Scrutinizer installation
<code>netflow.port</code>	the HTTP port through which Scrutinizer sends and receives communication
<code>netflow.username</code>	the username required to log in to Scrutinizer
<code>netflow.password</code>	the password required to log in to Scrutinizer

Splunk Settings

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 datacenter.

When values are provided for the Splunk settings listed below, the Splunk icon will appear in the **My Portal** panel beside the names of services that are in WARN or CRIT states. When you click the Splunk icon, you will be automatically logged in to your Splunk search page.

You can change your `up.time`-Splunk integration by manually inputting settings in the **up.time Configuration** panel, as outlined in [Modifying up.time Config Panel Settings](#).

[Changing Splunk Server Information for up.time](#)

You can enable automatic login to the Splunk search page, or modify an existing configuration through the following parameters:

Parameter	Description
<code>splunk.url</code>	the URL of the server on which your Splunk search page is hosted (e.g., <code>http://webportal:8000</code>)
<code>splunk.username</code>	the username required to log in to your Splunk search page
<code>splunk.password</code>	the password required to log in to your Splunk search page
<code>splunk.soapurl</code>	the URL that points to the SOAP management port that Splunk uses to communicate with the splunk daemon (e.g., <code>https://webportal:8089</code>).

In the URL, you must include the port on which the Splunk server listens for requests. See the [Splunk Admin Manual](#) for more information.

Archiving the DataStore

Depending on the amount of disk space available for the continuously growing DataStore, administrators can set an archive policy that determines how many month's worth of data is retained. Old performance data is automatically archived and removed from the DataStore. This archiving procedure works with all databases that are compatible with up.time.

The existing archive policy can be viewed and modified on the **Archive Policy** subpanel, which is accessed from the main **Config** panel. Here, the main archive categories are shown, along with the number of months for which collected data is retained in the DataStore.

Every month, up.time checks the DataStore's entries; data that is older than the limit set in the archive policy are written to XML files. The XML archives use the following format:

```
<table_name>_<date>.xml.gz
```

The archives created reflect the database table structure used to store performance data, as well as the date that the stored data represents:

```
performance_cpu_2006-09-13.xml.gz
```

The DataStore is trimmed and the XML files are compressed and stored in the `/archives` directory.

For example, if you installed up.time in the default location, the path to the archived data will be:

- Linux: `/usr/local/uptime/archives`
- Windows: `C:\Program Files\uptime software\uptime\archives`

Once backed up, archives can be stored offline. If required, they can be temporarily imported into the DataStore.

Archive Categories

The following table lists the statistical categories whose archiving can be configured, along with the corresponding DataStore database table:

Archive Policy Category	Database Table
Overall CPU/Memory	performance_cpu
Multi-CPU	performance_aggregate
Detailed Process	performance_psinfo
Disk Performance	performance_disk
File System Capacity	performance_fscap
Network	performance_network
User Information	performance_who
Volume Manager	performance_vxvol
Retained Data	erdc_int_data erdc_decimal_data erdc_string_data
vSphere Performance Data	vmware_perf_aggregate vmware_perf_cluster vmware_perf_datastore_usage vmware_perf_datastore_vm_usage vmware_perf_disk_rate vmware_perf_entitlement vmware_perf_host_cpu vmware_perf_host_disk_io vmware_perf_host_disk_io_adv vmware_perf_host_network

	vmware_perf_host_power_state vmware_perf_mem vmware_perf_mem_advanced vmware_perf_network_rate vmware_perf_vm_cpu vmware_perf_vm_disk_io vmware_perf_vm_network vmware_perf_vm_power_state vmware_perf_vm_storage_usage vmware_perf_vm_vcpu vmware_perf_watts vsync_update
vSphere Inventory Updates	virtual_inventory_update vmware_event table
Network Device Performance Data	net_device_perf_ping net_device_perf_port

Configuring an Archive Policy

To set an archive policy, do the following:

1. On the up.time tool bar, click **Config**.
2. In the tree panel, click **Archive Policy**.
3. For the following categories, specify the number of months worth of data that will be retained in the DataStore before being removed and archived:
 - Overall CPU/Memory Statistics
 - Multi-CPU Statistics
 - Detailed Process Statistics
 - Disk Performance Statistics
 - File System Capacity Statistics
 - Network Statistics
 - User Information Statistics
 - Volume Manager Statistics
 - Retained Data
 - vSphere Performance Data
 - vSphere Inventory Data
 - Network Device Performance Data
4. Ensure the **Enable Archiving** checkbox is selected.
5. Click **Set Archive Policy**.
6. Optionally, you can click the **Archive Now** button to immediately create archives of the data in your DataStore. up.time will check the DataStore entries and archiving anything that is older than the limits you have configured.

Restoring Archived Data

If you need to generate graphs or reports on older data that have already been archived and is no longer in the DataStore, you can import specific archives using the `restorearchive` command line utility. The command's parameters allow you to import archives in the following manner:

- a single archive that represents a specific archive category and date; the collected data for each archive category and 24-hour period is exported to individual XML files
- all archives for a specific date (i.e., 24-hour period)

Importing Archived Data into the DataStore

To import archived data into the DataStore, do the following:

1. Ensure the database hosting the DataStore is running.
2. At the command line, navigate to the up.time `/scripts` folder.
For example, if you installed the Monitoring Station in the default location on a Windows system, navigate to the following folder:
`C:\Program Files\uptime software\uptime\scripts\`
3. Run the `restorearchive` command with one or more of the following options:
 - `-f <filename>`
Imports a single file (i.e., an archive category's data for a single date). You must specify the full path to the file name.

- `-d <date>`
Imports all files with the specified date (in YYYY-MM-DD format).
- `-D <directory>`
The directory containing the archived files. Note that you must specify this option when using the `-d` option.
- `-c <directory>`
The full directory path to the `uptime.conf` file.

For example, the following command would import all of the data archived on September 18, 2006, which are located in the default directory for archived data:

```
restorearchive -d 2006-09-18 -D /usr/local/uptime/archives/ -c /usr/local/uptime
```



If you have deployed up.time UI instances, ensure you always run command-line scripts such as `restorearchive` on the primary Monitoring Station that is actually performing data collection.

Exporting and Importing the DataStore

In cases where you need to perform a wholesale backup of the existing DataStore (e.g., migrating your DataStore to another database), up.time includes two command-line utilities:

- `fulldatabasedump`: creates a compressed XML file of the contents of your DataStore
- `fulldatabaseimport`: imports the archived data back into your DataStore

Both utilities work with all of the databases supported by up.time.

Archiving the DataStore

To archive your DataStore, do the following:

1. Ensure the database hosting the DataStore is running.
2. Stop the up.time Data Collector service (i.e., the up.time Core).
See [Stopping and Restarting up.time Services](#) for more information.
3. At the command line, navigate to the up.time `/scripts` folder.
For example, if you installed the Monitoring Station in the default location on a Windows system, navigate to the following folder:
`C:\Program Files\uptime software\uptime\scripts\`
4. Run the following command:
`fulldatabasedump`



If you have deployed up.time UI instances, ensure you always run command-line scripts such as `fulldatabasedump` or `fulldatabaseimport` on the primary Monitoring Station that is actually performing data collection.

Depending on the size of your DataStore, this process can take anywhere from several minutes to several hours.

The utility creates the file `uptimedump_YYYY-MM-DD.xml.gz` (e.g., `uptimedump_2007-01-02.xml.gz`). This file is saved in up.time's root installation directory.

5. Restart the up.time Data Collector service.

Restoring the DataStore

To restore your DataStore, do the following:

1. Ensure the database hosting the DataStore is running.
2. Use the `resetdb` utility with the `really` and `nodata` options to delete, then recreate the database structure that is used by up.time by running one of the following commands:
 - Linux: `usr/local/uptime/resetdb --nodata really`
 - Windows: `C:\Program Files\uptime software\uptime\resetdb --nodata really`



If you have set up UI instances of up.time, ensure you always run command-line scripts such as `resetdb` on the primary Monitoring Station that is actually performing data collection.

3. Run the following command:

```
fulldatabaseimport path/<fileToImport>.xml.gz
```

Where `path/<fileToImport>.xml.gz` is the path to, and file name of, the archived contents of your DataStore. For example, to import an archive that is located in up.time's root installation directory, you would enter the following:

```
fulldatabaseimport uptimedump_2007-01-02.xml.gz
```

up.time Diagnosis

up.time's logs can assist you with diagnostic steps that you may need to perform should you encounter problems. Problem reports can be generated for up.time Customer Support if further analysis is required.

All up.time logs are written to the `/logs` directory, and problem reports to the `/GUI` directory, both of which are found in the up.time installation directory:

- Linux: `/usr/local/uptime/`
- Windows: `C:\Program Files\uptime software\uptime`

Logs

The following logs are available for troubleshooting. Depending on the type of investigation, output from multiple logs can be correlated.

Log Name	Description and Uses	uptime.conf parameter and values
uptime.log	<p>This is the base up.time log. System events are automatically recorded to these weekly logs, which follow the <code>uptime.log.<year>-<week>.log</code> naming format.</p> <p>You can determine the type of system information up.time writes to the log (ranging from verbose, to informational, to critical errors) by setting the logging level. The default setting, <code>INFO</code>, essentially logs all system event types that are higher than the service or thread level (which are logged at the <code>DEBUG</code> setting). To reduce the number of log entries, you can limit logging to events with a higher level of severity, from <code>WARN</code> to <code>FATAL</code>. Note that each severity level is a subset of higher levels (e.g., setting <code>loggingLevel</code> to <code>WARN</code> means any <code>WARN</code>-, <code>ERROR</code>- or <code>FATAL</code>-level events are written to the log).</p>	<p>loggingLevel=</p> <ul style="list-style-type: none"> • <code>DEBUG</code> • <code>INFO</code> (default) • <code>WARN</code> • <code>ERROR</code> • <code>FATAL</code> • <code>ALL</code> • <code>OFF</code>
uptime_diagnostics.log	<p>This log is similar to the <code>uptime.log</code>, but has a more detailed breakdown of system information to assist with troubleshooting. Additional information includes the name of associated thread, the name of the up.time component that logged the event, Element details, and if applicable, monitor, Element, VMware, user details.</p> <p>Like the <code>uptime.log</code>, the number of log entries is also set by the <code>loggingLevel</code> parameter.</p>	<p>loggingLevel=</p> <ul style="list-style-type: none"> • <code>DEBUG</code> • <code>INFO</code> (default) • <code>WARN</code> • <code>ERROR</code> • <code>FATAL</code> • <code>ALL</code> • <code>OFF</code>
uptime_exceptions.log	All <code>DEBUG</code> -level Java runtime exceptions evoked by up.time actions. Full stack traces are channeled to this log to lighten and accompany the core <code>uptime.log</code> and <code>uptime_diagnostics.log</code> files. Use the context marker in the core log to find the exception in this log.	N/A
uptime_console.log	All Java-related command-line feedback based on up.time activity is routed to this log, providing extra information that may not be captured in the standard up.time log.	N/A
audit.log	<p>up.time can record changes to the application's configuration in an audit log, and is essentially a record of which user performed which action, and when.</p> <p>The following is an example of an audit log entry:</p> <pre>2006-02-23 12:28:20,082 - kdawg: ADDSYSTEM [cfgcheck=true, port=9998, number=1, use-ssl=false, systemType=1, hostname=10.1.1.241, displayName=MailMain, systemSystemGroup=1, serviceGroup=, description=, systemSubtype=1]</pre> <p>There are many uses for the audit log. For example, you can use it to track changes to your up.time environment for compliance with your security or local policies. You can also use the audit log to debug problems that may have been introduced into your up.time installation by a specific configuration change; the audit log enables you to determine who made the change and when it took effect.</p> <p>By default, the <code>auditEnabled</code> parameter in <code>uptime.conf</code> is not defined, which means it is effectively disabled.</p>	<p>auditEnabled=</p> <ul style="list-style-type: none"> • <code>yes</code> • <code>no</code>
uptime_access.log	A summary of which up.time access-related actions, mainly database queries, were evoked by which service or user, and the execution time. This database-focused log can be used in conjunction with the more user-focused <code>audit.log</code> .	N/A
thirdparty.log	Aggregation of warnings and errors logged by thirdparty libraries that up.time is using, such as the iReasoning library for SNMP monitoring. Correlating these with the other logs may help with investigation.	N/A
uptime_sql.log	When SQL logging has been enabled with the assistance of up.time software Customer Support, these log shows all SQL queries, with and without execution time, respectively. Queries in <code>uptime_sql.log</code> are listed before execution, which can be compared with the second log to determine conflicts and deadlocks.	contact Customer Support

```
_sql_t
iming.
log
```

Problem Reports

When you encounter a problem with up.time, Customer Support needs a specific set of information to diagnose and fix the problem. up.time can automatically collect this information and compress it in an archive which you can send to Customer Support.

The archive contains the following:

- up.time configuration files
- system information
- log files
- database information and error files
- Java `hs_err_pid` error files
- a listing of the DataStore directory
- optionally, a copy of the configuration data from the DataStore

The archive is saved to the `GUI/problemreports` directory on the Monitoring Station and has a file name with the following format:

```
prYYYYMMDD-HHMMSS.zip
```

- YYYYMMDD is the date on which the report was generated (e.g., 20101224)
- HHMMSS is the time at which the report was generated (e.g., 202306)

Generating a Problem Report

To generate a problem report, do the following:

1. On the up.time tool bar, click **Config**.
2. In the tree panel, click **Problem Reporting**.
3. Configure the **Report Options**:
 - a. Indicate whether to **Include configuration and service monitor status history**, and if so, how many months' worth of data.
 - b. If configuration information is included, indicate whether to also **Include the last hour of performance data**. Adding performance data can result in a significantly larger problem report, requiring an appropriate amount of resources to generate, and time to download. This data, however, can help determine whether your up.time instance is running correctly.
 - c. Indicate whether to include the **database check output** in the problem report. When this option is enabled, up.time runs the `dbchecker` script with the default values on your DataStore. This integrity test allows you to ensure there are no database issues that are part of, or are at the root of the problem. Disable this check box to improve generation performance by skipping the database check.
4. Click the **Generate Report** button. When the report has been generated, it will appear in the **Existing Problem Reports** section below, along with problem reports that have been previously generated.
5. Click the name of the problem report to download it to your local file system, then send the archive to uptime software Customer Support at support@uptimesoftware.com.

up.time Measurement Tuning

In some cases, you can make measurement adjustments to up.time's default values. Changes can be made to the following:

- the number of threads allocated to service monitors
- Java heap size
- status thresholds in the **Resource Scan** and **Global Scan** dashboards
- how often performance and status are checked for monitored hosts

Service Monitor Thread Counts

By default, the number of Java threads allocated to service and performance monitors is 100. This can be modified with the following `uptime.conf` parameter:

```
serviceThreads=100
```

Java Heap Size

By default, the JVM's heap memory is to a maximum of 1 GB. If your monitoring deployment has a lot of service monitors running or reports to generate, you can increase the amount of Java heap memory (e.g., 1.5 GB) to improve performance.



When increasing the Java heap size, ensure your Monitoring Station resources can support the new setting. If the OS does not have the desired amount of memory available exclusively for up.time, the up.time Core service may become unstable and crash, despite starting up successfully.

Adjusting the Java Heap Size

The amount of memory allocated to the JVM can be adjusted by modifying one of the following parameters, depending on your Monitoring Station platform:

On Linux, edit the `<uptimeInstallDir>/uptime.jcnf` file and modify the following:

```
-Xmx1G
```

On Windows, edit the `<uptimeInstallDir>\UptimeDataCollector.inifile` and modify the following, which relates to the Java -Xmx option:

```
vm.heapsize.preferred=1024m
```

Note that the default heap size is measured in gigabytes in the Linux configuration file, and megabytes in the Windows configuration file.

Status Thresholds

The Global Scan threshold settings determine when a cell on the **Global Scan** dashboard changes state to reflect a host's status change: green represents normal status, yellow represents Warning status, and red represents Critical.

The Resource Scan threshold settings determine the size of the gauge ranges on the **Resource Scan** view: green represents normal status, yellow represents Warning status, and red represents Critical status.

You can change the thresholds used to determine status by manually inputting settings in the **up.time Configuration** panel, as outlined in [Modifying up.time Config Panel Settings](#).

Changing Global Scan Threshold Settings

You can modify the **Global Scan** threshold settings through the following parameters (default values are shown):

<code>globalscan.cpu.warn=70</code>	Warning-level status is reported when CPU usage is at 70% or greater
<code>globalscan.cpu.crit=90</code>	Critical-level status is reported when CPU usage is at 90% or greater
<code>globalscan.diskbusy.warn=70</code>	Warning-level status is reported when a disk on the host is busy for 70% or more of a five-minute time frame
<code>globalscan.diskbusy.crit=90</code>	Critical-level status is reported when a disk on the host is busy for 90% or more of a five-minute time frame
<code>globalscan.diskfull.warn=70</code>	Warning-level status is reported when 70% or more of the disk space on the host is used
<code>globalscan.diskfull.crit=90</code>	Critical-level status is reported when 90% or more of the disk space on the host is used
<code>globalscan.swap.warn=70</code>	Warning-level status is reported when 70% or more of the swap space on a disk is in use
<code>globalscan.swap.crit=90</code>	Critical-level status is reported when 90% or more of the swap space on a disk is in use



Changes to **Global Scan** thresholds are not retroactively applied to all Elements; only Elements added after threshold changes will reflect those changes.

Resource Scan Threshold Settings

You can modify the *Resource Scan* threshold settings through the following parameters (default values are shown):

<code>resourcescan.cpu.warn=70</code>	the Warning-level range in the CPU Usage gauge begins at this value (70%), and ends at the Critical-level range
<code>resourcescan.cpu.crit=90</code>	the Critical-level range in the CPU Usage gauge is between this value (90%) and 100%
<code>resourcescan.memory.warn=70</code>	the Warning-level range in the Memory Usage gauge begins at this value (70%), and ends at the Critical-level range
<code>resourcescan.memory.crit=90</code>	the Critical-level range in the Memory Usage gauge is between this value (70%) and 100%
<code>resourcescan.diskbusy.warn=70</code>	the Warning-level range in the Disk Busy gauge begins at this value (70%), and ends at the Critical-level range
<code>resourcescan.diskbusy.crit=90</code>	the Critical-level range in the Disk Busy gauge is between this value (70%) and 100%
<code>resourcescan.diskcapacity.warn=70</code>	the Warning-level range in the Disk Capacity gauge begins at this value (70%), and ends at the Critical-level range
<code>resourcescan.diskcapacity.warn=90</code>	the Critical-level range in the Disk Capacity gauge is between this value (70%) and 100%

Platform Performance Gatherer Check Intervals

The [Platform Performance Gatherer](#) is a core performance monitor that resides on all agent-based Elements.

By default, the Platform Performance Gatherer checks the host Elements' performance levels every 300 seconds. You can change the interval by manually inputting settings in the **up.time Configuration** panel, as outlined in [Modifying up.time Config Panel Settings](#).

Changing the Performance Monitor Check Interval

You can modify the Platform Performance Gatherer check interval through the following **up.time Configuration** parameter (the default value is shown):

```
performanceCheckInterval=300
```



A change to the Platform Performance Gatherer check interval is not retroactively applied to all Elements; only Elements added after an interval change will reflect that change.

Report Storage Options

When an up.time user generates a report, that report is stored in the `/GUI/reportcache` directory; when a scheduled report is automatically generated and published, it is stored in the `/GUI/published` directory. Both of these directory paths are found in the up.time installation directory:

- Linux: `/usr/local/uptime/`
- Windows: `C:\Program Files\uptime software\uptime`

By default, generated reports are cached on the Monitoring Station for 30 days; additionally, the location for published reports is also on the local Monitoring Station file system. Both options can be modified. In the latter case, automatically publishing reports to a publicly accessed directory on the network is an ideal way for non-IT staff to view them. See [Saving Reports to the File System](#) for more information.

Changing the Number of Days Reports Are Cached

You can change a report's expiry time limit by manually inputting settings in the **up.time Configuration** panel, as outlined in [Modifying up.time Config Panel Settings](#).

Change the expiry limit through the following parameter (the default value is shown):

```
reportCacheExpiryDays=30
```

Changing the Published Report Location

This can be modified with the following `uptime.conf` parameter:

```
publishedReportRoot=<location>
```

If the intended published report directory is on a system other than the Monitoring Station, the provided location should be a full network path to the system in addition to the directory path on that system.

Resource Usage Report Generation

Due to the large number of options available for the Resource Usage report, generating an extensive report for a large group of Elements can take several minutes. If exhaustive report generation is necessary, but taking too long, you can increase the number of report images (the default being "6") that *up.time* concurrently generates for this type of report.

Note that the default number is optimal in most cases; increasing the amount may improve performance, but the law of diminishing returns applies, as too many concurrent threads can tax the PDF generation process overall.

Logging is configured through the following `uptime.conf` parameter:

```
reporting.prefetch.images.threads = 6
```

Monitoring Station Interface Changes

Some configuration options affect the Monitoring Station interface. These can be modified by manually inputting settings in the *up.time Configuration* panel, as outlined in [Modifying up.time Config Panel Settings](#).

Status Alert Acknowledgement

When services reach a warning or critical state, administrators can flag an alert as "acknowledged," which prevents subsequent alerts from being broadcasted, giving them time to investigate the issue. See [Acknowledging Alerts](#) for more information.

Service status alert acknowledgements can be reported in the status tables on the **Global Scan** dashboard. By default, status alert acknowledgement counts are not shown; if enabled a new column (labelled ACK) appears in the Service Status section of **Global Scan**. When the current status of a monitor is acknowledged, it appears in the ACK column instead of in the WARN or CRIT column.

You can enable or disable status acknowledgement (i.e., add or remove the ACK column from the status tables) through the following parameter (the default value is shown):

```
acknowledgedSeparate=false
```

3D Graphs

When performance and availability graphs are generated, the Graph Editor is used to manipulate the appearance of graphed data (see [Using the Graph Editor](#)). Transformations from a three-dimensional perspective are possible if the user account permits it (see [Adding Users](#)), and the user is connecting to the Monitoring Station using Internet Explorer.

This 3D presentation option can be disabled outright. You can determine whether ActiveX graphs are displayed in 3D for users with Internet Explorer through the following parameter (the default value is shown):

```
default3DGraphs=true
```

Custom Dashboard Tabs

Custom dashboards can be added to **My Portal** to display custom content that is relevant to the particular user who is currently logged in. Up to 50 dashboards can be added, each of which is accessed through, and viewed in, its own tab at the top of **My Portal**.

A custom dashboard tab is configured by pointing up.time to a custom Web page, and indicating which User Group will be able to view it. You can enable and configure the first dashboard through the following parameters:

```
myportal.custom.tab1.enabled=true
myportal.custom.tab1.name=<DashboardNameOnTab>
myportal.custom.tab1.URL=<URLtoCustomPage>
myportal.custom.tab1.usergroups=<UserGroupName>
```

Values for the first three parameters are required. If no name is specified for the User Group parameter (or, if no User Groups have been defined), the custom dashboard will be visible to all *up.time* users. Thus, a User Group parameter is only required if you want to restrict or refine user access to a particular custom dashboard.

To create additional tabs, add the same set of parameters, but increment the tab count:

```
myportal.custom.tab2.enabled=true
myportal.custom.tab2.name=<DashboardNameOnTab>
myportal.custom.tab2.URL=<URLtoCustomPage>
```

License Information

If your up.time package did not come with a license key, then either contact your sales representative to request a key or send an email to support@uptimesoftware.com. You will need the host ID for the system so that a permanent license key can be generated. The host ID is displayed in the **License Information** subpanel, and is similar to the following:

```
001110bf101d
```



You do not need the host ID if you are evaluating up.time. The demo licenses expire after predetermined amounts of time and can run on any system.

Installing or Updating a License

To install or update a license, do the following:

1. In the *Tree* panel, click *License Info* .
If you currently have an *up.time* license, it is displayed in the *License Information* subpanel.
2. Paste the new or updated license into the *License Key* text box.
3. Click *Update License* .

Setting a Notification Group for vSync-Related Licensing Errors

In the *License Notification* section of the *License Information* page, you can select the Notification Group that receives alerts should there be any licensing errors related to syncing with VMware vSphere.

For more information, see [Managing vSync](#).