Contacting BMC Software

You can access the BMC Software Web site at http://www.bmc.com. From this Web site, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

<table>
<thead>
<tr>
<th>United States and Canada</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Address</strong></td>
</tr>
<tr>
<td>BMC Software, Inc.</td>
</tr>
<tr>
<td>2101 CityWest Blvd.</td>
</tr>
<tr>
<td>Houston TX 77042-2827</td>
</tr>
</tbody>
</table>

Copyright 2015 BMC Software, Inc. or licensors, as an unpublished work. All rights reserved.

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc.

All other trademarks belong to their respective companies.

BMC Software considers information included in this documentation to be proprietary and confidential. Your use of this information is subject to the terms and conditions of the applicable End User License Agreement for the product and the proprietary and restricted rights notices included in this documentation.

**Restricted Rights Legend**

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC Software, Inc., 2101 CityWest Blvd., Houston, TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer Support

You can obtain technical support by using the Support page on the BMC Software Web site or by contacting Customer Support by telephone or e-mail.

Support Web Site

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at http://www.bmc.com/support_home. From this Web site, you can:

- Read overviews about support services and programs that BMC Software offers
- Find the most current information about BMC Software products
- Search a database for problems similar to yours and possible solutions
- Order or download product documentation
- Report a problem or ask a question
- Subscribe to receive e-mail notices when new product versions are released
- Find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

You can also access product documents and search the Knowledge Base for help with an issue at http://www.sentrysoftware.com

Support by Telephone or E-mail

In the United States and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the United States and Canada, please contact your local support center for assistance. To find telephone and email contact information for the BMC Software support center that services your location, refer to the Contact Customer Support section of the Support page on the BMC Software Web site at http://www.bmc.com/support_home.
Table of Contents

- User Goals and Features .................................................................................................................. 12
- Product at a Glance ............................................................................................................................ 13
- Requirements ...................................................................................................................................... 14
- Credentials Management ................................................................................................................... 15
- String Search and Numeric Value Extraction ................................................................................... 16
- Thresholds and Alerts Actions ........................................................................................................ 17
- Importing the Monitoring Solution into Central Administration ................................................... 21
- Creating the Installation Package ..................................................................................................... 22
- Downloading the Installation Package ............................................................................................. 23
- Installing the Package ....................................................................................................................... 23
- Configuring Hosts and Groups .......................................................................................................... 27
  - Defining Host Settings .................................................................................................................... 28
  - Configuring a Monitor Group ......................................................................................................... 33
- Configuring Monitors ....................................................................................................................... 36
  - Analyzing Command Lines .......................................................................................................... 38
  - Analyzing a Database Query ......................................................................................................... 42
    - Performing a Query on a Microsoft SQL Server Database .......................................................... 43
    - Performing a Query on a MySQL Server Database ..................................................................... 45
    - Performing a Query on a PostgreSQL Database ........................................................................... 46
    - Performing a Query on an Oracle Database Server .................................................................... 47
    - Performing a Query on an Other Database .................................................................................. 49
- Monitoring Files (Flat and LOG) ...................................................................................................... 51
- Monitoring a File System ................................................................................................................ 54
- Monitoring Folders .......................................................................................................................... 56
- Leveraging Values from Other KMs’ Parameters ............................................................................. 59
- Monitoring Nagios Plugins ............................................................................................................. 63
- Monitoring Processes ..................................................................................................................... 66
- Monitoring SNMP Agents .............................................................................................................. 70
  - Polling SNMP Agents from a Single Numeric-based OID ............................................................. 72
  - Polling SNMP Agents from a Single String-based OID ............................................................... 73
  - Polling SNMP Agents from an SNMP Table ................................................................................ 74
- Listening for SNMP Traps ................................................................................................................. 76
Monitor Types and Attributes

Importing an Agent Configuration

Setting the Polling Interval

Configuring Thresholds

Configuring Global Advanced Settings

Configuring the SMTP Server

Enabling the Debug Mode

Configuring the Proxy Settings

Configuring Global Advanced Variables

Configuring Thresholds

Configuring Monitor Thresholds

Editing a Threshold Configuration

Deleting a Threshold Configuration

Defining Alert Actions

Alert Actions Macros

Format Symbols for %SEN_TIME:... Macros

Setting the Polling Interval

Importing an Agent Configuration

Monitor Types and Attributes

Studio Command Line

Studio Database Query

Studio File

Studio File System

Studio Folder

Studio Group

Studio Host

Monitoring Studio

Studio Multi-parameter Formula

Studio Nagios Performance Data
Release Notes for v9.3.00
What’s New

- **SWSY-180 - PSL Commands Analysis:** Monitoring Studio now supports PSL commands. The PSL Command Monitor makes it possible to execute PSL commands and analyze their result.
- **SWSY-1988 - Windows Event monitoring** now supports regular expressions for selecting and acknowledging specific event messages.
- **SWSY-2384 - Multi-Parameter Formula:**
  - supports the following new functions: swTableJoin(), swGetMaximum(), swGetMinimum(), swGetAverage().
  - it is now possible to create a String Search or a Numeric Value Extraction for this Monitor.
- **SWSY-1037 - The Web Request Analysis Monitor** now uses a java-based engine to improve performance and flexibility.
- **SWSY-1037 - Proxy Settings:** this new Global Advanced Settings option was added to let you configure the proxy server to be used to execute Web requests.
- **SWSY-1135 - The Discovery Status and Discovery Time attributes** were added to the Monitoring Studio monitor type to report on the global discovery.

Changes and Improvements

- **SWSY-2426 - Folders and Command Lines:** remote monitoring is now fully supported on Windows 2003.
- **SWSY-2226 - Web Requests (POST - Web Service):** Environment variables can now be used when specifying the path to the 'Web Service Body'.
- **SWSY-2242 - Monitoring Studio's overall performance and responsiveness** have been greatly improved to better handle large configurations.
- **SWSY-2211 - Using invalid Java credentials prevents data collection.** The message displayed in the annotation of the Collection Error Count attribute and in the KM debug file now clearly identifies this Java settings problem.
- **SWSY-2005 - The folder where temporary files are stored** is now emptied during the initial discovery cycle and temporary files are deleted at the end of each collection cycle.

Fixed Issues

- **SWSY-2316 - Monitoring Studio now supports multiple alert actions** used in combination with macros when importing a PATROL classic configuration into a CMA policy.
- **SWSY-2334 - When the polling interval was set to 'once a day' or 'once a week',** no data was collected after the first discovery.

Known Issue

- **SWSY-2384 - Multi-Parameter Formula:** Parameter path 'A' is not recorded in the version 10.5 of BMC TrueSight Operations Management.
Overview
What is Monitoring Studio?

**TrueSight Operations Management - Monitoring Studio** is a powerful tool designed to help administrators fulfill their custom monitoring needs. This toolbox enables you to monitor almost any technology (application, server, device, etc.) for which there is no out-of-the-box monitoring solution. In a few clicks, you can cover up to 100% of your technologies in your BMC monitoring environment. Compatible with Linux/UNIX and Windows, Monitoring Studio is a simple and effective way to rapidly deploy the monitoring of custom technologies without any coding. It also has the benefits of a “standard” solution: maintenance, updates, patches, etc. to further respond to growing technological needs for specific business-critical technologies.

**Monitoring Studio** is a monitoring solution designed to seamlessly integrate with Central Monitoring Administration and TrueSight Operations Management. Refer to the [Installing the Monitoring Solution](#) chapter for detailed information about the installation procedure.

What to Monitor with Monitoring Studio?

Depending on the nature of the technology you wish to monitor, **Monitoring Studio** offers a large choice of tools that you can easily configure.

**Monitoring Studio** allows you to create and monitor your own custom collection of components locally and remotely, such as software applications, processes, files, folders, Nagios plugins, Windows services, event logs, etc.

With **Monitoring Studio**, you will also be able to analyze database queries and Web requests for example, and perform string searches, extract numeric values from monitored components result outputs. See chapter [User Goals and Features](#) to know more about all the monitoring tools offered with **Monitoring Studio**.

Key Concepts and Terminology

Here is a list of definitions for concepts and terms used in this documentation:

- **Central Monitoring Administration** is the Console that allows Administrators to configure all the elements required to monitor technologies in an IT environment. Central Monitoring Administration is also referred to as **CMA** in this document. For detailed information about this product, refer to the official documentation provided on the BMC Web site.

- **TrueSight Operations Management** is the platform where the data collected by monitoring solutions, such as **Monitoring Studio**, is made available to Administrators.
• A **Technology** refers to any component that can be monitored via **Monitoring Studio**, for example an application, a server, a device, a database, a process, etc.

• **TrueSight Operations Management - Monitoring Studio** is the official name of the product described in this document. It can also be referred to as **Monitoring Studio**.

• A **Policy** refers to a group of user-defined settings that CMA uses to configure monitoring solutions, such as **Monitoring Studio**. A policy is applied when CMA receives a request from the BMC PATROL Agents that match the Agent selection properties in the policy.

• A **Monitor Group** is designed to gather a host and its related Monitors (monitoring tools). This grouping facilitates the management of Monitors. A Monitor Group can be named after the technology it monitors to easily distinguish Monitor Groups in the BMC Consoles.

• A **Host** is typically a device on which the technology you wish to monitor is installed. **Monitoring Studio** creates instances of hosts and displays them in TrueSight Operations Management.

• **Monitors** are the monitoring tools that you need to configure for monitoring a specific technology. They also refer to the instances of these monitoring tools.

**Documentation Scope**

This user guide provides detailed information about concepts and operating procedures directly in connection with the use of **TrueSight Operations Management - Monitoring Studio**. Other considerations relating to configuring or operating **Central Monitoring Administration** or **TrueSight Operations Management** are NOT included in this document. For information about BMC products, please refer to [BMC support Web site](https://www.bmc.com).
User Goals and Features

Depending on the nature of the technology you wish to monitor, Monitoring Studio offers a large choice of tools that you can easily configure to monitor system elements, query information about the targeted technology, and extract the relevant key metrics.

Monitoring Studio allows you to monitor:

- Commands lines
- Files
- File systems
- Folders
- Multi-parameter formulas
- Nagios plugins
- Processes
- PSL commands
- SNMP queries
- SNMP traps
- SQL queries
- WBEM queries
- Web requests
- Windows events
- Windows performance counters
- Windows services
- WMI queries

Monitoring Studio then allows you to analyze the results of some of the above Monitors by searching for strings and regular expressions, and extracting numeric values.

Then, you can easily define alert thresholds and alert actions to detect and react to critical conditions.
Product at a Glance

Product Integration within the BMC Framework

The diagram below shows how Monitoring Studio integrates within your BMC framework and shows interaction between all the components that compose your monitored environment.
Requirements

BMC Framework

- BMC ProactiveNet 9.5 and higher.
- BMC ProactiveNet Central Monitoring Administration 9.5 and higher.
- BMC TrueSight Operations Management.

PATROL Agent

TrueSight Operations Management - Monitoring Studio supports PATROL Agent v9.5.00 and higher.

JAVA

TrueSight Operations Management - Monitoring Studio requires Java 1.6 or higher and a Java Runtime Environment (JRE) to be installed on the same system that runs the PATROL Agent.

You can download the Java Runtime Environment along with the monitoring solution from the Sentry Software Web site.
Credentials Management

Monitoring a system thoroughly requires connecting to it and accessing some of its resources, which itself requires to be properly authenticated with this system. This is even more true for a system monitored remotely.

In Monitoring Studio, the credentials required to access a monitored system are stored and managed at the host level. For each monitored host, you will be able to define the "System Credentials", to access standard system resources on this host. You will also be able to define additional credentials that may be required to access specific resources (a database, a Web application, etc.).

System Credentials

System Credentials are provided at the Host level and can be shared by several Monitors. When the system credentials are changed, users only need to modify the credentials once for all the Monitors to inherit the changes.

The following Monitors can only use system credentials:

- File Systems
- Processes
- Windows Events
- Windows Performance Counters
- Windows Services
- WMI Queries

If the system credentials are not specified when monitoring a localhost, Monitoring Studio will use the PATROL Agent’s default account information defined via the Agent Properties panel of the Monitoring Policy Configuration page in CMA. For performing remote monitoring, system credentials are mandatory for the Monitors listed above; failing to provide this authentication information will prevent the Monitors to collect any data.

Specific Credentials

Some technologies may require additional privileges to allow access to their data. In this case, Monitoring Studio enables users to provide specific credentials that will apply to the host and its monitor types. Monitor types owned by the same host may use different credentials.
The following Monitors may require specific credentials. If specific credentials are not provided, the solution will automatically use the system credentials provided at the Host level:

- Command Lines
- Database Queries
- Files
- Folders
- Nagios Plugins
- WBEM Queries
- Web Requests

### String Search and Numeric Value Extraction

Some technologies report their bad health through repeated error messages or critical numbers which are hidden deep in log files, output of commands, database, Web page, etc. Monitoring Studio is capable of parsing all these data by means of **String Search and Numeric Value Extraction features** to detect the source of potential problems and alert you when they occur.

**String Search** and **Numeric Value Extraction** features are available for the following Monitors (also called information sources):
String Search and Numeric Value Extraction

- Command Line
- Database Query
- File
- Multi-Parameter Formula
- Nagios Plugin
- PSL Command
- SNMP Polling
- WBEM Query
- Web Request
- WMI Query

Searching for a Specific String

The **String Search** feature of **Monitoring Studio** enables you to run fast and powerful searches for strings on some of the monitored objects that you previously configured such as flat or log files, the output of a Web request or a database query, OID content, etc. You can then decide to trigger an alert and send a notification to your hypervision system when the string specified is found or not found in the return output of monitored instances.

Extracting Numeric Values

The **Numeric Value Extraction** feature of **Monitoring Studio** enables you to extract numeric values from a text input such as the output of a command, a Web page, the result of a WBEM or SQL query etc. All you need to do is indicate how to find the numeric values within the monitored object and **Monitoring Studio** will extract and report them as graphs in TrueSight Operations Management.

Thresholds and Alerts Actions

**Thresholds**

Thresholds let you define the status of an attribute based on specific conditions. TrueSight Operations Management - Monitoring Studio provides thresholds for most of the attributes managed by the solution. These thresholds are applied by default when you first start Monitoring Studio but can be easily tailored to your needs via the policies of Central Monitoring Administration. When a metric reaches the predefined warning or alert threshold, Monitoring Studio generates an alarm.
Alert Actions

Alert Actions enable you to configure specific actions to be executed when an alert is raised on an attribute. Monitoring Studio provides flexibility in fault management by triggering actions, such as triggering an event, annotating a graph, and executing a command, to notify you of the alerts generated while monitoring a technology. Alert actions can be set for all the instances of a Monitor Group as well as for individual Monitor instance.
Installing the Monitoring Solution
Once the latest version of the solution has been loaded into Central Monitoring Administration, administrators can create all the installation packages required for their different operating systems and platforms and save them for later use in the Monitoring Installation Packages list. These packages can then be deployed to multiple computers. Administrators just have to connect to TrueSight Operations Management from the server where they want to install the package, download it and launch the installation.

This section describes the different steps to follow to install Monitoring Studio:

- Importing Monitoring Studio into Central Monitoring Administration
- Creating the Installation Package
- Downloading the Installation Package
- Installing the Package
Importing the Monitoring Solution into Central Administration

The TrueSight Central Monitoring Repository includes the current versions of TrueSight Operations Management - Monitoring Studio that you can use with BMC TrueSight. If the version available in the Repository does not correspond to the latest one, you will have to manually import it:

1. Log on to the **BMC TrueSight Operations Management** Console.
2. Launch **Central Monitoring Administration**.
3. Click the **Repository** drawer and select **Manage Repository**.
4. Check that the version of the BMC component available is actually the latest one. If not, download the latest version corresponding to your operating system (Windows or UNIX/Linux) available on the [Sentry Software Website](#).
5. From **TrueSight Operations Management**, click **Import**.
6. Select **Single solution**.
8. Click **Import**.

The selected archive file is imported to the repository.
Creating the Installation Package

The installation package to deploy to managed systems can be created directly from TrueSight Operations Management:

1. Log on to TrueSight Operations Management
2. Click the Repository drawer and select Deployable Package Repository.
3. Click Add.
4. Select the operating system and platform for which you want to create a package. The components available in the repository for the selected operating system and platform are displayed.
5. Select the Installation Package Component:
   - From the Available components list, select the relevant component.
   - From the Version list, select the latest version.
   - Click the right arrow button to move the component into the Selected Components list. By default, the appropriate BMC PATROL Agent for the operating system and platform that you chose is included in the Selected components list.
   - Click Next. The Add Component Installation Package wizard are displayed.
6. Go through the wizard and specify the required PATROL information. The Installation Package Details is displayed.
7. Verify that:
   - the operating system and platform are correct
   - the components that you want to include are listed in the Included Components list.
8. Provide the following information:
   - Name: Enter a unique name for the package.
   - (Optional) Description: Enter a description of the package. The description is displayed in the Monitoring Installation Packages list on the Monitoring Repository window.
   - Format: Select a file compression format for the package.
9. Click Save Installation Package.
10. Click Close. The package is now available in the Monitoring Installation Packages list.
Downloading the Installation Package

You can download an installation package and install the components on one or more hosts. The installation runs silently with the information entered during package creation.

Recommendation
If you defined the BMC TrueSight Integration Service variable for PATROL Agents in the installation package, ensure the agents are started in phases. Do not start newly deployed agents all at once. Start and configure monitoring for the agents in planned phases to reduce the performance impact on the Integration Service nodes and on the BMC TrueSight Server associated with the automatic workflow process.

1. Log on to TrueSight Operations Management from the computer on which the PATROL Agent is installed or to be installed.
2. Click the Repository drawer and select Deployable Package Repository.
3. (Optional) To filter the list of installation packages, select an operating system from the Filter by Operating System list.
4. Click the link for the installation package that you want to download.
5. Through the browser’s download dialog box, save the installation package.

Installing the Package

This chapter provides a step by step procedure to install a monitoring solution package:
1. From the computer on which you want to install the package, log on to TrueSight Operations Management.
2. (Optional) To filter the list of installation packages, select an operating system from the Filter by Operating System list.
3. Click the link for the installation package that you want to download.
4. Through the browser’s download dialog box, save the installation package in a temporary file.
5. Extract the installation package that is appropriate for your operating system. The package is extracted to the bmc_products directory on the current host.
6. From the bmc_products directory, run the installation utility for your operating system:
   - (UNIX or Linux) RunSilentInstall.sh
   - (Microsoft Windows) RunSilentInstall.exe

The package is installed on the current host. If the package includes a BMC PATROL Agent, the agent sends a configuration request by passing its tags to Central Monitoring Administration, via the Integration Service. Central Monitoring Administration evaluates policies that match the tags, determines the final configuration to be applied, and sends the configuration information back to the agent. Monitoring is based on the configuration information received by the agent.
Monitoring with Monitoring Studio
Once the configuration of a Host and a Monitor Group is completed, you can then define the Monitors that will collect data from the targeted technologies you need to monitor. Monitoring Studio provides a large range of Monitors to cover the various technologies of your IT environment.

The following tables list the Monitors available for monitoring any technology according to the operation system (OS) they use (from source Agent to target Host).

### If Monitoring Studio is running on a Windows system:

<table>
<thead>
<tr>
<th>Monitors</th>
<th>Monitored Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Windows</td>
</tr>
<tr>
<td>Command Lines</td>
<td>X</td>
</tr>
<tr>
<td>Database Queries</td>
<td>X</td>
</tr>
<tr>
<td>File Systems</td>
<td>X</td>
</tr>
<tr>
<td>Files</td>
<td>X</td>
</tr>
<tr>
<td>Folders</td>
<td>X</td>
</tr>
<tr>
<td>Multi-Parameter Formulas</td>
<td>X</td>
</tr>
<tr>
<td>Nagios Plugins</td>
<td>X</td>
</tr>
<tr>
<td>Processes</td>
<td>X</td>
</tr>
<tr>
<td>PSL Command</td>
<td>X</td>
</tr>
<tr>
<td>SNMP Agents</td>
<td>X</td>
</tr>
<tr>
<td>SNMP Traps</td>
<td>X</td>
</tr>
<tr>
<td>WBEM Queries</td>
<td>X</td>
</tr>
<tr>
<td>Web Requests</td>
<td>X</td>
</tr>
<tr>
<td>Windows Events</td>
<td></td>
</tr>
<tr>
<td>Windows Performance Counters</td>
<td></td>
</tr>
<tr>
<td>Windows Services</td>
<td>X</td>
</tr>
<tr>
<td>WMI Queries</td>
<td></td>
</tr>
</tbody>
</table>
If Monitoring Studio is running on a UNIX/Linux system:

<table>
<thead>
<tr>
<th>Monitors</th>
<th>Monitored Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Windows</td>
</tr>
<tr>
<td>Command Lines</td>
<td></td>
</tr>
<tr>
<td>Database Queries</td>
<td></td>
</tr>
<tr>
<td>File System</td>
<td></td>
</tr>
<tr>
<td>Files</td>
<td></td>
</tr>
<tr>
<td>Folders</td>
<td></td>
</tr>
<tr>
<td>Multi-Parameter</td>
<td></td>
</tr>
<tr>
<td>Nagios Plugins</td>
<td></td>
</tr>
<tr>
<td>Processes</td>
<td></td>
</tr>
<tr>
<td>PSL Commands</td>
<td></td>
</tr>
<tr>
<td>SNMP Agents</td>
<td></td>
</tr>
<tr>
<td>SNMP Traps</td>
<td></td>
</tr>
<tr>
<td>WBEM Queries</td>
<td></td>
</tr>
<tr>
<td>Web Requests</td>
<td></td>
</tr>
<tr>
<td>Windows Events</td>
<td></td>
</tr>
<tr>
<td>Windows Performance Counters</td>
<td></td>
</tr>
<tr>
<td>Windows Services</td>
<td></td>
</tr>
<tr>
<td>WMI Queries</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Hosts and Groups

The monitoring of any technology can be configured through a Policy that you can create or edit via Central Monitoring Administration. You can then apply the policy to the PATROL Agents that share the same tag, or according to their IP address or hostname.

For detailed information about policies, please refer to the Central Monitoring Administration user documentation available on the BMC Web site.

To configure the monitoring of a technology with Monitoring Studio

1. Log on to Central Monitoring Administration.
2. In the Navigation pane, click the Policies drawer.
3. Expand the Monitoring folder and select a policy view (e.g. All).
4. Click to create a Policy that will be deployed on the PATROL Agents that share the same specified tag or according to their IP address, hostname, etc.
5. Click the Monitor Configuration link and click the button
6. In the Monitoring Solution field, select Monitoring Studio. The related Monitoring Profile, Version and Monitor Type information is automatically displayed.
7. Define the Monitoring Studio Configuration options:
   - Define a Host settings
   - Configure a Monitor Group
   - Create Monitors
   - Optional — Search Monitors' output for specified string
   - Optional — Extract numeric values from Monitors' output
   - Optional — Define the Monitor's polling interval
   - Optional — Set alert actions
   - Optional — Set global advanced settings
8. Click the Add to List button to complete the creation of the Monitor Group.
9. Once the policy is completed and deployed to the PATROL Agents it targets, Monitoring Studio will start collecting data from the managed technologies on local or remote hosts. The data will automatically be made available in TrueSight Operations Management.

You can also create a Monitor for an existing policy, in this case select the policy to which you want to add a monitor type and click the button. Once the Monitor Type panel is displayed, you can then proceed with the procedure described above from step 6.
Defining Host Settings

The first action you need to perform is to provide Monitoring Studio with the information related to the Host on which the technology you want to monitor is running. Once the Host settings are configured, its instance is automatically created in TrueSight Operations Management.

The Host instance contains all the information about the server where the technology to monitor is running (hostname, system type, connection credentials, host availability check information, etc.).

*A Host instance is always listed under a Group instance. A Group can only contain one host.*

To configure the connection settings for a host

1. In the Monitoring Studio **Configuration** panel, scroll to the **Host Settings** section.

2. Provide the following information:
   - **Hostname**: Enter the name (or IP address or Fully Qualified Domain Name) of the host on which the technology you wish to monitor is running. By default, **Monitoring Studio** offers to establish a connection to the server on which the PATROL Agent is installed (localhost).
   - **Optional — Description**: Enter a unique description to easily identify the host during the configuration import process.
   - **System Type**: Select the type of the operating system that is running on the host that will be monitored (Windows, UNIX/Linux, or Other). Select **Other** for any host that will not behave as a regular Linux or UNIX system would. For example, while a network device is likely to run a customized version of Linux or BSD, its restricted shell will not allow Monitoring Studio to use standard UNIX commands to perform the monitoring.

   *Monitoring Studio requires you to specify the type of the operating system to avoid a costly identification phase and to offer more robust monitoring options.*
- **System Username**: Enter the username that will be used by Monitors relying on System Credentials to collect data from the targeted host.
- **System Password**: Enter the password for the username provided above, that will be used by Monitors relying on System Credentials to collect data from the targeted host.
- **Optional — Associated OpenSSH Private Key File Path**: When monitoring remote hosts running UNIX, Linux or other types of operating systems that support SSH authentication key file, you may need to provide an OpenSSH private key file to establish a secured connection with the remote host. Enter the path of the OpenSSH private key file you wish to use to establish a connection with the remote host and enter the optional PassPhrase in the Password field.

⚠️ **The Private Key File should exists on the PATROL Agent node.**

When monitoring a local host, if credentials are not provided, Monitoring Studio will use the default PATROL Agent account. For remote monitoring, System Credentials are mandatory for the Monitors listed below; failing to provide this authentication information will prevent the solution from collecting any data for:
- File Systems
- Processes
- Windows Event Logs
- Windows Performance Counters
- Windows Services
- WMI Queries

- **Create a Device in the Console**: Select this option if you wish Monitoring Studio to create an instance of the device in TrueSight Operations Management. All associated Monitors and their events and attributes will be attached to this new device in TrueSight Operations Management. If this option is not selected, they will be attached to the device matching the PATROL Agent host.

3. **Optional — Configure the Host Availability Check**:
   - Select the method(s) Monitoring Studio should use to test the availability of the monitored host:
Host Availability

- **Ping Check**: Default option. Select this option to have Monitoring Studio ping the monitored host up to four times in a row. To be successful, the targeted host must respond to at least one ping command during each collection cycle.

- **Hostname Resolution Check**: Select this option to have Monitoring Studio resolve the hostname of the monitored host to an IP address.

- **SNMP Check**: Select this option to have Monitoring Studio check the monitored host availability via an SNMP session. For the SNMP availability check to be successful, the targeted host must respond to a "GETNEXT" request on either the OID 1.3.6.1 or the OID 1.3.6.1.4.1.

  It is necessary to provide the SNMP settings information for the SNMP check to be performed.

- **SSH Check**: Select this option to have Monitoring Studio check the targeted host availability (UNIX, Linux, or other systems) via the SSH protocol. For the SSH availability check to be successful, Monitoring Studio must be able to connect to the host using the provided System Credentials (required in the Host Settings).

- **TCP Check - Port Number**: Use this option to have Monitoring Studio check the targeted host availability via a TCP connection. You must provide a port number for the TCP connection.

- **WMI Check**: Select this option to have Monitoring Studio check the monitored host availability via the WMI protocol using the provided System Credentials (to use towards Windows-based host only).

- **Signature Files Check**: Click to open the Signature Files Check dialog box. If at least one of the signature files is present on the targeted host, the check is considered successful:
  - Provide the name and full path of the signature file on the targeted host.
  - Click the Add to List button to complete the configuration of the Signature File Check.
Configuring Hosts and Groups

31

**Signature Files Check**

**Signature Files**

- **Signature File**
  - Filename (Full Path)

  [Add to List] [Modify Selection] [Remove from List]

**List - Signature Files**

Wildcards are not supported in signature files

- **Disable Monitors When the Host is Unreachable**: Select this option if you wish Monitoring Studio to disable the monitoring of all the technology instances monitored on the target host. If any of the selected host availability check fails, note that the monitoring will resume automatically when the host becomes available again.

- **Optional** — Configure the **Host Advanced Variables**:
  - Enter the **Name** and **Value** of the configuration variable you need to configure.
  - Click **Add to List**.
  - You can easily modify or remove a variable by selecting it in the list and clicking either the **Modify Selection** or the **Remove from List** buttons.
  - Click **Close**.

- **The Variable field is case sensitive.**
These variables should only be set when instructed so by Sentry Software Support.

You are now ready to configure your Monitor Group.
Configuring a Monitor Group

A Monitor Group is designed to facilitate the management of the monitored technologies (application, server, device, etc.). It contains a Host and one or several Monitors. Typically, the Monitor Group display name is the name of the technology you wish to monitor, for example 'My Company's Web Site'. A Monitor Group can only contain one host.

To configure a monitor group

1. In the Monitoring Studio Configuration panel, scroll to the Monitor Group Settings section.

   Monitor Group Settings
   
<table>
<thead>
<tr>
<th>Internal ID</th>
<th>Display Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   Group Alert Actions  Group Constants
2. Enter the **Internal ID** that will be used internally by TrueSight Operations Management to identify the Monitor Group. The PATROL internal identifier of every object belonging to this Group (Hosts, Monitors, etc.) will include the internal ID of the Group. It is therefore recommended to keep this ID short.

3. Enter a **Display Name** that is significant to you and that identifies the technology you wish to monitor. This name will be used as a label to identify the Monitor Group in TrueSight Operations Management.

4. **Optional** — Set the **Group Alert Actions** options. Group Alert Actions are typical alert actions that apply to all Monitors attached to the Group when their thresholds are breached. **Note that these alert actions add up to the alert actions defined at the Monitor level.**

   Click the **Group Alert Actions** button to display the settings panel:

   Select one or several alert actions you wish **Monitoring Studio** to perform when a threshold is breached. Macros can be used to customize alert actions, refer to the **Alert Actions Macros** chapter for detailed information.

   - **Event**: select this option to have **Monitoring Studio** trigger a PATROL event. Use the **Event**
Content field to provide the string that will be displayed with the event. The \%(SEN_ALERT_DEFAULTCONTENT) macro is proposed by default to provide basic information about the event.

- **Annotation**: select this option to have Monitoring Studio annotate the attribute's graph. Use the Annotation Content field to provide the string that will be displayed at the annotation point. The \%(SEN_ALERT_DEFAULTCONTENT) macro is proposed by default to display basic information at the annotation point.

- **Local Command Line**: Select the Upon Thresholds Breach, Execute a Command option to have Monitoring Studio execute a command line on the system where the PATROL Agent is installed. Provide the Username and Password required to run the command line (or leave these fields blank to use the PATROL Agent's default account) as well as the Command Line you wish the solution to execute.

- **E-mail**: Select this option to have Monitoring Studio send an E-mail. Provide the sender and the recipient email address in the From and To fields. To send the email to multiple recipients, use the comma (,) or the semi-column (;) to separate the recipients' email addresses. The \%(SEN_ALERT_DEFAULTCONTENT) and \%(SEN_INFORMATION) macros are proposed by default to provide information about the event.

⚠️ An SMTP server is required to receive alerts by email. For more information refer to Configuring the SMTP Server.

5. **Optional** — Define the Group Constants for the Monitor Group. Constants are very useful for monitoring a technology whose properties may change from one system to another. Constants are defined at the Monitor Group level and can be reused in the Monitors related to the Group. Here are some examples of use:

**Example**: You want to specify the monitoring of a device through a command line interface (CLI). Depending on where the PATROL Agent is running, this CLI may be installed in a different directory. To avoid editing every Monitor using this CLI to change the path, you create a Group Constant MYCLI_PATH with the path to the CLI executable. Then you create the "Command Line Analysis" Monitor by specifying, for example: \%(MYCLI_PATH)" -option1 - option2

- Click the **Group Constants** button to open the settings panel and provide the required information:
Configuring Hosts and Groups

TrueSight Operations Management - Monitoring Studio 9.3.00

Configuring Hosts and Groups

Group Constants

- Constant name: Enter the name of the constant.
- Value: Set a value for the constant.
- Click the Add to List button to complete the creation of the Group Constant.

6. Click Close.

The next step consists in configuring Monitors for the technologies you wish to monitor.

Configuring Monitors

Once you have created the Monitor Group and defined the Host connection settings associated to your targeted technology, you need to choose and configure the tools, or Monitors, that Monitoring Studio will use to collect data from this technology.

To configure a monitor

1. Log on to Central Monitoring Administration.
2. In the Navigation pane, click the Policies drawer.
3. Expand the Monitoring folder and select a policy view (e.g. All).
4. Click + to create a Policy that will be deployed on the PATROL Agents that share the same specified tag or according to their IP address, hostname, etc.
5. Click the Monitor Configuration link and click the button.
6. In the Monitoring Solution field, select Monitoring Studio. The related Monitoring Profile, Version and Monitor Type information is automatically displayed. Monitoring Studio provides a large range of monitoring tools (Monitors) to watch over almost any component of your IT environment. Each Monitor is described in details in the Monitoring with Monitoring Studio chapter.
7. Define the targeted **Host Settings**.
8. Configure a **Monitor Group**.

    To modify the settings of an existing Monitor, simply edit the corresponding policy, then click the **Monitor Configuration** link, then select the **Monitoring Studio** Monitor Type and click 🖼. Scroll down to the **List – Monitor Groups** section and double-click the Monitor Group to which you want to add a Monitor.
Analyzing Command Lines

Even though TrueSight Operations Management - Monitoring Studio offers a variety of monitoring methods (Monitors), there might be an in-house script or command that you need to run and analyze on a regular basis to monitor a specific technology.

The Command Line Monitor allows you to trigger the periodic execution of a specified command line on the targeted host. This command can be a shell command, a shell script or an executable file with arguments.

You can define String Searches and Numeric Value Extractions that will parse the output of this command line.

To create a command line analysis:

1. Access the Monitoring Studio Configuration panel, as explained in the Configuring Monitors chapter.
2. In the Monitors section, click the Command Lines button.
3. The Command Lines panel is displayed to provide the connection credentials and define the command line settings.
4. Provide the Credentials required to establish a connection to the host:

<table>
<thead>
<tr>
<th>Credentials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
</tr>
<tr>
<td>Password</td>
</tr>
<tr>
<td>Associated OpenSSH Private Key File Path</td>
</tr>
</tbody>
</table>

   **Analyzing Command Lines — Credentials Settings**

   - **Username**: Enter the username to use to establish the connection with the Host. Leave this field blank to use the username provided for the Host at the Monitor Group level.
   - **Password**: Enter the password to use to establish the connection with the Host. Leave this field blank to use the password provided for the Host at the Monitor Group level.
   - **Optional — Associated OpenSSH Private Key File Path**: When monitoring remote hosts running UNIX, Linux or other types of operating systems that support SSH authentication key file, you may need to provide an OpenSSH private key file to establish a secured connection with the remote host. Enter the path of the OpenSSH private key file you wish to use to establish a connection with the remote host and enter the optional PassPhrase in the Password field.

   **The Private Key File should exists on the PATROL Agent node.**
5. Define the **Command Line Settings**:

**Command Line Settings**

<table>
<thead>
<tr>
<th><strong>Command Line</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Run this Command Locally</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Timeout (seconds)</strong></td>
<td>30</td>
</tr>
<tr>
<td><strong>Report Errors in Group’s “Collection Error Count”</strong></td>
<td></td>
</tr>
</tbody>
</table>

- **Command Line**: Enter the command line or the path to the script that **Monitoring Studio** will execute. You can use the following macros in the command line that will be resolved at the run time:
  - %\{SEN\_TIME\:<date-time-format>\}: Use this macro to insert the current date and time in the command line. You can specify the format of the date and time string that will be inserted, which follows UNIX asctime() format (%Y for year, %m for month, %D for day, %H for hours, %M for minutes, %S for seconds, etc.).
  - %\{SEN\_LASTTIME\:<date-time-format>\}: Use this macro to insert the date and time at which the command was last executed. This can be particularly useful when you need to specify a time range for the command, like listing events since the last time we checked. The format is the same as the UNIX asctime() format (%Y for year, %m for month, %D for day, %H for hours, %M for minutes, %S for seconds, etc.).

  **Warning**: When using this macro, the execution is skipped entirely the first time the Monitor runs (after the PATROL Agent starts). This is to ensure that an actual date and time is inserted with an actual value for this macro.

  - %\{SEN\_SCRIPTPATH\:<local-script-path>\}: Use this macro to copy a file stored on the PATROL Agent’s system to the monitored host before the command is executed. When the command is executed, the macro is replaced by the path to the copied file on the targeted host. This is particularly useful to trigger the execution of scripts that are stored on the PATROL Agent’s system without having to install these scripts on each monitored host. This macro is irrelevant when monitoring the localhost.
  - %\{HOSTNAME\}: This macro inserts the hostname of the targeted system, as specified in the host configuration (it therefore may be its IP address, FQDN or short name).

  **Warning**: The %\{SEN\_SCRIPTPATH\} macro should provide the script file path on the local Agent system.

  **Warning**: The %\{SEN\_SCRIPTPATH\} macro does not support local environment variables (e.g., %PATROL\_HOME%).

- %\{USERNAME\}: This macro inserts the username of the specified credentials in the command line to be executed.

- %\{PASSWORD\}: This macro inserts the password of the specified credentials in the command line to be executed. The password is inserted in clear text.
Passwords should never be sent in clear text. Passwords in command lines may be visible to non-root users. Use at your own risk.

Please note that macros are case sensitive and should then always be written in upper case.

- **Run this Command Locally**: Select this option to execute the command locally on the PATROL Agent while monitoring a remote host.
- **Timeout (seconds)**: Specify the time in seconds after which the command will be stopped (Default: 30 seconds). When the timeout is reached, Monitoring Studio will consider that the command has failed to execute properly and will set the value of the Status attribute to 2 (Failed). No further analysis will be performed (String Search or Numeric Value Extraction).
- **Report Errors in Group’s “Collection Error Count”**: Select this option to have the Collection Error Count attribute of the Group reflect possible alerts triggered upon the Command Line execution. The Collection Error Count attribute of the group reports on the collection errors of all Monitors, associated to the group, for which this option is available and selected, providing a global view of the collection errors for the whole group.

6. Define the **Command Execution Validation** criteria:

   - **Execution is Validated When Output Contains**: Enter the regular expression that needs to match the command output for the command to be considered successful. The regular expression entered here will be searched in the output of the command. If it is not found, the Status attribute of the Command Line Monitor is automatically set to 2 (Failed) indicating that the command failed to execute properly.

   This option can be particularly useful to ensure that the command has been properly executed and avoid false alerts triggered by the associated String Search/Numeric Value Extraction when an error is encountered during the command execution. For example, if the specified command must print a text banner, you will want to make sure that the text banner is found in the command output instead of an error message).

- **Exit Codes Below Mean the Command Execution**: Select an execution option (succeeded/failed), to state if the command line was properly executed or not. When one of the exit codes is found or not found, depending on the option selected, the ExitStatus attribute of the Command Line Monitor is automatically set to 1 (Failed) or 0 (Succeeded) indicating that the command failed or succeed to execute properly.

- **Exit Codes**: Enter one or several exit codes separated by commas. When one of the exit codes is found or not depending on the execution option selected above, the ExitStatus attribute will be set to 1 (Failed) or 0 (Successful).
7. Define the **Monitor Settings**:

<table>
<thead>
<tr>
<th><strong>Monitor Settings</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal ID</strong></td>
</tr>
<tr>
<td><strong>Display Name</strong></td>
</tr>
<tr>
<td><strong>Polling Interval</strong></td>
</tr>
<tr>
<td><strong>Alert Actions</strong></td>
</tr>
</tbody>
</table>

- **Internal ID**: Enter an ID to identify the managed command line instance in TrueSight Operations Management.
- **Display Name**: Enter a name to identify the managed command line instance in TrueSight Operations Management.
- **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
- **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for any attribute of this command line is breached.

8. **Optional — Define the Content Processing rules you wish to apply to the command line output**:

- **String Searches**: Define the search criteria for a specific string you wish to find or not find in the command line output.
- **Numeric Extractions**: Define any numeric value to be extracted from the command line output.

9. Click the **Add to List** button to complete the creation of the command line instance.
10. Click **Close**.
Analyzing a Database Query

The **Database Query Analysis** tool executes SQL queries on the most popular database servers currently available on the market (Microsoft SQL Server, MySQL, Oracle and PostgreSQL) and monitors their return output within your TrueSight Operations Management environment. You can then run string or numeric value searches on the return output to monitor the result in myriad ways.

If the technology you wish to monitor uses a database server, you can test this database by sending applicative queries to the database server, or by testing the content of some critical data tables. As query results are stored by **Monitoring Studio** in a pipe-separated table format, it is easy to specify strings to be searched or numeric values to be extracted from a database query. Please refer to the [String Search](#) and [Numeric Value Extraction](#) sections for more information.

**To analyze a database query:**

1. Access the **Monitoring Studio Configuration** panel, as explained in the Configure Monitors chapter.
2. In the **Monitors** section, click the **Database Queries** button.
3. The **Database Queries** panel is displayed to offer you several options. The procedure may differ according to the type of server you wish to query:

   ![Database Queries panel](image)

   *Command Line panel*

4. Click the type of database server you wish to query to display its specific configuration panel:
   - Performing a query on a Microsoft SQL Server database
   - Performing a query on a MySQL Server database
   - Performing a query on an Oracle Database Server
   - Performing a query on a PostgreSQL database
   - Performing a query on an Other Database
Performing a Query on a Microsoft SQL Server Database

This section details the various connection settings available for performing queries on a Microsoft SQL database server. First steps are common to all database queries and connection methods. They are described in the Analyzing a Database Query chapter.

⚠️ Performing a query on a Microsoft SQL Server 2000 is not possible with Java 1.8 and the SSL encryption enabled.

Configuring a connection to a Microsoft SQL database server

1. Define the Connection Settings to the Microsoft SQL database server:

   ![Connection Settings Table]

   - **SQL Server Instance Name**: Specify the SQL server instance name you wish to use, if there are several SQL Server instances installed. Leave "default" if there is a single instance.
   - **SQL Server Port Number**: Enter the number of the port that will be used to access the SQL database (default: 1433).
   - **SSL Encryption**: Select this option to encrypt the query with the HTTPS protocol. HTTPS is a secure version of the Hyper Text Transfer Protocol (HTTP) based on the SSL (Secure Sockets Layer) protocol. SSL creates a secure connection between a client and a server (SQL Server 2000 is not supported), over which any amount of data can be sent securely.
   - **Authentication mode (SQL Server/Windows)**:
     - Select **Windows** if you wish to connect to the database through a Windows user account. In that case, SQL Server validates the account name and password using the Windows identity validation (trusted connection).
     - Select **SQL Server** if you wish to connect to the database with a specified login name and password from a non-trusted connection. In that case, SQL Server performs the authentication itself by checking if a SQL Server login account has been set up and if the specified password matches the one previously recorded. If SQL Server does not have a login account set, authentication fails and you get an error message.
   - **Username**: Enter the username to use to connect to the database. Leave blank to use the username provided at the Host level.
- **Password**: Enter the password associated with the specified username. Leave blank to use the password provided at the Host level.

  * Connection Settings will be shared by all the queries defined within the Microsoft SQL Server Monitors.

2. Define the **SQL Query and Timeout** properties:

<table>
<thead>
<tr>
<th>SQL Query and Timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Query</td>
</tr>
<tr>
<td>Timeout (seconds)</td>
</tr>
<tr>
<td>Report Errors in Group’s CollectionErrorCount</td>
</tr>
</tbody>
</table>

   - **SQL Query**: Enter the SQL query you wish to perform.
   - **Timeout (seconds)**: Specify the time in seconds after which the query will be stopped. (Default: 30 seconds). If the query times out, the **Status** attribute of the **Studio Database Query** Monitor Type will be set to 2 (Failed) and an alarm will be triggered in TrueSight Operations Management.
   - **Report Errors in Group’s “CollectionErrorCount”**: Select this option to have the **Collection Error Count** attribute of the Group Monitor reflect possible alerts triggered upon the query execution. The **Collection Error Count** attribute of the Group reports on the collection errors of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.

3. Define the **Monitor Settings**:
   - **Internal ID**: Enter an ID to identify the managed SQL query instance in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the managed SQL query instance in TrueSight Operations Management.
   - **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional — Alert Actions**: Define the action(s) **Monitoring Studio** needs to perform when the thresholds for this SQL query instance is breached.

4. **Optional — Define the Content Processing rules you wish to apply to the SQL query output**:
   - **String Searches**: Define the search criteria for a specific string you wish to find or not find in the SQL query output.
   - **Numeric Extractions**: Define any numeric value to be extracted from the SQL query output.

5. Click the **Add to List** button to complete the creation of the SQL query instance.

6. Click **Close**.
Performing a Query on a MySQL Server Database

This section details the various connection settings available for performing queries on a MySQL server database. First steps are common to all database queries and connection methods. They are described in the Analyzing a Database Query chapter.

Configuring a connection to a MySQL server database

1. Define the Connection Settings to the MySQL server database:

   **Connection Settings**
   
   - **Database Name**: Enter the name of the database.
   - **MySQL Port Number**: Enter the port that will be used to access the MySQL server database.
   - **Username**: Enter the username to use to connect to the database.
   - **Password**: Enter the password associated with the specified username.

   *Connection Settings will be shared by all the queries defined within the MySQL Server Monitors.*

2. Define the SQL Query and Timeout properties:

   **SQL Query and Timeout**
   
   - **SQL Query**: Enter the SQL query you wish to perform.
   - **Timeout (seconds)**: Specify the time in seconds after which the query will be stopped (Default: 30 seconds). If the query times out, the Status attribute of the Studio Database Query Monitor Type will be set to 2 (Failed) and an alarm will be triggered in TrueSight Operations Management.

   - **Report Errors in Group's CollectionErrorCount**: Select this option to have the Collection Error Count attribute of the Group Monitor reflect possible alerts triggered upon the query execution. The Collection Error Count attribute of the Group reports on the collection errors of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.
3. Define the **Monitor Settings**:
   - **Internal ID**: Enter an ID to identify the managed MySQL query instance in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the managed MySQL query instance in TrueSight Operations Management.
   - **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional — Alert Actions**: Define the action(s) **Monitoring Studio** needs to perform when the thresholds for this MySQL query instance is breached.

4. **Optional** — Define the **Content Processing** rules you wish to apply to the MySQL query output:
   - **String Searches**: Define the search criteria for a specific string you wish to find or not find in the MySQL query output.
   - **Numeric Extractions**: Define any numeric value to be extracted from the MySQL query output.

5. Click the **Add to List** button to complete the creation of the MySQL query instance.
6. Click **Close**.

### Performing a Query on a PostgreSQL Database

This section details the various connection settings available for performing queries on a PostgreSQL database server. First steps are common to all database queries and connection methods. They are described in the [Analyzing a Database Query](#) chapter.

#### Configuring a connection to a PostgreSQL server database

1. Define the **Connection Settings** to the PostgreSQL server database:

<table>
<thead>
<tr>
<th><strong>Connection Settings</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
</tr>
<tr>
<td>PostgreSQL Port Number</td>
</tr>
<tr>
<td>Username</td>
</tr>
<tr>
<td>Password</td>
</tr>
</tbody>
</table>

   - **Database Name**: Enter the name of the PostgreSQL database.
   - **PostgreSQL Port Number**: Enter the port that will be used to access the PostgreSQL database.
   - **Username**: Enter the username to use to connect to the database.
   - **Password**: Enter the password associated with the specified username.

   *Connection Settings will be shared by all the queries defined within the PostgreSQL Server Monitor.*
2. Define the **SQL Query and Timeout** properties:

   ![SQL Query and Timeout](image)

   - **SQL Query**: Enter the SQL query you wish to perform.
   - **Timeout (seconds)**: Specify the time in seconds after which the query will be stopped (Default: 30 seconds). If the query times out, the **Status** attribute of the **Studio Database Query** Monitor Type will be set to 2 (Failed) and an alarm will be triggered in TrueSight Operations Management.
   - **Report Errors in Group’s "CollectionErrorCount"**: Select this option to have the **Collection Error Count** attribute of the Group Monitor reflect possible alerts triggered upon the query execution. The **Collection Error Count** attribute of the Group reports on the collection errors of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.

3. Define the **Monitor Settings**:

   - **Internal ID**: Enter an ID to identify the managed PostgreSQL query instance in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the managed PostgreSQL query instance in TrueSight Operations Management.
   - **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional — Alert Actions**: Define the action(s) **Monitoring Studio** needs to perform when the thresholds for this PostgreSQL query instance is breached.

4. **Optional** — Define the **Content Processing** rules you wish to apply to the PostgreSQL query output:

   - **String Searches**: Define the search criteria for a specific string you wish to find or not find in the PostgreSQL query output.
   - **Numeric Extractions**: Define any numeric value to be extracted from the PostgreSQL query output.

5. Click the **Add to List** button to complete the creation of the PostgreSQL query instance.

6. Click **Close**.

**Performing a Query on an Oracle Database Server**

This section details the various connection settings available for performing queries on an Oracle database server. First steps are common to all database queries and connection methods. They are described in the **Analyzing a Database Query** chapter.

**Configuring a connection to a Oracle database server**
1. Define the Connection Settings to the Oracle database server:

   **Connection Settings**
   
   - **Database Name**: Enter the name of the database.
   - **Oracle Port Number**: Enter the port to be used to access the Oracle database.
   - **Username**: Enter the username to use to connect to the database.
   - **Password**: Enter the password associated with the specified username.

   *Connection Settings will be shared by all the queries defined within the Oracle Database Server Monitors.*

2. Define the SQL Query and Timeout properties

   **SQL Query and Timeout**
   
   - **SQL Query**: Enter the SQL query you wish to perform.
   - **Timeout (seconds)**: Specify the time in seconds after which the query will be stopped (Default: 30 seconds). If the query times out, the **Status** attribute of the Studio Database Query Monitor Type will be set to 2 (Failed) and an alarm will be triggered in TrueSight Operations Management.
   - **Report Errors in Group’s CollectionErrorCount**: Select this option to have the Collection Error Count attribute of the Group Monitor reflect possible alerts triggered upon the query execution. The Collection Error Count attribute of the Group reports on the collection errors of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.

3. Define the SQL Query and Timeout properties
- **SQL Query**: Enter the SQL query you wish to perform.
- **Timeout (seconds)**: Specify the time in seconds after which the query will be stopped. Default is 30 seconds. If the query times out, the **Status** attribute of the **Studio Database Query** Monitor Type will be set to 2 (Failed) and an alarm will be triggered in TrueSight Operations Management.
- **Report Errors in Group’s "CollectionErrorCount"**: Select this option to have the **Collection Error Count** attribute of the Group Monitor reflect possible alerts triggered upon the query execution. The **Collection Error Count** attribute of the Group reports on the collection errors of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.

4. Define the **Monitor Settings**:
   - **Internal ID**: Enter an ID to identify the managed Oracle database query instance in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the managed Oracle database query instance in TrueSight Operations Management.
   - **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional — Alert Actions**: Define the action(s) **Monitoring Studio** needs to perform when the thresholds for this Oracle database query instance is breached.

5. **Optional — Define the Content Processing** rules you wish to apply to the Oracle database query output:
   - **String Searches**: Define the search criteria for a specific string you wish to find or not find in the Oracle database query output.
   - **Numeric Extractions**: Define any numeric value to be extracted from the Oracle database query output.

6. Click the **Add to List** button to complete the creation of the Oracle database query instance.
7. Click **Close**.

**Performing a Query on an Other Database**

This section details the various connection settings available for performing queries on database servers other than Microsoft SQL, MySQL, Oracle or PostgreSQL. First steps are common to all database queries and connection methods. They are described in the **Analyzing a Database Query** chapter.
Configuring a connection to another database server

1. Define the **Connection Settings** to the other database server:

   - **ODBC Connection String**: Enter the connection string that includes attributes such as the name of the driver, server, database and security information (username and password). In computing, a connection string is a string that specifies information about a data source and the means of connecting to it. It is passed in code to an underlying driver or provider in order to initiate the connection.
   - **Password**: Passwords are sensitive information that should not be displayed without being encrypted. When a password is included in the string you wish to execute, it is highly recommended to use the `%{SEN_PASSWORD}` macro in the string and enter the corresponding password in the Password field (encrypted display).

   *Connection Settings will be shared by all the queries defined within the database Monitor.*

   **Example**

   ```
   String = Driver={MySQL ODBC 3.51 Driver};Server=myserver;Database=mydatabase;User=user;Password=%{SEN_PASSWORD};Option=3;Password = *********
   ```

2. Define the **Monitor Settings**:

   - **Internal ID**: Enter an ID to identify the managed database query instance in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the managed database query instance in TrueSight Operations Management.
   - **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this database query instance is breached.

3. **Optional — Define the Content Processing rules you wish to apply to the database query output**:

   - **String Searches**: Define the search criteria for a specific string you wish to find or not find in the database query output.
   - **Numeric Extractions**: Define any numeric value to be extracted from the database query output.

4. Click the **Add to List** button to complete the creation of the database query instance.

5. Click **Close**.
Monitoring Files (Flat and Log)

The File Monitoring Tool is designed to monitor the presence, content, growth and change of a specific file. Therefore, the solution is able to instantly detect and alert when a critical file goes missing or if the file size is growing too fast, for example. It is one of the most important monitoring tools offered by TrueSight Operations Management - Monitoring Studio as a lot of technologies deal with files and many of them are critical. The most typical usage of file monitoring is parsing a log file. Most technologies use log files to trace their operations and notify operators when failures occur.

Selecting the right file type to monitor (flat or log) is essential to allow Monitoring Studio to read the monitored file correctly. Note that searching strings in flat or log files is also performed slightly differently, depending on the file type. Flat files are entirely updated and therefore need to be parsed entirely, as opposed to log files where new lines are appended at the end of the file - and hence only these new lines need to be analyzed.

The following procedure applies to the monitoring of both Flat and Log files.

To monitor a file (flat and log):

1. Access the Monitoring Studio Configuration panel, as explained in the Configure Monitors chapter.
2. In the Monitors section, click the Files (Flat) or Files (Log) button.
3. Provide the Credentials required to establish a connection to the host.

<table>
<thead>
<tr>
<th>Credentials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
</tr>
<tr>
<td>Password</td>
</tr>
<tr>
<td>Associated OpenSSH Private Key File Path</td>
</tr>
</tbody>
</table>

File (Flat or Log) — Credentials Settings

- **Username**: Enter the username to use to establish the connection with the Host. Leave this field blank to use the username provided for the Host at the Monitor Group level.
- **Password**: Enter the password for the username provided above, to use to establish the connection with the Host. Leave this field blank to use the password provided for the Host at the Monitor Group level.
- **Optional — Associated OpenSSH Private Key File Path**: When monitoring remote hosts running UNIX, Linux or other types of operating systems that support SSH authentication key file, you may need to provide an OpenSSH private key file to establish a secured connection with the remote host. Enter the path of the OpenSSH private key file you wish to use to establish a connection with the remote host and enter the optional PassPhrase in the Password field.
The Private Key File should exists on the PATROL Agent node.

4. Define the **File Settings:**

   - **Filename (Full Path):** Enter the path and name of the file you wish to monitor. You can use the ? wildcard to replace one character; the * wildcard to replace one or more characters or a date/time format to dynamically assign the current date or time in the file name. Simply insert the following string in the "Filename" field, where the date/time format appears, replacing the three dots by date format symbols: %[SEN_TIME:...]. For the complete list of format symbols, definitions and examples, please refer to the [Format Symbols](#) chapter.

   - The filename full path supports environment variables (example: %PATROL_HOME%).

5. Define the **Monitor Settings:**

   - **Internal ID:** Enter an ID to identify the managed monitored file instance in TrueSight Operations Management.
   - **Display Name:** Enter a name to identify the managed monitored file instance in TrueSight Operations Management.
   - **Optional — Polling Interval:** Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional — Alert Actions:** Define the action(s) Monitoring Studio needs to perform when the thresholds for this managed file instance is breached.

6. **Optional —** Define the **Content Processing** rules you wish to apply to the monitored file output:

   - **String Searches:** Define the search criteria for a specific string you wish to find or not find in the monitored file output.
The String Search for Flat and Log files is slightly different. Refer to the **Searching for a Specific String** chapter for detailed information.

- **Numeric Extractions**: Define any numeric value to be extracted from the monitored file output.

7. Click the **Add to List** button to complete the creation of File (Flat or Log) instance.
8. Click **Close**.
Monitoring a File System

Because file systems are often a critical resource for technologies, TrueSight Operations Management - Monitoring Studio provides a File System monitoring tool to rapidly identify which components are impacted when a file system is full.

To monitor a file system:

1. Access the Monitoring Studio Configuration panel, as explained in the Configure Monitors chapter.
2. In the Monitors section, click the File Systems button.
3. The File Systems panel is displayed to provide the connection credentials and define the file system settings.
4. Provide the File System information:
   
   - **File System**: Enter the name of the file system you wish to monitor.

   - **Monitor Settings**:

   
   - **Internal ID**: Enter an ID to identify the managed file system instance in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the managed file system instance in TrueSight Operations Management.
   - **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this file system instance is breached.

5. Configure the File System Cache refreshing frequency. Monitoring Studio relies on a cache mechanism to share the information among the Monitors in order to use as little resources as possible on the target host and over the network. The cache will be refreshed if one of the Monitors needs to collect data (polling interval reached) and the cache is older than the selected minimum cache refresh.
File System Cache

| Minimum Cache Refresh (seconds) | 15 |

- **Minimum Cache Refresh (seconds):** Use the spin button to set the minimum number of seconds Monitoring Studio must wait before refreshing the file system cache. Default is 15 seconds.

7. Click the **Add to List** button to complete the creation of the file system instance.
8. Click **Close**.
Monitoring Folders

Many applications store critical data as files in the filesystem (for example, each pending query is stored as a separate file in a specific folder). In such cases, monitoring the folders (directories) containing these files can prove very useful to detect an abnormal behavior (for example: many files are piling up in the "queue" folder).

The **Folder Monitoring** tool monitors folders (directories) that store files processed by the application. It measures their size, growth and flow (including how many files moved in, how many moved out, etc.).

This feature allows you to:

- Ensure that your application is not overloaded (number of files to be processed, e.g.).
- Measure the application activity (how many removed files, that is, how many have been processed).
- Check the age of the newest file (whether the data is coming in properly...).
- Check the age of the oldest file (whether the application is running late in processing queued files).

⚠️ **Folder monitoring may be time-consuming for large size folders. Therefore, the solution will automatically reduce its monitoring features when the monitored folder contains more than 1000 files. To change this limit, set the Host Advanced Configuration Variables: folderLimit, to a suitable value.**

To monitor a folder:

1. Access the **Monitoring Studio Configuration** panel, as explained in the Configure Monitors chapter.
2. In the **Monitors** section, click the **Folders** button.
3. The **Folders** panel is displayed to provide the connection credentials and define the folder settings.
4. Provide the **Credentials** information:

   ![Credentials](image)

   **Folders — Credentials Settings**

   - **Username**: Enter the username to use to establish the connection with the Host on which the folder you wish to monitor is located. Leave this field blank to use the username provided for the Host at the Monitor Group level.
   - **Password**: Enter the password to use to establish the connection with the Host on which the
folder you wish to monitor is located. Leave this field blank to use the password provided for the Host at the Monitor Group level.

- **Optional — Associated OpenSSH Private Key File Path:** When monitoring remote hosts running UNIX, Linux or other types of operating systems that support SSH authentication key file, you may need to provide an OpenSSH private key file to establish a secured connection with the remote host. Enter the path of the OpenSSH private key file you wish to use to establish a connection with the remote host and enter the optional PassPhrase in the Password field.

  \textbf{The Private Key File should exists on the PATROL Agent node.}

5. Define the **Folder Settings:**

<table>
<thead>
<tr>
<th>Folder Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Folder Path</strong></td>
</tr>
<tr>
<td><strong>Include Subfolders</strong></td>
</tr>
<tr>
<td><strong>Monitor Only Files Matching This Mask</strong></td>
</tr>
<tr>
<td><strong>When Folder Is Empty</strong></td>
</tr>
</tbody>
</table>

- **Folder Path:** Enter the name and path of the folder (directory) you wish to monitor. You can use the ? wildcard to replace one character; the * wildcard to replace one or more characters or a date/time format to dynamically assign the current date or time in the file name. Simply insert the following string in the "Folder Path" field, where the date/time format appears, replacing the three dots by date format symbols: `{SEN_TIME:...}`. For the complete list of format symbols, definitions and examples, please refer to the **Format Symbols** chapter.

  \textit{The folder path supports environment variables (example: `%PATROL_HOME`).}

- **Include Subfolders:** Select this option to monitor all the sub-folders of the above-specified folder.

- **Optional — Monitor Only Files Matching This Mask:** Specify the file types or enter masks to have Monitoring Studio only monitor the corresponding files (e.g.:*.txt;myFiles?.log;file.*). You can use wildcards such as "***" to replace any number of characters, or "?" to replace just one character. You may also use several masks separated by ";". **TrueSight Operations Management - Monitoring Studio** will only take into account the files matching the masks entered.

- **When Folder Is Empty:** Select the option corresponding to the action you want Monitoring Studio to perform on the time-based attributes when the monitored folder is empty (the time-based attributes are: Oldest Modified File Elapsed Time, Last Modified File Elapsed Time and Longest Time File Remains In Folder):
  - **Do not update the time-based attributes:** The attributes values are not updated, and the alerts are not cleared. (Default). In this case, the attributes keep the same value as previously set upon the last collect. If the last value set was within an alarm range, the alert remains active until the value is set again, that is when the folder will no longer be empty.
• **Suspend the time-based attributes**: The attributes values are not set, but any alert is cleared when the folder becomes empty. In this case, the attributes are suspended (i.e. deactivated) and immediately enabled again. The value of these attributes are not updated and the attributes remain offline in TrueSight OM until a new value is set, that is when the folder will no longer be empty. If the attributes were previously in alarm, the alert is not cleared (No PATROL event (STD/9) is triggered).

• **Set the time-based attributes to zero**: The attributes values are reset to zero and all alerts are automatically acknowledged. In this case, the attributes are set to zero as long as the folder remains empty. While the value could be considered improper, it ensures that previous alerts are cleared and that the corresponding PATROL event (STD/9) is triggered (assuming that zero is out of the alarm range).

6. Define the **Monitor Settings**:

![Monitor Settings](image)

- **Internal ID**: Enter an ID to identify the monitored folder instance in TrueSight Operations Management.
- **Display Name**: Enter a name to identify the monitored folder instance in TrueSight Operations Management.
- **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
- **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this managed folder instance is breached.

7. Click the **Add to List** button to complete the creation of the folder instance.
8. Click **Close**.
Leveraging Values from Other KMs' Parameters

The **Multi-Parameter Formula** feature can monitor all the KMs that are loaded in your PATROL Console. This feature more precisely extracts other KMs' Parameter values and computes them with a mathematical formula or a pre-defined PSL function. The returned value can be used to perform additional operations such as converting units, performing correlation, etc.

**Important**

This feature uses PATROL-based terminology and syntax: an 'Attribute' is referred to as a 'Parameter' and a 'Monitor Type' as a 'Class'. Therefore, you need to use the class and parameter names as they appear in the PATROL platform, to build a formula. We recommend you refer to the [Reference Guide](#) of the Monitoring Studio KM for PATROL documentation to retrieve the parameters and classes names you need to use.

The table below provide an example of the Monitor Type/Class and Attribute/Parameter equivalent in both TrueSight Operations Management and PATROL platforms:

<table>
<thead>
<tr>
<th>Monitor Type in TrueSight Operations Management</th>
<th>Class in PATROL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studio Folder</td>
<td>SEN_MS_FOLDER</td>
</tr>
<tr>
<td>Attribute in TrueSight Operations Management</td>
<td>Parameter in PATROL</td>
</tr>
<tr>
<td>File Count</td>
<td>FileCount</td>
</tr>
</tbody>
</table>

The objects created are instances of the *Studio Multi-Parameter Formula* monitor type.

**To create a multi-parameter formula:**

1. Access the **Monitoring Studio Configuration** panel, as explained in the Configure Monitors chapter.
2. In the **Monitors** section, click the **Multi-Parameter Formula** button.
3. The **Multi-Parameter Formula** panel is displayed to define the formula settings.
4. Specify the parameters to use in the formula:
   - **Optional** — Check the **Using Objects Display Names** option to allow the use of instance labels instead of PATROL IDs to define a parameter's path. This option may be particularly useful when an object ID is unknown.
   - Enter up to 10 parameters and their paths in the A to J fields using the following formats:
     - `<class-name>/<instance-name>/<parameter-name>` (if the **Using Objects Display Names** is
not selected).
- `<class-name>/<instance-label>/<parameter-name>` (if the Using Objects Display Names is selected).

5. Define the formula to apply to the monitored parameter(s):
   - **Formula**: Enter the formula you wish to apply to the parameter(s). The parameters are identified by a letter listed in the dialog box. Construct the required formula using:
     - The uppercase letter preceding the parameter (A, B, etc.).
     - Parameter path 'A' is not recorded in TrueSight Operations Management v10.5. To avoid this issue, leave the 'A' field blank and move the parameter paths to B, C, etc.
     - PSL operators: addition (+), subtraction (-), multiplication (*), division (/) concatenation (.), bitwise (&), etc.
     - Predefined PSL functions: `swTableJoin()`, `swGetMinimum()`, `swGetMaximum()`, and `swGetAverage()`.

Here are some examples of use for each predefined PSL function:

**To join two different tables from text parameters:**

The `swTableJoin()` PSL function is used to join two different tables from text parameters:
- `swTableJoin(tableA, separatorsA, keyColumnA, tableB, separatorsB, keyColumnB)`
defaultRightLine, keyType, timeout

where:
- **tableA** is the left table derived from text parameter A
- **separatorsA** are the separators that separate the columns in tableA
- **keyColumnA** is the key column number in tableA used for matching the key columns in tableB
- **tableB** is the right table derived from text parameter B
- **separatorsB** are the separators that separate the columns in tableB
- **keyColumnB** is the key column number in tableB used for matching the key columns in tableA
- **defaultRightLine** (optional) default rightTable line, when a match is not found
- **keyType** (optional) key type such as wbem used by Matsya TableJointClient used by the Java client.
- **timeout** (optional) timeout for the table joint query.

**Example:**
```
swTableJoin(A, "|", 1, B, "|", 1)
```

where:
- **Table 1** (in parameter A: `path1/Result`):
  - key 1|A|B|C|key 2|aa|bb|cc|key 3|1|2|3|
- **Table 2** (in parameter B: `path2/Result`):
  - key 1|X|Y|Z|key 2|xx|yy|zz|key 3|4|5|6|

The returned output is a table that is set to the `Result` text parameter:
```
key 1;A;B;C;key 1;X;Y;Z;key 2;aa;bb;cc;key 2;xx;yy;zz;key 3;1;2;3;key 3;4;5;6;
```

**To find the minimum value among several parameters:**

Use the **swGetMinimum()** PSL function to find the minimum value for the chosen list of parameters (where A and B are number parameters):
```
swGetMinimum([A, B])
```

**To find the maximum value among several parameters:**

Use the **swGetMaximum()** PSL function to find the maximum value for the chosen list of parameters (where A and B are number parameters):
```
swGetMaximum([A, B])
```

**To find the average value among several parameters:**

Use the **swGetAverage()** PSL function to find the average value for the chosen list of
parameters (where A and B are number parameters):

- \( \text{swGetAverage}([A, B]) \)

The returned output of the \( \text{swGetMinimum} \), \( \text{swGetMaximum} \), and \( \text{swGetAverage} \) PSL functions will be displayed by the Value attribute.

- **If the formula or parameters entered are not PSL compatible, an error will be reported in the Group’s CollectionErrorCount parameter.**

- **Do Not Collect When Empty**: Select this option if you wish to skip the collect when one or more parameters is not populated (usually after a PATROL Agent restart).

6. **Define the Monitor Settings**:

   - **Internal ID**: Enter an ID to identify the Multi-Parameter Formula instance in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the Multi-Parameter Formula instance in TrueSight Operations Management.
   - **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this Multi-Parameter Formula instance are breached.

7. **Optional — Define the Content Processing rules you wish to apply to the Multi-Parameter Formula output**:

   - **String Searches**: Define the search criteria for a specific string you wish to find or not find in the Multi-Parameter Formula output.
   - **Numeric Extractions**: Define any numeric value to be extracted from the Multi-Parameter Formula output.

8. Click the **Add to List** button to complete the creation of the Multi-Parameter Formula instance.

9. **Click Close.**
Monitoring Nagios Plugins

Monitoring Studio provides powerful monitoring capabilities to support your existing custom scripts built for Nagios. Through Nagios Plugin Monitors, Monitoring Studio extends the monitoring coverage of your IT environment by supporting any existing Nagios Plugins. This Monitor is designed to integrate with the Nagios server, execute any Nagios Plugins and monitor the results directly from your TrueSight Operations Management environment.

To monitor Nagios plugins:

1. Access the Monitoring Studio Configuration panel, as explained in the Configure Monitors chapter.
2. In the Monitors section, click the Nagios Plugin button.
3. Provide the required information to define the Nagios plugin credentials that you want to use to execute a command line:

<table>
<thead>
<tr>
<th>Credentials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
</tr>
<tr>
<td>Password</td>
</tr>
<tr>
<td>Associated OpenSSH Private Key File Path</td>
</tr>
</tbody>
</table>

Nagios Plugins - Credentials Settings

- **Username**: Enter the username to use to execute the command line.
- **Password**: Enter the password to use to execute the command line.
- **Optional** — **Associated OpenSSH Private Key File Path**: When monitoring remote hosts running UNIX, Linux or other types of operating systems that support SSH authentication key file, you may need to provide an OpenSSH private key file to establish a secured connection with the remote host. Enter the path of the OpenSSH private key file you wish to use to establish a connection with the remote host and enter the optional PassPhrase in the **Password** field.

⚠️ The Private Key File should exists on the PATROL Agent node.

4. Provide the required information to configure the Nagios plugin settings:

<table>
<thead>
<tr>
<th>Nagios Plugin Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nagios Plugin Folder Path</td>
</tr>
<tr>
<td>Nagios Plugin Command</td>
</tr>
<tr>
<td>Run this Command Locally</td>
</tr>
<tr>
<td>Timeout (seconds)</td>
</tr>
<tr>
<td>Report Unknown (3) Service Status</td>
</tr>
</tbody>
</table>

Nagios Plugins — Settings
- **Nagios Plugin Folder Path**: Enter the path to the folder where the Nagios plugin files are stored. For example: `/usr/local/nagios/libexec`.
- **Nagios Plugin Command**: Enter the command line to execute. For example: `check_disk -w 50% -c 30%`.
- **Run this Command Locally**: Check this option if you want the command line to be executed on the PATROL Agent's system and not on the targeted host.
- **Timeout (seconds)**: Enter the time in seconds after which the command will be stopped (Default: 30 seconds). If the timeout is reached, a new error is logged in *Collection Error Count* attribute of the monitored group, indicating that the command failed to execute properly.
- **Report Unknown (3) Service Status**: Select how you want an unknown status to be reported in your TrueSight Operations Management environment:
  - as a new error in the Collection Error Count attribute of the monitored group.
  - as a specific value in the Status attribute.

5. **Configure the Performance Data Settings**:

![Nagios Performance Data Monitoring](image)

- **Value**: Monitored
- **Turn on Default Alerts on "Value"**: Checked
- **Percentage**: Monitored
- **Delta**: Not Monitored
- **Delta Per Second**: Not Monitored
- **Present**: Monitored

6. **Monitor Performance Data**: Check this option if you want each performance data returned by the plugin to be instantiatted and monitored.

7. Indicate if you wish the following attributes to be monitored or not: **Value** (*Optional* — Check the Turn on Default Alerts on "Value" option to use the default alerts for the Value attribute based on the returned performance data), **Percentage**, **Delta**, **Delta Per Second**, **Present**.

8. Click the **Rescaling Value** button to customize the way **Monitoring Studio** will report on the retrieved value:
  - **No Rescaling**: The value remains unchanged and is reported as collected.
  - **Convert to MB**: The value is converted into megabytes.
  - **Convert to GB**: The value is converted into gigabytes.
- **Convert to TB**: The value is converted into terabytes.
- **Divide by the Value Below**: The value is divided by the number you provide.

9. Click the **Missing Data Detection** button to specify a period of time after which Nagios Performance instances with missing performance data will be deleted:
   - **After the Number of Hours Specified Below**: Enter the number of hours after which the instance will be deleted.
   - **Never**: To prevent instances with missing performance data to be deleted.
   - **Immediately**: To delete instances with missing performance data immediately.

10. Define the **Monitor Settings**:

    | Monitor Settings |
    |------------------|
    | **Internal ID**  |
    | **Display Name** |

    *Polling Interval* [ ]  
    *Alert Actions* [ ]

    **Nagios Plugins — Monitor Settings**

    - **Internal ID**: Enter an ID to identify the monitored Nagios Performance Data instance in TrueSight Operations Management.
    - **Display Name**: Enter a name to identify the monitored Nagios Performance Data instance in TrueSight Operations Management.
    - **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
    - **Optional — Alert Actions**: Define the action(s) **Monitoring Studio** needs to perform when the thresholds for this managed process instance is breached.

11. **Optional — Define the Content Processing rules you wish to apply to the command line output**:

    **Analyzing Command Lines — Content Processing Settings**

    - **String Searches**: Define the search criteria for a specific string you wish to find or not find in the command line output.
    - **Numeric Value Extractions**: Define any numeric value to be extracted from the command line output.

12. Click the **Add to List** button to complete the creation of the Nagios Plugin instance.
13. Click Close.

Monitoring Processes

When monitoring a technology, you typically want to check that its processes are running properly; yet spotting specific processes amongst all the running processes can sometimes be challenging.

The Process Monitoring tool provided by TrueSight Operations Management - Monitoring Studio allows you to easily check the presence of Windows, UNIX, or Linux processes by specifying one or more of the following criteria:

- The process name (as it appears in `ps` or in the Windows Task Manager)
- The Command line that was used to spawn the process
- The Username the process is run as
- The PID file path that contains the process ID

*While the first 3 criteria can be combined to identify a process, the PID must always be used alone.*

To monitor a process:

1. Access the Monitoring Studio Configuration panel, as explained in the Configure Monitors chapter.
2. In the Monitors section, click the Processes button.
3. The Processes panel is displayed to define the process monitoring settings and specify the process detection details method.
4. Define the Detection by Process Details information:

```
<table>
<thead>
<tr>
<th>Detection by Process Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Name</td>
</tr>
<tr>
<td>Command Line</td>
</tr>
<tr>
<td>Process Runs as This User</td>
</tr>
</tbody>
</table>
```

*Processes — Detection by Process Details*

- **Process Name**: Enter the name of the process you wish to monitor (including the extension for Windows processes).
- **Command Line**: Specify the command line that launched the process.
- **Process Runs as This User**: Enter the username the monitored process is running as.

Examples of settings for detection by process details

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Processes that match the criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td></td>
</tr>
</tbody>
</table>
### Criteria

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Processes that match the criteria</th>
</tr>
</thead>
</table>
| - Process name MUST BE EXACTLY patrolagent.exe  
- Command-line = <nothing>  
- User Identity = <nothing> | PatrolAgent.exe  
C:\Patrol\PatrolAgent.exe -p 3181  
C:\Patrol\PatrolAgent.exe |

**Example 2**

- Process name MUST BE EXACTLY patrolagent.exe  
- Command-line MUST MATCH THE REGULAR EXPRESSION -[pP] 3181  
- User Identity = <nothing> | C:\Patrol\PatrolAgent.exe -p 3181 |

**Example 3**

- Process name MUST CONTAIN Pat  
- Command-line = <nothing>  
- User Identity = <nothing> | PatrolAgent.exe  
PatProcess.exe  
PatrolPerf.exe |

⚠️ **To monitor all processes of a selected user, enter only the "User Identity" and leave "Process name" and "Command-line" empty.**

⚠️ **Search criteria are case-sensitive on UNIX and Linux.**

ℹ️ **None of the details listed above are mandatory, but at least one of them must be specified.**

5. Or, define the **Detection by PID File** option. Some technologies record their PID (process ID) in a pre-defined file. To make sure these technologies are operating properly, **Monitoring Studio** can read the PID from this file and monitor the corresponding process.

```
Detection by PID File

PID File Path
```

**Processes — Detection by PID File**

- **PID File Path**: Enter the path of the PID file. At each polling, the solution reads this file, retrieves the PID number and checks whether this process PID exists or not. Normally, the process PID is dynamically allocated. The process PID number should be at the very beginning of the file’s content.

6. **Enable/Disable the Child Processes option**:

```
Child Processes

Include Child Processes
```

**Processes — Detection by PID File**

- **Include Child Processes**: Select this option to have **Monitoring Studio** include the child processes associated to the current main process in the calculation of the process performance statistics.
7. Configure the **Status Interpretation** settings (UNIX/Linux only). This feature allows you to qualify the **Status** attribute of a process according to its state. The solution can then trigger an alert according to the **Status** attribute value. For example, if you apply **Suspicious** to the **Status** attribute for the **Stopped** state, an alert (**Warning**) will be triggered when the process is stopped and its **Status** attribute will be set to **Suspicious**.

   - **Optional** — For each of the process possible state, select the value of the **Status** attribute: **OK**, **Suspicious** or **Failed**. By default, the overall process state will be interpreted as listed in the table below:

<table>
<thead>
<tr>
<th>Process State</th>
<th>Attribute Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>OK</td>
</tr>
<tr>
<td>Sleeping</td>
<td>OK</td>
</tr>
<tr>
<td>Queued</td>
<td>OK</td>
</tr>
<tr>
<td>Waiting</td>
<td>OK</td>
</tr>
<tr>
<td>Intermediate</td>
<td>OK</td>
</tr>
<tr>
<td>Stopped</td>
<td>Suspicious</td>
</tr>
<tr>
<td>Growing</td>
<td>Suspicious</td>
</tr>
<tr>
<td>Unknown</td>
<td>Suspicious</td>
</tr>
<tr>
<td>Terminated</td>
<td>Failed</td>
</tr>
<tr>
<td>Not Running</td>
<td>Failed</td>
</tr>
</tbody>
</table>

8. Define the **Monitor Settings**:

   - **Internal ID**: Enter an ID to identify the monitored process in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the monitored process in TrueSight Operations Management.
   - **Optional** — **Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional** — **Alert Actions**: Define the action(s) **Monitoring Studio** needs to perform when the thresholds for this managed process instance is breached.
9. Set the **Process Cache** time. **Monitoring Studio** relies on a cache mechanism to share the information among the Monitors in order to use as little resources as possible on the target host and over the network. The cache will be refreshed if one of the Monitors needs to collect data (polling interval reached) and the cache is older than the selected minimum cache refresh.

<table>
<thead>
<tr>
<th>Process Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Cache Refresh (seconds)</td>
</tr>
</tbody>
</table>

**Processes — Monitor Settings**

- **Minimum Cache Refresh (seconds):** Collected process data are cached to the namespace under each monitored Host instance. These data are shared between other process Monitors under the same Host. A cache most recent than this value will not be refreshed even if a collector tries to refresh it. Enter the number of seconds after which the cache will be refreshed. Default is 15 seconds.

10. Click the **Add to List** button to complete the creation of the process instance.
11. Click **Close**.
Monitoring SNMP Agents

Many devices use SNMP to report their health and operations. They often use SNMP traps to notify failures but the best case is when they embed a true SNMP agent and a documented MIB which specifies the meaning of each SNMP OID. Polling SNMP agents is a good method to ensure that your device is operating properly.

The SNMP Monitor is designed to poll the SNMP agent and retrieve the values of a given OID (object identifier), or the values of an SNMP table, thereby enabling you to be notified when a problem occurs or just be informed of the status of the monitored device/attribute.

SNMP polling supports SNMP v1, v2c, and v3.

To monitor an SNMP agent:

1. Access the Monitoring Studio Configuration panel, as explained in the Configure Monitors chapter.
2. In the Monitors section, click the SNMP button. The SNMP panel is displayed.
3. Define the SNMP Information:
   - Select the version of the SNMP protocol used by the device to be monitored. Possible values are 1, 2c, or 3.
   - Use the spin button to indicate the SNMP port number you wish to use to perform SNMP queries. By default the SNMP queries are performed through port 161.

4. If you selected SNMP Version 1, indicate the SNMP Community string to use to perform SNMP v1 queries:

5. If you selected SNMP Version 2c, indicate:
   - the SNMP Community string to use to perform SNMP v2c queries.
   - the number of seconds Monitoring Studio will wait for the completion of the SNMP polling (default: 120 seconds). This timeout must be long enough to complete the polling of an
entire SNMP table.

**SNMP v2c Connection Settings**

<table>
<thead>
<tr>
<th>Community</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Timeout (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
</tr>
</tbody>
</table>

Using the SNMP Protocol - SNMP v2c Connection Settings

6. If you selected **SNMP Version 3**, indicate:
   - the **Username** to be used to perform the SNMP v3 queries.
   - the **Authentication Protocol** to be used to authenticate the SNMP v3 messages. Possible values are: None, MD5, SHA.
   - the **Authentication password** to be used to authenticate the SNMP v3 messages.
   - the **Privacy protocol** to be used to encrypt SNMP v3 messages. Possible values are: None, AES, DES.
   - the **Privacy password** associated with the privacy protocol.
   - the **Context name** accessible to the SNMP entity.
   - the number of seconds Monitoring Studio will wait for the completion of the SNMP polling (default: 120 seconds). This timeout must be long enough to complete the polling of an entire SNMP table.

**SNMP v3 Connection Settings**

<table>
<thead>
<tr>
<th>Username</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Authentication Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authentication Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Privacy Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Privacy Password</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Context Name</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Timeout (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
</tr>
</tbody>
</table>

Using the SNMP Protocol - SNMP v3 Connection Settings

7. Click the button corresponding to the OID to be polled (**Numeric**) or (**String**), the **SNMP Table**, or the **Traps** to listen:

- Single OID (Numeric)
- Single OID (String)
- Table
- Traps

**Polling SNMP Agents — Configuration Panel**

- **SNMP trap listening only supports SNMP version 1. SNMP v2c and v3 traps are not supported.**

8. Click **Close** to validate.
Polling SNMP Agents from a Single Numeric-based OID

To poll SNMP agents from a single OID (numeric):

1. The Single OID (Numeric) panel allows you to configure the polling of SNMP Agents from one OID (Numeric):

   ![OID Settings](image)

   **OID Settings**
   - **OID**: Enter the OID (object identifier) to poll, as provided by the Management Information Base (MIB).
   - **Monitored Value**: Select the value you wish Monitoring Studio to collect:
     - **Raw Value**: Reports on the actual value collected upon data polling.
     - **Delta**: Reports on the difference between values collected during two consecutive polling.
     - **Delta per second/minute/hour**: Reports on the value resulting of the division of the Delta value by the number of seconds/minutes/hour elapsed between the collection times.
   - **Report Errors in Group’s “Collection Error Count”**: Select this option to have the Collection Error Count attribute of the Group Monitor reflect possible alerts triggered upon the SNMP monitoring execution. The Collection Error Count attribute of the Group reports on the collection errors of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.

2. **OID**: Enter the OID (object identifier) to poll, as provided by the Management Information Base (MIB).

3. **Monitored Value**: Select the value you wish Monitoring Studio to collect:
   - **Raw Value**: Reports on the actual value collected upon data polling.
   - **Delta**: Reports on the difference between values collected during two consecutive polling.
   - **Delta per second/minute/hour**: Reports on the value resulting of the division of the Delta value by the number of seconds/minutes/hour elapsed between the collection times.

4. **Report Errors in Group’s “Collection Error Count”**: Select this option to have the Collection Error Count attribute of the Group Monitor reflect possible alerts triggered upon the SNMP monitoring execution. The Collection Error Count attribute of the Group reports on the collection errors of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.

5. Define the **Monitor Settings**:

   ![Monitor Settings](image)

   **Monitor Settings**
   - **Internal ID**: Enter an ID to identify the managed SNMP Agent in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the managed SNMP Agent in TrueSight Operations Management.
   - **Optional** — **Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional** — **Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this managed SNMP Agent is breached.
6. Click the Add to List button to complete the creation of the SNMP Single OID (Numeric) instance.

Polling SNMP Agents from a Single String-based OID

To poll SNMP agents from a single OID (string):

1. The Single OID (String) panel allows you to configure the polling of SNMP Agents from one OID (String):

   **OID Settings**

   - **OID**: Enter the OID (object identifier) to poll, as provided by the Management Information Base (MIB).
   - **Report Errors in Group’s “Collection Error Count”**: Select this option to have the Collection Error Count attribute of the Group reflect possible alerts triggered upon the SNMP monitoring execution. The **Collection Error Count** attribute of the Group reports on the collection errors of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.

2. Define the **Monitor Settings**:

   **Monitor Settings**

   - **Internal ID**: Enter an ID to identify the managed SNMP Agent in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the managed SNMP Agent in TrueSight Operations Management.
   - **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this managed SNMP Agent is breached.

3. **Optional — Define the Content Processing rules you wish to apply to the SNMP Agent monitoring output:**
### Polling SNMP Agents — Content Processing

- **String Searches**: Define the search criteria for a specific string you wish to find or not find in the SNMP Agent monitoring output.
- **Numeric Extractions**: Define any numeric value to be extracted from the SNMP Agent monitoring output.

4. Click the **Add to List** button to complete the creation of the SNMP Single OID (String) instance.
5. Click **Close**.

## Polling SNMP Agents from an SNMP Table

### To poll SNMP agents from an SNMP table:

1. The **Table** panel allows you configure the polling of SNMP Agents from an SNMP Table:

   - **Root OID**: Enter the Table OID (object identifier) to poll, as given by the Management Information Base (MIB). This OID should always end with ".1".
   - **Column Numbers**: Enter the column numbers in the SNMP table that Monitoring Studio will poll. Column numbers must be delimited by commas (Example: 4,8,9). Leave the field blank to retrieve values from the entire row. Enter "ID" to retrieve the row identifier.
   - **Report Errors in Group’s "Collection Error Count"**: Select this option to have the Collection Error Count attribute of the Group reflect possible alerts triggered upon the SNMP monitoring execution. The Collection Error Count attribute of the Group reports on the collection errors of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.

2. Define the **Monitor Settings**:

   - **Internal ID**: Enter an ID to identify the managed SNMP Agent in TrueSight Operations.
Management.

- **Display Name**: Enter a name to identify the managed SNMP Agent in TrueSight Operations Management.
- **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
- **Optional — Alert Actions**: Define the action(s) **Monitoring Studio** needs to perform when the thresholds for this managed SNMP Agent is breached.
4. **Optional** — Define the **Content Processing** rules you wish to apply to the SNMP Agent monitoring output:

   - **String Searches**: Define the search criteria for a specific string you wish to find or not find in the SNMP Agent monitoring output.
   - **Numeric Extractions**: Define any numeric value to be extracted from the SNMP Agent monitoring output.

5. Click the **Add to List** button to complete the creation of the SNMP Monitor instance.
6. Click **Close**.

### Listening for SNMP Traps

Many devices use SNMP to report their health and operations. They often use SNMP traps to notify failures but the best case is when they embed a true SNMP agent and a documented MIB which specifies the meaning of each SNMP OID. Listening to SNMP traps is a good method to ensure that your device is operating properly.

The **SNMP Trap Listening** tool monitors and listens for SNMP traps and enables rapid recovery actions depending on the traps received, thereby ensuring optimal functioning of applications or devices that send SNMP traps.

- **SNMP trap listening only supports SNMP version 1. SNMP v2c and v3 traps are not supported.**

- **The SNMP Agent emitting the traps should be configured to send them to the PATROL Agent where Monitoring Studio is installed and running, otherwise, the solution will not be able to receive any SNMP trap. No other Trap Listener should be running at the same time.**

### To listen for SNMP traps:

1. Access the **Monitoring Studio Configuration** panel, as explained in the **Configure Monitors** chapter.
2. In the **Monitors** section, click the **SNMP** button.
3. The **SNMP** panel is displayed. Click the **Traps** button to display the **Traps** configuration panel.
4. Provide the two standard SNMP Trap identifiers in the **Traps Settings** section:

   - **Enterprise OID**: 1.3.6.1.4.1.
   - **Trap Number**: 

---

**Listening for SNMP Traps — Traps Settings**
- **Enterprise OID**: Enter the ID of the SNMP Trap. You can use regular expressions.
- **Trap Number**: Enter the number identifying the SNMP Trap.

5. Configure the **Varbinds Settings**. A Varbind or Variable Binding is a sequence of two specific fields. The first field is an OID, which addresses a specific attribute. The second field contains the Value of the specified attribute:

<table>
<thead>
<tr>
<th>Varbind Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Varbind 1 OID</td>
</tr>
<tr>
<td>Varbind 1 Value</td>
</tr>
<tr>
<td>Varbind 2 OID</td>
</tr>
<tr>
<td>Varbind 2 Value</td>
</tr>
</tbody>
</table>

   - **Varbind 1 OID**: Enter the identifier for the first Varbind.
   - **Varbind Value**: Enter the value for the first Varbind.
   - **Varbind 2 OID**: Enter the identifier for the second Varbind.
   - **Varbind2 Value**: Enter the value for the second Varbind.

6. Define the **Acknowledgment Rule** to make Monitoring Studio acknowledge alerts according to the following settings:

<table>
<thead>
<tr>
<th>Acknowledgment Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledge Alert After (minutes)</td>
</tr>
<tr>
<td>When Acknowledging</td>
</tr>
<tr>
<td>Acknowledge Alert if the Following Trap is Received</td>
</tr>
</tbody>
</table>

   - **Acknowledge Alert After (minutes)**: Specify the number of minutes after which you wish Monitoring Studio to automatically acknowledge the alerts. Default is 120 minutes.
   - **When Acknowledging**: Specify the action you wish Monitoring Studio to perform when acknowledging an alert:
     - Select the **Reset "Matching Trap Count"** option to have Monitoring Studio automatically reset the counter of the **Matching Trap Count** attribute to zero.
     - Select the **Reset "Matching Trap Count by One"** option to have Monitoring Studio automatically decrease by one the value of the counter of the **Matching Trap Count** attribute. Use this option if you need the solution to acknowledge each SNMP Trap.
   - **Acknowledge Alert if the Following Trap is Received**: This option enables you to have Monitoring Studio acknowledge an alert for a specific Trap:
     - **Trap Number**: Enter the number identifying the SNMP Trap.
7. Configure the **Alert Action Execution** settings:

<table>
<thead>
<tr>
<th>Alert Actions Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute Alert Actions</td>
</tr>
</tbody>
</table>

- **Execute Alert Actions**: Select the condition on which you wish Monitoring Studio to execute an alert action: **When Thresholds are Breached** or for **Each Matching Trap**.

8. Define the **Monitor Settings**:

- **Internal ID**: Enter an ID to identify the managed SNMP Trap instance in TrueSight Operations Management.
- **Display Name**: Enter a name to identify the managed SNMP Trap instance in TrueSight Operations Management.
- **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this SNMP Trap instance is breached or for each matching trap, according to the option selected in the **Execute Alert Actions** section.

9. Click the **Add to List** button to complete the creation of the SNMP Trap instance.
10. Click **Close**.
Analyzing WBEM Queries

WBEM (Web-Based Enterprise Management) is a set of systems management technologies developed to unify the management of distributed computing environments that provides users with information about the status of local or remote computer systems.

Monitoring Studio can execute WBEM queries on your system and consolidate these queries within your TrueSight environment along with the application monitoring under a single icon. It can also query the WBEM repository for class and instance information.

To analyze a WBEM query:

1. Access the Monitoring Studio Configuration panel, as explained in the Configure Monitors chapter.
2. In the Monitors section, click the WBEM button.
3. The WBEM panel is displayed to provide connection information and credentials.
4. Configure the Connection Settings:

<table>
<thead>
<tr>
<th>Connection Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>WBEM Port Number</td>
</tr>
<tr>
<td>Encryption</td>
</tr>
<tr>
<td>Username</td>
</tr>
<tr>
<td>Password</td>
</tr>
</tbody>
</table>

- **WBEM Port Number**: Enter the port number you wish to use for the connection. By default, in standard environments, the port 5988 is used for non-encrypted data, while port 5989 is used for encrypted data.
- **Encryption**: Select this option to encrypt the query with the HTTPS protocol. HTTPS is a secure version of the Hyper Text Transfer Protocol (HTTP) based on the SSL (Secure Sockets Layer) protocol. SSL creates a secure connection between a client and a server, over which any amount of data can be sent securely.
- **Username**: Enter the username to use to perform the WBEM query. Leave blank to use the username provided at the host level.
- **Password**: Enter the password to use to perform the WBEM query. Leave blank to use the password provided at the host level.
5. Define the **WBEM Query** settings:

   **WBEM Query Settings**

   - **Namespace**: Enter the WBEM namespace (*Example: root\emc*). A namespace is a logical group of related monitor types representing a specific technology or area of management.
   - **WBEM Query**: Enter your query (*Example: SELECT * FROM Win32 process*).
   - **Timeout (seconds)**: Specify the time in seconds after which the query will be stopped (Default: 30 seconds). If the query times out, the **Status** attribute of the Studio WBEM Query Monitor Type will be set to 2 (Failed) and an alarm will be triggered in TrueSight Operations Management.
   - **Report Errors in Group’s “Collection Error Count”**: Select this option to have the **Collection Error Count** attribute of the Group reflect possible alerts triggered upon the WBEM query execution. The **Collection Error Count** attribute of the Group reports on the collection errors of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.

6. Define the **Monitor Settings**:

   **Monitor Settings**

   - **Internal ID**: Enter an ID to identify the managed WBEM query instance in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the managed WBEM query instance in TrueSight Operations Management.
   - **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional — Alert Actions**: Define the action(s) **Monitoring Studio** needs to perform when the thresholds for this WMI query instance is breached.
7. *Optional* — Define the **Content Processing** rules you wish to apply to the WBEM query output:

**Content Processing**

- **String Searches**: Define the search criteria for a specific string you wish to find or not find in the WBEM query output.
- **Numeric Extractions**: Define any numeric value to be extracted from the WBEM query output.

8. Click the **Add to List** button to complete the creation of the WBEM query instance.
9. Click **Close**.
Analyzing Web Requests

The Web Request Analysis tool allows you to extract data from any Web-based interface. You can either monitor the availability of a Web page, extract useful information from a Web-based administration UI, or extract data from a Web service or a REST API. If your IT environment uses a proxy server, please ensure Proxy Settings are properly configured.

⚠️ The Web Request Monitor requires Java 1.8 or higher in specific circumstances:

- When the Web Request requires "NTLM" or "Negotiate" HTTP authentication.
- When the proxy authentication requires "NTLM" or "Negotiate" HTTP authentication.
- When the HTTP server requires at least TLSv1.1 or TLSv1.2 to establish an SSL connection.

ℹ️ Web Requests are always executed locally.

To analyze a Web request:

1. Access the Monitoring Studio Configuration panel, as explained in the Configure Monitors chapter.
2. In the Monitors section, click the Web Requests button.
3. The Web Requests panel is displayed to define the web request settings.

### Web Request Settings

<table>
<thead>
<tr>
<th><strong>URL</strong></th>
<th>http://</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Request Type</strong></td>
<td>GET</td>
</tr>
<tr>
<td><strong>Timeout (seconds)</strong></td>
<td>30</td>
</tr>
<tr>
<td><strong>Report Errors in Group’s “Collection Error Count”</strong></td>
<td>☑</td>
</tr>
</tbody>
</table>

**URL**: Enter the URL of the resource that needs to be polled and monitored. It is possible to poll a secure web site by using the "https" method.

**Request Type**: Select the method to be used:

- **GET**: this mode is the standard way to query a Web page from a Web server.
- **POST (Variables)**: this mode is classically used to post a form to a Web server and obtain the result of the processing of the form data. If you choose this mode you need to specify which variables with what values must be passed to the Web server.
- **POST (Web Service)**: this mode is used to query a Web service (SOAP, REST, etc.). If you choose this mode you need to specify the content-type, header and body that must be passed to the Web server.

**Timeout (seconds)**: Specify the time in seconds after which the request will be stopped (Default: 30 seconds). If the request times out, the **Status** attribute of the Studio Web Request Monitor will be set to 2 (Failed) and an alarm will be triggered in TrueSight Operations Management.

**Report Errors in Group’s "Collection Error Count"**: Select this option to have the Collection Error Count attribute of the Group reflect possible alerts triggered upon the Web Request execution. The Collection Error Count attribute of the Group reports on the collection errors...
of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.

⚠️ Some Web forms can be passed to the server with the HTTP GET method. In this case, the form data goes through the URL (http://server/form.php?varA=valueA&varB=valueB&...).

⚠️ To post a form to a Web server, you need to enter the URL of the script/CGI/page that will actually process the data, which may be different from the Web page URL that shows the form itself. Again, you need to check the Web form HTML source to identify the URL to query.

⚠️ A Web form HTML source should contain a statement like the following one: <FORM METHOD="POST" ACTION="/urlToQuery.php" >.

4. Optional — If you configured Monitoring Studio to go through a proxy for the Web Requests, you can choose to bypass the proxy for this Web Request specifically. This can be useful when the resource is located on the internal network and the proxy refuses to serve it.

![Proxy Settings](image)

**Web Request — Proxy Settings**

- **Bypass Proxy:** Check this option if you want to bypass the proxy server configured under Global Advanced Settings.

5. Specify the **HTTP Authentication** credentials:

![HTTP Authentication](image)

**Web Request — HTTP Authentication**

- **Username:** Enter the username for HTTP authentication.
- **Password:** Enter the password for HTTP authentication.

ℹ️ Refer to the HTTP Authentication chapter for further details.

6. Define the **Form Inputs for POST Requests** (if you have selected the POST (variables from a form) type in the Web Requests Settings section):

![Form Inputs for POST Request](image)

**Web Request — POST Request Settings**

- **Form Input Field Name:** Specify the variable to be transmitted to the Web server.
- **Value:** Specify the value for the specific variable you have entered.
- Click the **Add to List** button to complete the creation of the POST form.
7. Define the **Web Service Information for POST Requests** (if you have selected the **POST (Web Service)** type in the Web Requests Settings section).

   **Web Service Information for POST Requests**

   | Content-Type |   |
   | Header       |   |
   | Body         |   |

   **Web Request — Web Service Information for POST Requests**

   - **Content-Type**: Specify the MIME type for the message and the character encoding used for the XML body of the request.
   - **Optional — Header**: Enter application-specific information (like authentication, payment, etc) about the Web Service message to be sent with the request.
   - **Body**: Enter the body of the Web Service request, or the path to the file (on the PATROL Agent) containing the body of the request. The file path can include environment variables.

   **Example**: `%PATROL_HOME%/tmp/body.txt`

   ```xml
   <?xml version="1.0"?>
   <soap:Envelope
   xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
   soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
   <soap:Body xmlns:m="http://www.example.org/stock">
   <m:GetStockPrice>
   <m:StockName>IBM</m:StockName>
   </m:GetStockPrice>
   </soap:Body>
   </soap:Envelope>
   ```

8. Define the **Monitor Settings**:

   **Monitor Settings**

   | Internal ID |   |
   | Display Name |   |

   **Web Request — Monitor Settings**

   - **Internal ID**: Enter an ID to identify the managed Web Request instance in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the managed Web Request instance in TrueSight Operations Management.
   - **Optional — Polling Interval**: Set the frequency at which the data collection will be
performed. Default is 2 minutes.

- **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this Web Request instance is breached.

9. **Optional — Define the Content Processing rules you wish to apply to the Web Request output:**

![Content Processing](image)

- **Content to BeParsed**: Select what you would like to retrieve from the Web page returned by the Web server: *Entire HTTP Response*, *Web Page with HTML Tags* or *Web Page without HTML Tags*.
- **String Searches**: Define the search criteria for a specific string you wish to find or not find in the Web Request output.
- **Numeric Extractions**: Define any numeric value to be extracted from the Web Request output.

10. Click the **Add to List** button to complete the creation of the Web Request instance.

11. Click **Close**.
Monitoring Windows Event Logs

The **Windows Event Monitoring** tool monitors events posted by your technology to consolidate the monitoring under a single icon and avoid you to look up in the Windows Event Log. It also enables you to define automatic acknowledgment of previously triggered alerts by specifying the Windows event that will acknowledge the alert.

ℹ️ This function is only available to agents running on Windows systems. Windows 2003 cannot be monitored remotely.

To monitor a Windows event log:

1. Access the **Monitoring Studio Configuration** panel, as explained in the *Configure Monitors* chapter.
2. In the **Monitors** section, click the **Windows Event Logs** button.
3. The **Windows Event Logs** panel is displayed. Provide the required Windows event log information.
4. Define the **Event Settings**:

<table>
<thead>
<tr>
<th>Event Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Log Name</td>
</tr>
<tr>
<td>Provider Name</td>
</tr>
<tr>
<td>Count Events with These Event IDs</td>
</tr>
<tr>
<td>But Exclude These Event IDs</td>
</tr>
<tr>
<td>Event Message</td>
</tr>
<tr>
<td>- Must Contain/Must Not Contain: Enter a string or regular expression to look for, and specify whether or not it should be found in the event message.</td>
</tr>
</tbody>
</table>

- **Event Log Name**: Enter the name of the Windows event log you wish to monitor.
- **Provider Name**: Enter the name of the event provider. Typically, the software or driver that triggers the event.
- **Count Events with These Event IDs**: Enter the ID(s) of the event(s) for which *Monitoring Studio* will trigger an alert.
- **But Exclude These Event IDs**: Enter the ID(s) of the event(s) for which *Monitoring Studio* will NOT trigger an alert.

ℹ️ Use a comma (,) to separate several IDs or a hyphen (-) between the first and the last ID to indicate a range. Example: 4372,4375,4380-4385.

- **Event message - Must Contain/Must Not Contain**: Enter a string or regular expression to look for, and specify whether or not it should be found in the event message.
5. Configure the **Event Level** settings. This option allows you to select the Windows Event type you wish to monitor. Available options are: **Critical, Error, Warning, Information**.

   ![Event Level Settings](image)

6. **Optional** — Define the **Acknowledgment Rule**. The automatic acknowledging feature allows you to manage the alerts for the **Matching Event Count** attribute:

   ![Acknowledgment Rule Settings](image)

   - **Acknowledge Alert After (minutes)**: Specify the number of minutes after which you wish **Monitoring Studio** to automatically acknowledge the alerts. Default: 120 minutes.
   - **Acknowledge on These Event IDs**: Enter the ID(s) of the event(s) for which **Monitoring Studio** will automatically acknowledge the alerts.
   - **Event Message - Must Contain/Must Not Contain**: Enter a string or regular expression to look for, and specify whether or not it should be found in the event message.

   **Example**

   6005 is the ID of the event that occurs when the Event log service is started. The first line of the description of such an event is "The Event log service was started." The Event ID and the event description can be used by product support representatives to troubleshoot system problems.

Finally, specify the action you wish **Monitoring Studio** to perform **when acknowledging** an alert:

- Select the **Reset "Matching Event Count"** option to have **Monitoring Studio** automatically reset the counter to zero.
- Select the **Decrease "Matching Event Count by One"** option to have **Monitoring Studio** automatically decrease the value of the counter by one. Use this option if you need the
solution to acknowledge each event count to get a close follow-up on the log activity.

7. Define the Monitor Settings:

- **Internal ID**: Enter an ID to identify the managed event log instance in TrueSight Operations Management.
- **Display Name**: Enter a name to identify the managed event log instance in TrueSight Operations Management.
- **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
- **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this event log instance is breached.

8. Configure the Windows Event Cache refreshing frequency:

- **Minimum Cache Refresh (seconds)**: Use the spin button to set the minimum number of seconds Monitoring Studio must wait before refreshing the event cache. Default is 15 seconds. Monitoring Studio relies on a cache mechanism to share the information among the Monitors in order to use as little resources as possible on the target host and over the network. The cache will be refreshed if one of the Monitors needs to collect data (polling interval reached) and the cache is older than the selected minimum cache refresh time.

9. Click the Add to List button to complete the creation of the Window event log instance.

10. Click Close.
Monitoring Windows Performance Counters

The **Windows Performance Counter** feature is designed to monitor any commercial Windows-based technology or any custom technology relying on a Windows-based middleware which is instrumented through Windows Performance Counters.

The **Windows Performance Counter** Monitor collects information about objects on your Windows systems and measures them. These objects can be processors, threads, processes, memory, etc., with each one having an associate set of counters and possibly instances. The data gathered by the Windows performances about specific components can be used to identify problems and bottlenecks within your technology and plan ahead for your future needs.

The **Windows Performance Counters** tool brings Windows performance data, important to the proper functioning of your applications, within your BMC framework and automatically notifies you when a value breaches a specific threshold.

ℹ️ *This function is only available to agents running on Windows systems.*

To monitor a Windows performance counter:

1. Access the **Monitoring Studio Configuration** panel, as explained in the Configure Monitors chapter.
2. In the **Monitors** section, click the **Windows Perf Counters** button.
3. The **Windows Performance Counters** panel is displayed to specify the performance counter to monitor.
4. Provide the **Performance Counter Settings** information for the Windows performance counter you wish to monitor:

   - **Performance Object**: Enter the name of the Windows performance object that contains the counter you wish to monitor, for example: `Win32_PerfRawData_PerfOS_Processor`.
   - **Instance Name**: Enter the name of the instance you wish to monitor. This name is used to distinguish between multiple performance objects of the same type on a single device, for example: `0`.
   - **Performance Counter**: Enter the name of the performance counter you wish to monitor, for example: `PercentUserTime`.

---

[Image of the Windows Performance Counters — Settings panel with tables for Performance Object, Instance Name, and Performance Counter.]

---

 Sentry Software
1. Refer to our KB article to learn how you can easily get the list of Performance Objects and Performance Counters available for your system.

5. Optional — Select a Rescaling option, if needed. A scale can be used to divide or multiply the Windows performance value by a specific number.

   ![Rescaling](image1)

   Windows Performance Counters — Rescaling Settings

   To do so, simply select the Divide by the Value Below or Multiply by the Value Below option and type in the number you wish the value to be divided/multiplied by. Leave this option set to No Rescaling (default) if you wish the solution to report on the performance counter raw value.

6. Define the Monitor Settings:

   ![Monitor Settings](image2)

   Windows Performance Counters — Monitor Settings

   - **Internal ID**: Enter an ID to identify the monitored performance counter in TrueSight Operations Management.
   - **Display Name**: Enter a name to identify the monitored performance counter in TrueSight Operations Management.
   - **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
   - **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this managed performance counter instance is breached.

7. Click the Add to List button to complete the creation of the performance counter instance.

8. Click Close.
## Monitoring a Windows Service

Many software technologies that run on any Windows operating system run as Windows services, running as background processes with no direct user interface and no logged-on user. Typically, they start automatically and are expected to stay running without human intervention. When one of these critical services fails, many users and external services can immediately be affected. Therefore, making sure that these services are seamlessly running is a key requirement for most system administrators.

> *This function is only available to agents running on Windows systems.*

### To monitor a Windows service:

1. Access the Monitoring Studio Configuration panel, as explained in the Configure Monitors chapter.
2. In the Monitors section, click the Windows Services button.
3. The Windows Services panel is displayed to specify the service to monitor.
4. Provide the Name of the Windows service you wish to monitor.

5. Configure the Status Interpretation settings. This feature allows you to qualify the Status attribute of a Windows service according to its state. The solution can then trigger an alert according to the Status attribute value. For example, if you apply **Failed** to the Status attribute for the Stopped state, an alert (Alarm) will be triggered when the Windows service is stopped and its Status attribute will be set to **Failed**.

- **Optional** — For each of the process possible state, select the value of the Status attribute: **OK**, **Suspicious** or **Failed**. By default, the overall service state will be interpreted as listed in the table below:

<table>
<thead>
<tr>
<th>Service State</th>
<th>Attribute Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>OK</td>
</tr>
<tr>
<td>Paused</td>
<td>Suspicious</td>
</tr>
<tr>
<td>Stopped</td>
<td>Failed</td>
</tr>
<tr>
<td>Pending Start</td>
<td>Suspicious</td>
</tr>
<tr>
<td>Pending Continue</td>
<td>Suspicious</td>
</tr>
<tr>
<td>Pending Pause</td>
<td>Suspicious</td>
</tr>
<tr>
<td>Pending Stop</td>
<td>Suspicious</td>
</tr>
<tr>
<td>Not Installed</td>
<td>Failed</td>
</tr>
<tr>
<td>Unknown</td>
<td>Suspicious</td>
</tr>
</tbody>
</table>
6. Define the **Monitor Settings:**

<table>
<thead>
<tr>
<th><strong>Monitor Settings</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal ID</strong></td>
</tr>
<tr>
<td><strong>Display Name</strong></td>
</tr>
<tr>
<td><strong>Polling Interval</strong></td>
</tr>
<tr>
<td><strong>Alert Actions</strong></td>
</tr>
</tbody>
</table>

**Windows Service — Monitor Settings**

- **Internal ID**: Enter an ID to identify the monitored service in TrueSight Operations Management.
- **Display Name**: Enter a name to identify the monitored service in TrueSight Operations Management.
- **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
- **Optional — Alert Actions**: Define the action(s) **Monitoring Studio** needs to perform when the thresholds for this managed service instance is breached.

7. Click the **Add to List** button to complete the creation of the Windows service instance.
8. Click **Close**.
Analyzing PSL Commands

PSL (PATROL Script Language) is a language developed by BMC for writing complex application discovery procedures, parameters, arbitrary commands, and tasks. This language is commonly used for KM development but can also be useful to IT administrators as they can access the internal information of the PATROL Agent through PSL commands.

The PSL Command Monitor fills the gap of what other Monitors cannot offer you. This feature allows you to run PSL commands on the local PATROL Agent system using the default PATROL Agent credentials. You can then define String Searches and Numeric Value Extractions that will help you parse and analyze the output of PSL commands.

⚠️ The PSL Command Monitor is intended for advanced users with solid experience in PSL.

To create a PSL command:

1. Access the Monitoring Studio Configuration panel, as explained in the Configure Monitors chapter.
2. In the Monitors section, click the PSL Commands button.
3. The PSL Commands panel is displayed. Define the PSL Command Settings:

   - **PSL Command:** Enter the PSL command or the full path to the PSL file that Monitoring Studio will execute. PSL commands are executed on local host, where the PATROL Agent is running.

   ⚠️ The PSL file path can include environment variables (example: %PATROL_HOME%\lib\patrol\patrol\pslfiles.psl).

You can for example execute simple PSL commands to check the health of the PATROL Agent:

   - `system("%PSLPS");`  # Reports the PSL processes and their status
   - `system("%DUMP RUNQ");`  # Reports the list of items scheduled in the run queue
   - `system("%STAT ALL");`  # Reports all Agent memory usage statistics

   - **Timeout (seconds):** Enter the time in seconds after which the PSL Command will be stopped (Default: 30 seconds). When the timeout is reached, Monitoring Studio will consider that the PSL command has failed to execute properly and will set the value of the Status attribute to 2 (Failed). No further analysis will be performed.
4. Define the **Monitor Settings**:

<table>
<thead>
<tr>
<th>Monitor Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal ID</strong></td>
</tr>
<tr>
<td><strong>Display Name</strong></td>
</tr>
</tbody>
</table>

- **Internal ID**: Enter an ID to identify the managed PSL Command instance in TrueSight Operations Management.
- **Display Name**: Enter a name to identify the managed PSL Command instance in TrueSight Operations Management.
- **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
- **Optional — Alert Actions**: Define the action(s) **Monitoring Studio** needs to perform when the thresholds for any attribute of this PSL Command is breached.

5. **Optional — Define the Content Processing rules you wish to apply to the PSL Command output**:

<table>
<thead>
<tr>
<th>Content Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>String Searches</strong></td>
</tr>
<tr>
<td><strong>Numeric Extractions</strong></td>
</tr>
</tbody>
</table>

- **String Searches**: Define the search criteria for a specific string you wish to find or not find in the PSL Command output.
- **Numeric Extractions**: Define any numeric value to be extracted from the PSL Command output.

6. Click the **Add to List** button to complete the creation of the PSL Command instance.
7. Click **Close**.
Analyzing WMI Queries

WMI (Windows Management Instrumentation) is the Microsoft implementation of WBEM (Web Based Enterprise Management) that provides users with information about the status of local or remote computer systems.

TrueSight Operations Management - Monitoring Studio can execute WMI queries on your system and consolidate them within your TrueSight Operations Management environment. It can also query the WMI repository for monitor type and instance information. You can for example request the WMI that returns all the objects representing shut-down events from your desktop system.

To analyze a WMI query:

1. Access the Monitoring Studio Configuration panel, as explained in the Configure Monitors chapter.
2. In the Monitors section, click the WMI button.
3. The WMI panel is displayed. Provide the connection credentials and define the WMI query settings.
4. Provide the WMI Query information:

<table>
<thead>
<tr>
<th>WMI Query Settings</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Namespace</td>
<td>root\cimv2</td>
</tr>
<tr>
<td>WQL Query</td>
<td></td>
</tr>
<tr>
<td>Timeout (seconds)</td>
<td>30</td>
</tr>
<tr>
<td>Report Errors in Group’s “Collection Error Count”</td>
<td></td>
</tr>
</tbody>
</table>

- **Namespace**: Enter the WMI namespace (Example: root\cimv2). A namespace is a logical group of related monitor types representing a specific technology or area of management.
- **WQL Query**: Enter your query (Example: SELECT * FROM Win32 process). In case you need help to build your WMI query, you could download WMI CIM Studio, which is one of the WMI Administrative tools on the Microsoft site.
- **Timeout (seconds)**: Specify the time in seconds after which the query will be stopped (Default: 30 seconds). If the query times out, the Status attribute of the Studio WMI Query Monitor Type will be set to 2 (Failed) and an alarm will be triggered in TrueSight Operations Management.
- **Report Errors in Group’s “Collection Error Count”**: Select this option to have the Collection Error Count attribute of the Group reflect possible alerts triggered upon the WMI query execution. The Collection Error Count attribute of the Group reports on the collection errors of all Monitors, associated to the Group, for which this option is available and selected, providing a global view of the collection errors for the whole Group.
Basic WMI Queries
Queries may be issued against WMI resources using WMI Query Language (WQL). WQL is a subset of SQL designed to retrieve information from WMI. A simple example of a WMI query would be: SELECT * FROM Win32_Process. This retrieves all attributes (the * is used as a wildcard) for all processes currently running on the computer. Win32_Process is the name of the WMI class for Windows processes.
WMI queries of this type are often issued from a script using Windows Script Host or from any technology or tool that can access WMI. Queries retrieve specific information from instances of WMI resources or execute methods against instances to perform such actions as stopping services, or starting processes.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Example code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>SELECT *</td>
<td>Specifies what properties are returned. Typically * is used to simply retrieve all.</td>
</tr>
<tr>
<td>FROM</td>
<td>FROM __InstanceCreationEvent</td>
<td>Specifies the event class to query. This will be the extrinsic or intrinsic event class.</td>
</tr>
<tr>
<td>WHERE</td>
<td>WHERE TargetInstance ISA 'Win32_Process' AND TargetInstance.Name = 'notepad.exe'</td>
<td>Filters the results. For intrinsic events, it will usually include the ISA keyword to specify the class of the TargetInstance. Note: WMI queries support dot characters in the WHERE statement.</td>
</tr>
</tbody>
</table>

5. Define the Monitor Settings:

- **Internal ID**: Enter an ID to identify the managed WMI query instance in TrueSight Operations Management.
- **Display Name**: Enter a name to identify the managed WMI query instance in TrueSight Operations Management.
- **Optional — Polling Interval**: Set the frequency at which the data collection will be performed. Default is 2 minutes.
- **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this WMI query instance is breached.

6. **Optional — Define the Content Processing rules you wish to apply to the WMI query output:**
**Content Processing**

- **String Searches**: Define the search criteria for a specific string you wish to find or not find in the WMI query output.
- **Numeric Extractions**: Define any numeric value to be extracted from the WMI query output.

7. Click the **Add to List** button to complete the creation of the WMI query instance.
8. Click **Close**.
Analyzing Information Sources
Searching for a Specific String

The **String Search** Monitor allows you to run fast and powerful searches for strings on the information sources that you previously configured (flat or log files, output of a Web request or a database query, OID content, etc).

Please note that the **String Search** Monitor works slightly differently on "running sources" (Log files) than on flat sources (flat files, command lines, Web requests, etc.):

- **On "running sources"** (Log files):
  - the strings are searched only in new lines since the last polling. For a String Search in a running source, two graphs are built:
    - Number of matching lines since the last acknowledgment or Number of matches for the current collect.
    - Number of matches per minute since the last polling.
  - you can specify auto-acknowledging strings that will automatically reset the graph to the "number of matches".

    Refer to the **String Searches for log Files** chapter for detailed information.

- **On "flat sources"** (any other source):
  - the strings are searched in the entire source every time (the whole file, the whole standard output, the whole HTTP response, the whole dataset). For a String Search in a flat source, one graph is built: **Number of matches at the current polling**.
  - you cannot specify auto-acknowledging strings since the attribute is recalculated from "0" at each polling.
  - you can specify where information should be searched in the file (n lines, pre-filter, etc.).

To search for a specific string:

1. Log on to **Central Monitoring Administration**.
2. Create (or edit) a **Policy** that will be deployed on the PATROL Agents that share the same specified tag or according to their IP address, hostname, etc.
3. Click the **Monitor Configuration** link and click the **(or** button.
4. In the **Monitoring Solution** field, select **Monitoring Studio**. The related **Monitoring Profile**, **Version** and **Monitor Type** information is automatically displayed.
5. Select the **Monitoring Studio** Monitor Type and click ****.
6. Select the **Monitoring Studio** Monitoring Solution.
7. From a specific **Monitors** panel (command lines, files, queries, etc.), click the **String Searches** button.
8. The **String Searches** panel is displayed to define the String Search settings.
9. Provide the **String Search** information:

   ![String Search Settings](image)

   - **Considered Line Numbers**: By default, the string(s) will be searched for in all the lines of the specified source but you can also specify the line numbers to be scanned. Line numbers are specified as follows:
     - x, y: line x and line y
     - x-y: all lines from x to y inclusive
     - x: only line x
     - x-: all lines from x to the end of the file inclusive
   
   - **Count Lines Matching With**: Enter the regular expression that needs to be found for the line to be counted.
   
   - **But Exclude Those Matching With**: Enter the regular expression that needs to be found for the line NOT to be counted.

10. Define the **Monitor Settings**:

   ![Monitor Settings](image)

   - **Internal ID**: Enter an ID to identify the managed String Search instance in TrueSight Operations Management.
   
   - **Display Name**: Enter a name to identify the managed String Search instance in TrueSight Operations Management.
   
   - **Optional — Alert Actions**: Define the action(s) **Monitoring Studio** needs to perform when the thresholds for this String Search instance is breached.

11. Click the **Add to List** button to complete the creation of the String Search instance.
12. Click **Close**.
String Searches for Log Files

Since a log file is constantly evolving in terms of contents, the String Search option performs slightly differently for log files than for any other sources (see Searching for a Specific String for detailed information about the String Search option for 'flat sources').

To search for a specific string in a log file:

1. Log on to Central Monitoring Administration.
2. Create (or edit) a Policy that will be deployed on the PATROL Agents that share the same specified tag or according to their IP address, hostname, etc.
3. Click the Monitor Configuration link and click the (or ) button.
4. In the Monitoring Solution field, select Monitoring Studio. The related Monitoring Profile, Version and Monitor Type information is automatically displayed.
5. Select the Monitoring Studio Monitor Type and click .
7. Click the Files (Log) button
8. In the list of files, select the file on which you wish to perform a String Search and click the String Searches button. You can also configure a String Search when you create a brand new File (Log) Monitor instance.
9. The String Searches panel is displayed to define the String Search settings.
10. Provide the String Search information:

   ![String Search Settings Table]

   - **Count Lines Matching With**: Enter the regular expression that needs to be found for the line to be counted.
   - **But Exclude Those Matching With**: Enter the regular expression that needs to be found for the line NOT to be counted.
   - **"Matching Lines Count" Report Matches**: Select the period Monitoring Studio will consider for counting the number of lines matching the String Search:
     - **Since Last Acknowledge (Incremental)**: Select this option to count the lines matching the String Search since the last time the Matching Line Count attribute was reset.
     - **In the Current Collect Only**: Select this option to count the lines matching the String Search during the current collect.
11. Define the **Acknowledgment Rule** settings:

- **Acknowledge Alert After (minutes)**: Enter the number of minutes after which **Monitoring Studio** will automatically acknowledge an alert triggered on the Matching Lines Count attribute. Default is 120 minutes.
- **Acknowledge Alert When a Line Matches With**: Enter the string that, if found, will automatically make **Monitoring Studio** acknowledge an alert on the Matching Lines Count attribute.
- **When Acknowledging**: Specify the action you wish **Monitoring Studio** to perform when acknowledging an alert:
  - Select the **Reset "Matching Line Count"** option to have **Monitoring Studio** automatically reset the counter of the Matching Line Count attribute to zero.
  - Select the **Reset "Matching Line Count by One"** option to have **Monitoring Studio** automatically decrease by one the value of the counter of the Matching Line Count attribute. Use this option if you need the solution to acknowledge each event count to get a close follow-up on the log activity.

12. Define the **Alert Actions Execution** criteria:

- **Execute Alert Actions**: Select the condition that needs to be met for the defined alert action to be performed: **When the Thresholds are Reached** or **Every Time a Matching Line is Found**. Note that when using the later option, the solution will perform as much Alert Actions as the number of matching lines found.

13. Define the **Monitor Settings**

- **Internal ID**: Enter an ID to identify the managed String Search instance in TrueSight Operations Management.
- **Display Name**: Enter a name to identify the managed String Search instance in TrueSight Operations Management.
Operations Management.

- **Optional — Alert Actions:** Define the action(s) **Monitoring Studio** needs to perform when the thresholds for this String Search instance is breached.

14. Click the **Add to List** button to complete the creation of the String Search instance.
15. Click **Close**.

**Extracting Numeric Values**

Problems with technologies are not always as simple as a sentence explaining that an "error has occurred." Sometimes, a technology reports its health by providing critical numbers, like a queue length, a processing time, a utilization percentage, etc. These numbers may be reported by the technology through its files, in the output of commands, in a database, in a Web page, etc. **TrueSight Operations Management - Monitoring Studio** extracts these values and reports them as graphs in TrueSight Operations Management. This feature is called **Numeric Value** extraction as its purpose is to extract numeric values from a text input (the output of a command, a Web page, the result of a SQL query, or WBEM query, etc.).

All you need to do is indicate how to find the numeric value(s) within the information source. The basic mechanism is:

1. Specify an information source.
2. Specify with a regular expression the location of the numeric value within the searched lines.
3. Indicate the numeric value’s position in these lines i.e. before/after the string; column number, etc.
4. Indicate which numbers are to be considered if several lines contain the searched numeric value: first value, last value, calculate average; highest value; lowest value.

The numeric value searching function works a bit differently on "running sources" (Log files and never-ending command lines) than on flat sources (flat files, command lines, Web requests, etc.):

- On "running sources" (Log files), the numeric values are searched only in new lines since the last polling.
- On "flat sources" (flat files, command lines, Web requests, database queries), the numeric values are searched in the entire source (the whole file, the whole standard output, the whole HTTP response, the whole data-set).

If several values are found, it is possible to select which value should be kept: the last value found, the average of all values, the minimum or maximum or a total of all values. A graph is then built with this value.

**To extract a numeric value:**

1. Log on to **Central Monitoring Administration**.
2. Create (or edit) a **Policy** that will be deployed on the PATROL Agents that share the same specified tag or according to their IP address, hostname, etc.
3. Click the **Monitor Configuration** link and click the **Add to List** (or **Edit**) button.
4. In the **Monitoring Solution** field, select **Monitoring Studio**. The related **Monitoring Profile**,
Version and Monitor Type information is automatically displayed.

5. Select the **Monitoring Studio** Monitor Type and click 🖋️.

6. Select the **Monitoring Studio** Monitoring Solution.

7. Select a Monitor Group in the **Monitor Groups List** and click the specific **Monitor** (command lines, files, queries, etc.) for which you want to perform a numeric extraction, then click the **Numeric Extractions** button.

8. The **Numeric Extractions** panel is displayed to define the numeric extractions settings.

### Numeric Extraction Settings

- **Extract Numeric Values From**: Select the option corresponding to the lines that will search for the numeric extraction:
  - **All Lines**: All lines of the text will be scanned for the Numeric value search. There is no need to provide further information.
  - **The Line Numbers Below**: Only specific line numbers will be scanned. The line numbers must be specified in the text field below. Line numbers are specified as follows:
    - `x, y`: line x and line y
    - `x-y`: all lines from x to y inclusive
    - `x`: only line x
    - `x-`: all lines from x to the end of the file inclusive
    
    If the "Skip blank lines" option is selected, empty lines will be ignored in the line-count.
  - **Lines Matching the Regular Expression Below**: The lines scanned will be the lines matching the regular expression specified in the field.

- **The Numeric Values are Located**: Select an option and enter a value in the field provided below:
  - **After/Before the String…**: Indicate if the numeric value is located after or before the string to be entered in the text field.
  - **At the Character Offset…**: The value should be at a specific character offset in the line. Enter the offset number in the box. The solution will look for the numeric value at that exact character offset in the line. If no numeric data is found, no value will be collected.
  - **In the Column Number…**: The previously selected lines contain several columns identified
by a separator character. Enter the column number that should contain the value and click the Column Separators button to specify how columns are separated.

- **Expected Format**: This option allows you to extract numeric values that use blank, comma or points as decimal and thousand separators. Monitoring Studio will look for a number in the location specified previously. When a character that does not match the expected format is found, the parsing stops. So, if the expected format is 1,000 and Monitoring Studio finds 1000, the returned value will be 1.
  
  Select the expected format for the value to be searched.

- **In Case of Multiple Values**: Select the option corresponding to the action you wish the solution to perform when multiple values are extracted:
  
  - **Keep the Last**: Only the latest value found will be kept (default).
  - **Calculate the Average**: The values found will be averaged and only the result will be kept.
  - **Calculate the Sum**: The values found will be summed and only the result will be kept.
  - **Keep the First**: Only the first value found will be kept.
  - **Select the Lowest**: Only the lowest value found will be kept.

- **Monitored Values**: Select the option corresponding to the post-processing action you wish the solution to perform:
  
  - **Raw Value**: The values are reported as collected. No post-processing is performed.
  - **Delta**: The value reported corresponds to the difference between values collected during two consecutive polling.
  - **Delta per second/minute/hour**: The value reported corresponds to the result of the division of the Delta by the number of seconds/minutes/hour elapsed between the collection times.

  *Delta calculation cannot be processed until at least two collects are performed.*

9. Specify the Column Separators options. Click the Column Separators button to display the following panel:

```
  Column Separators

  Blank Space
  Tab
  Semicolon (;)
  Comma (,)
  Pipe (|)
  Other Separators
  Consecutive Separators
  Quotes
```

*Extracting Numeric Value — Column Separator Configuration*

- **Blank Space**: A blank space will be considered as column separator.
- **Tab**: A tab will be considered as column separator.
- **Semicolon (;)**: A semicolon will be considered as column separator.
Comma (,): A comma will be considered as column separator.

Pipe (|): A pipe will be considered as column separator.

Other Separators: Enter the character(s) that will be considered as column separator(s). If you need to provide more than one character, simply type the separators one after another.

Consecutive Separators: Select the way consecutive separator should be processed.

Must be Treated as a Single One: Each consecutive separators must be treated as a single separator.

Mean Empty Columns: Each separator is treated as an individual column separator and the column is considered empty.

Quotes: Indicate how quotes will be considered:

Do Not Interpret Quotes: All applicable separators will be considered as a column separator even when enclosed in quotes.

Consider Text in "Double Quotes" as a Single Column: Numeric values enclosed in double quotes will be considered as a single column. Any separator found within double quotes will not be considered as a column separator.

Consider Text in 'Single Quotes' as a Single Column: Numeric values enclosed in single quotes will be considered as a single column. Any separator found within single quotes will not be considered as a column separator.

Click Close.

10. Specify the Rescaling options. Click the Rescaling button to display the following panel:

Rescaling

Rescaling: Select an option to rescale the value that is being extracted in order to have a more readable graph in TrueSight Operations Management:

No rescaling

Divide the value by a value that then gives you the reading in terms best suited to you

Multiply the extracted value by a constant factor

This can be useful if you extract numeric values in bytes but prefer to show a graph in megabytes. In such a case, you would divide the values by 1048576 (1024*1024).

11. Define the Monitor Settings:

Internal ID: Enter an ID to identify the monitored object.
- **Display Name**: Enter a name to identify the monitored object in the Console.
- **Optional — Alert Actions**: Define the action(s) Monitoring Studio needs to perform when the thresholds for this instance is breached.

12. Click the **Add to List** button to complete the creation of the **Numeric Extraction** instance.
13. Click **Close**.

### Configuring Global Advanced Settings

Configuring Global Advanced Settings consists in:

- configuring the SMTP server
- enabling the debug mode
- configuring the proxy server
- configuring advanced variables

![Global Advanced Settings](image)

### Configuring the SMTP Server

An SMTP server is required to receive alerts by email. If you wish to configure e-mail alert actions, you will have to specify the SMTP server to be used.

**To configure the SMTP server:**

1. In the **Global Advanced Settings** section, click **SMTP Server**.
2. Enter a fully qualified SMTP server host.

![SMTP Server](image)

3. Click **Close**.
Enabling the Debug Mode

When you encounter an issue and wish to report it to Sentry Software, you will be asked to enable the Debug Mode and provide the debug output to the Sentry Software support team.

To enable the debug mode:

1. In the Global Advanced Settings section, click Debug.

2. Check the Enable Debug Mode option. The solution will store debug information in a log file. By default debug files are stored in the %PATROL_HOME%/log or $PATROL_HOME/log folder.

3. In the Debug End Time field, enter the date and time at which the system must stop logging debug information. The required format is: yyyy/mm/dd hh:mm:ss based on a 24 hour-day.

   For the debug mode to be enabled, the Enable Debug Mode must be checked and the Debug End Time must be properly set to a date and time in the future.

4. Click Close to save your settings.

   The following debug files are generated:
   - SEN_MS_debug_km_<PatrolAgent_Port>_YYYY-mm-dd-HH-MM>.log, with the debug output of the KM (example: SEN_MS_debug_km_3181_2016-02-12-17-25.log)
   - SEN_MS_CollectionHub_debug_psl_<PatrolAgent_Port>_YYYY-mm-dd-HH-MM>.log, with the PSL debug output of the Collection Hub (example: SEN_MS_CollectionHub_debug_psl_3181_2016-01-27-09-27.log)
   - SEN_MS_CollectionHub_debug_java_<PatrolAgent_Port>_YYYY-mm-dd-HH-MM>.log, with the Java debug output of the Collection Hub (example: SEN_MS_CollectionHub_debug_java_3181_2016-01-27-09-27.log)

5. In the Add Monitor Types dialog box, click Add.
Configuring the Proxy Settings

In most IT environments, a proxy server is now present and acts as an intermediary between computers and the Internet. In such environments, configuring your proxy settings is required to enable Monitoring Studio to send Web requests to servers located on the Internet. These settings can then be shared or bypassed when configuring the Web Request Monitor.

To configure the proxy server:

1. In the Global Advanced Settings section, click Proxy Settings.

2. Provide the following proxy server information:
   - **Proxy Hostname**: Enter the hostname of the proxy server used to connect to Web sites.
   - **Proxy Port number**: Enter the port to be used to access the proxy server. Default: 3128.

3. Enter the **Username** and **Password** to use for proxy authentication and click **Close**.

   *When logging into TrueSight Operations Management v10.5, please note that if you choose to save your credentials when prompted by your web browser, the Username and Password fields of the proxy will be filled by default with this information.*
Configuring Global Advanced Variables

Advanced configuration variables are used to manually set variables that are normally not available through the standard interface.

⚠️ These variables should only be set when instructed so by Sentry Software Support.

To configure a variable:

1. In the Global Advanced Settings section, click Advanced Variables.

2. In the Advanced Variables section, enter the variables you need to configure and enter the value to be set:

⚠️ The Variable field is case sensitive.
3. Click **Add** to List.
4. Click **Close** to save your settings.

You can easily modify or remove a variable by selecting it in the list and clicking either the **Modify Selection** or the **Remove from List** buttons.
Configuring Other Monitoring Settings
Configuring Thresholds

When you are creating or editing a policy, you can add and configure monitor thresholds. The Add Instance Thresholds dialog box presents threshold configuration fields for TrueSight Operations Management monitors that are configured through Central Monitoring Administration.

For details about baselines and Key Performance Indicators (KPI), see TrueSight Operations Management or Central Monitoring Administration user documentation or refer to the Managing Baselines and Key Performance Indicators chapter. For details about the monitored attributes, refer to the specific monitor type in the Reference Guide chapter.

Before you begin

Ensure that your monitor configuration is complete before thresholds are applied. Thresholds cannot be applied to monitors that are not configured through Central Monitoring Administration. If you have not done so already, please refer to the Configuring Monitor Settings chapter.

Recommended Thresholds

Thresholds define acceptable high and/or low values for the data collected. Thresholds can be created as part of a policy that can be applied to multiple monitor types on multiple BMC PATROL Agents.

The Reference Guide chapter provides a list of monitor types with their respective attributes and the thresholds applied by default.

⚠️ Do not set server thresholds for availability or Boolean oriented parameters or any other parameters that will have events generated for them by the PATROL Agents.

Configuring Monitor Thresholds

To configure monitor thresholds:

1. Log on to Central Monitoring Administration.
2. Edit the Monitoring Policy that applies to the PATROL Agent for which you need to configure thresholds:
   - In the Navigation pane, click the Policies drawer
   - Expand the Monitoring folder and select a policy view (e.g. All).
   - Select your policy and click 📝.
3. Check the Server Threshold Configuration box.
4. Click the Server Threshold Configuration link.
5. In the Server Threshold Configuration page, click.
6. Select the relevant Solution, Version, and Monitor Type.
7. In the Instance Name field, specify the instance to which the threshold configuration will be applied. You can either use a string or the following regular expression patterns: \?, +, *, ( ), |, [ ], { }, ^, $.
8. Check the Match Device Name box if you want the entire <deviceName><instanceName> string to be considered.
9. From the Attribute list, select a monitor attribute.
10. In the Threshold section:
    - Select the threshold type.
    - Set the threshold parameters values.
    - Click Add.
11. Resume the procedure to configure all the monitor thresholds required.
12. Click Close. The configuration details are displayed in the Server Threshold Configuration page.
13. Click Finish.

New threshold configurations are pushed to BMC PATROL Agents with matching Central Monitoring Configuration tags, IP address, hostname, etc.

**Editing a Threshold Configuration**

To edit a threshold configuration:

1. Log on to Central Monitoring Administration.
2. Edit the Monitoring Policy that applies to the PATROL Agent for which you need to edit thresholds:
   - In the Navigation pane, click the Policies drawer
   - Expand the Monitoring folder and select a policy view (e.g. All).
   - Select your policy and click.
3. Click the Server Threshold Configuration link.
4. On the Server Threshold Configuration page, select an Instance Name and click.
5. Edit the threshold configuration values and click Update.
6. When you finish editing the threshold configuration, click Close.
7. Click Finish.

Updated threshold configurations are pushed to BMC PATROL Agents with matching Central Monitoring Configuration tags.
Deleting a Threshold Configuration

To delete a threshold configuration:

1. Log on to **Central Monitoring Administration**.
2. Edit the **Monitoring Policy** that applies to the PATROL Agent for which you need to edit thresholds:
   - In the **Navigation** pane, click the **Policies** drawer
   - Expand the **Monitoring** folder and select a policy view (e.g. **All**).
   - Select your policy and click .
3. Click the **Server Threshold Configuration** link.
4. On the **Server Threshold Configuration** page, select an **Instance Name** and click .
5. Click **Yes** to confirm deletion.
6. After you delete one or more threshold configurations, click **Finish** to save changes to the policy. Deleted threshold configurations are removed from BMC PATROL Agents with matching Central Monitoring Configuration tags.

Defining Alert Actions

**Alert Actions** enable you to choose specific actions to be executed when a failure is detected. With Alert Actions, it is possible to customize the way a problem notification is performed.

**Monitoring Studio** can be configured to run one, several, or all types of Alert Actions when an alert is triggered for a monitored technology. Alert actions can be specified for a Monitor Group or for individual Monitors. The settings and functionality of these two features are similar but, the Group Alert Actions apply to all the Monitors of a group while individual Alert Actions apply to the Monitor for which they are defined.

---

To avoid redundancy, we have chosen to illustrate this chapter with the Group Alert Action panel. Note that the options for configuring the Monitor Alert Actions are identical and operate in the same manner.

To configure group or monitor alert actions:

1. Edit or create the **Monitor Group** or the **Monitor** for which you want to define alert actions.
2. Click the **Group Alert Actions/Alert Actions** button to display the configuration panel.
3. Select one or several alert actions you wish **Monitoring Studio** to perform when a threshold is breached. Macros can be used to customize alert actions, refer to the **Alert Actions Macros** chapter for detailed information:
   - **Event**: select the **Upon Threshold Breach, Trigger an Event** option to have **Monitoring Studio** trigger a PATROL event. Use the **Event Content** field to provide the string that will be displayed with the event.
**Event**

- Upon Thresholds Breach, Trigger an Event
- Event Content: `%{SEN_ALERT_DEFAULTCONTENT}`

- **Annotation**: select the **Upon Threshold Breach, Annotate the Graph** option to have Monitoring Studio annotate the attribute’s graph. Use the **Annotation Content** field to provide the string that will be displayed at the annotation point.

- **Local Command Line (Group Alert Actions)**: Select the **Upon Thresholds Breach, Execute a Command** option to have Monitoring Studio execute a command line on the system where the PATROL Agent is installed. Provide the **Username** and **Password** required to run the command line (or leave these fields blank to use the PATROL Agent’s default account) as well as the **Command Line** you wish the solution to execute.

- **Command Line (Alert Actions)**: Select the **Upon Thresholds Breach, Execute a Command** option to have Monitoring Studio execute a command line. Provide valid credentials to run the command line on the target host as well as the **Command Line** you wish the solution to execute. For remote UNIX/Linux and other platforms, you may need to provide the **Associated OpenSSH private key Path**, and the optional PassPhrase in the **Password** field. Select **Run this Command Locally** to execute the command on the PATROL Agent.

- **E-mail**: Select this option to have Monitoring Studio send an E-mail. Provide the sender and the recipient email address in the **From** and **To** fields. To send the email to multiple recipients, use the comma (,) or the semi-column (;) to separate the recipients' email addresses. The `%{SEN_ALERT_DEFAULTCONTENT}` and `%{SEN_INFORMATION}` macros are
Defining Alert Actions

An SMTP server is required to receive alerts by email. For more information refer to Configuring the SMTP Server.

4. Click Close.

Alert Actions Macros

A macro is a variable whose value is replaced when an Alert Action is triggered. Macros can be used to customize the content of each Alert Action. For example: `%{VALUE}` is replaced by the actual current value of the attribute that triggered the alert.

Each macro listed in the tables below contains information about what triggered the alert. Some macros are "general" or "common" - these can be used for any object, and some are "object-specific" macros that are specific to the object, such as databases or files, etc.

Macro syntax supports white spaces.

Please note that macros are case sensitive and should then always be written in upper case.

General Macros

The macros listed in the table below can be used with alert actions on any object.

<table>
<thead>
<tr>
<th>General Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_ALERT_DEFAULTCONTENT}</td>
<td>Default alert content suitable for events and annotations, which resolves to the following macro syntax: %{SEN_GROUP_NAME} %{SEN_HOST_NAME} %{SEN_PARAMETER_STATUS} of %{SEN_PARAMETER_NAME} of %{SEN_OBJECT_LABEL}.</td>
</tr>
<tr>
<td>%{SEN_ALERT_DEFAULTLOGMESSAGE}</td>
<td>Default alert content suitable for log files, which resolves to the following macro syntax: %SEN_TIME:%Y-%m-%d %H:%M:%S %{SEN_GROUP_NAME} %{SEN_HOST_NAME} %{SEN_PARAMETER_STATUS} of %{SEN_PARAMETER_NAME} of %{SEN_OBJECT_LABEL}.</td>
</tr>
<tr>
<td>%{SEN_CREDENTIALS}</td>
<td>Username of the Host as provided in CMA for a Monitor Group.</td>
</tr>
<tr>
<td>General Macro</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>%{SEN_DATE}</code></td>
<td>Date on which the alert action is performed (YYYY-MM-DD format).</td>
</tr>
<tr>
<td><code>%{SEN_GROUP_CLASS}</code></td>
<td>Class name of the Monitor Group.</td>
</tr>
<tr>
<td><code>%{SEN_GROUP_COLLECTIONERRORS}</code></td>
<td>All collection errors collected for the Monitor Group at the time of the alert action.</td>
</tr>
<tr>
<td><code>%{SEN_GROUP_CONTACT}</code></td>
<td>Contact information for the Monitor Group, if set under PATROL configuration /SENTRY/STUDIO/&lt;groupID&gt;/contact.</td>
</tr>
<tr>
<td><code>%{SEN_GROUP_DESCRIPTION}</code></td>
<td>Description for the Monitor Group, if set under PATROL configuration /SENTRY/STUDIO/&lt;groupID&gt;/description.</td>
</tr>
<tr>
<td><code>%{SEN_GROUP_ID}</code></td>
<td>PATROL identifier of the Monitor Group.</td>
</tr>
<tr>
<td><code>%{SEN_GROUP_LABEL}</code></td>
<td>Name of the Monitor Group triggering the alert action.</td>
</tr>
<tr>
<td><code>%{SEN_GROUP_TYPE}</code></td>
<td>Type of the Monitor Group triggering the alert.</td>
</tr>
<tr>
<td><code>%{SEN_HOST_DOMAIN}</code></td>
<td>Domain of the targeted host.</td>
</tr>
<tr>
<td><code>%{SEN_HOST_FQDN}</code></td>
<td>Fully qualified domain name of the targeted host.</td>
</tr>
<tr>
<td><code>%{SEN_HOST_IPADDRESS}</code></td>
<td>IP address of the targeted host.</td>
</tr>
<tr>
<td><code>%{SEN_HOST_NAME}</code></td>
<td>Name of the targeted host.</td>
</tr>
<tr>
<td><code>%{SEN_HOST_SNMPCOMMUNITY}</code></td>
<td>SNMP community set for the SNMP Agent on the targeted host.</td>
</tr>
<tr>
<td><code>%{SEN_HOST_SYSTEMTYPE}</code></td>
<td>Operating system type of the targeted host.</td>
</tr>
<tr>
<td><code>%{SEN_HOSTNAME}</code></td>
<td>Name of the monitored host as provided in CMA.</td>
</tr>
<tr>
<td><code>%{SEN_INFORMATION}</code></td>
<td>Detailed information about the alert.</td>
</tr>
<tr>
<td><code>%{SEN_INFORMATIONONELINE}</code></td>
<td>Detailed information about the alert in a single line (no carriage return).</td>
</tr>
<tr>
<td><code>%{SEN_NEWLINE}</code></td>
<td>Inserts carriage return.</td>
</tr>
<tr>
<td><code>%{SEN_OBJECT_CLASS}</code></td>
<td>Class name of the object to which the alert action belongs.</td>
</tr>
<tr>
<td><code>%{SEN_OBJECT_ID}</code></td>
<td>PATROL identifier of the object triggering the alert.</td>
</tr>
<tr>
<td>General Macro</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>%{SEN_OBJECT_LABEL}</td>
<td>Display name of the object triggering the alert.</td>
</tr>
<tr>
<td>%{SEN_OBJECT_TYPE}</td>
<td>Type of the object triggering the alert (&quot;Process&quot;, &quot;String&quot;, etc.).</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_ALARM1MAX}</td>
<td>Alarm1 maximum range of the parameter triggering the alert.</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_ALARM1MIN}</td>
<td>Alarm1 minimum range of the parameter triggering the alert.</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_ALARM1NTIMES}</td>
<td>Number of consecutive times the parameter triggering the alert must have a value within the alarm1 range before the alert occurs.</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_ALARM1TYPE}</td>
<td>Alarm alert type of the parameter triggering the alert (OK, WARN, ALARM).</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_ALARM2MAX}</td>
<td>Alarm2 maximum range of the parameter triggering the alert.</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_ALARM2MIN}</td>
<td>Alarm2 minimum range of the parameter triggering the alert.</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_ALARM2NTIMES}</td>
<td>Number of consecutive times the parameter triggering the alert must have a value within the alarm2 range before the alert occurs.</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_ALARM2TYPE}</td>
<td>Alarm2 alert type of the parameter triggering the alert (OK, WARN, ALARM).</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_BORDERMAX}</td>
<td>Border maximum range of the parameter triggering the alert.</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_BORDERMIN}</td>
<td>Border minimum range of the parameter triggering the alert.</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_BORDERNTIMES}</td>
<td>Number of consecutive times the parameter triggering the alert must have a value outside the border range before the alert occurs.</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_BORDERTYPE}</td>
<td>Border alert type of the parameter triggering the alert (OK, WARN, ALARM).</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_NAME}</td>
<td>Display name of the parameter triggering the alert.</td>
</tr>
<tr>
<td>General Macro</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_STATUS}</td>
<td>Status of the parameter.</td>
</tr>
<tr>
<td>% {SEN_PARAMETER_VALUE}</td>
<td>Value of the parameter triggering the alert.</td>
</tr>
<tr>
<td>% {SEN_PARENT_&lt;macro-name-without-SEN_&gt;}</td>
<td>Gets the parent’s object’s macro. The name of the parent’s macro is copied in the syntax without the SEN_. Example: In the case of a String Search performed on a Command Line (parent) output, to read the full output from the alert action in LastMatchingLines, the solution needs to read parent’s % {SEN_RESULT}, which can be accessed using % {SEN_PARENT_RESULT}.</td>
</tr>
<tr>
<td>% {SEN_PARENT_CLASS}</td>
<td>Class name of the parent object to which the alert action belongs.</td>
</tr>
<tr>
<td>% {SEN_PARENT_ID}</td>
<td>PATROL identifier of the object’s parent.</td>
</tr>
<tr>
<td>% {SEN_PARENT_LABEL}</td>
<td>Name of the object’s parent triggering the alert.</td>
</tr>
<tr>
<td>% {SEN_PARENT_TYPE}</td>
<td>Type of the object’s parent triggering the alert (&quot;File&quot;, &quot;CommandLine&quot;, etc.).</td>
</tr>
<tr>
<td>% {SEN_PASSWORD}</td>
<td>Encrypted password of the targeted host.</td>
</tr>
<tr>
<td>% {SEN_RESULT}</td>
<td>Query result received for the monitored object during data collection, when available.</td>
</tr>
<tr>
<td>% {SEN_STATUS INFORMATION}</td>
<td>Provides detailed information about the Status attribute of the monitored object, when available.</td>
</tr>
<tr>
<td>% {SEN_TIME}</td>
<td>Time of the alert action (in HH-MM-SS format).</td>
</tr>
<tr>
<td>% {SEN_TIME:&lt;date-time-format&gt;}</td>
<td>Time of the alert action with a configurable time format as described in the Format Symbols chapter. Example: % {SEN_TIME:%H%M%S} may read 094517 at run time.</td>
</tr>
<tr>
<td>% {SEN_USERNAME}</td>
<td>Username to use to connect to the targeted host.</td>
</tr>
<tr>
<td>%{/...}</td>
<td><strong>Recommended for advanced users only.</strong> Provides an internal instance variable name to be inserted. The path is relative to the object triggering the alert. Example: %{/worstParam} will contain the name of the worst parameter on this instance, which is an application instance built-in variable (see the &quot;PATROL Script Language Reference&quot; document).</td>
</tr>
</tbody>
</table>
Object Specific Macros

The macros listed in the tables below can be used with alert actions specifically for their respective object type.

**Command Line Macros**

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_COMMANDLINE}</td>
<td>Command line being executed and analyzed.</td>
</tr>
<tr>
<td>%{SEN_EXITSTATUSCODE}</td>
<td>Exit status returned by the system after executing the command.</td>
</tr>
</tbody>
</table>

**Database Macros**

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_DATABASENAME}</td>
<td>Name of the database the SQL query is sent to. For example, the database name for SQL Server, or the Oracle SID for Oracle.</td>
</tr>
<tr>
<td>%{SEN_DATABASETYPE}</td>
<td>Type of the database.</td>
</tr>
<tr>
<td>%{SEN_QUERY}</td>
<td>SQL statement sent for execution.</td>
</tr>
</tbody>
</table>

**File Macros**

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_FILENAME}</td>
<td>Name of the monitored file as entered in CMA.</td>
</tr>
<tr>
<td>%{SEN_MONITOREDFILE}</td>
<td>Current file being monitored.</td>
</tr>
</tbody>
</table>

**File System Macros**

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_FILESYSTEM}</td>
<td>Name of the monitored file system.</td>
</tr>
</tbody>
</table>

**Folder Macros**

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_FOLDER}</td>
<td>Name of the folder being monitored.</td>
</tr>
<tr>
<td>%{SEN_OLEDESTREMAININGINFOLDER}</td>
<td>Name of the oldest remaining file in the folder.</td>
</tr>
</tbody>
</table>
## Host Macros

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_AVAILABILITYCHECKS}</td>
<td>List of availability checks, separated by commas.</td>
</tr>
<tr>
<td>%{SEN_CREDENTIALSLIST}</td>
<td>List of credentials, separated by commas.</td>
</tr>
<tr>
<td>%{SEN_SIGNATUREFILES}</td>
<td>List of signatures files, separated by commas.</td>
</tr>
<tr>
<td>%{SEN_TCP_PORT}</td>
<td>Port number used to perform the TCP availability check.</td>
</tr>
</tbody>
</table>

## Multi-Parameter Formula Macro

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_FORMULA}</td>
<td>User-defined formula used to calculate the parameter value.</td>
</tr>
</tbody>
</table>

## Process Macros

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_COMMANDLINE}</td>
<td>Process command line being searched for, as entered in CMA.</td>
</tr>
<tr>
<td>%{SEN_MATCHINGPROCESSES}</td>
<td>List of all the processes that match the search criteria.</td>
</tr>
<tr>
<td>%{SEN_PIDFILE}</td>
<td>Path to the PID file whose corresponding process is being monitored.</td>
</tr>
<tr>
<td>%{SEN_PROCESSNAME}</td>
<td>Process name being searched for, as entered in CMA.</td>
</tr>
<tr>
<td>%{SEN_USERID}</td>
<td>Process user ID being searched for, as entered in CMA.</td>
</tr>
<tr>
<td>% {SEN_WORSTPROCESS_COMMANDLINE}</td>
<td>Command line of the first worst process.</td>
</tr>
<tr>
<td>%{SEN_WORSTPROCESS_NAME}</td>
<td>Name of the first worst process.</td>
</tr>
<tr>
<td>%{SEN_WORSTPROCESS_PID}</td>
<td>PID of the first worst process.</td>
</tr>
<tr>
<td>%{SEN_WORSTPROCESS_PPID}</td>
<td>PPID of the first worst process.</td>
</tr>
<tr>
<td>%{SEN_WORSTPROCESS_STATE}</td>
<td>State of the first worst process.</td>
</tr>
<tr>
<td>% {SEN_WORSTPROCESS_USERNAME}</td>
<td>Username of the first worst process.</td>
</tr>
<tr>
<td>%{SEN_WORSTPROCESSES}</td>
<td>A list of worst processes, semicolon delimited, containing PID, process name, username, PPID, state and command line.</td>
</tr>
</tbody>
</table>

## SNMP Polling Macros

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_CONTENT}</td>
<td>Content of the OID being polled.</td>
</tr>
</tbody>
</table>
### Macros

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_OID}</td>
<td>SNMP OID being polled.</td>
</tr>
</tbody>
</table>

### SNMP Trap Macros

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_CONTENT}</td>
<td>Content of the found trap.</td>
</tr>
<tr>
<td>%{SEN_ENTERPRISEID}</td>
<td>Enterprise ID (OID) of the SNMP traps being looked for.</td>
</tr>
<tr>
<td>%{SEN_FOUNDIP}</td>
<td>Actual originating IP address of the trap that has been received.</td>
</tr>
<tr>
<td>%{SEN_FOUNDTRAPNUMBER}</td>
<td>Actual SNMP trap number that has been received and matches the entered criteria.</td>
</tr>
<tr>
<td>%{SEN_TRAPNUMBER}</td>
<td>SNMP Trap numbers (specific numbers) being looked for.</td>
</tr>
</tbody>
</table>

### String Macros

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_LASTMATCHINGLINE}</td>
<td>Last line that matches with the String search criteria.</td>
</tr>
<tr>
<td>%{SEN_LASTMATCHINGLINES}</td>
<td>Last 10 lines that match with the String search criteria.</td>
</tr>
<tr>
<td>%{SEN_STRING1}</td>
<td>First regular expression being searched for.</td>
</tr>
<tr>
<td>%{SEN_STRING2}</td>
<td>Second regular expressions being searched for.</td>
</tr>
</tbody>
</table>

### Web Request Macros

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_HTTPMETHOD}</td>
<td>Displays whether the GET or POST request type is used to perform the query.</td>
</tr>
<tr>
<td>%{SEN_URL}</td>
<td>URL of the targeted Web page.</td>
</tr>
</tbody>
</table>

### WBEM Macros

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_NAMESPACE}</td>
<td>Namespace of the WBEM query.</td>
</tr>
<tr>
<td>%{SEN_QUERY}</td>
<td>Executed WBEM query.</td>
</tr>
</tbody>
</table>

### Windows Event Macros

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_MATCHINGEVENTS}</td>
<td>List of matching events.</td>
</tr>
</tbody>
</table>
### Macros

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_PROVIDER}</td>
<td>Name of the event provider.</td>
</tr>
<tr>
<td>%{SEN_EVENTID}</td>
<td>Monitored event IDs as configured in the Monitor.</td>
</tr>
<tr>
<td>%{SEN_EVENTLOG}</td>
<td>Name of the monitored event log.</td>
</tr>
<tr>
<td>%{SEN_CONTENT}</td>
<td>Message content of the last matching event.</td>
</tr>
<tr>
<td>%{SEN_RECORDNUMBER}</td>
<td>Last matching event record number.</td>
</tr>
</tbody>
</table>

### Windows Performance Macros

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_PERFORMANCECOUNTER}</td>
<td>Windows performance counter being monitored.</td>
</tr>
<tr>
<td>%{SEN_PERFORMANCEINSTANCE}</td>
<td>Windows performance object instances being monitored.</td>
</tr>
<tr>
<td>%{SEN_PERFORMANCEOBJECT}</td>
<td>Windows performance object name being monitored.</td>
</tr>
</tbody>
</table>

### Windows Service Macros

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_SERVICENAME}</td>
<td>Name of the monitored Windows service.</td>
</tr>
</tbody>
</table>

### WMI Macros

<table>
<thead>
<tr>
<th>Macros</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%{SEN_NAMESPACE}</td>
<td>Namespace of the WMI query.</td>
</tr>
<tr>
<td>%{SEN_QUERY}</td>
<td>Executed WMI query.</td>
</tr>
</tbody>
</table>

### Format Symbols for %{SEN_TIME:...} Macros

The following table lists all of the time formats available for the %{SEN_TIME:...} macro.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>This symbol allows you to use a percent sign (%) in the format of a date string.</td>
</tr>
<tr>
<td>%a</td>
<td>Locale’s abbreviated name of the day of week.</td>
</tr>
<tr>
<td>%A</td>
<td>Locale’s full name of the day of week.</td>
</tr>
<tr>
<td>%b</td>
<td>Locale’s abbreviated name of the month.</td>
</tr>
<tr>
<td>%B</td>
<td>Locale’s full name of the month.</td>
</tr>
<tr>
<td>%c</td>
<td>Locale’s appropriate date and time representation.</td>
</tr>
<tr>
<td>%C</td>
<td>Data and time as %c.</td>
</tr>
<tr>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>%d</td>
<td>Day of month [1,31]; single digits are preceded by 0.</td>
</tr>
<tr>
<td>%D</td>
<td>Date as %m/%d/%y.</td>
</tr>
<tr>
<td>%e</td>
<td>Day of month [1,31]; single digits are preceded by a space.</td>
</tr>
<tr>
<td>%h</td>
<td>Locale's abbreviated name of the month.</td>
</tr>
<tr>
<td>%H</td>
<td>Hour (24-hour clock) [0,23]; single digits are preceded by 0.</td>
</tr>
<tr>
<td>%I</td>
<td>Hour (12-hour clock) [1,12]; single digits are preceded by 0.</td>
</tr>
<tr>
<td>%j</td>
<td>Day of year [1,366]; single digits are preceded by 0.</td>
</tr>
<tr>
<td>%k</td>
<td>Hour (24-hour clock) [0,23]; single digits are preceded by a space.</td>
</tr>
<tr>
<td>%l</td>
<td>Hour (12-hour clock) [1,12]; single digits are preceded by a space.</td>
</tr>
<tr>
<td>%m</td>
<td>Month as a decimal number [1,12]; single digits are preceded by 0.</td>
</tr>
<tr>
<td>%M</td>
<td>Minute [0,59]; leading zero is permitted but not required.</td>
</tr>
<tr>
<td>%n</td>
<td>Insert a new line.</td>
</tr>
<tr>
<td>%p</td>
<td>Locale’s equivalent of either AM or PM.</td>
</tr>
<tr>
<td>%r</td>
<td>Appropriate time representation in 12-hour clock format with %p.</td>
</tr>
<tr>
<td>%R</td>
<td>Time as %H:%M.</td>
</tr>
<tr>
<td>%S</td>
<td>Seconds [0,61].</td>
</tr>
<tr>
<td>%t</td>
<td>Insert a tab.</td>
</tr>
<tr>
<td>%T</td>
<td>Time as %H:%M:%S.</td>
</tr>
<tr>
<td>%u</td>
<td>Day of week as a decimal number [1,7], with 1 representing Monday.</td>
</tr>
<tr>
<td>%U</td>
<td>Week of the year as a decimal number [0,53], with Sunday as the first day of week 1.</td>
</tr>
<tr>
<td>%V</td>
<td>Week of the year as a decimal number [01,53], with Monday as the first day of the week. If the week containing 1 January has four or more days in the new year, then it is considered week 1; otherwise, it is week 53 of the previous year, and the next week, is, week 1.</td>
</tr>
<tr>
<td>%w</td>
<td>Day of week as a decimal number [0,6], with 0 representing Sunday.</td>
</tr>
<tr>
<td>%W</td>
<td>Week of the year as a decimal number [0,53], with Monday as the first day of week 1.</td>
</tr>
<tr>
<td>%x</td>
<td>Locale’s appropriate date representation.</td>
</tr>
<tr>
<td>%X</td>
<td>Locale’s appropriate time representation.</td>
</tr>
<tr>
<td>%y</td>
<td>Year within century [0,99].</td>
</tr>
<tr>
<td>%Y</td>
<td>Year, including the century (for example 1993).</td>
</tr>
<tr>
<td>%Z</td>
<td>Abbreviated or full name of time zone, or no bytes if no information of the time zone exists.</td>
</tr>
<tr>
<td>%Ec</td>
<td>Locale’s alternative appropriate date and time representation.</td>
</tr>
<tr>
<td>%EC</td>
<td>Name of the base year (period) in the locale’s alternative representation.</td>
</tr>
<tr>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>%Ex</td>
<td>Locale's alternative date representation.</td>
</tr>
<tr>
<td>%EX</td>
<td>Locale's alternative time representation.</td>
</tr>
<tr>
<td>%Ey</td>
<td>Offset from %EC (year only) in the locale's alternative representation.</td>
</tr>
<tr>
<td>%EY</td>
<td>Alternative representation of the year in full.</td>
</tr>
<tr>
<td>%Od</td>
<td>Day of the month using the locale's alternative numeric symbols.</td>
</tr>
<tr>
<td>%Oe</td>
<td>Same as %Od.</td>
</tr>
<tr>
<td>%OH</td>
<td>Hour (24-hour clock) using the locale's alternative numeric symbols.</td>
</tr>
<tr>
<td>%OI</td>
<td>Hour (12-hour clock) using the locale's alternative numeric symbols.</td>
</tr>
<tr>
<td>%Om</td>
<td>Month using the locale's alternative numeric symbols.</td>
</tr>
<tr>
<td>%OM</td>
<td>Minutes using the locale's alternative numeric symbols.</td>
</tr>
<tr>
<td>%OS</td>
<td>Seconds using the locale's alternative numeric symbols.</td>
</tr>
<tr>
<td>%OU</td>
<td>Week of the year (Sunday as the first day of the week) using the locale's alternative numeric symbols.</td>
</tr>
<tr>
<td>%Ow</td>
<td>Day of week (Sunday=0) using the locale's alternative numeric symbols.</td>
</tr>
<tr>
<td>%OW</td>
<td>Week of the year (Monday as the first day of the week) using the locale's alternative numeric symbols.</td>
</tr>
<tr>
<td>%Oy</td>
<td>Year (offset from %C) in the locale's alternative representation and using the locale's alternative numeric symbols.</td>
</tr>
</tbody>
</table>
Setting the Polling Interval

A polling interval defines how often new data is collected. A new collect can be performed from once every second, to once in a day. Polling intervals can be set for objects created by TrueSight Operations Management - Monitoring Studio that collect data (files, processes, command lines, SNMP polling etc.). By default, the polling interval is set to 2 minutes on all objects, which can be modified at any time.

To set the polling interval:

1. Access the Monitoring Studio Configuration panel, as explained in the Configure Monitor Settings chapter.
2. In the Monitors section, select the monitoring method for which you need to set the polling interval.
3. The related panel is displayed. Click the Polling Interval button.

⚠️ The option to set polling intervals is not available for string searches, numeric values and SNMP trap instances, since either they do not have collectors, or as in the case of SNMP traps – have collectors that react to events.

4. Configure the polling interval options:

   ![Polling Interval Settings](image)

   5. **Collect**: Select the frequency of the polling:
      - ✓ Every...: This option allows you to specify the number of hours, minutes, and/or seconds at which the polling operation will be performed.
      - ✓ Collect once a day at: This option allows you to specify the time of day at which the polling operation will be performed, by setting the hour (24 hours), minutes and seconds (for example: if you wish the polling to be performed once a day at 2:30 pm, set the timer to: 14:30:00).
      - ✓ Collect once a week on <weekday> at: This option allows you to specify the time of a specific day at which the polling operation will be performed, by setting the hour (24 hours), minutes
and seconds (for example: if you wish the polling to be performed once a week at 2:30 pm, set the timer to: 14:30:00).

6. Click OK.

**Importing an Agent Configuration**

Basically, importing an Agent configuration consists in importing a configuration file (.cfg) that has been previously "exported" from another Agent and saved. The exporting procedure must be performed from a PATROL Console. Refer to the **Backing Up an Entire Monitoring Studio Configuration** or **Exporting a Group Configuration** chapters of the Monitoring Studio KM for PATROL User Documentation available on Sentry Software’s Website.

⚠️ This operation is only possible if you have previously exported an entire Monitoring Studio configuration or a single Monitoring Studio Group configuration (without clearing Host information) from a PATROL Console.

**To import an agent configuration:**

1. Log on to Central Monitoring Administration.
2. In the Navigation pane, click the Policies drawer.
3. Expand the Monitoring folder and select a policy view (e.g. All).
4. Click + to create a Policy that will be deployed on the PATROL Agents that share the same specified Tag or according to their IP address, hostname, etc.
5. Only select the Configuration Variables option.
6. Click the Import button and select the exported Monitoring Studio configuration file (.cfg). This will import the full Monitoring Studio configuration.
7. Recheck the imported configuration values and change them if required.
8. Check that the configuration variable: `/SENTRY/STUDIO/forceClassicConfigMode` is set to 1. If it is missing, simply create it and set it to 1.

⚠️ Please note that after setting the `forceClassicConfigMode` configuration variable to 1, all other configurations will be deactivated in CMA.

9. Deploy the policy.

⚠️ You may import a PATROL Agent configuration (".cfg") with any type of thresholds. If necessary, Monitoring Studio will convert the thresholds to the type of thresholds currently used at the next discovery (within an hour).
Reference Guide
Monitor Types and Attributes

Introduction

This chapter lists all the monitor types and attributes provided by TrueSight Operations Management - Monitoring Studio to monitor your systems.

Please note that depending on the type of managed systems, some attributes may not be available.

Monitor Types

- Studio Command Line
- Studio Database Query
- Studio File
- Studio File System
- Studio Folder
- Studio Group
- Studio Host
- Monitoring Studio
- Studio Multi-parameter Formula
- Studio Nagios Plugin
- Studio Number Extract
- Studio Process
- Studio PSL Command
- Studio SNMP Polling
- Studio SNMP Trap
- Studio String Search
- Studio WBEM Query
- Studio Web Request
- Studio Windows Event
- Studio Windows Performance Counter
- Studio Windows Service
- Studio WMI Query
### Studio Command Line

#### Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution Time*</td>
<td>Time taken by the command to be executed. Value set by commandLineColl.</td>
<td>Seconds</td>
<td>Warning ≥ 30</td>
<td>Response Time</td>
</tr>
<tr>
<td>Exit Code</td>
<td>Exit code returned by the executed command. Value set by commandLineColl.</td>
<td>None</td>
<td>None</td>
<td>Availability</td>
</tr>
<tr>
<td>Exit Status</td>
<td>Status of the command exit code. Value set by commandLineColl.</td>
<td>{0 = Successful ; 1 = Failed}</td>
<td>Alert = 1</td>
<td>Availability</td>
</tr>
<tr>
<td>Result</td>
<td>Displays the return output of the command. Value set by commandLineColl.</td>
<td>n/a</td>
<td>None</td>
<td>--</td>
</tr>
<tr>
<td>Status*</td>
<td>Status of the command line execution. Value set by commandLineColl.</td>
<td>{0 = OK ; 1 = Suspicious ; 2 = Failed}</td>
<td>Warning ≥ 15 Alarm ≥ 60</td>
<td>Availability</td>
</tr>
</tbody>
</table>

For detailed information about KPI, see Managing Baselines and Key Performance Indicators.

* Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.

### Studio Database Query

#### Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution Time*</td>
<td>Time taken by the database query to be executed. Value set by dbQueryColl every 2 minutes.</td>
<td>Seconds</td>
<td>Warning ≥ 15 Alarm ≥ 60</td>
<td>Response Time</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the database query execution. Value set by dbQueryColl every 2 minutes.</td>
<td>{0 = OK ; 1 = Suspicious ; 2 = Failed}</td>
<td>None</td>
<td>Availability</td>
</tr>
</tbody>
</table>

For detailed information about KPI, see Managing Baselines and Key Performance Indicators.

* Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.
### Studio File

#### Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists*</td>
<td>Indicates whether the file exists or not. Value set by fileColl.</td>
<td>{0 = File exists; 1 = File does not exist}</td>
<td>Alarm = 1</td>
<td>Availability</td>
</tr>
<tr>
<td>Growth Percentage</td>
<td>File growth percentage. Value set by fileColl.</td>
<td>Percentage per minute (%/min)</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Growth Speed</td>
<td>File growth speed. Value set by fileColl.</td>
<td>Kilobytes per minute (KB/min)</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Last Changed</td>
<td>Elapsed time since the file was modified. Value set by fileColl.</td>
<td>Minutes</td>
<td>Warning ≥ 7 200 Alarm ≥ 14 400</td>
<td>Statistics</td>
</tr>
<tr>
<td>Size*</td>
<td>File size. Value set by fileColl.</td>
<td>Kilobytes (KB)</td>
<td>Alarm = 0</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.

### Studio File System

#### Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Capacity Percentage*</td>
<td>Percentage of capacity not consumed in the File System. Value set by fileSystemColl.</td>
<td>Percentage (%)</td>
<td>Warning ≤ 10 Alarm ≤ 1</td>
<td>Statistics</td>
</tr>
<tr>
<td>Available Inodes Percentage</td>
<td>Percentage of available inodes on UNIX and Linux platforms. Value set by fileSystemColl.</td>
<td>Percentage (%)</td>
<td>Warning ≤ 10 Alarm ≤ 1</td>
<td>Statistics</td>
</tr>
</tbody>
</table>
### Monitors

**Monitor Types and Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumed Capacity Growth Percentage</td>
<td>Percentage of the capacity that is actually consumed per hour in the File System. Value set by fileSystemColl.</td>
<td>Percentage per hour (%/h)</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Consumed Capacity Growth Speed</td>
<td>Speed at which the capacity is actually consumed in the File System. Value set by fileSystemColl.</td>
<td>Megabytes per hour (MB/h)</td>
<td>None</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.*

### Studio Folder

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deleted File Rate</td>
<td>Displays the number of deleted files per minute. Value set by folderColl.</td>
<td>Files per minutes (Files/min)</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>File Count*</td>
<td>Displays the current number of files in a folder (includes sub-folders, if any, when the option is activated). Value set by folderColl.</td>
<td>File(s)</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Folder Exists*</td>
<td>Indicates whether the folder exists or not since the last collection. Value set by folderColl.</td>
<td>{0 = Folder exists ; 1 = Folder does not exist}</td>
<td>Alarm = 1</td>
<td>Availability</td>
</tr>
<tr>
<td>Growth Percentage</td>
<td>Displays the percentage of the folder size growth per minute since the last polling. Value set by folderColl.</td>
<td>Percentag e per minute (%/min)</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Growth Speed</td>
<td>Displays the folder size growth per minute since the last polling. Value set by folderColl.</td>
<td>Kilobytes per minute (KB/min)</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Last Modified File Elapsed Time</td>
<td>Displays the elapsed time since the last modification of any file in this folder. Value set by folderColl.</td>
<td>Minutes</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Units</td>
<td>Default Alert Conditions</td>
<td>Attribute Type</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>-------</td>
<td>--------------------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>
| Longest Time File Remains In Folder | Displays the longest time an existing file has been placed in the folder.  
*Note: User-defined settings could impact the way Monitoring Studio manages alerts and values for this attribute (see Monitoring Folders for more information).* | Minutes | None | Statistics |
| Modified File Rate | Displays the rate of modified files per minute. Value set by folderColl. | Files per minutes (Files/min) | None | Statistics |
| New File Rate | Displays the rate of new files per minute. Value set by folderColl. | Files per minutes (Files/min) | None | Statistics |
| Oldest Modified File Elapsed Time | Displays the elapsed time since the oldest modification of any file in this folder (or sub-folder, when the option is activated). Value set by folderColl.  
*Note: User-defined settings could impact the way Monitoring Studio manages alerts and values for this attribute (see Monitoring Folders for more information).* | Minutes | None | Statistics |
| Folder Size* | Total size of all the files in the folder (include sub-folders if any) in MB. Value set by folderColl. | Megabytes (MB) | None | Statistics |

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.*

### Studio Group

#### Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection Error Count*</td>
<td>Number of collection problems that occurred on the Host and the related Monitors attached to its group. This attribute is cumulative, each new reported errors increases the value of this</td>
<td>Errors</td>
<td>Alarm ≥ 1</td>
<td>Collecton Status</td>
</tr>
</tbody>
</table>
attribute by one.

The Collection Error Count value will be reset after a given time if no new errors are found. The default timeout is 15 minutes but can be configured with the : /SENTRY/STUDIO/<groupID>/collectionErrorCountAutoAcknowledgeTime variable.

Value set by collectionErrorColl.

* Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.

### Studio Host

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status*</td>
<td>Status of the host availability. Value set by the availabilityCheckColl.</td>
<td>{0 = OK ; 1 = Signature Files not Present ; 2 = Unreachable}</td>
<td>Warning = 1 Alarm = 2</td>
<td>Availability</td>
</tr>
</tbody>
</table>

* Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.

### Monitoring Studio

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection Error Count*</td>
<td>Number of errors that prevent Monitoring Studio from operating properly. This attribute is cumulative, new errors increase the value of the attribute.</td>
<td>Errors</td>
<td>Alarm ≥ 1</td>
<td>Collection Status</td>
</tr>
</tbody>
</table>

* Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.
Monitor Types and Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection Error Count</td>
<td>The Collection Error Count value will be reset after a given time if no new errors are found. The default timeout is 135 minutes but can be configured with the following variable: <code>/SENTRY/STUDIO/collectionErrorCountAutoAcknowledgeTime</code> Value set by discoveryColl.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Debug Status</td>
<td>Indicates whether the debug mode has been enabled or not. Value set by studioColl.</td>
<td>{0 = Off ; 1 = On}</td>
<td>None</td>
<td>Availability</td>
</tr>
<tr>
<td>Discovery Status</td>
<td>Indicates if the global discovery is currently running. Value set by discoveryColl.</td>
<td>{0 = Not Running ; 1 = Running}</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Discovery Time</td>
<td>Time taken to execute the global discovery. Value set by discoveryColl.</td>
<td>seconds</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Host Count</td>
<td>Total number of Hosts managed by the solution. Note: Identical hosts are counted as if they were unique. Value set by studioColl.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor Count*</td>
<td>Total number of Monitors managed by the solution. Value set by studioColl.</td>
<td>Monitors</td>
<td>None</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.

**Studio Multi-parameter Formula**

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value*</td>
<td>Derived (numeric) value from the formula based on the input parameters. Value set by formulaColl.</td>
<td>Value</td>
<td>None</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in BMC TrueSight Operations Management.
**Studio Nagios Performance Data**

### Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delta</td>
<td>Displays the difference between values collected during two consecutive pollings. Value set by nagiosPluginColl.</td>
<td>Delta</td>
<td>None</td>
<td>Delta</td>
</tr>
<tr>
<td>DeltaPerSecond</td>
<td>Displays the value corresponding to &quot;Delta&quot; divided by the elapsed time in seconds between the collection times. Value set by nagiosPluginColl.</td>
<td>Delta/seconds (delta/s)</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Percentage*</td>
<td>Displays the percentage of the Value against the maximum, if a maximum value is found in the performance data. Value set by nagiosPluginColl.</td>
<td>Percent (%)</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Present</td>
<td>Monitors whether the performance object is present in the performance data received. Value set by nagiosPluginColl.</td>
<td>{0 = Found; 1 = NotFound}</td>
<td>Alarm = 1</td>
<td>Availability</td>
</tr>
<tr>
<td>Value*</td>
<td>Value interpreted from the performance data. Value set by nagiosPluginColl.</td>
<td>Value</td>
<td>Set dynamically</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in BMC TrueSight Operations Management*

**Studio Nagios Plugin**

### Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution Time*</td>
<td>Time taken to execute the Nagios plugin. Value set by nagiosPluginColl.</td>
<td>Seconds</td>
<td>Warning ≥ 30</td>
<td>Response Time</td>
</tr>
<tr>
<td>Status*</td>
<td>Exit code returned by the Nagios plugin. Value set by nagiosPluginColl.</td>
<td>{0 = OK ; 1 = Warning ; 2 = Critical}</td>
<td>1 = Warning 2 = Alarm</td>
<td>Availability</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in BMC TrueSight Operations Management*
# Studio Number Extract

## Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value*</td>
<td>Value of the extracted Numeric Value (no value will be reported if no number is found). Value set by the collector of the parent’s object.</td>
<td>Value</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Value Found</td>
<td>States if a numeric value has been found.</td>
<td></td>
<td>None</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.

# Studio Process

## Attributes

<p>| Name                       | Description                                                                 | Units                  | Default Alert Conditions | Attribute Type |
|----------------------------|-----------------------------------------------------------------------------|                       |                          |                |
| Child Count                | Displays the number of child processes of the matching process(es). Value set by processColl. | Processes             | None                      | Statistics     |
| Count*                     | Displays the number of processes that match the criteria. Value set by processColl. | Processes             | Alarm = 0                | Statistics     |
| Handle Count (Windows only)| Displays the number of handles opened by the matching process(es). Value set by processColl. | Handles               | None                      | Statistics     |
| Page Faults Per Seconds (Windows only) | Displays the number of page faults per second caused by the matching process(es). Value set by processColl. | Faults per seconds (Faults/s) | None                      | Statistics     |
| Page File Bytes (Windows only) | Displays the page file used by the matching process(es). Value set by processColl. | Megabytes (MB)         | None                      | Statistics     |
| Private Bytes (Windows only) | Displays the amount of memory that has been allocated by this process and that cannot be shared with others. Value set by processColl. | Megabytes (MB)         | None                      | Statistics     |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor Time*</td>
<td>Displays the processor time percent used by the matching process(es). Value set by processColl.</td>
<td>Percentage (%)</td>
<td>Warning ≥ 100</td>
<td>Statistics</td>
</tr>
<tr>
<td>Status* (UNIX/Linux only)</td>
<td>Worst status of all matching processes. Value set by processColl.</td>
<td>{0 = OK ; 1 = Suspicious ; 2 = Failed}</td>
<td>Warning = 1 Alarm = 2</td>
<td>Availability</td>
</tr>
<tr>
<td>Thread Count (Windows only)</td>
<td>Displays the number of threads of the matching process(es). Value set by processColl.</td>
<td>Threads</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Virtual Bytes</td>
<td>Displays the virtual memory used by the matching process(es). Value set by processColl.</td>
<td>Megabytes (MB)</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Working Set (Windows only)</td>
<td>Displays the working set size of the matching process(es). Value set by processColl.</td>
<td>Megabytes (MB)</td>
<td>None</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

For detailed information about KPI, see Managing Baselines and Key Performance Indicators.

* Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.

## Studio PSL Command

### Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution Time*</td>
<td>Time taken by the PSL command to be executed. Value set by pslCommandColl.</td>
<td>Seconds</td>
<td>Warning ≥ 30</td>
<td>Response Time</td>
</tr>
<tr>
<td>Status*</td>
<td>Status of the execution. Value set by pslCommandColl.</td>
<td>{0 = OK ; 1 = Suspicious ; 2 = Failed}</td>
<td>Warning = 1 Alarm = 2</td>
<td>Availability</td>
</tr>
</tbody>
</table>

* Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.
### Studio SNMP Polling

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution Time*</td>
<td>Time taken by the SNMP polling to be executed. Value set by snmpPollingColl.</td>
<td>Seconds</td>
<td>Warning ≤ 30</td>
<td>Response Time</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the SNMP polling execution. Value set by snmpPollingColl.</td>
<td>{0 = OK; 1 = Suspicious; 2 = Failed}</td>
<td>Warning = 1 Alarm = 2</td>
<td>Availability</td>
</tr>
<tr>
<td>Value*</td>
<td>Value of the SNMP polling returned output. Value set by snmpPollingColl.</td>
<td>Value</td>
<td>None</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

Note: Only applicable to SNMPs of integer type.

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.*

For detailed information about KPI, see [Managing Baselines and Key Performance Indicators](#).

### Studio SNMP Trap

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching Trap Count*</td>
<td>Number of SNMP traps matching the search. Value set by snmpTrapColl.</td>
<td>Traps</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Matching Trap Rate*</td>
<td>Number of SNMP traps per minute matching the search. Value set by snmpTrapColl.</td>
<td>Traps per minute (Traps/min)</td>
<td>None</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.*
### Studio String Search

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching Line Count*</td>
<td>Number of lines matching the string search. Value set by the collector of the parent object.</td>
<td>Lines</td>
<td>Alarm ≥ 1</td>
<td>Statistics</td>
</tr>
<tr>
<td>Matching Line Rate*</td>
<td>Number of lines matching the string search per minute. Value set by the collector of the parent object. The Matching Line Rate attribute is only activated for String searches in log files.</td>
<td>Lines per minutes (lines/min)</td>
<td>None</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.*

### Studio WBEM Query

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution Time*</td>
<td>Time taken by the WBEM query to be executed. Value set by wbemQueryColl.</td>
<td>Seconds</td>
<td>Warning ≤ 30</td>
<td>Response Time</td>
</tr>
<tr>
<td>Status*</td>
<td>Status of the WBEM query execution. Value set by wbemQueryColl.</td>
<td>{0 = OK ; 1 = Suspicious ; 2 = Failed}</td>
<td>Warning = 1 Alarm = 2</td>
<td>Availability</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.*

### Studio Web Request

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution Time*</td>
<td>Time taken by the Web request to be executed. Value set by webRequestColl.</td>
<td>Seconds</td>
<td>Warning ≥ 15 Alarm ≥ 30</td>
<td>Response Time</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Units</td>
<td>Default Alert Conditions</td>
<td>Attribute Type</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------</td>
<td>--------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>HTTP Status*</td>
<td>Web request status. Value set by webRequestColl.</td>
<td>{0 = OK ; 1 = Degraded ; 2 = Failed}</td>
<td>Warning = 1 Alarm = 2</td>
<td>Availability</td>
</tr>
<tr>
<td>HTTP Status Code</td>
<td>Web response Status code. Value set by webRequestColl.</td>
<td>n/a</td>
<td>None</td>
<td>Statistics</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the Web request execution. Value set by webRequestColl.</td>
<td>{0 = OK ; 1 = Suspicious ; 2 = Failed}</td>
<td>Warning = 1 Alarm = 2</td>
<td>Availability</td>
</tr>
</tbody>
</table>

For detailed information about KPI, see Managing Baselines and Key Performance Indicators.

* Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.

### Studio Windows Event

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching Event Count*</td>
<td>Number of Matching Windows Events. Value set by winEventColl.</td>
<td>Events</td>
<td>Alarm ≥ 1</td>
<td>Statistics</td>
</tr>
<tr>
<td>Matching Event Rate*</td>
<td>Rate at which Windows Events are found. Value set by winEventColl.</td>
<td>Events per minutes (events/min)</td>
<td>None</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

* Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.

### Studio Windows Performance Counter

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value*</td>
<td>Value of the monitored Windows Performance counter. Value set by winPerfColl.</td>
<td>Attribut e specific</td>
<td>None</td>
<td>Statistics</td>
</tr>
</tbody>
</table>

* Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.
### Studio Windows Service

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status*</td>
<td>Status of the Windows Service. Value set by winServiceColl.</td>
<td>{0 = OK ; 1 = Suspicious ; 2 = Failed}</td>
<td>Warning = 1 Alarm = 2</td>
<td>Availability</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.

### Studio WMI Query

**Attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Default Alert Conditions</th>
<th>Attribute Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution Time*</td>
<td>Time taken by the WMI query to be executed. Value set by wmiQueryColl.</td>
<td>Seconds</td>
<td>Warning ≤ 30</td>
<td>Response Time</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the WMI query execution. Value set by wmiQueryColl.</td>
<td>{0 = OK ; 1 = Suspicious ; 2 = Failed}</td>
<td>Warning = 1 Alarm = 2</td>
<td>Availability</td>
</tr>
</tbody>
</table>

*Attributes marked with an asterisk are used by default when visualizing the corresponding monitor instance in TrueSight Operations Management.

For detailed information about KPI, see Managing Baselines and Key Performance Indicators.
Managing Baselines and Key Performance Indicators

To detect abnormalities on the monitored environment, BMC TrueSight Operations Management calculates baselines per attribute based on values collected over a specified period of time to determine a normal operating range. When the collected values for these parameters are out of range, an alert is triggered. Some attributes are identified by default as Key Performance Indicators (identified with the 🌐 icon) and automatically included in the baselining calculation.

Managing baselines

The baseline is the expected normal operating range for an attribute of a monitor. There are two baselines: **Baseline High** and **Baseline Low**. **Baseline High** represents the point at which 95% of the weighted average of the historical values fall below this value for the selected time period; **Baseline Low** represents the point at which 90% of the weighted average of historical values for the selected time period fall above this line.

Baselines are generated for KPI attributes that have an active abnormality thresholds.

Managing Key Performance Indicators

Starting from v9.5 of BPPM, attributes that have not been initially designated in the KM as Key Performance Indicators (KPIs) cannot be flagged as KPIs from BPPM/TrueSight. Although enabling baseline is possible through the **Options > Administration > Intelligent Event Thresholds** feature available in the Infrastructure Management Server operator console, BMC **does not** recommend doing it.

⚠️ For more information, refer to the BMC TrueSight Operations Management documentation available from [docs.bmc.com](http://docs.bmc.com).
# Configuration Variables

## Global Advanced Configuration Variables

The following table sums up the configuration variables used by Monitoring Studio globally, i.e. that apply to all of the monitored hosts. These configuration variables are supported in the Advanced Settings under the Global Advanced Settings.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addTimestampToLastMatchingLines</td>
<td>When set to ‘1’, adds a timestamp to the matching lines when stored in the PATROL Agent namespace. Default: unset.</td>
</tr>
<tr>
<td>collectionErrorCountAutoAcknowledgeTime</td>
<td>Number of seconds after which the SEN_MS_MAIN\CollectionErrorCountParameter is reset if no new error is detected. Default: 8100 (2 hours and 15 minutes).</td>
</tr>
<tr>
<td>collectionHubOverrideJavaCommandLine</td>
<td>Command line used by the KM to launch the Java Collection Hub. Default: Not set This variable should only be set if instructed by Sentry Support.</td>
</tr>
<tr>
<td>completeCommandLineOnAIX</td>
<td>When set to '1', the process information is obtained using the standard (system V) &quot;ps&quot; command instead of the custom AIX command.</td>
</tr>
<tr>
<td>dfCommand</td>
<td>df command executed on Linux and UNIX systems to retrieve file systems information.</td>
</tr>
<tr>
<td>dfCommand2</td>
<td>df command executed to retrieve filesystem inode usage information for the given Linux and UNIX systems.</td>
</tr>
<tr>
<td>disableI2D</td>
<td>When set to 1, no MetaTokenID and no MetaFQDN information will be set in the PATROL namespace for any host.</td>
</tr>
<tr>
<td>disableJRECheck</td>
<td>To disable the validation tests of the JRE used by the KM to run Java code and therefore force the KM to use a non-Sun or non-Oracle JRE.</td>
</tr>
<tr>
<td>disablePslExecuteBugWorkaround</td>
<td>When set to ‘1’, disables the workaround in the KM for a bug in the PslExecute() PSL function. If the KM detects that the version of the PATROL Agent is affected by the PslExecute() bug, it uses an alternate technique to create asynchronous threads with the event_trigger() function and the RemPsl standard event. The disablePslExecuteBugWorkaround variable disables this workaround. Default: Not set.</td>
</tr>
<tr>
<td>doNotUsePslExecuteForChildren</td>
<td>When set to ‘1’, string searches, numeric value extractions, dynamic objects and text transforms collects are not called through a PSL execute call. This should be used only if the PATROL Agent is affected by the PSL execute bug.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>fileFindCommand</td>
<td>On Linux and UNIX systems, command line to retrieve all files matching the monitored file mask. Default: find %{FILENAME} -prune.</td>
</tr>
<tr>
<td>fileFindCommandOneDay</td>
<td>On Linux and UNIX systems, command line to retrieve the list of files matching the monitored file mask that were created or modified less than 24 hours ago. Default: find %{FILENAME} -prune -mtime -1.</td>
</tr>
<tr>
<td>fileFindCommandSevenDays</td>
<td>On Linux and UNIX systems, command line to retrieve the list of files matching the monitored file mask that were created or modified less than 7 days ago. Default: find %{FILENAME} -prune -mtime -8.</td>
</tr>
<tr>
<td>folderLimit</td>
<td>Maximum number of files that should be analyzed when monitoring a Folder. Default: 1000.</td>
</tr>
<tr>
<td>folderUnixLsCommand</td>
<td>On Linux and UNIX systems, command line to retrieve all files in the monitored folder. Default: /bin/ls -atp1 %{FOLDERPATH}.</td>
</tr>
<tr>
<td>folderUnixRecursiveLsCommand</td>
<td>On Linux and UNIX systems, command line to retrieve all files in the monitored folder, including sub-folders. Default: /bin/ls -atpR1 %{FOLDERPATH}.</td>
</tr>
<tr>
<td>forceSnmpSerialization</td>
<td>When set to '1', forces the serialization of the SNMP requests. Several SNMP requests may be sent at the same time on multi-processor computers. Some poorly written SNMP agents may not support this. Default: Not set.</td>
</tr>
<tr>
<td>httpRequestConnectionType</td>
<td>HTTP request connection method: sopen or telnet. Default: auto-detect.</td>
</tr>
<tr>
<td>javaPassword</td>
<td>Password used for running java command. By default the default PATROL user account is used.</td>
</tr>
<tr>
<td>javaPath</td>
<td>Path of the java binaries. By default it is discovered automatically.</td>
</tr>
<tr>
<td>javaUsername</td>
<td>Username used for running java command. By default the default PATROL user account is used.</td>
</tr>
<tr>
<td>LastMatchingLinesNumber</td>
<td>Maximum number of matching lines that should be stored in the PATROL Agent namespace. Default: 50.</td>
</tr>
<tr>
<td>maxConcurrentHostDiscoveryThreads</td>
<td>Maximum number of concurrent host discovery threads. Default: 50.</td>
</tr>
<tr>
<td>maxConcurrentHostSSHConnections</td>
<td>Maximum number of concurrent SSH connections to a host. Default: 10.</td>
</tr>
<tr>
<td>maxConcurrentTcpChannels</td>
<td>Maximum number of concurrent socket channels open when performing a host TCP availability check. Default: 10.</td>
</tr>
<tr>
<td>maxConcurrentWmiQueries</td>
<td>Maximum number of concurrent WMI queries. Default: 10.</td>
</tr>
</tbody>
</table>
### Configuration Variables

#### TrueSight Operations Management - Monitoring Studio 9.3.00

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxFileSizeRead</td>
<td>Maximum numbers of characters that should be read from a LOG file in a single collect to perform string searches, numeric value extractions, etc. Remaining text will be read at the next collect. Default: 33554432 bytes (32MB).</td>
</tr>
<tr>
<td>maxParameterValueLength</td>
<td>Maximum number of characters a text parameter should contain. Once this limit is reached, the value is truncated. Default: 1048576.</td>
</tr>
<tr>
<td>newerFileFindCommand</td>
<td>On Linux and UNIX systems, command line to retrieve all files matching the monitored file mask that are newer than the currently monitored file. Default: <code>find %{FILENAME} -prune -newer %{MONITOREDFILE}</code>.</td>
</tr>
<tr>
<td>psCommand</td>
<td>Ps command executed on Linux and UNIX systems to retrieve process information.</td>
</tr>
<tr>
<td>psCommand2</td>
<td>Additional &quot;ps&quot; command executed on Linux and UNIX systems to retrieve additional process information.</td>
</tr>
<tr>
<td>psCommand3</td>
<td>Ps command executed on Linux and UNIX systems to retrieve process performance data.</td>
</tr>
<tr>
<td>restartFromStartUponFileSizeDecrease</td>
<td>When set to '1', LOG files are read from the start if their size decreases, indicating the file was purged. Default: 1.</td>
</tr>
<tr>
<td>temporaryFolder</td>
<td>Path to the folder where temporary files used by the KM are stored. By default, this is set to C:\Windows\Temp on Windows based PATROL Agents and /var/tmp for UNIX/Linux based Agents.</td>
</tr>
<tr>
<td>ThresholdsManagementMode</td>
<td>Threshold management mode: tuning or as. Default: auto-detect.</td>
</tr>
<tr>
<td>wmiQueryColumnSeparator</td>
<td>When multiple values are returned by a WMI query, separator used to split the values. Default: '.</td>
</tr>
</tbody>
</table>

#### Host Advanced Configuration Variables

The following table sums up the configuration variables used by Monitoring Studio for each monitored system.

These configuration variables are supported in the **Host Advanced Settings** under **Host Settings**.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>completeCommandLineOnAIX</td>
<td>When set to '1', process information on the given host is obtained using the normal (system V) &quot;ps&quot; command instead of the custom AIX command.</td>
</tr>
<tr>
<td>dfCommand</td>
<td>Df command executed to retrieve filesystems information for the given Linux and UNIX systems.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>dfCommand2</td>
<td>df command executed to retrieve filesystem inode usage information for the given Linux and UNIX systems.</td>
</tr>
<tr>
<td>fileFindCommand</td>
<td>On Linux and UNIX systems, command line to retrieve all files matching the monitored file mask. Default: find %{FILENAME} -prune.</td>
</tr>
<tr>
<td>fileFindCommandOneDay</td>
<td>On Linux and UNIX systems, command line to retrieve the list of files matching the monitored file mask that were created or modified less than 24 hours ago. Default: find %{FILENAME} -prune -mtime -1.</td>
</tr>
<tr>
<td>fileFindCommandSevenDays</td>
<td>On Linux and UNIX systems, command-line to retrieve the list of files matching the monitored file mask that were created or modified less than 7 days ago. Default: find %{FILENAME} -prune -mtime -8.</td>
</tr>
<tr>
<td>folderLimit</td>
<td>Maximum number of files that should be analyzed for the given host when monitoring a folder. Default: 1000.</td>
</tr>
<tr>
<td>maxFileSizeRead</td>
<td>Maximum numbers of characters that should be read from a LOG file in a single collect to perform string searches, numeric value extractions, etc. Remaining text left will be read at the next collect. Default: 33554432 bytes (32MB).</td>
</tr>
<tr>
<td>newerFileFindCommand</td>
<td>On Linux and UNIX systems, command line to retrieve all files matching the monitored file mask that are newer than the currently monitored file. Default: find %{FILENAME} -prune -newer %{MONITOREDFILE}.</td>
</tr>
<tr>
<td>psCommand</td>
<td>ps command executed to retrieve process information for the given Linux or UNIX host.</td>
</tr>
<tr>
<td>psCommand2</td>
<td>Additional ps command executed to retrieve additional process information for the given Linux or UNIX host.</td>
</tr>
<tr>
<td>psCommand3</td>
<td>ps command executed to retrieve process performance data for the given Linux and UNIX systems.</td>
</tr>
<tr>
<td>restartFromStartUponFileSizeDecrease</td>
<td>When set to ‘1’, LOG files on this host are read from the start if their size decreases, indicating the file was purged. Default: 1.</td>
</tr>
</tbody>
</table>
About...  

About HTTP Authentication

The HTTP authentication is a login/password-based mechanism implemented in the HTTP protocol itself. A Web server that requires an HTTP authentication will display a dialog box in a Web browser.

⚠️ HTTP authentication has nothing to do with an authentication system of a Web page with a form asking for user credentials like any public Web mail service, for example.

There are several HTTP authentication schemes, depending on the way the proxy server has been configured. Here is the list of HTTP authentication supported by Monitoring Studio:

- Basic
- Digest
- NTLM (Windows-integrated)
- Negotiate

For Basic HTTP authentication, the password is sent in a Base64-encoded form and is therefore very easily decoded.

⚠️ Please note that Monitoring Studio requires Java 1.8 (or higher) for HTTP authentication to perform properly.

The proxy authentication also supports Basic, Digest, NTLM, and Negotiate authentication schemes. It can be configured for accessing Web sites through the **Proxy Settings** in the **Global Advanced Settings** section.

About Monitor Internal Identifiers

In TrueSight Operations Management - Monitoring Studio, all objects are identified with a unique ID. When configuring Monitoring Studio using a third-party tool, it is important to have a clear understanding of the ID formats.

There are four different ID formats which vary according to the type of objects:

Group ID

The Group ID is the internal identifier as entered when configuring the **Monitor Group Settings** in the **Monitoring Studio Configuration** panel. It is referred to as `<groupId>`.
Host ID

The Host ID is referred to as `<hostID>` and always follows this format: `<groupId>@<hostname>`
where:

- `<groupId>` is the Internal ID as entered when configuring the Group Settings.
- `<hostname>` is the Hostname/IP address/FQDN of the host on which the technology you wish to monitor is running.

Monitor ID

For all Monitors attached to a Host, the Monitor ID format is as follows:

```
<groupId>@<hostID>:<monitorType>:<monitorID>
```

where:

- `<groupId>` is the Internal ID as entered when configuring the Monitor Group Settings.
- `<hostID>` is the `<groupId>@<hostname>` (Hostname/IP address/FQDN).
- `<monitorType>` is the type of Monitor (Command Line, Web Request, Folder, Process, etc.).
- `<monitorID>` is the internal ID as entered when configuring the Monitor Settings.

**Example**

If you configure a File Monitor with the ID "myFile" under the Host "myHost" which belongs to the Group "myGroup", the complete ID of the instance of the File monitoring type will be:

```
myGroup@myHost:File:myFile
```

Monitor Children ID

For String Search and Numeric Value Extraction tools, the Monitor Children ID format is as follows:

```
<parentID>:<monitorType>:<monitorID>
```

where:

- `<parentID>` is the `<groupId>@<hostID>:<monitorType>:<monitorID>`.
- `<monitorType>` is the type of Monitor: String Search or Numeric Value Extraction.
- `<monitorID>` is the internal ID as entered when configuring the Monitor Settings.

**Example**

If you configure a String Search Monitor with the ID "myStringSearch" under the File "myFile" which belongs to the Host "myHost" which itself belongs to the Group "myGroup", the complete ID of the instance of the String Search monitoring type will be:

```
myGroup@myHost:File:myFile:StringSearch:myStringSearch
```
About Processes

What’s a process

In practice, a process is basically a binary code being executed by processors. Processes are launched by the operating system (since the operating system controls the execution flow) and have several properties: PID (unique identifier of a process); Name; User ID; Command line that was used to launch the process (arguments passed to the binary); Environment; CPU and memory usage; Other various OS-specific properties.

How to identify a process

When you monitor an application, you typically want to check that the application's processes are running properly. The problem lies in how to identify the processes of this application, how to recognize them amongst all of the running processes.

The only thing that really identifies a process is its PID (Process ID). But since the PID is an integer number randomly set upon the process startup, most often we cannot use it to identify the processes of an application (unless the application gives you its PID in a so-called PID file).

In general, you recognize application processes by their name if this criterion is enough to distinguish them from other processes. If the name of the process is not sufficient, you can identify application processes by parsing the process's command lines. This is typically useful with scripts and java processes, whose process names are the same: java, CSCRIPT.EXE, etc.

Process name

Under Windows, the name of a process is basically the file name of the binary file which is being executed: Java.exe, lisAdmin.exe. It always includes the "\.EXE" extension. Process names can easily be shown in Windows Task Manager.

Under UNIX, the process name could be either the file name of the binary being executed, including the path or not, or something completely different (e.g. Oracle processes).

⚠️ The naming of processes is highly platform dependent. Linux processes are not named in the same way as on HP-UX servers, for example. Under UNIX, process names can be shown by executing the "ps –e –o name" command line.

Process command line

Every process is launched through a command line, which consists of the file path to the binary
which has to be executed, and arguments that have to be passed to the binary: <path to the binary file> <argument1> <argument2> etc.

If the directory of the binary file is in the PATH environment variable, the path may not be included in the command line: <binary file name> <argument1> <argument2> etc.

This is the only way to distinguish Java processes and scripts from others, because their process names are all identical (Java.EXE). Unfortunately, in Windows, there is no easy way to see the command lines of the currently running processes. Under UNIX, processes command lines can be shown by executing the "ps –e –o comm" command.

**Process user ID**

On both Windows and UNIX systems, processes run "as" a user. Depending on this, the process may be allowed to access various system resources (files, network, databases, etc.). In secured environments, most applications processes have to run as a specific user to let them access the application resources. If the processes run as another user, the application is very likely to fail and not run properly. This is why it could be important to check that the processes of the application you want to monitor are running as the appropriate user.

**PID file**

A classic way for applications to indicate they are running is to write the PID of their process into a given file. In this case we only need to read this file and check whether the PID written in the file corresponds to a running process. Please note that now the PID file is not provided for all the applications and most Windows applications do not provide PIDs.
About Regular Expressions

Regular expressions are used in Monitoring Studio to define strings to be searched for. A regular expression is:

- A string formatted with a specific syntax.
- It is intended to select some lines in a text, which will match the regular expression.

Regular expressions are commonly used in pattern matching, and especially on UNIX systems with the grep, awk and sed commands. You can use regular expressions in Monitoring Studio in order to:

- Find a process
- Search for strings in a file
- Check a web page
- Parse a table in a database
- Retrieve numbers, etc.

The following table describes the regular expression syntax that is supported in Monitoring Studio.

<table>
<thead>
<tr>
<th>Character</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>. (dot)</td>
<td>Match any single character. Example: Err.. will match Err01, Err02 or ErrAB, etc.</td>
</tr>
<tr>
<td>[xyz]</td>
<td>Match any character in the brackets. Example: Err[123] will match Err1, Err2 or Err3 [Ee]rror will match either error or Error</td>
</tr>
<tr>
<td>[^xyz]</td>
<td>Match any character not in the brackets. Example: Err[^12345] will match Err0, Err6, Err7, etc. but not Err1</td>
</tr>
<tr>
<td>[a-z]</td>
<td>Match any character in the range in the brackets. Example: Err[0-9] will match Err0, Err1, etc. and Err9 Err[A-Z][0-9] will match ErrA0, ErrA1, ErrS9, Err20, etc. but not Err1A Err[A-Z0-9] will match ErrA0, ErrA1, etc. and Err1A</td>
</tr>
<tr>
<td>[^a-z]</td>
<td>Match any character not in the range in the brackets. Example: Application[^0-9] will match ApplicationA, ApplicationB, Application1 but not Application1</td>
</tr>
<tr>
<td>*</td>
<td>Match zero or more repetitions of the preceding. Example: Err[0-9A-F]* will match Err, Err0, ErrA, Err11, ErrBF0001, etc. Error.*ApplicationABC will match all lines that contains Error and ApplicationABC further (Critical Error 0x000295F0 on ApplicationABC)</td>
</tr>
<tr>
<td>+</td>
<td>Match one or more repetitions of the preceding. Example: Err[0-9A-F]+ will match Err0, ErrA, Err11, ErrBF0001, etc. but not Err</td>
</tr>
<tr>
<td>Character</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>^</td>
<td>Match the beginning of the line. Example: ^Err will match all lines that begin with Err</td>
</tr>
<tr>
<td>$</td>
<td>Match the end of the line. Example: [0-9]+ connections$ will match all lines that end with xxx connections where xxx is an integer</td>
</tr>
<tr>
<td>&lt;</td>
<td>Match the beginning of a word. Example: &lt;set will match any line that contains a word that begins with set. It will not match a line that only contains the word unset.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Match the end of a word. Example: [Aa]pplication&gt; will match all lines that contain the word Application or application but not ApplicationAA</td>
</tr>
<tr>
<td>(expr)</td>
<td>Defines an expression which has to be processed as a unit regarding the modifier *, + and | Example: ([a-zA-Z0-9])+ will match only sequences like _patrol, _patrol_agent, _patrol_console, etc.</td>
</tr>
<tr>
<td>exprA|exprB</td>
<td>Match either exprA or exprB Example: (firewall)|(antivirus) will match all lines that contains either the word firewall or the word antivirus</td>
</tr>
<tr>
<td>\</td>
<td>Avoid the meaning of the following character. Example: . will match the single character dot (.) C:\Program Files will match C:\Program Files</td>
</tr>
</tbody>
</table>
About WMI

Definition

Windows Management Instrumentation (WMI) is a set of specifications from Microsoft for consolidating the management of devices and applications in a network from Windows computing systems. WMI is the Microsoft implementation of Web Based Enterprise Management (WBEM), which is built on the Common Information Model (CIM), a computer industry standard for defining device and application characteristics so that system administrators and management programs can control devices and applications from multiple manufacturers or sources in the same way.

What does it do?

WMI provides users with information about the status of local or remote computer systems. It also supports such actions as the configuration of security settings, setting and changing system properties, setting and changing permissions for authorized users and user groups, assigning and changing drive labels, scheduling processes to run at specific times, backing up the object repository, and enabling or disabling error logging. You can use WMI to manage both local and remote computers.

The word "Instrumentation" in WMI refers to the fact that WMI can get information about the internal state of computer systems, much like the dashboard instruments of cars can retrieve and display information about the state of the engine. WMI "instruments" by modeling objects such as disks, processes, or other objects found in Windows systems. These computer system objects are modeled using classes such as Win32_LogicalDisk or Win32_Process; as you might expect, the Win32_LogicalDisk class models the logical disks installed on a computer, and the Win32_Process class models any processes currently running on a computer. Classes are based on the extensible schema called the Common Information Model (CIM). The CIM schema is a public standard of the Distributed Management Task Force (http://www.dmtf.org/). WMI capabilities also include eventing, remoting, querying, views, user extensions to the schema, instrumentation, and more.

WMI Concepts

CIM Repository

CIM stands for Common Information Model and the repository is the WMI schema that stores the class definitions that model WMI-managed resources. The repository holds the information required to work with live resources in the computing environment. It does not contain actual data about these resources since this data is dynamically retrieved as required. It is this schema that allows the wide variety of different resources to be uniformly managed.
Namespace

CIM classes are organized into namespaces. Each namespace in the CIM contains a logical group of related classes representing a specific technology or area of management. Anytime a connection is made to WMI, a namespace must be specified. Only the classes contained within this namespace may be accessed by the connection. The most common namespace used for Windows management is root\cimv2. This contains the classes with the Win32_ prefix representing various components of the Windows operating system and hosting computer. Examples include Win32_Process (running processes in Windows), Win32_LogicalDisk (Windows logical disk drives), and Win32_ComputerSystem (the computer hosting Windows).

The namespace also includes the CIM_DataFile class which can be used to monitor files and folders. The following table lists common namespaces.

<table>
<thead>
<tr>
<th>Namespace</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>root\cimv2</td>
<td>Contains the most useful classes including all Win32_ classes</td>
</tr>
<tr>
<td>root\default</td>
<td>Contains registry events</td>
</tr>
</tbody>
</table>

Class

Every resource managed by WMI is defined by a class. A class is a template for each type of resource and defines the properties that will be collected for that resource. Examples of common WMI classes are shown in the table below:

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win32_Process</td>
<td>Processes running on a Windows computer</td>
</tr>
<tr>
<td>Win32_ComputerSystem</td>
<td>The computer running a Windows operating system</td>
</tr>
<tr>
<td>CIM_DataFile</td>
<td>A file stored on a disk</td>
</tr>
</tbody>
</table>
**Instance**

An Instance is a unique occurrence of a particular class. For example, each service installed on a Windows computer is an instance of the Win32_Service class. The C: drive is an instance of the Win32_LogicalDrive class.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Winmgmt</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Windows Management Instrumentation</td>
</tr>
<tr>
<td>PathName</td>
<td>C:\WINDOWS\system32\svchost.exe -k netsvcs</td>
</tr>
<tr>
<td>StartMode</td>
<td>Auto</td>
</tr>
<tr>
<td>State</td>
<td>Running</td>
</tr>
</tbody>
</table>

**Property**

A property is unique piece of information about an instance. All instances of a class will have the same set of properties although the values each instance’s properties may differ. Sample Properties of the Win32_Service class are shown in the table below:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Unique name of the service.</td>
</tr>
<tr>
<td>DisplayName</td>
<td>Displayed name of the service.</td>
</tr>
<tr>
<td>PathName</td>
<td>The command line path that was executed to start the service.</td>
</tr>
<tr>
<td>StartMode</td>
<td>Startup type of the service (Auto, Manual, or Disabled)</td>
</tr>
<tr>
<td>State</td>
<td>Current state of the service (Running, Stopping, or Stopped)</td>
</tr>
</tbody>
</table>
Basic WMI Queries

Queries may be issued against WMI resources using WMI Query Language (WQL). WQL is a subset of SQL designed to retrieve information from WMI. A simple example of a WMI query would be:

```
SELECT * FROM Win32_Process.
```

This retrieves all attributes (the * is used as a wildcard) for all processes currently running on the computer. Win32_Process is the name of the WMI class for Windows processes.

WMI queries of this type are often issued from a script using Windows Script Host or from any application or tool that can access WMI. Queries retrieve specific information from instances of WMI resources or execute methods against instances to perform such actions as stopping services, or starting processes.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Example code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>SELECT *</td>
<td>Specifies what properties are returned. Typically * is used to simply retrieve all.</td>
</tr>
<tr>
<td>FROM</td>
<td>FROM __InstanceCreationEvent</td>
<td>Specifies the event class to query. This will be the extrinsic or intrinsic event class.</td>
</tr>
<tr>
<td>WHERE</td>
<td>WHERE TargetInstance ISA 'Win32_Process' AND TargetInstance.Name = 'notepad.exe'</td>
<td>Filters the results. For intrinsic events, will usually include the ISA keyword to specify the class of the TargetInstance.</td>
</tr>
</tbody>
</table>

In case you need help to build your WMI query, you could download [WMI CIM Studio](http://www.microsoft.com) – which is one of the WMI Administrative tools on the Microsoft site.
Index

- % -

%{SEN_TIME:...}
Macros 123

- A -

About
  Process 150
  WMI 154
Advanced
  Variables 109
Agent Configuration
  Importing 127
Alert Actions
  Defining 114
  Group 114
  Macros 116
Analysis
  PSL Command 93
Analyzing
  Command Lines 38
  Database Query 42
  WBEM Queries 79
  Web Requests 82
  WMI Queries 95
Architecture 13
Attributes 129
  Available Capacity 131
  Available Capacity Percentage 131
  Available Inodes Percentage 131
  Child Count 137
  Collection Error Count 133, 134
  Consumed Capacity Growth Percentage 131
  Consumed Capacity Growth Speed 131
  Count 137
  Debug Status 134
  Deleted File Rate 132
  Delta 136
  DeltaPerSecond 136
  Discovery Status 134
  Discovery Time 134
  Execution Time 130, 139, 140, 142
  ExecutionTime 136, 138
  Exists 131
  Exit Code 130
  Exit Status 130
  File Count 132
  Folder Exists 132
  Folder Size 132
  Growth Percentage 131, 132
  Growth Speed 131, 132
  Handle Count 137
  Host Count 134
  HTTP Status 140
  HTTP Status Code 140
  Last Changed 131
  Last Modified File Elapsed Time 132
  Longest Time File Remains In Folder 132
  Matching Event Count 141
  Matching Event Rate 141
  Matching Line Count 140
  Matching Line Rate 140
  Matching Trap Count 139
  Matching Trap Rate 139
  Modified File Rate 132
  Monitor Count 134
  New File Rate 132
  Oldest Modified File Elapsed Time 132
  Page Faults Per Seconds 137
  Page File Bytes 137
  Percentage 136
  Present 136
  Private Bytes 137
  Processor Time 137
  Result 130
  Size 131
  Status 130, 134, 136, 137, 138, 139, 140, 142
  Thread Count 137
  Value 135, 136, 137, 139, 141
  Value Found 137
  Virtual Bytes 137
  Working Set 137
Authentication
  HTTP 148

- B -

Baselines 143
- C -

CIM  
- CIM contains 154  
- CIM Repository 154  
- CIM stands 154  
- CIM_DataFile 154  

Class  
- CIM_DataFile 154  
- Win32_ComputerSystem 154  
- Win32_Process 154  
- CIMRepository 154  
- CIM_DataFile 154  
- CIMRepository 154  
- CIM_DataFile 154  
- CIMRepository 154

- D -

Database Query  
- Analyzing 42  
- Microsoft SQL Server Database 43  
- MySQL Server Database 45  
- Oracle Database Server 47  
- Other Database 49  
- PostgreSQL Database 46  
- Deleting a Threshold Configuration 114  
- dfCommand 146  
- Documentation Scope 10  
- Downloading 23

- E -

Edit  
- Multi-Parameter Formula 59  
- Editing a Threshold Configuration 113  
- Event Logs  
- Monitoring 86  
- Execution  
- Timeout 93  
- Extracting  
- Numeric Values 102

- F -

Features 12  
File  
- PID 150  
- fileFindCommand 146  
- fileFindCommandOneDay 146  
- fileFindCommandSevenDays 146  
- File Systems  
- Monitoring 54  
- Flat Files  
- Monitoring 51  
- folderLimit 146  
- Folders  
- Monitoring 56  
- Formula  
- Multi-Parameter 59
- **G** -

Group
- Alert Actions 114
- Group Alert Actions 33
- Group Constants 33

- **H** -

Host
- OpenSSH 28
- PassPhrase 28
- Private Key 28
- Settings 28
- Host Availability Check 28
HTTP
- Authentication 148

- **I** -

Identifiers 148
Identify
- Process 150
- Importing the Monitoring Solution 21
Installing 23
- Installing the Monitoring Solution 20
Instance
- DisplayName 154
- Name 154
- PathName 154
- StartMode 154
- State 154
- Internal 148

- **J** -

JAVA 14

- **K** -

Key Concepts 10
Key Performance Indicators 143
KPI 143

- **L** -

Log Files
- Monitoring 51

- **M** -

Macros
- \%{SEN\_TIME::...} 123
- Format Symbols for \%{SEN\_TIME::...} 123
- General 116
- Object Specific 116
- maxFileSizeRead 146
Monitor
- Microsoft SQL Server Database 43
- MySQL Server Database 45
- Oracle Database Server 47
- Other Database 49
- PostgreSQL Database 46
Monitor Group
- Settings 33
- Monitor Internal Identifiers 148
- Monitor thresholds 112
Monitor Types 129
- Monitoring Studio 134
- Studio Command Line 130
- Studio Database Query 130
- Studio File 131
- Studio File System 131
- Studio Folder 132
- Studio Group 133
- Studio Host 134
- Studio Multi-parameter Formula 135
- Studio Nagios Performance Data 136
- Studio Nagios Plugin 136
- Studio Number Extract 137
- Studio Process 137
- Studio PSL Command 138
- Studio SNMP Polling 139
- Studio SNMP Trap 139
- Studio String Search 140
- Studio WBEM Query 140
- Studio Web Request 140
- Studio Windows Event 141
- Studio Windows Performance Counter 141
- Studio Windows Service 142
- Studio WMI Query 142
Monitoring
  Command Lines  38
  Event Logs     86
  File Systems   54
  Flat Files     51
  Folders        56
  Log Files      51
  Performance Counters 89
  Processes      66
  SNMP Agents    70
  SNMP Agents from a Single Numeric-based OID 72
  SNMP Agents from a Single String-based OID 73
  SNMP Agents from an SNMP Table 74
  SNMP Traps     76
  WBEM Queries   79
  Windows Event Logs 86
  Windows Performance Counters 89
  Windows Service 91
  WMI Queries    95

Monitoring Studio
  Settings       93

Monitors
  Command Lines  38
  Event Logs     86
  Files (Flat and Log) 51
  File Systems   54
  Folders        56
  Performance Counters 89
  Processes      66
  SNMP Agents    70
  SNMP Agents from a Single Numeric-based OID 72
  SNMP Agents from a Single String-based OID 73
  SNMP Agents from an SNMP Table 74
  SNMP Traps     76
  WBEM Queries   79
  Web Requests   82
  Windows Event Logs 86
  Windows Performance Counters 89
  Windows Service 91
  WMI Queries    95

- N -
  Namespace 154
  newerFileFindCommand 146
  Numeric Values
    Extract 102

- O -
  OpenSSH 28, 38, 51, 56
  Operating Systems 14
  Overview 10

- P -
  Package 22, 23
  PassPhrase 28, 38, 51, 56
  PATROL Agent 14
  Performance Counters
    Monitoring 89

Performing
  Microsoft SQL Server DatabaseQuery 43
  MySQL Server DatabaseQuery 45
  Oracle Database Server 47
  Other Database 49
  PostgreSQL DatabaseQuery 46

PID
  File 150
  Polling Interval
    Setting 126
  Private Key 28, 38, 51, 56

Process
  About 150
  Command Line 150
  Identify 150
  Name 150
  User ID 150

Processes
  Monitoring 66
  Product at a Glance 13

Properties
  Command line 150
  Name 150
  PID 150
  User ID 150
Property
- DisplayName 154
- Name 154
- PathName 154
- StartMode 154
- State 154
- Proxy server 108

Proxy Settings
- Configure 108
- psCommand 146
- psCommand2 146
- psCommand3 146
- PSL Command Analysis 93

- Q -

Queries
- WBEM 79
- WMI 95

- R -

Regular Expressions
- $ 152
- * 152
- . (dot) 152
- [^a-z] 152
- [^xyz] 152
- [a-z] 152
- [xyz] 152
- \ 152
- \(expression\) 152
- \< 152
- \> 152
- ^ 152
- + 152
- exprA\exprB 152

Requirements 14
- restartFromStartUponFileSizeDecrease 146

- S -

Searching
- String 98, 100

Settings
- Monitoring Studio 93

Signature Files Check 28
- SMTP
  - Server 106
- SMTP Server Configuring 106
- SNMP Agents
  - Monitoring 70
  - Polling 72, 73, 74
  - Single Numeric-based OID 72
  - Single String-based OID 73
  - SNMP Table 74
- SNMP Trap
  - Listening 76

String
- Log Files 100
- Search 98, 100

Supported BMC Platforms 14
- UNIX 150, 152
- Windows 150

- T -

Terminology 10
- Thresholds 112
  - Custom 59
  - Default 59

Timeout
- Execution 93

- U -

User Goals 12

- V -

Variables
- Advanced 109
- Configuration 146
- Configuring 109
- Group Specific 146
- Host Specific 146

- W -

WBEM 154
- WBEM Queries
WBEM Queries
  Monitoring    79
Web Requests
  Analyzing    82
Windows Event Logs
  Monitoring    86
Windows Performance Counters
  Monitoring    89
Windows Service
  Monitoring    91
WMI
  About    154
WMI Queries
  Monitoring    95
WMI Query
  FROM    154
  SELECT    154
  WHERE    154
About Sentry Software™
Sentry Software, a strategic Technology Alliance Partner of BMC Software, provides comprehensive multi-platform monitoring solutions that enable management of the hardware and software aspects of all servers and SANs and covering up to 100% of custom applications within the BMC TrueSight environment. Sentry Software also develops adapters for BMC Atrium Orchestrator that enables IT administrators to automate the execution of common requests and tasks that occur in the daily course of IT operations. Combined with BMC’s servers and network automation tools, the adapters allow IT administrators to implement provisioning and decommissioning workflows that cover all layers of their IT infrastructure. Finally, Sentry Software designs connectors that bring storage capacity metrics into BMC TrueSight Capacity Optimization to ensure IT administrators that their storage infrastructure is properly sized for their current and future needs.

The combination of its monitoring, automation, and capacity optimization capabilities for IT

About BMC Software™
BMC Software helps leading companies around the world put technology at the forefront of business transformation, improving the delivery and consumption of digital services. From mainframe to cloud to mobile, BMC delivers innovative IT management solutions that have enabled more than 20,000 customers to leverage complex technology into extraordinary business performance—increasing their agility and exceeding anything they previously thought possible. For more information about BMC Software, visit www.bmc.com.

BUSINESS RUNS ON I.T.
I.T. RUNS ON BMC SOFTWARE
Business thrives when IT runs smarter, faster and stronger. That’s why the most demanding IT organizations in the world rely on BMC Software across distributed, mainframe, virtual and cloud environments. Recognized as the leader in Business Service Management, BMC offers a comprehensive approach and unified platform that helps IT organizations cut cost, reduce risk and drive business profit. For the four fiscal quarters ended September 30, 2011, BMC revenue was approximately $2.2 billion.

LEARN MORE
To learn more about our solutions, please visit: www.sentrysoftware.com/solutions

Like us on Facebook: facebook.com/sentrysoftware
Follow us on Twitter: twitter.com/sentrysoftware