TIBCO Hawk®

Console User's Guide

Software Release 6.2
September 2019

TIBCO®

Important Information

Contents

Figures

Tables

Preface

This manual describes the functionality of TIBCO Hawk® Console, a web based tool for monitoring and managing applications.

Topics

Related Documentation

This section lists documentation resources you may find useful.

TIBCO Hawk Documentation

The following documents form the TIBCO Hawk documentation set:

- *TIBCO Hawk Release Notes:* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.
- *TIBCO Hawk Concepts:* This manual includes basic descriptions of TIBCO Hawk concepts.
- *TIBCO Hawk Installation, Configuration, and Administration:* Read this book first. It contains step-by-step instructions for installing TIBCO Hawk software on various operating system platforms. It also describes how to configure the software for specific applications, once it is installed. An installation FAQ is included.
- *TIBCO Hawk Microagent Reference*: A reference to the microagents and methods used by a TIBCO Hawk Agent for system and application monitoring.
- *TIBCO Hawk WebConsole User's Guide:* This manual includes complete instructions for using TIBCO Hawk WebConsole.
- *TIBCO Hawk Programmer's Guide*: All programmers should read this manual. It contains detailed descriptions of Application Management Interface (AMI), Application Programming Interface (API) concepts, and the TIBCO Hawk security framework and its classes. It also contains detailed descriptions of each class and method for the following APIs:

  — AMI API

  Java, C++ and C API

  — Console API

  Java API

  — Configuration Object API

  Java API

  Programmers should refer to the appropriate language reference sections for the AMI API details. The TIBCO Hawk Application Management Interface (AMI) exposes internal application methods to TIBCO Hawk.

- *TIBCO Hawk Plug-in Reference Guide:* Contains details about the Enterprise Message Service, Messaging and JVM microagents methods that are used to administer and monitor the TIBCO Enterprise Message Service server.
- *TIBCO Hawk Plug-ins for TIBCO Administrator:* Contains detailed descriptions of the TIBCO Hawk plug-ins accessed via TIBCO Administrator.
- *TIBCO Hawk HTTP Adapter User's Guide:* Contains information about performing discovery, monitoring of agent status, monitoring of agent alerts, method invocation, method subscription, and many more activities on TIBCO Hawk and third-party products.
- *TIBCO Hawk Admin Agent Guide:* Contains basic configuration details for TIBCO Hawk Admin Agent and complete instructions for using the web interface of TIBCO Enterprise Administrator for TIBCO Hawk.
- *TIBCO Hawk Security Guide:* Provides guidelines to ensure security within the components of TIBCO Hawk and within the communication channels between the components.

Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products:

- TIBCO® Enterprise Administrator
- TIBCO ActiveSpaces®
- TIBCO Rendezvous®
- TIBCO Enterprise Message Service™

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
ENV_HOME *TIBCO_HOME* *HAWK_HOME* *CONFIG_FOLDER*	TIBCO products are installed into an installation environment. A product installed into an installation environment does not access components in other installation environments. Incompatible products and multiple instances of the same product must be installed into different installation environments. An installation environment consists of the following properties: • **Name** Identifies the installation environment. This name is referenced in documentation as *ENV_NAME*. On Microsoft Windows, the name is appended to the name of Windows services created by the installer and is a component of the path to the product shortcut in the Windows Start > All Programs menu. • **Path** The folder into which the product is installed. This folder is referenced in documentation as *TIBCO_HOME*. TIBCO Hawk installs into a directory within a *TIBCO_HOME*. This directory is referenced in documentation as *HAWK_HOME*. The default value of *HAWK_HOME* depends on the operating system. For example on Windows systems, the default value is `C:\tibco\hawk\6.0`. A TIBCO Hawk configuration folder stores configuration data generated by TIBCO Hawk. Configuration data can include sample scripts, session data, configured binaries, logs, and so on. This folder is referenced in documentation as *CONFIG_FOLDER*. For example, on Windows systems, the default value is `C:\ProgramData\tibco\cfgmgmt\hawk`.
`code font`	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use `MyCommand` to start the foo process.

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
`bold code font`	Bold code font is used in the following ways: • In procedures, to indicate what a user types. For example: Type **`admin`**. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled: MyCommand [**`enable`** | disable]
italic font	Italic font is used in the following ways: • To indicate a document title. For example: See *TIBCO BusinessWorks Concepts*. • To introduce new terms For example: A portal page may contain several portlets. *Portlets* are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand` *pathname*
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

TIBCO Product Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website mainly in the HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit https://docs.tibco.com.

Documentation for TIBCO Hawk is available on the TIBCO Hawk Product Documentation page.

How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit https://www.tibco.com/services/support.
- For accessing the Support Knowledge Base, viewing the latest product updates that were not available at the time of the release, and getting personalized content about products you are interested in, visit the TIBCO Support portal at https://support.tibco.com.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to https://support.tibco.com. If you do not have a user name, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to https://community.tibco.com.

Chapter 1 TIBCO Hawk Console Dashboard

The TIBCO Hawk Console is a web application that provides a central view of all the distributed components interacting within the TIBCO Hawk system.

The landing page of Hawk Console displays a heat map of all alerts and the information cards for each registered Hawk domain.

For more details about the Hawk Console features, see *TIBCO Hawk Concepts Guide*.

Topics

Starting Hawk Console

Start the Hawk Console to view all the information of the Hawk system in its web interface.

Prerequisite

Ensure that the transport parameters are setup in the Hawk Console configuration file (`hawkconsole.cfg`). For details about Hawk Console configurations, see *TIBCO Hawk Installation, Configuration, and Administration Guide*.

Procedure

1. Start the Hawk Console by using either of the following steps:
   - Run `tibhawkconsole.exe`. (or `tibhawkconsole.sh`, depending on your operating system) from *HAWK_HOME*`\bin\`.
   - (Windows only) Click **Start > All Programs > TIBCO > *HAWK_HOME* > TIBCO Hawk > Start Hawk Console**.
2. In a web browser enter the URL `http://`*<Console_host_IP>*`:`*<Host_port>*`/HawkConsole`.
3. On the Hawk Console login page, enter your login credentials for the Hawk Console and click **Login**.

Result

The Hawk Console dashboard is displayed with information about the domain and their alerts, see Chapter 1, TIBCO Hawk Console Dashboard, on page 1.

When you start Hawk Console for the first time, you see information on only the default domain.

What to do Next

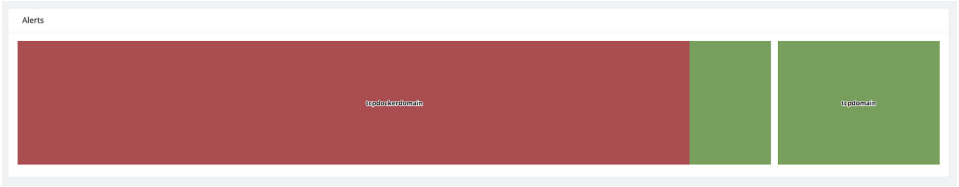
After logging in to the Hawk Console, you can perform either of the following major tasks:

- Configuring a Domain to Hawk Console on page 7
- Viewing Alerts for the Hawk Agent on page 16.
- Creating an Alert Rule for a Hawk Agent on page 39

Alerts Heat Map

The heat map is a graphical representation of alerts and notifications in the entire monitoring ecosystem (across agents and domains).

Figure 1 Hawk Console Alerts Heat map



The color of the individual cell in the map represents different alert levels. The size of the individual cell is directly proportional to the number of alerts/notifications of that type. The color scheme of the alerts indicate the following type of alerts:

- [Red] High
- [Orange] Medium
- [Yellow] Low
- [Green] Notification

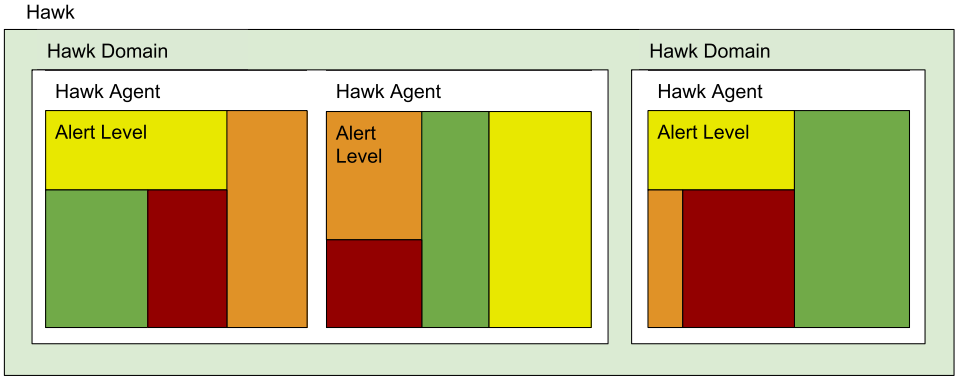
Heat Map Hierarchy

Heat map implementation in Hawk Console is a treemap representation. The treemap charts display hierarchical data in a set of nested rectangles. You can click any rectangle to drill down to its nested rectangles (levels). A rectangle's size is directly proportional to the specified dimension of the data.

The dimensions of hierarchy in Hawk Console is:

- Hawk domain
- Hawk agent
- Alert level

Figure 2 Hawk Console Alerts Heat Map Hierarchy



Drill Down Capability

The top-level heat map shows all alerts in all agents in all domains. You can drill down to any level of the hierarchy (dimension) to see the details. For example, if there are four domains, you can drill down to one domain to see all the agents in the domain in an expanded form. You can further drill down to an agent in the domain to see all alerts in expanded view. You can drill down to the last level in the hierarchy which is the cell for Alert category for an Agent. On clicking this cell, the user is navigated to Agent Alerts Details page filtered with the Alert Category.

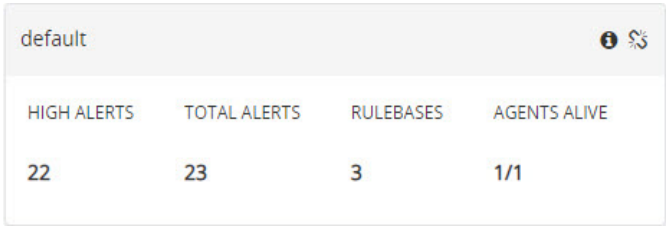
Heat Map Auto Updates

The heat map are autoupdated after every 20 seconds. If the agent previously had less number of high alerts, the size of high alerts cell for the agent was small as compared to other cells. However, if the agent starts generating a large number of high alerts, then the size of high alerts cell for the agent starts growing dynamically and all other cells in the entire heat map are automatically adjusted accordingly.

Domain Information Cards

The landing page of Hawk Console also displays information card for each registered Hawk domain. Information cards list key information about each registered Hawk domain.

Figure 3 Domain Information Cards in Hawk Console



The following information is displayed in each domain card:

- **Domain name** - The registered name of the domain.
- **High Alerts** - Number of alerts with the *high* status generated for all agents in the domain.
- **Total Alerts** - Total number of alerts generated for all agents in the domain.
- **Rulebases** - Total number of rulebases defined for all agents in the domain.
- **Agents Alive** - Number of agents that are in the running state out of the total number of agents in the domain.

For domain details, in the upper-right corner of domain information card, click the icon. The following domain details are included:

- Domain Type
- Domain Status
- Transport Type
- Daemon URL
- Self URL
- Security Policy Used

Actions

On the domain information section, you can perform the following actions:

- **Configure a domain** - You can also configure a Hawk domain to the Hawk Console and start monitoring that Hawk domain by using the web interface. For details, see Configuring a Domain to Hawk Console on page 7.
- **Unregister a domain** - Click the unregister icon in the domain information card to unregister the domain from the Hawk Console. After successful deregistration, you can not monitor Hawk agents in that domain through the current Hawk Console.
- **View domains in a list** - If needed, you can also view the domain information in a table. Click the **List View** icon on the right to switch to the list view for the domains. The List View icon is now toggled to the **Cards View** icon and all domain information cards are merged in a single table. You can sort the table rows based on any column. The columns available in the table are same as the information available in the information card. You can also switch back to the Information cards by clicking on the **Cards View** icon.

For more details about any particular domain, click the domain card (in card view) or the domain name (in list view) for drilling down to that domain. For details, see Viewing the Agent Details on page 12.

Configuring a Domain to Hawk Console

In the Hawk Console, you can configure a domain through the web interface. After configuration, you can also monitor that domain.

Prerequisites

Ensure that the Hawk domain that you are configuring is already running.

Ensure to take the backup of the DomainTransportConfig.yml file before configuring a Hawk domain to the Hawk Console. The Configure Domain option removes all the commented configurations from the `DomainTransportConfig.yml` file.

Procedure

1. Start the Hawk Console and sign in with your user name and password. The Hawk Console dashboard is displayed with the Alerts heat map and domain information cards.
2. In the Domains section, click the plus icon.

   The Configure Domain window is displayed with options to register a domain to the Hawk Console.
3. In the Configure Domain window, enter the details for registering the domain on the Hawk Console and click **Configure**. For details, see The Domain Configuration Reference on page 8.

Result

The information card for the newly configured domain is displayed on the dashboard.

What to do Next

After domain registration you can either view the alerts for the agent or create new alert rules:

- Viewing Alerts for the Hawk Agent on page 16.
- Creating an Alert Rule for a Hawk Agent on page 39

The Domain Configuration Reference

From the Hawk Console web interface, on the Configure Domain window, you can configure a Hawk domain.

For more information about Hawk domains and their transport configuration, see *TIBCO Hawk Installation, Configuration, and Administration* guide.

Table 2 Configure Domain Fields

Field	Description
Domain Type	Specify whether the Hawk domain to be registered is a regular domain or a proxy domain. Based on the domain type, the fields are displayed on the Configure Domain window. The values are: • `proxy` - For details about fields for the proxy domain type, see Table 3, Configure Domain Fields for Proxy Domain Type, on page 9. • `regular` - For details about fields for the regular domain type, see Table 4, Configure Domain Fields for Regular Domain Type, on page 9.
Domain Name	Specify the Hawk domain name.
Security Policy	Select the security policy that you want to apply to a domain: • **Default:** None • **Trusted:** Select this option to apply Trusted Security policy. • **Trusted with Domains (only for Windows XP domains):** Select this option for Microsoft Windows XP domains only. • **Custom:** To apply a custom security policy select this policy and enter the name of the custom security policy. For more information about Hawk Trusted Security Model, see *TIBCO Hawk Installation, Configuration, and Administration guide*.

Table 3 Configure Domain Fields for Proxy Domain Type

Field	Description
Host URL	URL of the domain that needs to be registered.
Username	User name required to log in to the domain.
Password	Password for the domain Username.
Secured Channel	Select the check box for connecting to the domain by using a secure channel.

Table 4 Configure Domain Fields for Regular Domain Type

Field	Description
Transport	Type of transport that the Hawk domain is using. The following transport types are available: • TCP - TCP Transport for TIBCO Hawk • RV - TIBCO Rendezvous Transport • EMS - TIBCO Enterprise Message Service (EMS) Transport Based on the transport type selected, transport configuration fields are displayed.
TCP Transport for TIBCO Hawk	
Self Url	Unique socket address of the Hawk Console for connecting to the TCP Transport for TIBCO Hawk cluster.
Daemon Url	The socket address of the Cluster Manager acting as the seed node for the TCP Transport for TIBCO Hawk cluster.
Additional transport options	Select the check box to provide additional details for SSL based TCP transport for the domain. The following fields are displayed after you select the check box:
Key store	Absolute path of the keystore that contains the Monitoring Console certificate and key to be loaded while communicating with the Hawk domain. You must provide a custom keystore that has a password protected key.

Table 4 Configure Domain Fields for Regular Domain Type

Field	Description
Key store password	Password to access the keystore.
Key password	Password to access the private key.
Trust store	Absolute path to trust store which will be used to validate the Hawk component certificates while communicating with the Hawk domain. For example, the default trust store of LogLogic LMI: `/loglogic/tomcat/conf/truststore`
Trust store password	Password to access the trust store.
SSL protocol	This is an optional field. Only TLSv1.2 protocol is supported. If you want to specify a protocol in this field, then it must be TLSv1.2.
SSL Enabled Algorithms	This is an optional field. Default value: `TLS_RSA_WITH_AES_128_CBC_SHA`
	You can use LogLogic LMI certificates to establish a secure communication between the Hawk components and the Monitoring Console. If the Hawk components are already configured to use the different CA certificates then you must add them to the LogLogic LMI trust store. There is no need to restart the LogLogic LMI engines after enabling SSL configuration for the Hawk domain. To add the Hawk CA certificates to the default LogLogic LMI trust store, perform the following steps on the UI of LogLogic LMI: **Note:** Use the **Trusted Certificate** tab to create a trusted relationship between the Monitoring Console and the remote Hawk components. 1. On the top navigation bar, click **Administration > SSL Certificate > Trusted Certificate.** 2. Copy the Hawk component certificate and paste it in the **Import Trusted** Certificate text box. 3. Click the **Import** button.

Table 4 Configure Domain Fields for Regular Domain Type

Field	Description
TIBCO Rendezvous Transport	
RV Service	Specify the service that the Rendezvous daemon uses to convey messages on this transport. You can specify the port number as the service to be used, for example, `7474`.
RV Network	Specify the network that the Rendezvous daemon uses for all communications involving this transport. The network parameter consists of up to three parts, separated by semicolons: network, multicast groups, and send address.
RV Daemon	Specify the socket address of the Rendezvous daemon.
TIBCO Enterprise Message Service (EMS) Transport	
EMS Server URL	Specify the location of the EMS server.
EMS Username	Specify the user name to login to the EMS server.
EMS Password	Specify the password for the **EMS Username**.
Additional transport options	Select the check box to provide additional details for SSL based EMS transport for the domain. The following fields are displayed after you select the check box: • EMS SSL Vendor • EMS SSL Trace • EMS SSL Trusted • EMS SSL Private Key • EMS SSL Expected Hostname • EMS SSL Password

Viewing the Agent Details

In Hawk Console, you can create rulebases and rules to monitor a Hawk agent. Also, you can view all the alerts related to the Hawk agent.

Procedure

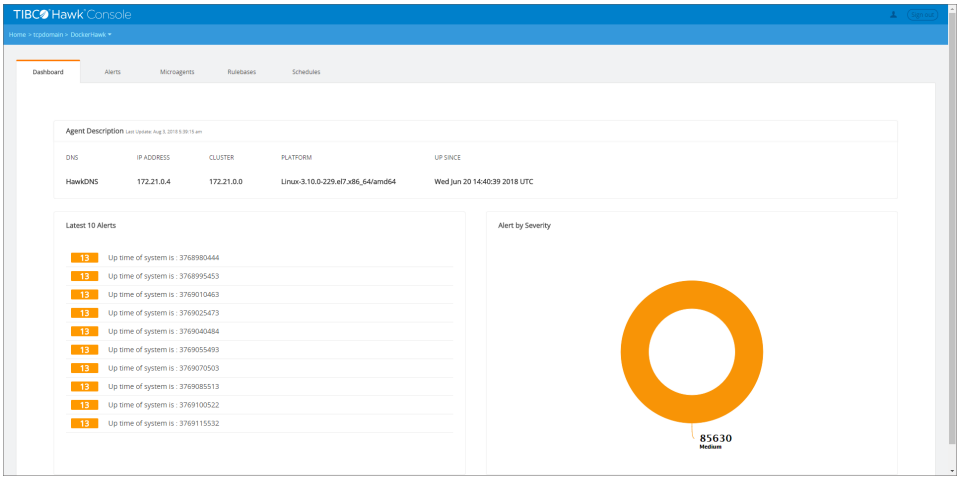
1. Start the Hawk Console and sign in with your user name and password. For steps, see Starting Hawk Console on page 2.
   The Hawk Console dashboard is displayed with the Alerts heat map and domain information cards.
2. On the Hawk Console dashboard, click the information card for the domain whose Hawk agent you want to view.
3. On the Domain's page, click the information card for the Hawk agent for which you want to view the details.
   The Hawk Agents page with multiple tabs (for Hawk agent actions and information) is displayed.

Result

The Agents page displays the following tabs to perform various functions:

- **Dashboard** - It displays agent's and alerts information in a single view. For details, see Dashboard Tab on page 14.
- **Alerts** - The Alerts tab lists all the alerts for the Hawk agent in a table. For details, see Chapter 2, Alert Messages, on page 15.
- **Microagents** - In the Microagents tab, you can view microagents and their methods for the Hawk agent. For details, see Chapter 3, Microagent Management, on page 21.
- **Rulebases** - The Rulebases page shows all the rulebases for the agent. For details, see Chapter 4, Rulebase Management, on page 29.
- **Schedules** - The Schedules tab enables you to define a schedule and deploy the schedule to the Hawk agent. For details, see Chapter 5, Schedule Management, on page 65.

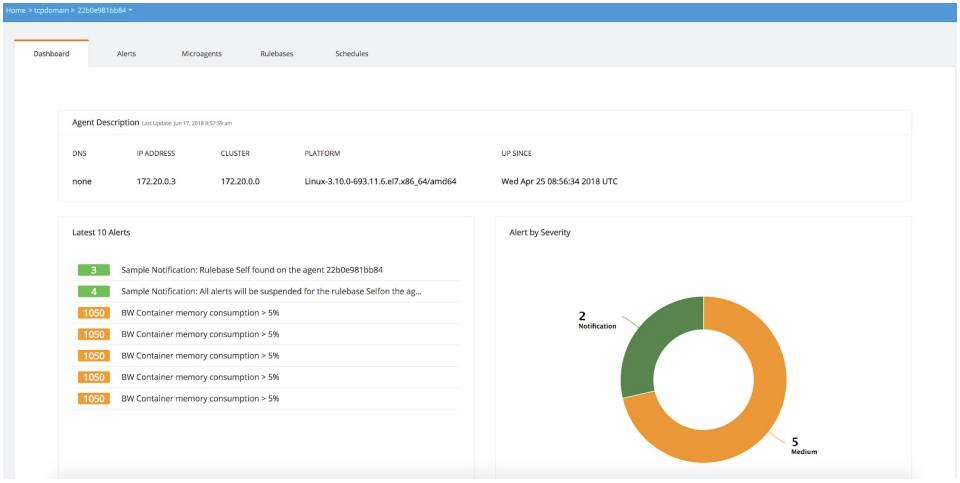
Figure 4 Agent's Details Page



Dashboard Tab

The Dashboard tab displays agent information and alerts information in a single view.

Figure 5 The Dashboard Page for the Hawk Agent



The following sections are displayed on the Dashboard tab:

- **Agent Description** - The section displays the infrastructure details of the Hawk agent. The following field values are displayed for the agent:
  - DNS
  - IP Address
  - Cluster
  - Platform
  - Up Since
- **Latest 10 Alerts** - The section lists most recent ten alerts for the Hawk agent.
- **Alert by Severity** - The section shows the doughnut chart for the alerts based on their severity. Each colored section denotes different severity. Click on any section of the doughnut chart to open the list of alert message of that severity.

Chapter 2 Alert Messages

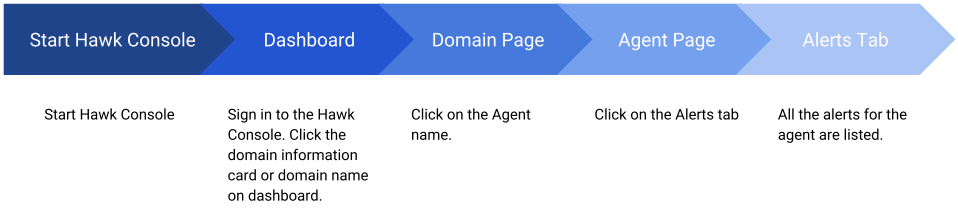
Alerts are messages an agent sends to TIBCO Hawk Console. Alerts originate from rulebases when a specified condition occurs that enforces your monitoring criterion.

Topics

Viewing Alerts for the Hawk Agent

By using the Hawk Console you can view all the alerts generated for a Hawk agent based on the rules deployed for the Hawk agent.

The following procedure helps you to view all the alerts for the Hawk agent; however, if you want to see the filtered result only based on the alert type, you can use the Alerts heat map.



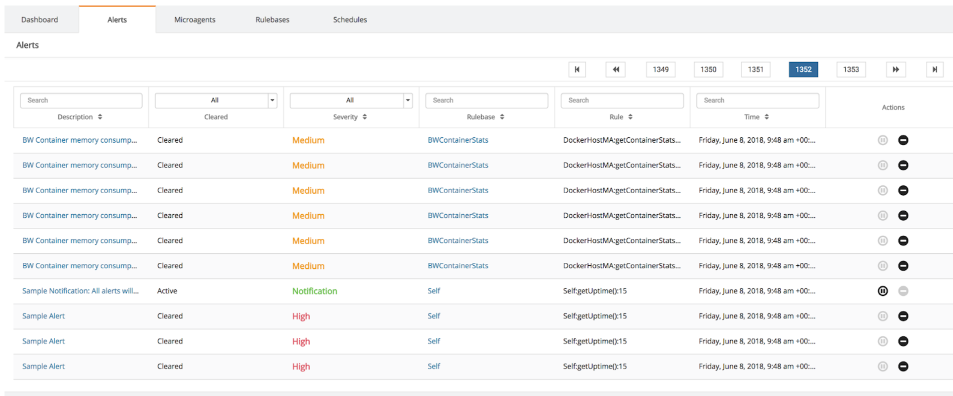
Procedure

1. Start the Hawk Console and sign in by using your user name and password.
   The Hawk Console dashboard is displayed with the Alerts heat map and domain information cards.
2. Click the domain information card for the domain of your Hawk agent.
   The domain details page is displayed with Hawk agents information card.
3. Click the information card for the Hawk agent for which you want to see alerts.
   The Agents page with several tabs for various operations is displayed. For details about the Agents page, see Viewing the Agent Details on page 12.
4. Click the **Alerts** tab.
   All the alerts for the Hawk agent are listed in a table. For details, see Alerts Tab on page 17.

Alerts Tab

The Alerts tab lists all the alerts for the Hawk agent in a table. You can sort and filter these alerts by using these columns.

Figure 6   The Alerts Page for the Hawk Agent



For each alert, the following details are provided and you can filter out the results based on these details:

- **Description** - A string that describes the alert. Click the description link to view more details about the alert. The following details are displayed:
  - — Description
  - — Alert ID
  - — Rulebase name
  - — Data source
  - — Rule (that triggered this alert)
  - — Test condition
  - — Action
  - — DataIndex
- **Cleared** - It specifies if the alert has been cleared or not.
- **Severity** - The type of severity which can be one of High, Medium, Low, or Notification.
- **Rulebase** - The name of the rulebase which generated this alert. Click the Rulebase link to get the details of the rulebase that triggered the alert. The rulebase details are displayed in the Rulebase tab.

- **Rule** - The name of the rule that triggered the alert.
- **Time** - Timestamp when the alert was generated. For filtering alerts based on their timestamp, you can use the date and time picker to select a range.
- **Actions** - The action that you want to take on this alert.
  - Suspend the alert for a specified amount of time. For details, see Suspending an Alert on page 19.
  - Purge suspended alerts from the alerts list. For details, see Purging Suspended Alerts on page 20.

Suspending an Alert

If an alert might interrupt another monitoring task, you can temporarily suspend it.

For example, if a condition such as a process failure is generating a high-level alert with a warning bell and the problem is being worked on, you can suspend the alert until the problem is resolved. Suspension details are added to the properties of the message. These details are visible to you, other Hawk Console users, and Console API applications.

Suspending an alert message affects only the action of the generated alert. If the condition that generates the alert message also generates another type of action, such as attempting to restart the process, that action is unaffected.

Procedure

1. In the Hawk Console, open the Hawk Agent page for which you want to suspend an alert.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Alerts** tab.

   All the alerts for the Hawk agent are listed in the tab. For details, see Alerts Tab on page 17.
3. In the Alerts tab, click the **Suspend** icon for the alert that you want to suspend.
4. In the Suspend Alert window, enter the **Time** (in minutes) for which you want to suspend the alert.
5. Specify a valid reason for suspending the alert in the **Reason** field and click **Suspend**.

   On successful suspension, the successful message is displayed.

Result

All the alerts with the specified `AlertID` are suspended. The Cleared column value is changed to `Cleared`. Also, for all the suspended alerts the **Purge** icon becomes active.

What to do Next

You can purge the suspended alerts from the alerts list. For details, see Purging Suspended Alerts on page 20.

Purging Suspended Alerts

You can purge all the suspended alerts to clean up the alerts list. You can only purge suspended alerts.

Procedure

1. In the Hawk Console, open the Hawk Agent page for which you want to purge suspended alerts.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Alerts** tab.

   All the alerts for the Hawk agent are listed in the tab. For details, see Alerts Tab on page 17.
3. In the Alerts tab, click the **Purge** icon for the suspended alert that you want to purge. The suspended alert have the Cleared column value as `Cleared`.

   A Confirmation dialog box is displayed to confirm the purging of the alerts with the displayed alert ID.
4. In the Confirmation dialog box, click **Ok** to purge all the cleared alerts with same `AlertID`.

   On successful purging, the successful message is displayed with the number of alerts purged. Click **Ok** to close the dialog box.

Result

All purged alerts are removed from the alerts list in the **Alerts** tab.

Chapter 3 Microagent Management

This chapter contains steps to perform operations supported on the Microagent tab.

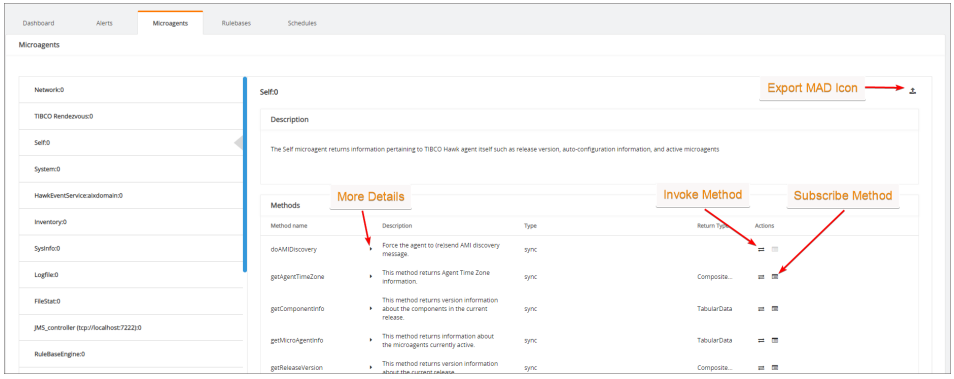
Topics

Microagents Tab

Each agent has a set of default microagents, which is discovered by agents when it is started. If you install and start an adapter or gateway, or instrument an application with AMI, microagents for these objects are dynamically added to the agent. In the Microagents tab, you can view microagents and their methods for the Hawk agent.

For more details about microagents, see *TIBCO Hawk Concepts Guide*.

Figure 7 The Microagents Page for the Hawk Agent



The left panel on the page lists all the microagents available for the Hawk agent. Select any microagent to view its description and all the available methods on the right panel.

For each method, the following details are displayed in a table:

- **Method Name** - Displays the name of the method. To see more technical specifications of the method, click the right-pointing triangle in front of the method name. It displays more details about the method, such as arguments and returns. For details about microagent and methods, see *TIBCO Hawk Microagent Reference Guide*.
- **Description** - A short description of what the method does.
- **Type** - Specifies whether the subscription information is returned synchronously, on a regular time schedule, or asynchronously, when data becomes available.
- **Return Type** - Specifies if the data is returned as Tabular Data or Composite data.

- **Action** - You can perform the following actions for each method.
  - **Invoke** - Use the Invoke action to immediately view the results. Invoking is useful when you want to test a method before using it in a rule, or to check a return value for troubleshooting purposes, see Invoking a Microagent Method on page 24.
  - **Subscribe** - Use the Subscribe action to view the microagent method results over time. Creating a subscription is useful when you want to test a range of return values before specifying boundaries in a rule, or to identify general patterns of activity, see Subscribing to a Microagent Method on page 26.

Exporting the Description for a Microagent

You can either view or download the microagent description to a file. Click the **Export MAD** icon for the microagent and select the location to save the microagent description (`.hmd`) file.

Invoking a Microagent Method

Invoke a microagent method to immediately view its result. Invoking is useful when you want to test a method before using it in a rule, or to check a return value for troubleshooting purposes.

The invoke results are displayed in the Invoke window. The results vary, depending on the arguments required for the invoked method. For information about all the microagent methods, see *TIBCO Hawk Microagent Reference Guide*.

For all methods that have some return values, the result of the method is displayed on the window and for all the methods without any return values, no result is displayed.

Procedure

1. In the Hawk Console, open the Hawk Agent page for which you want to invoke the microagent method.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Microagents** tab.

   All the microagents for the Hawk agent are displayed in the tab. For details, see Microagents Tab on page 22.
3. Select the microagent whose method you want to invoke.

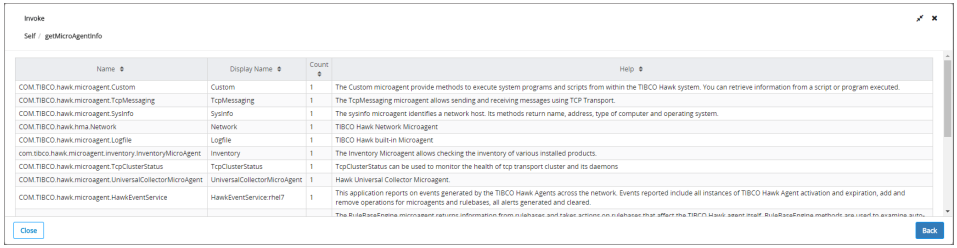
   The right panel displays the microagent details and all its methods.
4. Under the Actions column, click the **Invoke** icon to invoke the microagent method.

   The Invoke window is displayed with all the required parameters for the microagent method.
5. In the Invoke window, enter the details of the fields and click **Invoke**.
6. Click **Close** or, to define parameters for the method, click **Back**.

Example

For example, the following figure shows the sample result for invoking the `Self:getMicroAgentInfo` method without any supplied argument.

Figure 8 Sample Result of Invoking getMicroAgentInfo Method



Subscribing to a Microagent Method

Subscribe to a microagent method to view its results over time. Creating a subscription is useful when you want to test a range of return values before specifying boundaries in a rule or to identify general patterns of activity.

Prerequisites

Ensure that the return type is defined for the microagent method that you want to subscribe to.

You cannot subscribe to microagent methods that do not have a return type.

For more information on the microagent methods, either view the microagent method details on the **Microagents** tab or see *TIBCO Hawk Microagent Reference Guide*.

Procedure

1. In the Hawk Console, open the Hawk Agent page for which you want to subscribe to a microagent method.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Microagents** tab.

   All the microagents for the Hawk agent are displayed in the tab. For details, see Microagents Tab on page 22.
3. Select the microagent whose method you want to invoke.

   The right panel displays the microagent details and all its methods.
4. Under the Actions column, click the **Subscribe** icon to start a subscription of the microagent method.

   The Subscribe window is displayed with all the required parameters for the microagent method.
5. In the Subscribe window, enter the details of the fields, select the subscription interval, and click **Start Subscription**.
   The fields displayed on the window vary depending on the arguments required for the subscribed method. For reference information on all the microagent methods, either view the microagent method details on the **Microagents** tab or see *TIBCO Hawk Microagent Reference Guide*.

The result of the method is displayed on the window with a new line added after each subscription interval. If required, you can also select the **Update same line** option to display every result after updating the same line.

6. Click **Minimize** icon to minimize the Subscription window while the results are published in the background.

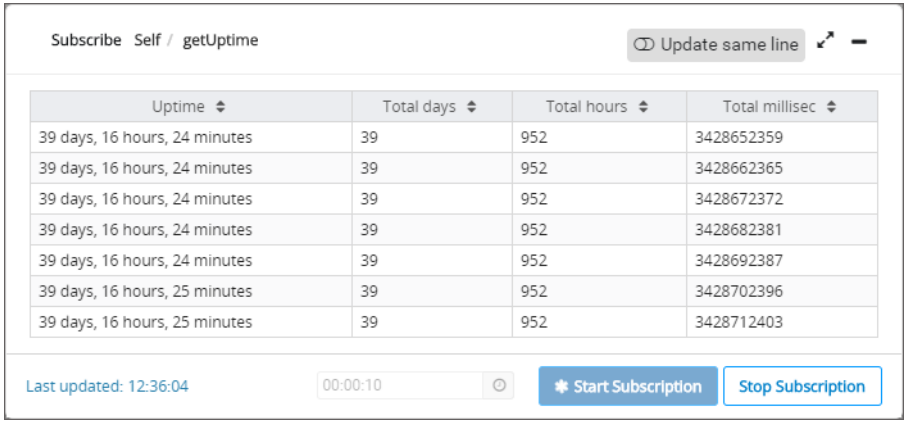
   After minimizing, the **Subscribe** icon for the method is changed to a gear icon. Click the gear icon to restore the Subscription window.

7. In the Subscription window, click **Stop Subscription** to stop receiving the subscription results. After stopping the subscription, click the **Close** icon to close the subscription window.

Example

For example, the following figure shows the sample subscription result for the `getUptime` method of the `Self` microagent with the subscription interval set to 10 seconds.

Figure 9 Sample Result for Subscription of getUptime Method



Chapter 4 Rulebase Management

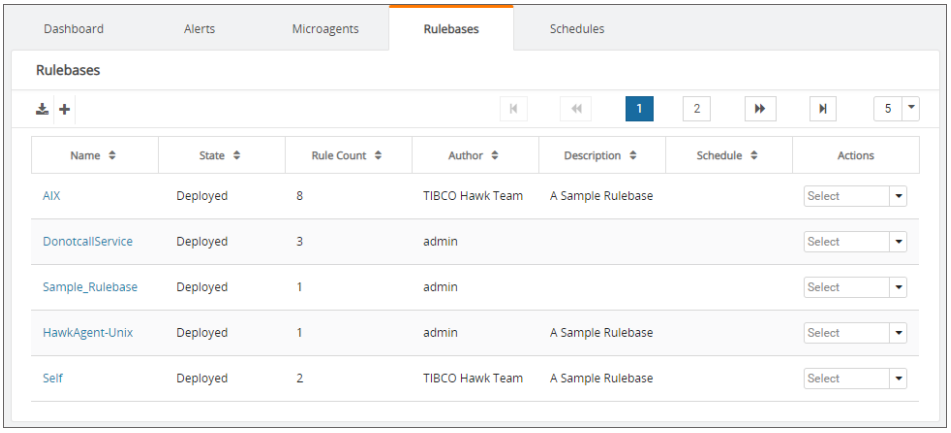
This chapter contains steps for the operations supported on the **Rulebases** tab.

Topics

Rulebases Tab

The **Rulebases** tab shows all the rulebases for the agent. You can select the number of rulebases displayed on a page by selecting 5, 10, 25, 50 or 100 from the records per page drop-down menu.

Figure 10 The Rulebases Page for the Hawk Agent



The following information is displayed for each rulebase:

- **Name** - the name of the Rulebase
- **State** - whether the rulebase is deployed or undeployed
- **Rule Count** - the number of rules in the Rulebase
- **Author** - name of person or entity that created the Rulebase
- **Description** - text used to describe the Rulebase
- **Schedule** - the name of the schedule that is used by the Rulebase

- **Actions** - you can take the following actions on the Rulebase:
  - get alerts
  - edit the rulebase
  - deploy or undeploy the rulebase on the agent
  - delete the rulebase
  - export the rulebase to a `.hrb` file
  - derive a new rulebase based on an existing one
  - deploy the rulebase to an agent other than the agent on which the rulebase exists
  - undeploy the rulebase from an agent on which it was previously deployed

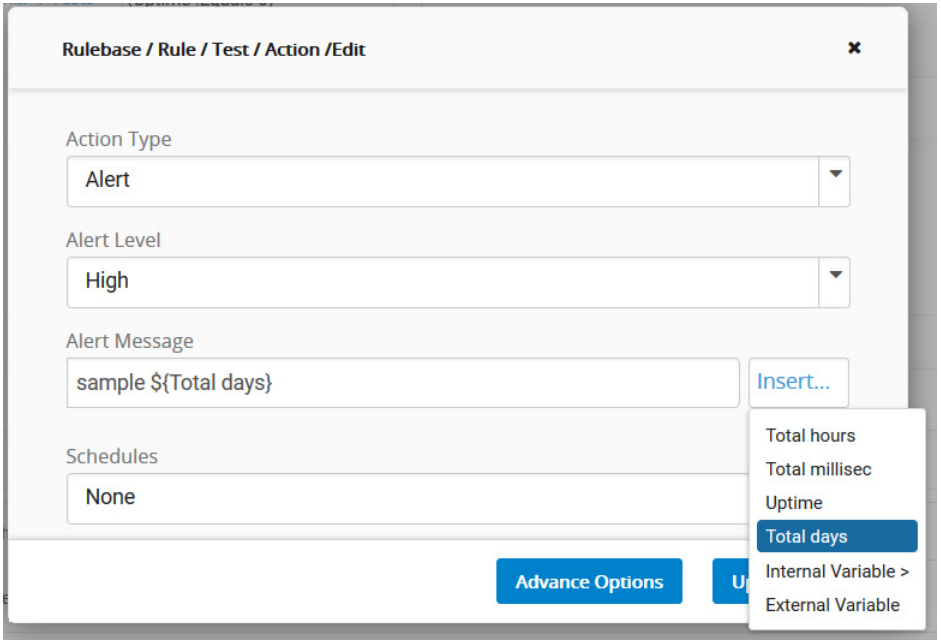
Apart from this information, can also perform two more operations on the **Rulebases** tab:

- **Import a rulebase** - Click the Import icon to import an existing rulebase (the `.hrb` file) to the Hawk agent. For details, see Importing a Rulebase to the Hawk Agent on page 43.
- **Create a new rulebase** - Click the Add icon to create a new rulebase for the Hawk agent, see Adding a Rulebase to the Hawk Agent on page 38.

Rulebase Variables in Alert Messages

You can use rulebase internal variables while defining rulebase alert actions message, along with microagent specific variables in the alert message text. For example, in a rulebase with Self microagent as the data source, you can choose rulebase internal variables such as rulebase name, rule name, test name, and so on. You can also use Self microagent specific variables such as Uptime, TotalHours, and so on in your alert message text. For details, see Variables in a Rulebase on page 61.

Figure 11 Variables Usage in Alert Messages



Drilling Down Rulebases

In the Rulebase tab, you can drill down the rulebase details to the action level. You can drill down the rulebase details in the following hierarchy:

Figure 12 Drilling Down Rulebases

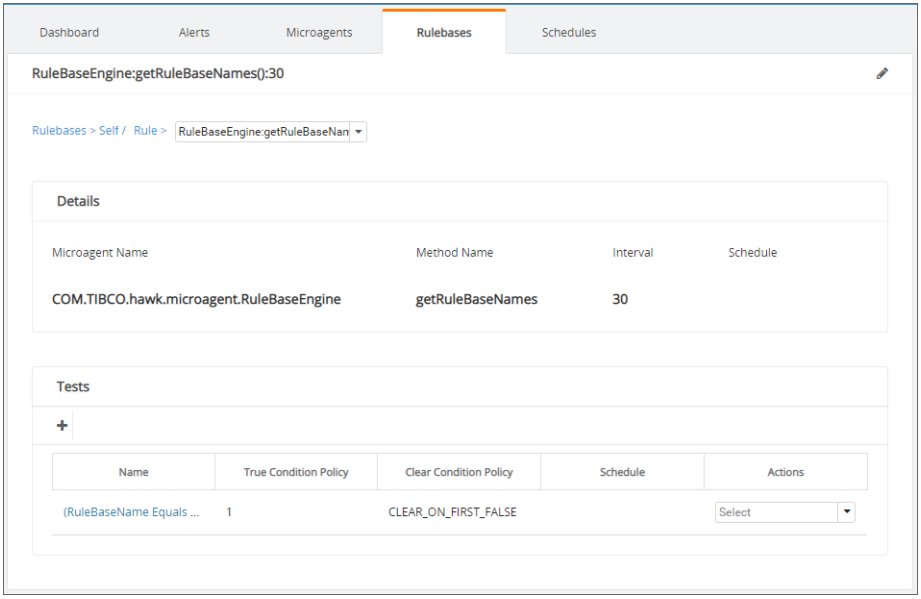
At each level, you can view the details of that entity and list of its subsequent entities. For example, when you click a rulebase name, the rulebase details are displayed and also the list of rules associated with it is displayed.

Also, at each level, you can perform some actions specific to that level. For example, on the **Rulebases** tab, click the plus icon to add a new rulebase, when you drill down to the rulebase details page, you can click the plus icon to add a new rule to the rulebase.

Rule Details Page

In the Rule details page, you can view all the details of a rule as well as you can add test conditions for the rule.

Figure 13 Rule Details Page



Details

- **Microagent Name** - The microagent whose methods can act as the data source for the rule.
- **Method Name** - The microagent method which acts as the data source for the rule.
- **Interval** - The time interval after which Hawk Console checks the rule.
- **Schedule** - The schedule applied to the rule. For details about schedule, see Chapter 5, Schedule Management, on page 65.
- **Tests** - List of all the tests, associated with the rule, and their details. For tests details, see Test Details Page on page 35.

Actions

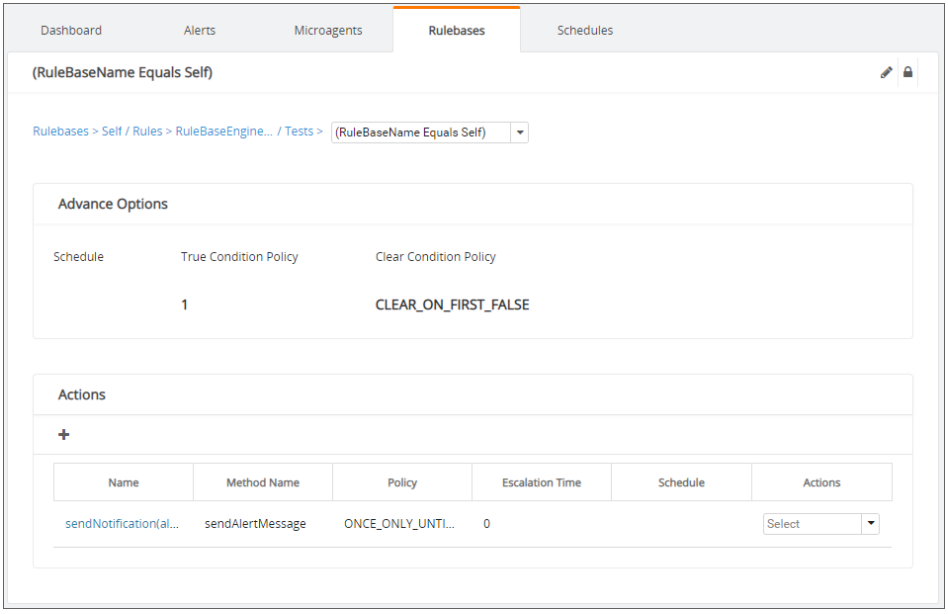
- Add an test condition - Click the plus icon (+) to add a new test condition to the rule. For details, see Creating a Test in the Rule on page 47.
- Edit the rule - Click the edit icon to edit the details of the rule.
- Edit a test condition - Select the **Edit** option under the Actions column for an test condition to edit it.

- Derive a test condition - Select the **Derive** option under the Actions column to duplicate the test.
- Delete a test condition - Select the **Delete** option under the Actions column for a test condition to delete it.

Test Details Page

In the Test details page, you can view details of the test condition and add an action for the test condition.

Figure 14 Test Details Page



Details

- **Schedule** - The schedule applied to the rule. For details about schedule, see Chapter 5, Schedule Management, on page 65.
- **True Condition Policy** - A counter which specifies after how many times, when the condition is true, the action is triggered.
- **Clear Condition** - The condition which when true triggers a clear action.

- **Actions** - List of all the actions, associated with the test condition, and their details.
  - Name
  - Method Name
  - Policy
  - Escalation Time
  - Schedule
  - Actions

  Click the action **Name** to view the Action details page, see Actions Details Page on page 36.

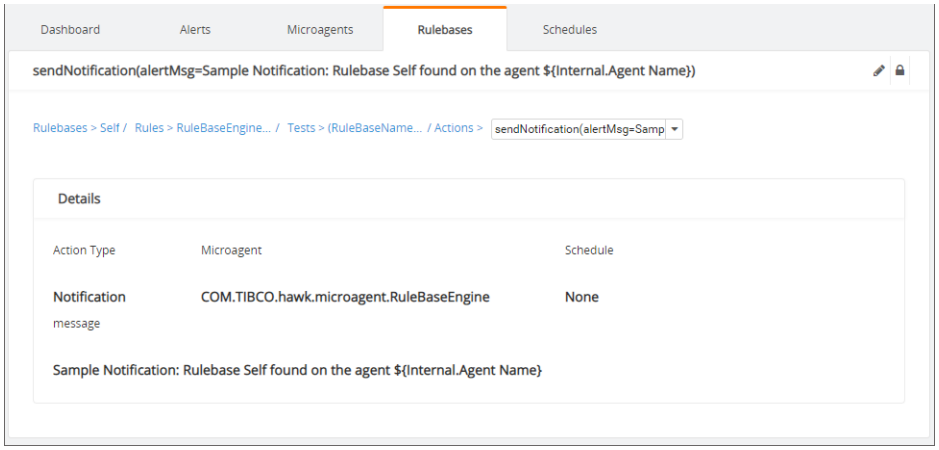
Actions

- Add an action - Click the Add Action icon to add a new action to be taken when the test condition is true. For details, see Creating an Action for a Test Condition on page 56.
- Edit the test condition - Click the edit icon to edit the details of the test condition.
- Deploy the rulebase - Click the deploy icon to deploy the rulebase to the Hawk agent. For details, see Deploying a Rulebase to the Associated Hawk Agent on page 45.
- Edit an action - Select the **Edit** option under the Actions column for an action to edit it.
- Derive an action - Select the **Derive** option under the Actions column to duplicate the action.
- Delete an action - Select the **Delete** option under the Actions column for an action to delete it.

Actions Details Page

In the Action details page, you can view details of the action configured for the test condition.

Figure 15 Action Details Page



Details

- **Action type** - Type of the action configured. Based on the action type other fields are displayed.
- **Alert Level** - Severity of the alert.
- **Microagent** - The microagent associated with the rulebase.
- **Message** - Alert message to be displayed.

Actions

- Edit the action - Click the edit icon to edit the details of the test condition.
- Deploy the rulebase - Click the deploy icon to deploy the rulebase to the Hawk agent. For details, see Deploying a Rulebase to the Associated Hawk Agent on page 45.

Adding a Rulebase to the Hawk Agent

A rulebase is a collection of rules. To add rules to any Hawk agent, you must create a rulebase first.

Procedure

1. In the Hawk Console, open the Hawk Agent page to which you want to add a rulebase.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Rulebases** tab.

   All the rulebases for the Hawk agent are displayed in the tab. For details, see Rulebases Tab on page 30.
3. Click the **Add Rulebase** icon to open the New Rulebase wizard.
4. In the New Rulebase wizard, enter the following details:
   - **Name** - A name of the new rulebase.
   - **Description** - A brief description about the rulebase.
   - **Schedules** - Select the schedule to associate with the rulebase. The rulebase is active only for the inclusion period defined in the selected schedule.
5. Click **Create Rulebase**.

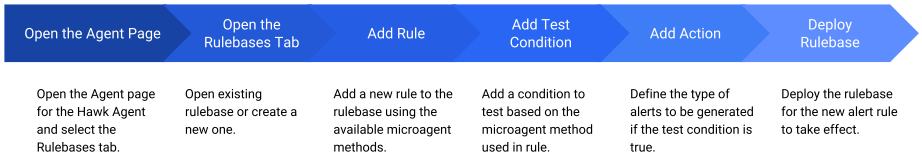
   The newly created Rulebase is listed on the **Rulebases** tab.

What to do Next

Add rules to the rulebase that you can apply for your monitoring requirement. For details, see Creating an Alert Rule for a Hawk Agent on page 39.

Creating an Alert Rule for a Hawk Agent

In the Hawk Console, you can define rules to generate alerts or emails based on predefined test condition. The test condition can be designed by using the rulebase and microagent variables.



Procedure

1. In the Hawk Console, open the Hawk Agent page for which you want to create the rule.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent's page, select the **Rulebases** tab.
   All the rulebases for the Hawk agent are displayed. For details, see Rulebases Tab on page 30.
3. Click on an existing rulebase name to which you want to add the new rule.

   Or, you can add a new rulebase to the agent and add a new rule to it, see Adding a Rulebase to the Hawk Agent on page 38.

   Details of the rulebase and list of all its rules are displayed on the **Rulebase** tab.
4. In the **Rules** section, click the plus icon.
   The New Rule wizard opens.
5. In the New Rule wizard, enter the value for the fields and click **Create and Add Test**.
   Some of the common fields are listed in the following table.

Table 5 New Rule Wizard Common Fields

Fields	Description
Microagents	Select the microagent whose method you want to use for the rule.
Methods	Select the microagent method that you want to use in the rule. The list displays only those methods that are relevant to the microagent selected. Based on the microagent method selected, some more fields might be displayed on the wizard.
Interval	Enter the time interval after which the rule runs. The default value is 60 (seconds).
Schedules	Select the name of an existing schedule to apply to this rule. This is an optional field. By default, the rule is always active.

The New Test window opens to enter the details for the condition to test for the rule.

6. In the New Test window, create the test condition by using the existing fields and click **Create and Add Action**. For details of defining the test condition, see Creating a Test in the Rule on page 47.

   The New Action window opens to add an appropriate action for the rule if the test condition is true.

7. In the New Action window, select the action you want to perform and enter the details to relevant fields. The following actions are available:

   — Alert

   — Notification

   — Method

   — Email

   — Post-Condition

   For details, see Creating an Action for a Test Condition on page 56.

8. Click **Create Action** to create a new action for the test condition.
   The New Rule wizard closes and the action is created for the test condition created.

What to do Next

Deploy the rulebase to the domain for the new rule to take effect, see Deploying a Rulebase to the Associated Hawk Agent on page 45.

Exporting a Rulebase to a File

To add a rulebase to a Hawk agent that is similar to the one already defined in another Hawk agent, you can export the existing rulebase and import the same to the Hawk agent.

The rulebase is exported in a `.hrb` file. This exported file contains all the details of the rulebase and all its rules. You can also select multiple rulebases and export them in a `.zip` file.

Procedure

1. In the Hawk Console, open the Hawk Agent page from which you want to export the rulebase.
   For steps, see Viewing the Agent Details.
2. In the Agent details page, select the **Rulebases** tab.

   All the rulebases for the Hawk agent are displayed in the tab. For details, see Rulebases Tab.
3. On the **Rulebases** tab, export the rulebase by using either of the following ways:
   - From the rulebases list, under the **Actions** column, select the **Export** option for the rulebase that you want to export.
   - Click the rulebase name that you want to export. Now, in the rulebase details page, click the export icon. The Save As window opens to save the exported rulebase (`.hrb`) file.
   - To export multiple rulebases, select the check boxes next to rulebases which you want to export and click the export icon in the top left corner. The `.zip` is downloaded to the Downloads folder of your machine which contains exported rulebase (`.hrb`) files.
4. In the Save As window, browse to the location where you want to save the `.hrb` or `.zip` file and click **Save**.

Importing a Rulebase to the Hawk Agent

If you want to add a rulebase that is similar to the one already defined in another Hawk agent, you can import the rulebase from the other Hawk agent.

The rulebase is exported in a `.hrb` file. This exported file contains all the details of the rulebase and all its rules. You have to import the `.hrb` file and deploy (with or without modifications) to your Hawk agent for applying all the rules of the rulebase to the Hawk agent. You can also import multiple rulebases (.hrb files) in a `.zip` file.

Prerequisites

You must have the `.hrb` file that contains the exported rulebase.

For the procedure to generate this exported file, see Exporting a Rulebase to a File on page 42.

Procedure

1. In the Hawk Console, open the Hawk Agent page to which you want to import the rulebase.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent details page, select the **Rulebases** tab.

   All the rulebases for the Hawk agent are displayed on the tab. For details, see Rulebases Tab on page 30.
3. On the **Rulebases** tab, click the import icon in the top left corner.
   - — To import single rulebase, select the rulebase (`.hrb`) file that you want to import.
   - — To import multiple rulebases at the same time, select the `.zip` file that contains all the rulebase files that you want to import.
4. Click **Open**.

Result

If the import is successful, the imported rulebase is listed on the **Rulebases** tab.

What to do Next

If required, modify the rulebase rules and deploy the imported rulebase to the Hawk agent. For details, see Deploying a Rulebase to the Associated Hawk Agent on page 45.

Deploying a Rulebase to the Associated Hawk Agent

For the rules of a rulebase to be activated for the Hawk agent, deploy the rulebase to the Hawk agent.

Procedure

1. In the Hawk Console, open the Hawk Agent page to which you want to deploy the rulebase.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Rulebases** tab.

   All the rulebases for the Hawk agent are displayed in the tab. For details, see Rulebases Tab on page 30.
3. On the **Rulebases** tab, deploy the rulebase by using either of the following ways:
   - From the rulebases list, under the **Actions** column, select the **Deploy** option for the rulebase that you want to deploy.
   - Click the rulebase name that you want to deploy. Now, in the rulebase details page, click the **Deploy** icon.

   The deployment confirmation dialog box is displayed.
4. In the deployment confirmation dialog box, click **Yes**.

Result

The successful deployment message is displayed.

Deploying a Rulebase to Another Hawk Agent

If you want to activate the rules of a rulebase for any other Hawk agent added to the Hawk Console, you can do that by using the deploy-to option.

Procedure

1. In the Hawk Console, open the Hawk Agent page from which you want to deploy the rulebase to another Hawk agent.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Rulebases** tab.

   All the rulebases for the Hawk agent are displayed in the tab. For details, see Rulebases Tab on page 30.
3. On the **Rulebases** tab, deploy the rulebase to another Hawk agent by using either of the following ways:
   - From the rulebases list, under the **Actions** column, select the **Deploy To** option for the rulebase that you want to deploy to another Hawk agent.
   - Click the rulebase name that you want to deploy. Now, in the rulebase details page, click the **Deploy To** icon.
4. In the Deploy To window, select the Hawk agents to which the rulebase must be deployed and click **Deploy**.

Result

A message indicating successful deployment is displayed.

Creating a Test in the Rule

In the Hawk Console, you can define test condition for the rules. The alerts are generated based on a predefined test condition. The test condition can be designed by using the rulebase and microagent variables.

Procedure

1. In the Hawk Console, open the Hawk Agent page to which you want to create a test condition.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent's page, select the **Rulebases** tab.
   All the rulebases for the Hawk agent are displayed. For details, see Rulebases Tab on page 30.
3. On the **Rulebases** tab, click the rulebase name which you want to edit.

   All the rules in the rulebase are listed. For details, see Rule Details Page on page 33.
4. Click the rule name to which you want to add the test condition.

   All the test conditions in the rule are listed. For details, see Test Details Page on page 35.
5. Under the Tests section, click the **New Test** icon.
6. In the New Test window, define a test condition by using the test builder.

   For details about the test condition builder elements, see Test Condition Builder Reference on page 48.
7. (Optional) Click **Advance Options** to add extra conditions to the test.

   For the list of Advance Options fields, see Table 7, Test Condition Advance Options Fields, on page 51.
8. (Optional) Click **Create and Add Action**.

   The New Action window opens to add an appropriate action for the rule if the test condition is true. For details, see Creating an Action for a Test Condition on page 56.
9. Click **Create Test**.

   On successful creation of the test condition, the New Test window closes.

Result

The new test condition is now listed on the Rule details page.

What to do Next

Create a new action for the test condition. For details, see Creating an Action for a Test Condition on page 56.

Test Condition Builder Reference

You can define a simple test condition by using just a test variable, a test operator, and a test value. If needed, you can also define a compound test condition by using multiple test expressions at multiple levels joined by logical operators.

Test Condition Builder Core Elements

The following figure shows a sample compound test condition and has numbers to mark different test elements. The following table lists the test elements marked in the figure that you can use to define the test condition.

Figure 16 Sample Test Condition with Element Markers

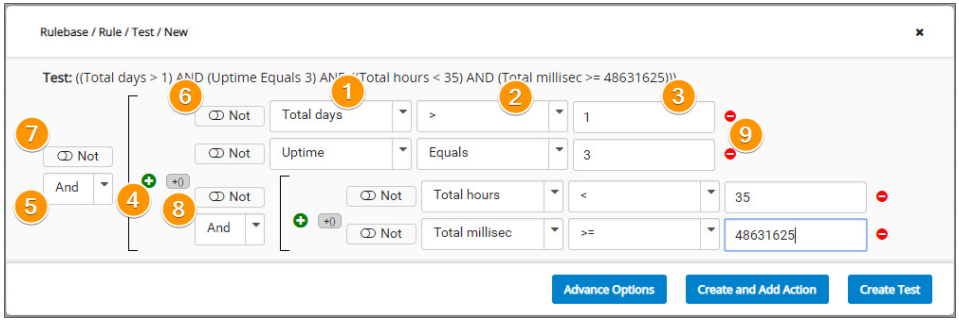


Table 6 Test Condition Elements

Legend No.	Test Element	Description
1	Test variable	Select the test variable for your test expression. The drop-down lists all the result fields of the microagent method used in the parent rule. For example, the following variables are listed for the `Self:getUptime()` method: • `Uptime` • `Total days` • `Total hours` • `Total millisec`
2	Test operator	Select the test operator for your test expression. The drop-down lists all the valid test operators based on the data type of the test variable. For the list of operators displayed based on the data type, see Test Operators Reference on page 52.
3	Test value	Enter the value of the test variable that you want to check for the test condition. Provide the values according to the data type of the test variable. When a test variable, the test operator, and a test value are provided, a test expression is created. For example, `(Total days > 1)`
4	Add Expression	Click the **Add Expression** icon to add one more expression to the test condition at the same level. The expressions are joined by using the logical operator specified (`AND` or `OR`) For example, the following test condition joins two expressions by using the `AND` operator: `((Total days > 1) AND (Total hours < 35))`

Table 6 Test Condition Elements

Legend No.	Test Element	Description
5	Logical operator	Select the logical operator to join two expressions. The values are: AND and OR. You can use the **Add Expression** icon to add one more expression to the test condition. For example, the following test condition joins two expressions by using the AND operator: `((Total days > 1) AND (Total hours < 35))`
6	NOT operator for expression	Select this NOT operator for the expression if you want to negate the result of that single expression. For example, the following test condition has an expression whose result is negated by using the NOT operator: `((Total days > 1) AND (NOT (Total hours < 35)))`
7	NOT operator for expressions set	Select this NOT operator for the set of expressions if you want to negate the result of that set of expressions. For example, the following is the test expression when the NOT operator is selected: `(NOT ((Total days > 1) AND (Total hours < 35)))`
8	Add sub-expression	Click to add a sub-expression, that is another set of test expression that is one level under (nested). For example, the following test condition has two expressions at the same level but another set of expression is nested: `((Total days > 1) AND (Uptime Equals 3) AND ((Total hours < 35) AND (Total millisec >= 48631625)))`
9	Remove Expression	Click the Remove Expression icon to remove the test expression from the test condition. **Note:** You must have at least one test expression to create a test condition.

Advance Options Fields

The following table lists the Advance Options fields displayed for the new test.

Table 7 Test Condition Advance Options Fields

Fields	Description
True Count Threshold	Enter the number of true evaluation for the test condition after which the action is triggered. For example, to check for consistently high CPU usage and ignore any brief spikes, you can set the true test counter for the test to five. The action is triggered when the test expression (CPU use high) is true for five consecutive test evaluations. The default value is `1`.
Schedules	Select the schedule that you want to apply for the test. The drop-down lists all the schedules deployed on the Hawk agent. The drop-down also lists the negative of the schedules as well, which when selected means that the conditions are checked at times other than the schedule. For example, if the `Weekend` schedule defines the time interval for every Saturday and Sunday then the `!Weekend` schedule means the time interval other than every Saturday and Sunday. By default, the test is always active. For more details about schedules, see Chapter 5, Schedule Management, on page 65.

Table 7 Test Condition Advance Options Fields

Fields	Description
Clear Condition Policy	Select a clear condition for the test. The values are: • (Default)**`CLEAR_ON_FIRST_FALSE`** - After the test becomes `true`, the test is cleared when the first time the test changes from `true` to `false`. This is the default behavior for a test with a synchronous data source. • **`CLEAR_TIMER`** - Specify a wait interval in seconds. After the test becomes `true` it remains `true` until this interval has passed without an additional `true` test. This is the default behavior for a test with an asynchronous data source, and the default wait interval is 900 seconds (15 minutes). • **`CLEAR_TEST`** - Specify an extra test expression for clearing the test. After the test becomes `true`, it becomes `false` only when the clear test expression becomes `true`. The clear test uses the microagent method result fields of the data source as input. For example, a test monitors each line in a log file for the string `Feed Line Down`. If this string is found, an alert is generated. A clear test for the original test checks for a log file line that signals the condition is resolved, such as `Feed Line Up`. When the clear test evaluates to `true`, the original alert message is cleared.

Test Operators Reference

The following tables describe the test operators you can apply to numeric, text and Boolean test variables while building test expressions.

Table 8 Test Operators for Numeric Method Results

Operator	Description
== ! =	The test expression is true when the value of the test parameter is (equal to, not equal to) the operator value.
< <= > >=	The test expression is true when the value of the test parameter is (less than, less than or equal to, greater than, greater than or equal to) the operator value.

Table 8 Test Operators for Numeric Method Results (Cont'd)

Operator	Description
InRange	The test expression is true when the value of the test parameter is between two extremes of a range. Endpoints are included.
OutOfRange	The test expression is true when the value of the test parameter is outside the range of two operator values. Endpoints are excluded.
Increase	The test expression is true when the value of the test parameter has increased at least by the operator value between two successive test evaluations. For example, the amount of disk space in use has increased by more than 10 MB in a sample period.
%Increase	The test expression is true when the value of the test parameter increases by at least the operator value as a percentage (the increase divided by the previous value times 100) between two successive test evaluations. For example, the amount of disk space in use has increased by more than 10 percent in a sample period.
Decrease	The test expression is true when the value of the test parameter decreases by at least the operator value between two successive tests.
%Decrease	The test expression is true when the value of the test parameter decreases by at least the operator value as a percentage (the decrease divided by the previous value times 100) between two successive test evaluations.
NetChange	The test expression is true when the value of the test parameter increases or decreases by at least the operator value between two successive test evaluations. The operator value specifies the absolute value of the increase or decrease.
%NetChange	The test expression is true when the value of the test parameter increases or decreases by at least the operator value as a percentage (the increase or decrease divided by the previous value times 100) between two successive test evaluations. The operator value specifies the absolute value of the percentage increase or decrease.

Table 8 Test Operators for Numeric Method Results (Cont'd)

Operator	Description
postedConditionExists	The test expression is true when the specified posted condition exists. This operator displays when a posted condition is selected in the parameter list. For more information, see Creating and Using Posted Conditions in Hawk Console
!postedConditionExists	The test expression is true when the specified posted condition does not exist. This operator displays when a posted condition is selected in the parameter list. For more information, see Creating and Using Posted Conditions in Hawk Console

Table 9 Test Operators for Text String Results

Operator	Description
Equals	The test expression is true when the value of the test parameter exactly matches the operator value. This is a case-sensitive match.
!Equals	The test expression is true when the value of the test parameter does not exactly match the operator value. This is a case-sensitive match.
StartsWith	The test expression is true when the value of the test parameter starts with the operator value. This is a case-sensitive match.
Contains	The test expression is true when the value of the test parameter contains the operator value. This is a case-sensitive match.
!Contains	The test expression is true when the value of the test parameter does not contain the operator value. This is a case-sensitive match.
Perl5 PatternMatch	The test expression is true when a match is found by using a regular expression as an operator value.

Table 10 Test Operators for Boolean Results

Operator	Description
isTrue	The test expression is true when the value of the test parameter is true.
isFalse	The test expression is true when the value of the test parameter is false.

Creating an Action for a Test Condition

In the Hawk Console, you can define an action to take when a test condition for a rule of the Hawk agent becomes true.

Procedure

1. In the Hawk Console, open the Hawk Agent page to which you want to add an action.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent's page, select the **Rulebases** tab.
   All the rulebases for the Hawk agent are displayed. For details, see Rulebases Tab on page 30.
3. On the **Rulebases** tab, click the rulebase name which you want to edit.

   All the rules in the rulebase are listed. For details, see Rule Details Page on page 33.
4. Click the rule name for which you want to add the action.

   All the test conditions in the rule are listed. For details, see Test Details Page on page 35.
5. Click the test condition name to which you want to add action.

   All the actions in the test are listed. For details, see Actions Details Page on page 36.
6. Under the Actions section, click the **New Action** icon.

   The New Action window opens.
7. In the New Action window, select the action you want to perform and enter the details to relevant fields. The following table lists all the available actions.

Table 11 Action Types in the Action Editor

Action Type	Result	Usage Notes
Alert (default)	Sends an alert message to Hawk Console	In the Message field, type the alert text that you want to display in the **Alerts** tab. Specify an alert level: high (default), medium or low.
Notification	Sends a notification message to Hawk Console	In the Notification field, type the notification text that you want to display in the Hawk Console **Alerts** tab.

Table 11 Action Types in the Action Editor (Cont'd)

Action Type	Result (Cont'd)	Usage Notes (Cont'd)
Method	Invokes a microagent method on the TIBCO Hawk agent machine	Select a microagent and method from the Microagent Info panel. Specify any required arguments.
Email	Sends an email message.	Specify a message recipient as *recipient@domain.com*. Specify a subject string, an SMTP mail server for sending the message, and message text.
Post Condition	Creates a posted condition to use in another rule in the same rulebase	In the Posted Condition field, type a label for the posted condition. For more information, see Creating and Using Posted Conditions in Hawk Console on page 64.

8. Select the **Schedule** that you want to apply to the action. For details about the schedules, see Chapter 5, Schedule Management, on page 65.
9. (Optional) Click **Advance Options** to add flexibility in timing when an action is performed.

   The following table lists the Advance Options fields displayed for the new test.

Table 12 New Rule Wizard Common Fields

Fields	Description
Escalation Period	To escalate a problem, type a wait interval in seconds in the Escalation Period field. The action is not performed the first time the associated test is `true`, but instead starts an internal timer. When the action is triggered by a test transition from `false` to `true`, the timer is started. If the associated test remains `true` for another evaluation after the specified interval, then the action is performed. You can use an escalation period to respond to continuing or deteriorating conditions.

Table 12 New Rule Wizard Common Fields

Fields	Description
Action Policy	Specify how actions are performed. The values are: • **`ALWAYS`** - The action is performed each time the associated test is evaluated as `true`, even if the test was `true` in the last evaluation. • **`COUNT_ON_INTERVAL`** - Specify the maximum number of times the action can be performed in the **Max Count** field, no matter how long the associated test continues to remain `true`. If the test becomes `false`, the counter is reset. Specify the number of seconds to wait between actions, in the **Interval** field, as long as the test is `true`. The related action can only be triggered at a test evaluation, so the actual interval between actions might be longer than the specified **Interval**. This option is useful when the action runs a paging script. A single page might be lost, but paging at each test evaluation (such as once per minute) is too often. With this option, you can send the page every five minutes until it is likely to be received. • **`ONCE_ONLY_UNTIL_MESSAGE_CHANGE`** - The first time this action is triggered by a test, the action is performed. On subsequent `true` evaluations, the action is performed only until there is a change in the alert message. This option is applicable only if the associated action creates an alert message with some string variables. The action is performed each time the value of the string variable changes resulting in a change in the alert message. Substituting variables in alert messages overrules this feature. • **`ONCE_ONLY`** - The first time this action is triggered by a test, the action is performed. On subsequent `true` evaluations, the action is not performed. The action is not performed again until the test becomes `false` and then `true` again. This is the default behavior for all actions. Substitution of variables in alert messages has no impact on this feature.

Table 12 New Rule Wizard Common Fields

Fields	Description
Max Count	Use this field when the **Action Policy** is `COUNT_ON_INTERVAL`. Specify the maximum number of times the action can be performed, no matter how long the associated test continues to remain `true`. If the test becomes `false`, the counter is reset.
Interval	Use this field when the **Action Policy** is `COUNT_ON_INTERVAL`. Specify the number of seconds to wait between actions, as long as the test is `true`.

10. Click **Create Action**.
    The New Action window closes and the action is created for the test condition.

What to do Next

Deploy the rulebase to the domain for the new action to take effect, see Deploying a Rulebase to the Associated Hawk Agent on page 45.

Variables in a Rulebase

You can reference several kinds of variables in a rulebase. By referencing variables, the rulebase can adapt to changes on multiple machines. For example, not all machines store log files or temporary files in the same directory. Also, rulebases used on multiple platforms have subtle differences in how path names are expressed. You can use variables rather than specifying this information manually.

When an action contains variable substitution, a new alert is generated each time the test is true and the value of the variable changes. Variable substitution is most useful for values that are slowly changing, very important or both.

Variable substitution affects the performance of rulebase processing. Therefore, you must reference a variable only when it provides a clear benefit.

Supported Variables Types

The following types of variables are supported in a TIBCO Hawk rulebase:

- External, such as user-defined variables
- Internal, such as the name of a test in a rule
- Data source, such as a microagent method result field (Data source variables can be referenced in actions only)

Referencing these variables outside of a rulebase is not supported.

External Variables

External variables are variables defined by a user on the machine where the TIBCO Hawk agent runs.

First, you define the variable values in a properties file on the local machine. Then you specify the variable file by using the `-variable` option when starting Hawk agent. Then you can reference the external variable in a rulebase. For more information on agent startup parameters, see *TIBCO Hawk Installation Configuration and Administration Guide*.

After variable values are defined and the properties file is specified to the agent, you can reference external variables in a rulebase by using the following syntax:

```
${External.<variable-name>}
```

where *variable-name* is the name of an environment variable defined in the properties file. The file uses a standard Java property file format, with one line per variable defined. Each entry is a name-value pair in the following format: *<variable-name>=<value>*

You can reference external variables in string arguments of actions and in data source method string arguments. For example, the Hawk Services sample rulebase provides a rule for sending a high-level alert. Without variable substitution, the text of the alert is generic. With variable substitution, the alert includes information specific to the generating condition.

Restrictions

In Microsoft Windows, the following restrictions apply to external variables:

- The variables file to support External variables in the agent must conform to the Java properties file format.
- Variables and variable names cannot include spaces or any of the following characters: equals sign (=), period (.), or forward slash(\).
- Any special characters must be escaped to be evaluated properly.

On UNIX systems, the `env` command outputs environment values in the correct format.

Internal Variables

Internal variables refer to elements of the current rulebase. This type of variable is defined internally by the TIBCO Hawk agent and requires no properties file. Values are assigned to variables when the rule is processed.

Like external variables, internal variables can be referenced in string arguments of methods used as a rule's data source or in string arguments of actions. You can manually type internal variable syntax in the string argument of a method, or, for action arguments, TIBCO Hawk Console provides a dropdown list of internal variables.

Manually entering variables

To manually enter internal variables, specify the variable by using the following syntax:

```
${Internal.<variable>}
```

where *<variable>* can be `Agent Name`, `Agent IP Address`, or so on.

The variables are substituted with the appropriate value before the command runs. For example, the command `Telnet ${Internal.Agent Name}` runs as `Telnet kimyou` if the command runs for agent `kimyou` from the Agent page.

Data Source Variables

Data source variables are Hawk variables that represent the return fields of a microagent method. The method must be used as the data source of the current rule. You can reference data source variables only in actions.

For example, the Hawk Services sample rulebase provides a rule for monitoring an event log and sending a high-level alert message when an error is written to the log. The Alert action type used in this rule allows you to specify a text string for the alert message. In this example, the text string is:

```
Hawk Agent : ${nextLine}
```

where `${nextLine}` is the text of the error message in the log. `nextLine` is a label for values returned by the microagent method that extracts information from the log file. Without variable substitution, you can include only static text, such as `High level alert` or a similar string, in the alert message.

How Variable Substitution Affects Actions

Action text strings can include variable references, where you include pertinent information from the data source in the alert text.

For example, the alert text:

```
Disk space on ${Instance} is at ${% Free Space}%
```

might display as:

```
Disk space on C: is at 10.2%
```

when generated. Or, if you call a script named `ClearTempFiles.exe` in an action whose data source provides information on disk partitions, you can specify the following command syntax:

```
ClearTempFiles.exe ${Instance}
```

and the agent inserts the name of the logical drive into the command line.

Variable substitution can cause actions to be taken more than once. If an action raises an alert with a variable reference, a new alert is generated at each test evaluation when the text message is different until the alert is cleared, even if the action that raises the alert was configured to take place only once.

Creating and Using Posted Conditions in Hawk Console

You can use posted condition to test for conditions in more than one managed object. A posted condition is an internal status message, similar to an alert message. Posted conditions are the result of actions in a rule and can pass status information to other rules in the same rulebase. Each rule uses only a single data source for input, so the posted condition serves as a link between rules with different data sources. For more information about posted conditions, see *TIBCO Hawk Concepts Guide.*

Procedure

1. When creating an action for a test condition, select **Post-Condition** as an action type. For steps, see Creating an Action for a Test Condition. You can use this post condition in another rule in the same rulebase.
2. To use post condition in another rule, in the test condition builder, click the **Add Expression** icon and, from the list, select the post condition that you have already created. You can apply `postedConditionExists`, `!postedConditionExists` and other numeric operators while building test expression. For details about the test condition builder elements, see Test Condition Builder Reference.

You can create and use multiple post conditions.

Chapter 5 Schedule Management

This chapter contains simple examples that demonstrate the operations supported on the Schedules tab.

Topics

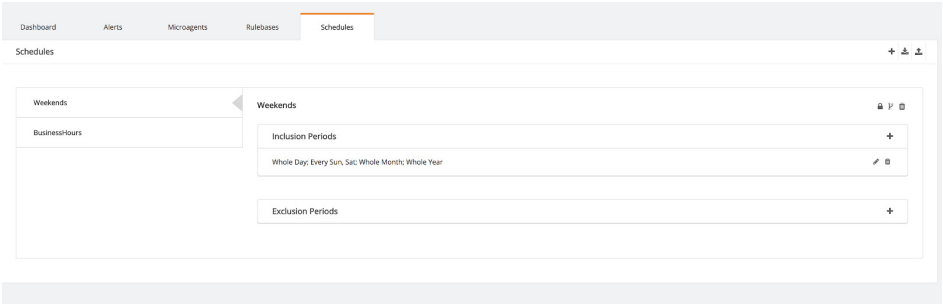
Schedules Tab

A schedule is a configuration object that defines when a rulebase, rule, test, or action is active. The schedule tab enables you to define a schedule and deploy the schedule.

For more details about schedules, see *TIBCO Hawk Concepts Guide*.

The **Schedules** tab lists the schedules in the left pane. Select the schedule to view its inclusion and exclusion period.

Figure 17 The Schedules Page for the Hawk Agent



Actions

The following actions are available on the Schedules tab for schedules:

- **Add new schedule** - Click the plus (+) icon to create a new schedule for the Hawk agent. For details, see Adding a Schedule on page 68.
- **Export schedules** - Use this option if want to create same schedules in another Hawk agent. You can use the exported schedule (.hsf) file to import these schedules to another Hawk agent and this saves you time to recreate the same schedules. For details, see Exporting a Schedule on page 74.
- **Import schedules** - Use this option if you have some exported schedules from another Hawk agent which you want to use in your Hawk agent. For details, see Importing a Schedule on page 75.

The following actions are available for each schedule:

- **Deploy schedule** - Click the deploy icon ( ) to deploy the schedule to the Hawk agent. After deployment, the schedule can be applied to the rulebases and rules for the Hawk agent. For details, see Deploying a Schedule to the Associated Hawk Agent on page 76.

- **Deploy schedule to another agent** - Click the deploy-to icon ( ) to deploy the schedule to another Hawk agent registered in the same Hawk Console. After deployment, the schedule can be applied to the rulebases and rules for the Hawk agent to which the schedule is deployed. For details, see Deploying a Schedule to Another Hawk Agent on page 77.
- **Delete schedule** - Click the delete icon ( ) to delete the schedule. For details, see Deleting a Schedule on page 78.
- **Add inclusion period** - Click the plus icon ( ) on the inclusion period, for the selected schedule, to specify the time period when you would like the system to apply the rule or rulebases depending on whether the conditions are met. You can define the following parameters in the inclusion period of a schedule:
  - Time of the day
  - Day of the month
  - Week day of the month
  - Month of the year

  For details, see Adding Inclusion Period to a Schedule on page 68.
- **Add exclusion period** - Click the plus icon ( ) on the exclusion period for the selected schedule to specify the time period when you would like the system to ignore the rule or rulebases. For details, see Adding Exclusion Period to a Schedule on page 70.

Adding a Schedule

A schedule is a configuration object that defines when a rulebase, rule, test, or action is active. The Schedules tab enables you to define a schedule and deploy the schedule. Then you can send the schedule to one or more Hawk agents, and apply the schedule to rulebase objects.

Procedure

1. In the Hawk Console, open the Hawk Agent page for which you want to create the schedule.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Schedules** tab.

   All the schedules for the Hawk agent are displayed in the left panel of the tab. For details, see Schedules Tab on page 66.
3. In the Schedules tab, click the **Add** icon

   The Add Schedule window opens to specify schedule details.
4. In the Add Schedule window, enter the new **Schedule Name** and select the **Time Zone** of the schedule.
5. Click **Save**.

Result

The new schedule with the specified **Schedule Name** is listed in the left panel.

What to do Next

Define the inclusion period of the schedule. For details, see Adding Inclusion Period to a Schedule on page 68.

Adding Inclusion Period to a Schedule

You can define a period when you want the system to apply the rulebases and rules whenever the conditions are met.

Procedure

1. In the Hawk Console, open the Hawk Agent page for which you want to add the inclusion period of a schedule.
   For steps, see Viewing the Agent Details on page 12.

2. In the Agent page, select the **Schedules** tab.

   All the schedules for the Hawk agent are displayed in the left panel of the tab. For details, see Schedules Tab on page 66.

3. In the **Schedules** tab, select the schedule for which you want to add the inclusion period.

   The list of inclusion periods and exclusion periods are displayed in the right panel.

4. On the left panel, click the **Add** icon for the Inclusion Periods.

   The Period Details window opens to define a new period.

5. In the Period Details window, define the following parameters in the inclusion period of a schedule:

   — **Time of Day** - Select the starting and ending time interval for the period. If you want to specify multiple time intervals for the schedule, you must add multiple Inclusion periods.

   — **Day of Month** - Select the dates for the schedule to be active. Click **Select All** to select all days of the month.

   — **Week Day of Month** - Select the days of the week and weeks in the month for the period.

   — **Month of Year** - Select the month of the year for the period. Click **Select All** to select all months of the year.

   The Week Day of Month and Day of Month selections must overlap in order for a day to be selected.

6. Click **Save**.

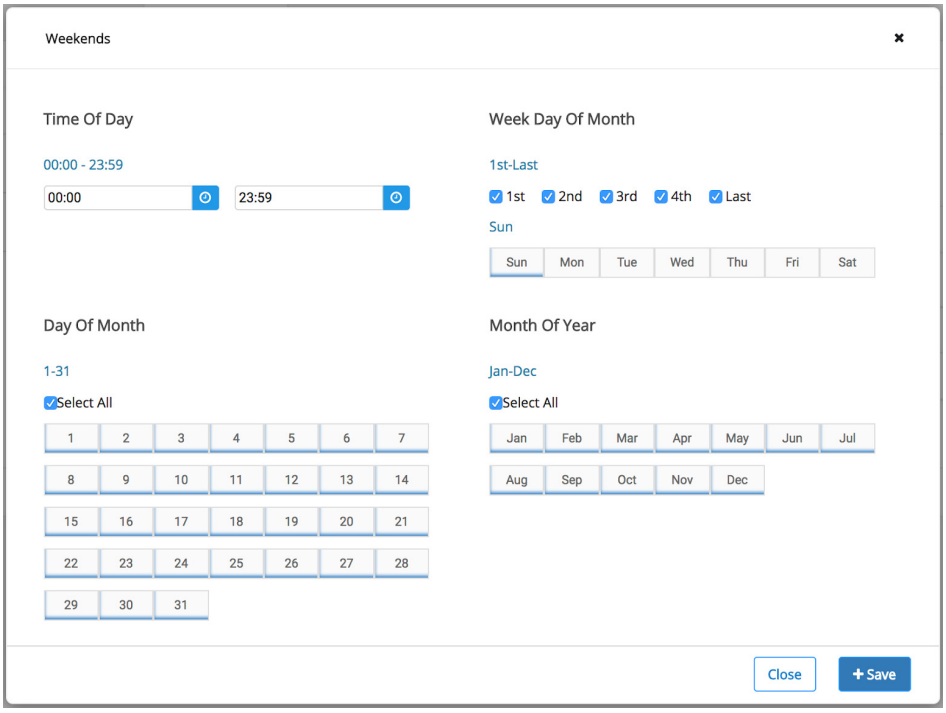
Result

The new inclusion period is listed under the **Inclusion Periods** list for the schedule.

Example

For example, the following figure shows the selection for the inclusion period of a *Weekend* schedule, where the applied rules and rulebases are active on every Sunday for 24 hours.

Figure 18 Weekend Schedule Inclusion Period



What to do Next

- (Optional)If you want to define a period when you want the system to ignore the rulebases and rules, add an exclusion period for the schedule. For details, see Adding Exclusion Period to a Schedule on page 70.
- Deploy the schedule to the Hawk agent for applying it to all rules and rulebases of the Hawk agent. For details, see Deploying a Schedule to the Associated Hawk Agent on page 76.

Adding Exclusion Period to a Schedule

You can define a period when you want the system to ignore the rulebases and rules.

Procedure

1. In the Hawk Console, open the Hawk Agent page for which you want to add the exclusion period of a schedule.
   For steps, see Viewing the Agent Details on page 12.

2. In the Agent page, select the **Schedules** tab.

   All the schedules for the Hawk agent are displayed in the left panel of the tab. For details, see Schedules Tab on page 66.

3. In the **Schedules** tab, select the Schedule for which you want to add the exclusion period.

   The list of inclusion periods and exclusion periods are displayed in the right panel.

4. On the right panel, click the **Add** icon for the Exclusion Periods.

   The Period Details window opens to define a new period.

5. In the Period Details window, define the following parameters in the exclusion period of a schedule:

   — **Time of Day** - Select the starting and ending time interval for the period. If you want to specify multiple time intervals for the schedule, you must add multiple Exclusion periods.

   — **Day of Month** - Select the dates for the schedule to be active. Click **Select All** to select all days of the month.

   — **Week Day of Month** - Select the days of the week and weeks in the month for the period.

   — **Month of Year** -Select the month of the year for the period. Click **Select All** to select all months of the year.

   The **Week Day of Month** and **Day of Month** selections must overlap in order for a day to be selected.

6. Click **Save**.

Result

The new exclusion period is listed under the **Exclusion Periods** list for the schedule.

What to do Next

- If you want to define a period when you want the system to apply the rulebases and rules whenever conditions are met, add an inclusion period for the schedule. For details, see Adding Inclusion Period to a Schedule on page 68.
- Deploy the schedule to the Hawk agent for applying it to all rules and rulebases of the Hawk agent. For details, see Deploying a Schedule to the Associated Hawk Agent on page 76.

Editing a Schedule

If needed, you can edit a schedule and redeploy it to a Hawk agent. You cannot modify the name of a schedule but you add or edit the inclusion and exclusion periods.

Procedure

1. In the Hawk Console, open the Hawk Agent page for which you want to edit the schedule.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Schedules** tab.

   All the schedules for the Hawk agent are displayed in the left pane of the tab. For details, see Schedules Tab on page 66.
3. In the **Schedules** tab, select the Schedule which you want to edit.

   The list of inclusion periods and exclusion periods are displayed in the right panel.
4. On the right panel, you can perform either of the following operations to edit the schedule:
   - Click the **Add** icon for the Inclusion Periods to add a new inclusion period, see Adding Inclusion Period to a Schedule on page 68.
   - Click the **Edit** icon for the inclusion period to edit the period details, see step 5 and step 6 in Adding Inclusion Period to a Schedule on page 68.
   - Click the **Delete** icon for the inclusion period to delete that inclusion period.
   - Click the **Add** icon for the Exclusion Periods to add a new exclusion period, see Adding Exclusion Period to a Schedule on page 70.
   - Click the **Edit** icon for the exclusion period to edit the period details, see step 5 and step 6 in Adding Exclusion Period to a Schedule on page 70.
   - Click the **Delete** icon for the exclusion period to delete that exclusion period.

Result

The new or updated inclusion and exclusion periods are listed for the schedule under the **Schedules** tab.

What to do Next

Deploy the updated schedule to the Hawk agent for applying it to all rules and rulebases of the Hawk agent. For details, see Deploying a Schedule to the Associated Hawk Agent on page 76.

Exporting a Schedule

If needed, you can also apply the same schedule that you defined for a Hawk agent to another Hawk agent. You can export the schedules from a Hawk agent to another Hawk agent.

Procedure

1. In the Hawk Console, open the Hawk Agent page from which you want to export the schedule.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Schedules** tab.

   All the schedules for the Hawk agent are displayed in the left pane of the tab. For details, see Schedules Tab on page 66.
3. In the Schedules tab, click the **Export Schedule** icon.
4. In the Save As window, browse to the location where you want to save the schedules file (`.hsf`), enter a name for the file, and click **Save**.

Result

The schedules file (`.hsf`) is saved in the location specified.

What to do Next

You might want to import the exported schedules file (`.hsf`) to another Hawk agent registered in another Hawk Console. For details, see Importing a Schedule on page 75.

Importing a Schedule

If needed, you can also apply the same schedule that was defined for another Hawk agent to your Hawk agent. You can import the schedules, exported from another Hawk agent, to your Hawk agent.

Procedure

1. In the Hawk Console, open the Hawk Agent page to which you want to import the schedule.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Schedules** tab.

   All the schedules for the Hawk agent are displayed in the left panel of the tab. For details, see Schedules Tab on page 66.
3. In **Schedules** tab, click the **Import Schedule** icon.

   The Open window opens to select the schedules file (`.hsf`).
4. In the Open window, browse to the location of the schedules file (`.hsf`), select the schedules file (`.hsf`), and click **Open**.

Result

The schedules from the schedules file (`.hsf`) are listed in the Schedules tab.

What to do Next

Deploy the schedule to the Hawk agent for applying it to all rules and rulebases of the Hawk agent. For details, see Deploying a Schedule to the Associated Hawk Agent on page 76.

Deploying a Schedule to the Associated Hawk Agent

For applying the schedules to your Hawk agent, you must first deploy them on the Hawk agent.

Procedure

1. In the Hawk Console, open the Hawk Agent page to which you want to deploy the schedule.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Schedules** tab.

   All the schedules for the Hawk agent are displayed in the left panel of the tab. For details, see Schedules Tab on page 66.
3. In the **Schedules** tab, select the schedule that you want to deploy.

   The list of inclusion and exclusion periods for the schedule is displayed in the right panel.
4. Click the **Deploy Schedule** icon for the schedule.

   A Confirmation dialog box is displayed.
5. In the Confirmation dialog box, click **Yes** to deploy the schedule.

Result

After successful deployment, the success message dialog box is displayed. Click **Ok** to close the success message dialog box.

Deploying a Schedule to Another Hawk Agent

To apply the schedules to another Hawk agent registered on Hawk Console, you must first deploy them on that Hawk agent.

Procedure

1. In the Hawk Console, open the Hawk Agent page which has the schedule you want to deploy.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Schedules** tab.

   All the schedules for the Hawk agent are displayed in the left panel of the tab. For details, see Schedules Tab on page 66.
3. In the **Schedules** tab, select the schedule that you want to deploy.

   The list of inclusion and exclusion periods for the schedule is displayed in the right panel.
4. Click the **Deploy To** icon for the schedule.

   The Deploy To window is displayed to select the Hawk agent.
5. In the Deploy to window, select Hawk agents to which you want to deploy the schedule and click **Yes**.

Result

After successful deployment, the success message dialog box is displayed. Click **Ok** to close the success message dialog box.

Deleting a Schedule

If you don't require a schedule for the Hawk agent, you can delete the schedule from the Hawk Console.

Procedure

1. In the Hawk Console, open the Hawk Agent page from which you want to delete the schedule.
   For steps, see Viewing the Agent Details on page 12.
2. In the Agent page, select the **Schedules** tab.

   All the schedules for the Hawk agent are displayed in the left panel of the tab. For details, see Schedules Tab on page 66.
3. In the **Schedules** tab, select the schedule that you want to delete.

   The list of inclusion and exclusion periods for the schedule is displayed in the right panel.
4. Click the **Delete** icon for the schedule.

   A Confirmation dialog box is displayed.
5. In the Confirmation dialog box, click **Yes** to delete the schedule.

   After successful deletion, the success message dialog box is displayed. Click **Ok** to close the success message dialog box.

Result

The deleted schedule is also deleted from the schedules list under the **Schedules** tab.

Chapter 6 Rulebase Repository Management

This chapter contains information about the rulebase repository that stores configuration objects. The configuration objects include rulebases, schedules, and rulebase map.

Topics

Overview

The Hawk Console contains a rulebase repository that stores configuration objects and then distributes and deploys them to Hawk agents. The configuration objects include rulebases, schedules, and rulebase map. You can create new objects or update existing objects in the repository. Hawk Console deploys the configuration objects from the repository to selected Hawk agents when they start.

Rulebase Mapping

A rulebase mapping defines a mapping between rulebases and Hawk agents. The Hawk Console deploys the mapped rulebases to the Hawk agent during start. The mapping can be between a rulebase and Hawk agent, or between a rulebase and a group of Hawk agents. You can create a rulebase mapping by using the Hawk Console.

For details about creating a rulebase mapping, see Creating Rulebase Mapping, on page 83.

Agent Groups

The Hawk agents that have similar rulebase needs are grouped together in an agent group. The agent groups are of two types: system-defined and user-defined.

System-defined groups

Hawk automatically creates the following agent groups:

- Operating system group — Hawk groups Hawk agents based on their operating system. The name of agent groups starts with a ++ symbol followed by the name of the operating system.

  For example, all the Hawk agent running on the Solaris operating system are part of the `++Solaris` group.

- All group — Hawk groups all registered Hawk agents into one group. The name of the agent group is ++.

User-defined groups

You can group any Hawk agent into a group. The name of the group starts with the + symbol.

For details about creating an agent group, see Creating Agent Groups, on page 82.

Rulebases and Schedules

In the rulebase repository, rulebase and schedule in a rulebase repository work in the same way as they work in Hawk agents.

You can also use existing rulebases and schedules from Hawk agents and store them in a rulebase repository. For details, see Migrating Rulebases and Schedules from Hawk Agent to Rulebase Repository, on page 84.

Also, for details about the effect of actions performed on rulebases and schedules in the rulebase repository, see Actions on Rulebase Repository Configuration Objects, on page 86.

Rulebase Repository Configuration

You must configure the repository in the Hawk Console by using the `hawk_console_repository_path` environment variable. It specifies the path of the repository on the Hawk Console machine. Hawk Console loads and saves the configuration objects in the repository at `<hawk_console_repository_path>/<domain_name>`.

Rulebase Repository Actions

Hawk Console updates the configuration objects based on the action performed on the configuration objects.

For details about the effect of actions performed in repository, see Actions on Rulebase Repository Configuration Objects, on page 86.

Creating Agent Groups

The Hawk agents that have similar rulebase needs are grouped together in an agent group. You can create an agent group by using the Hawk Console.

To create an agent group:

1. In Hawk Console, open the domain for which you have to create an agent group.
2. On the **Domain** page, click **Rulebase Repository**.
3. On the **Rulebase Repository** page, select the **Groups** tab.
4. On the **Groups** tab, click the **Add Group** icon  .
5. Enter the name of the new agent group and click **OK**.
6. From the **Available Members** list, click the **Add Members** icon  for the Hawk agent that you want to add to the agent group.
7. Verify the **Members Mapped** list for the agent group and click **Save Mapping**.

You can now map the rulebase to this agent group. For more details about rulebase mapping, see Creating Rulebase Mapping, on page 83.

Creating Rulebase Mapping

A rulebase mapping defines a mapping between rulebases and Hawk agents. You can create a rulebase mapping by using the Hawk Console.

To create a rulebase mapping:

1. In Hawk Console, open the domain for which you have to create a rulebase mapping.
2. On the **Domain** page, click **Rulebase Repository**.
3. On the **Rulebase Repository** page, select the **Rulebase-Map** tab.
4. On the **Rulebases-Map** tab, select a rulebase which you want to map to Hawk agents or groups.
5. From the **Available Members** list, click the **Add Members** icon  for the group or Hawk agent that you want to map to the rulebase.
6. Verify the **Members Mapped** list for the rulebase and click **Save Mapping**.

Migrating Rulebases and Schedules from Hawk Agent to Rulebase Repository

For the optimized use of the rulebase repository, you must migrate existing rulebases and schedules from Hawk agents to the rulebase repository. In Hawk Console, you can create a rulebase mapping for these migrated rulebases. Based on these rulebase mapping, the rulebase repository deploys the rulebases and schedules to Hawk agents.

You can migrate the rulebases and schedules from Hawk agents to a rulebase repository by following either of these procedures:

Table 13 Migrating rulebases and schedules from Hawk agents to a rulebase repository

Scenario	Steps
Move each rulebase one by one	1. In Hawk Console, open the Hawk Agent page from which you want to migrate the rulebase. 2. On the Agent page, select the **Rulebases** tab. 3. On the **Rulebases** tab, from the rulebases list, under the **Actions** column, select the **Send to Repository** option for the rulebase that you want to migrate. 4. Click **Yes** to confirm the migration.
Move each schedule one by one	1. In Hawk Console, open the Hawk Agent page from which you want to migrate the schedule. 2. On the Agent page, select the **Schedules** tab. 3. On the **Schedules** tab, select the schedule that you want to migrate. 4. Click the **Send to Repository** icon  . 5. Click **Yes** to confirm the migration.

Table 13 Migrating rulebases and schedules from Hawk agents to a rulebase repository

Scenario	Steps
Move rulebases and schedules in bulk	1. Copy all the rulebase files (`.hrb`) and schedule files (`.hsf`) from your Hawk agent to the domain folder in the Hawk Console repository path. The repository path is specified by the `hawk_console_repository_path` environment variable. Thus, the path to copy the rulebase and schedule files is `<hawk_console_repository_path>/<domain_name>`. 2. Start the Hawk Console to load these rulebases and schedules in the rulebase repository.

The rulebase repository in Hawk Console, lists all the migrated rulebases and schedules. You can then perform different operations on these rulebases and schedules.

Actions on Rulebase Repository Configuration Objects

Hawk Console contains a rulebase repository that stores configuration objects and then distributes and deploys them to Hawk agents. The configuration objects include rulebases, schedules, and rulebase map.

Rulebase Repository Configuration Objects Files

You must configure the repository in Hawk Console by using the `hawk_console_repository_path` environment variable. Hawk Console loads and saves configuration objects in the repository at `<hawk_console_repository_path>/< domain_name>`. Hawk Console stores the configuration objects in the following files:

Configuration Object	File Extension
Rulebase mapping and agent group mapping	`.hrm` The default file is `rbmap.hrm`.
Rulebase	`.hrb`
Schedule	`.hsf` The default file is `schedules.hsf`.

Hawk Console Actions

The following table lists the effect of the action performed on the Hawk Console to the configuration objects:

Table 14 Effects of Actions in Rulebase Repository

Event / Action	Effect on Rulebase Map	Effect on Rulebase	Effect on Schedule
Hawk Console starts up	Hawk Console loads rulebase mapping and agent groups from the `rbmap.hrm` file present in the rulebase repository.	Hawk Console loads all rulebase files from the rulebase repository.	Hawk Console loads schedules from the `schedules.hsf` file present in the rulebase repository.

Table 14 (Cont'd)Effects of Actions in Rulebase Repository

Event / Action	Effect on Rulebase Map	Effect on Rulebase	Effect on Schedule
Hawk agent starts up	If the agent group for Hawk agent operating system is not present, Hawk Console adds a new agent group to the list of operating system groups of the rulebase repository.	Hawk Console deploys rulebases to the respective Hawk agent based on the rulebase mapping.	Hawk Console deploys all the schedules from the rulebase repository to the Hawk agent.
Create and save a new rulebase or schedule	Not applicable	The Hawk Console adds the new rulebase to the rulebase repository and saves it to a `.hrb` file. **What to do next:** To deploy the new rulebase to Hawk agents, you must create a rulebase mapping for it, see Creating Rulebase Mapping, on page 83.	Hawk Console adds a new schedule to the rulebase repository and saves it to the schedule.hsf file. **What to do next:** To deploy this new schedule to all Hawk agents in the domain, on the **Schedules** tab, click the **Deploy Schedule** icon and confirm the action.
Update and save an existing rulebase or schedule	Not applicable	The Hawk Console updates the rulebase in the rulebase repository and saves the update to the respective `.hrb` file. **What to do next:** To deploy the updated rulebase to all mapped Hawk agents, on the **Rulebase Mapping** tab, click **Save Mapping** for the rulebase.	Hawk Console updates the schedule in the rulebase repository and saves the update to the `schedule.hsf` file. **What to do next:** To deploy this updated schedule to all Hawk agents in the domain, on the **Schedules** tab, click the **Deploy Schedule** icon and confirm the action.

Table 14 (Cont'd)Effects of Actions in Rulebase Repository

Event / Action	Effect on Rulebase Map	Effect on Rulebase	Effect on Schedule
Delete a rulebase	Hawk Console removes all the rulebase mappings for the rulebase and updates the `rbmap.hrm` file.	Hawk Console deletes the rulebase file (`.hrb`) from the rulebase repository. If the deleted rulebase was mapped to Hawk agents or agent groups, then these mapped rulebases are undeployed from Hawk agents after they are restarted.	Not applicable
Delete a schedule	Not applicable	Not applicable	Hawk Console deletes the schedule from the rulebase repository and the `schedule.hsf` file. If the deleted schedule was deployed to Hawk agents or agent groups, then these schedules are undeployed from Hawk agents after they are restarted.
Create and save a rulebase mapping	Hawk Console adds rulebase mapping to the rulebase repository and updates the `rbmap.hrm` file.	Hawk Console deploys the rulebase to the mapped Hawk agents and agent groups.	Not applicable

Table 14 (Cont'd)Effects of Actions in Rulebase Repository

Event / Action	Effect on Rulebase Map	Effect on Rulebase	Effect on Schedule
Update and save existing rulebase mapping	Hawk Console updates rulebase mapping in the rulebase repository and updates the `rbmap.hrm` file.	If Hawk agents or agent groups are added to the rulebase mapping, Hawk Console deploys the rulebase to new members. If Hawk agents or agent groups are removed from the rulebase mapping, Hawk Console undeploys the respective rulebase from those Hawk agents or members of agent group.	Not applicable
Create a new agent group	Hawk Console creates an agent group in memory only. **What to do next:** To save the agent group information in the `rbmap.hrm` file, either add Hawk agents to the agent group or map the group to a rulebase. For details, see Creating Agent Groups, on page 82 and Creating Rulebase Mapping, on page 83.	Not applicable	Not applicable

Table 14 (Cont'd)Effects of Actions in Rulebase Repository

Event / Action	Effect on Rulebase Map	Effect on Rulebase	Effect on Schedule
Add new Hawk agents to the agent group and save the agent group mapping	Hawk Console updates the group mapping information in the `rbmap.hrm` file in the rulebase repository.	If the agent group is already mapped to rulebases, then these mapped rulebases are not automatically deployed to new agents. **What to do next:** To deploy the mapped rulebase to new Hawk agents, on the **Rulebase Mapping** tab, click **Save Mapping** for the rulebases that are mapped to the updated agent group.	Not applicable
Remove Hawk agents from the agent group and save the agent group mapping	Hawk Console updates the group mapping information in the `rbmap.hrm` file in the rulebase repository.	If the agent group is already mapped to rulebases, then these mapped rulebases are not automatically undeployed from the removed Hawk agents. **What to do next:** To undeploy the mapped rulebases from the removed Hawk agents, on the **Rulebase Mapping** tab, click **Save Mapping** for those mapped rulebases.	Not applicable

Table 14 (Cont'd)Effects of Actions in Rulebase Repository

Event / Action	Effect on Rulebase Map	Effect on Rulebase	Effect on Schedule
Delete an agent group	Hawk Console deletes the agent group from the rulebase repository and updates the `rbmap.hrm` file.	If the deleted agent group was mapped to rulebases, then these mapped rulebases are undeployed from Hawk agents after they are restarted.	Not applicable

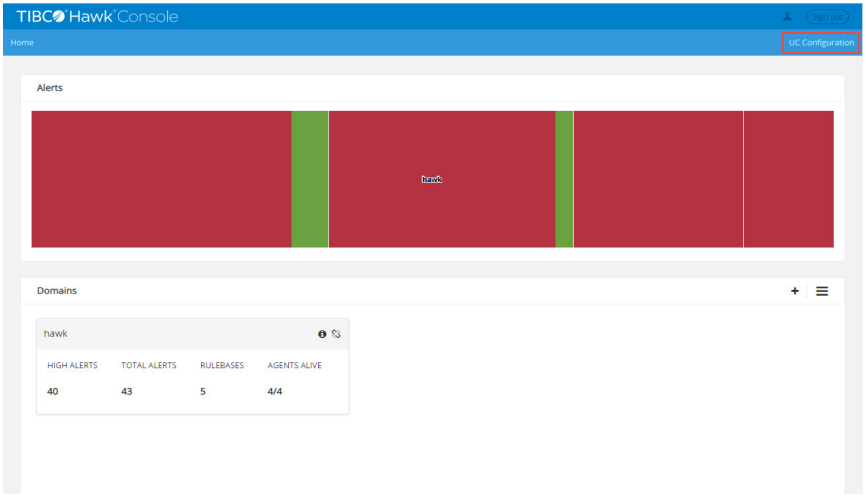
Chapter 7 Universal Collector Microagent Management

You can configure and manage Universal Collector microagent using Hawk Console. Universal Collector microagent is available as a Hawk plugin. To configure the plug-in into your Hawk installation, see *Hawk Plug-in reference guide*. This chapter discusses the collection of logs, rulebases and metrics from different sources and forwarding logs to LogLogic LMI or Syslog server.

Topics

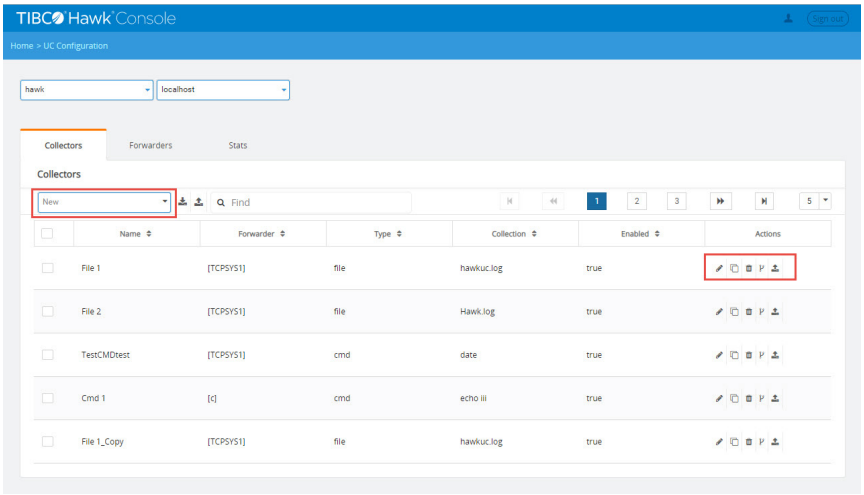
Collecting Logs

You can collect logs from different sources: Real-Time File logs, Syslog, Remote files, Command logs. You can also collect metrics from Hawk Rulebases and Microagents. Click **UC Configuration** tab in the upper-right corner to view the list of collectors and forwarders.



Creating and Configuring Log Sources

You can add, copy, delete, import, export, and deploy log sources.



Adding a New Log Source

You can add a new log source by using the following steps:

1. In the Hawk Console, click the **UC Configuration** tab in the upper-right corner.
2. Select **Domain** and **Agent** from dropdown list. List of collectors and forwards is displayed.
3. From **New** dropdown list, select the type of log sources you want to add:

- Real Time File
- Syslog
- Remote Files
- Cmd
- Hawk Rulebase
- Hawk Metrics

4. In the Log Source Configure screen, enter the relevant information as explained in Editing Log Sources on page 97.
5. Click **Configure** to save the log source.

Result

A new log source is added in the list of log sources.

Copying Log Source

To copy a log source, click **Copy** icon in the Actions column of the log source.

The copied log sources are displayed below the list of log sources with the name *Log_source_name*_copy.

Deleting Log Source

To delete a log source, click Delete icon in the Actions column of the log source.

Deploying Log Source

You can deploy log source to one or more hawk agents.

Procedure

1. Open the Universal Collector microagent Console by clicking the **UC Configuration** tab in the upper-right corner.

2. Click Deploy to icon in the Actions column of the log source you want to deploy.

3. In Deploy Collector to window, select Hawk agents to which you want to deploy the log source and click **Yes**.

Importing Log Sources

Procedure

1. Open the Universal Collector microagent Console by clicking the **UC Configuration** tab in the upper-right corner.

2. On the Collectors tab, click import icon beside New dropdown list.

- To import single collector, select the earlier exported collector (`.xml`) file.
- To import multiple collectors at the same time, select the .zip file which contains multiple collector (`.xml`) files.

3. Click **Open**.

Result

If the import is successful, the imported Collector is listed on the **Collectors** tab.

Exporting Log Sources

Procedure

1. Open the Universal Collector microagent Console by clicking the **UC Configuration** tab in the upper-right corner.

2. On the **Collectors** tab, export the collector by using either of the following ways:

- From the collectors list, under the Actions column, select the Export option for the collector that you want to export.
- To export multiple collectors, select the check boxes next to collectors which you want to export and click the export icon in the top beside new dropdown list. The .zip is downloaded to the Downloads folder of your machine which contains exported collectors (.xml) files.

3. In the Save As window, browse to the location where you want to save the .xml or .zip file and click **Save**.

Editing Log Sources

You can edit log source configuration using edit icon in the **Actions** column of Log Source list.

Editing a Real-Time File Log Source

You can modify the following fields of the Real-Time File log source:

Option	Description
General	
Log Source Enabled	Click toggle button **Yes** or **No** to define whether the current Log Source is enabled or disabled.
Name	Name of the Log Source.
Description	Description of the Log Source.
Forwarders	
Select Forwarder	Select the Forwarding connection to which you want to forward collected RT File logs from dropdown list.
Universal Collector Collection date	Define whether the log message sent to the LogLogic LMI server remains in a local system time zone or is converted into UTC time zone.
Message Filter	
Message Filter	Click Yes or No to activate or deactivate the option.

Option	Description
Collect messages	Define whether you collect messages that: • Collect logs matching regex pattern • Not matching Regex (that is, filter the logs that match the regex)
Filter	Enter a case insensitive regular expression to specify the messages to be matched. For example, if “Not matching regex is selected”: "packet accepted" means that all the lines containing packet accepted are filtered. "^64\.242" means that all the lines that are beginning exactly with 64.242 are filtered. "846$" means that all the lines that are ending exactly with 846 are filtered. For example, if “Matching regex is selected”: "packet accepted" means that only the lines containing packet accepted are kept. "^64\.242" means that only the lines that are beginning exactly with 64.242 are kept. "846$" means that only the lines that are ending exactly with 846 are kept.
Collection	
File Path	Specify the path of the log file to be collected. **NOTE:** Log file must be present at host machine where Hawk Agent is running. If the log file is rotated, you may enter [id] or [date] or both in the file name and configure the File rotation parameters. For example, `c:\temp\logFile[date].log` to obtain file names such as `logFile20170521.log`
File rotation	Click Yes or No to activate or deactivate the option.
(If File rotation is active) Date pattern	Select the check box and enter the date format you want to use for the [date] parameter. For example, yyyyMMdd for 20170421.

Option	Description
(If File rotation is active) Max number of digits	Select the check box and enter the maximum number of digits you want for the [id] parameter. Universal Collector microagent can collect any file with an [id] whose number of digits is between 1 and 9 inclusive. For example, If you set 5, the following [id] will be taken into account: 1, 054, 586, 00599, 78945, etc.
File change notification	Click Yes or No to activate or deactivate the option. This option allows you to monitor file changes. If set Yes, a notification will be sent to LogLogic LMI through the uc.log file when the modified date of the specified file changes. The notification includes the changed content and time. A new log is recorded for the notification when Universal Collector microagent internal logs are forwarded to LogLogic LMI. The file changes are not monitored for rotated files. In this case, the File change notification option is disabled. The specified file size must be less than the default size (10MB). If the file size is more than 10MB, the notification does not include changed content. Before activating this monitoring option, ensure that you set the **LMI Connection > Forwarding > Forward UC Internal Logs** option is ON.
Multiline messages	Click Yes or No to activate or deactivate the option to define whether the single message has several lines.
(If Multiline messages is ON) Multiline Header Type	Select the type of multi-line logs. For example, 'jboss', 'tomcat', 'weblogic', 'websphere' or 'custom'.
(If Multiline messages is ON) Custom Header regex	Set a regular expression matching the header of the first line of a log.
(If Multiline messages is ON) Custom Separator	Specify a custom delimiter to use as a separator for multiple lines. The default separator is \r\n. If the field is empty, a space is added in the message.
(If Multiline messages is ON) Send orphaned lines	Indicate whether you want Universal Collector microagent to send messages that do not match the Header Regex.

Option	Description
(If Multiline messages is ON) Multiline timeout after detected header	Indicate the number of seconds after which the multi-line logs are ready to be sent.
Advanced	
Host name	Enter the name of the host used to pair logs on the LogLogic LMI server. For example, customHostname.com If you enter an IPv4 address, the device to be displayed in LogLogic LMI will be referred with this IP address.
Application name	Enter the name of the application used to identify logs on the LogLogic LMI server. For example, customApplicationName
Maximum message length	Indicate the possible maximum length for the message (in bytes). The maximum supported value is 1048576. Default value: 64000 To specify a message length of more than 64000 bytes, ensure that you use LogLogic LMI 6.2.0 and later versions.
Charset	Select the data format. Default value: Use local system charset

Editing a Syslog Log Source

You can modify the following fields of the Syslog log source:

Option	Description
General	
Log Source Enabled	Click toggle button **Yes** or **No** to define whether the current Log Source is enabled or disabled.
Name	Name of the Log Source.

Option	Description
Description	Description of the Log Source.
Forwarders	
Select Forwarder	Select the Forwarding connection from dropdown list to which you want to forward collected Syslog logs.
Universal Collector Collection date	Define whether the log message sent to the LogLogic LMI server remains in a local system time zone or is converted into UTC time zone.
Collection	
Protocol	Define whether the Log Source uses the UDP/TCP SYSLOG protocol. To listen on both UDP and TCP protocols, you must create two Syslog Log Sources.
Port	Enter the port to listen to the Syslog flow. Default value: 514
Binding interface	If there are multiple network interfaces, enter the IP address to listen to the Syslog flow. Only one IP address is possible. To listen to all network interfaces for IPv4, use 0.0.0.0. To listen to a specific interface for IPv4, use an address like 192.168.11.10 Default value: 0.0.0.0 When there are multiple syslog collectors, if one of the collectors has been bound to a specific interface, all remaining collectors cannot be bound to 0.0.0.0. The remaining collectors must be bound to other specific interfaces.
Message Filtering	
Filtering	Click Yes or No to activate or deactivate the option. If Message Filtering is set on OFF, messages with a 'debug' severity are not collected (max severity set to 6). If a message has neither severity nor facility, Universal Collector microagent automatically allocates the local use 7 facility and the debug severity to the message. It will then be automatically filtered.

Option	Description
Maximum Severity	Select the maximum accepted severity (numerical code, see RFC 3164) 0 - Emergency: system is unusable 1 - Alert: action must be taken immediately 2 - Critical: critical conditions 3 - Error: error conditions 4 - Warning: warning conditions 5 - Notice: normal but significant condition 6 - Informational: informational messages 7 - Debug: debug-level messages Default value: 6 - Informational: informational messages

Option	Description
Authorized facilities	Select one accepted facility (see RFC 3164). The logs with these facilities are kept. 0 - kernel messages 1 - user-level messages 2 - mail system 3 - system daemons 4 - security/authorization messages (note 1) 5 - messages generated internally by syslogd 6 - line printer subsystem 7 - network news subsystem 8 - UUCP subsystem 9 - clock daemon (note 2) 10 - security/authorization messages (note 1) 11 - FTP daemon 12 - NTP subsystem 13 - log audit (note 1) 14 - log alert (note 1) 15 - clock daemon (note 2) 16 - local use 0 (local0) 17 - local use 1 (local1) 18 - local use 2 (local2) 19 - local use 3 (local3) 20 - local use 4 (local4) 21 - local use 5 (local5) 22 - local use 6 (local6) 23 - local use 7 (local7) Default value: 0-23
Authorized IP addresses	Enter the regular expression to filter the accepted IP addresses and to filter the accepted host. All the logs from all IP addresses are collected if the field is blank (default).

Editing a Hawk Rulebase Log Source

Option	Description
General	
Log Source Enabled	Click toggle button **Yes** or **No** to define whether the current log source is enabled or disabled.
Name	Name of the Log Source.
Description	Description of the Log Source.
Collection	
Select Rulebases	Select one or more rulebases from dropdown list. Data sources configured in hawk rulebase are used for collecting data. If no rulebase is selected then all available rulebases are used for collecting data.
Forwarders	
Select forwarder	Select the Forwarding connection to which you want to forward collected rulebase metrics.
Advanced	
Host name	Enter the name of the host used to pair logs on the LogLogic LMI server. For example, customHostname.com If you enter an IPv4 address, the device to be displayed in LogLogic LMI will be referred with this IP address.
Application name	Enter the name of the application used to identify logs on the LogLogic LMI server. For example, customApplicationName

Editing a Hawk Metrics Log Source

Option	Description
General	
Log Source Enabled	Click toggle button **Yes** or **No** to define whether the current Log Source is enabled or disabled
Name	Name of the Log Source
Description	Description of the Log Source
Collection	
Add New Combination	You can specify the multiple Microagents and methods to collect the metrics data.You can also specify the columns that you want to collect from each Microagent method. When specifying multiple Microagents and methods, the following points must be considered: • Multiple microagent method combinations can be specified only if all the methods return the CompositeData. • Only one Microagent method can be specified if that method returns the TabularData. • If all the methods in the combination are Synchronized methods, then all the methods will be subscribed to at given interval. Since the methods are synchronized, their data will be regularly available at given interval. This data from multiple methods will be combined and sent to LogLogic LMI. • If some methods in the combination are asynchronized and some are synchronized, then synchronized methods will return the data regularly at given interval. But asynchronized methods may not return data regularly. So in this case, whenever data is available from all the methods, it will be combined and sent to LogLogic LMI.
Select microagent	Select microagent from microagent dropdown list.
Select method	Select method of the microagent from dropdown list to collect the metrics data
Select metrics	Select the data returned by the method which you want to collect, from metrics dropdown list

Editing a Command Line Log Source

You can modify the following fields of the Command line log source.

Option	Description
General	
Log Source Enabled	Click toggle button **Yes** or **No** to define whether the current Log Source is enabled or disabled.
Name	Name of the Log Source.
Description	Description of the Log Source.
Forwarders	
Select Forwarder	Select the Forwarding connection to which you want to forward collected RT File logs from dropdown list.
Universal Collector Collection date	Define whether the log message sent to the LogLogic LMI server remains in a local system time zone or is converted into UTC time zone.
Collection	
Command	Enter the command line script path. If the script path or argument contains empty spaces, it must be entered in double quotation marks. On Windows, if the script path and argument contain empty spaces, you must enter the command as shown below: "D:\folder name\Hello World.py" "hello world" (double quotation marks for the whole command) or D:\"folder name"\"Hello World.py" "hello world"
Multiline messages	Click Yes or No to activate or deactivate the option to define whether the single message has several lines.
(If Multiline messages is ON) Multiline timeout after detected header	Indicate the number of seconds after which the multi-line logs are ready to be sent.

Option	Description
Advanced	
Host name	Enter the name of the host used to pair logs on the LogLogic LMI server. For example, customHostname.com If you enter an IPv4 address, the device to be displayed in LogLogic LMI will be referred with this IP address.
Application name	Enter the name of the application used to identify logs on the LogLogic LMI server. For example, customApplicationName
Maximum message length	Indicate the possible maximum length for the message (in bytes). The maximum supported value is 1048576. Default value: 64000
Run once	Click Yes or No to activate or deactivate the option to define whether the script must be run once or multiple times.
Schedule	Select the collection period, either per minute, hour, daily, or weekly at a specific hour.

Editing a Remote File Log Source

You can modify the following fields of the Remote file log source:

Option	Description
General	
Log Source Enabled	Click toggle button **Yes** or **No** to define whether the current Log Source is enabled or disabled.
Name	Name of the Log Source.
Description	Description of the Log Source.
Forwarders	
Select Forwarder	Select the Forwarding connection to which you want to forward collected RT File logs from dropdown list.

Option	Description
Universal Collector Collection date	Define whether the log message sent to the LogLogic LMI server remains in a local system time zone or is converted into UTC time zone.
Collection	
Host IP/Name	Enter the IP or name of the remote log source.
Protocol	Define whether the Log Source uses the FTP, SFTP, CIFS or file protocol. On Windows, Remote file collection by using file protocol is unavailable on network shared and Network File System (NFS) mounted drives.
(If ftp is selected) Server Time Zone	Select the time zone of the remote log source.
User ID	Enter the User ID to connect to the remote log source.
User password	Enter the user password.
File Path	Specify the path of the log file to be collected. If the log file is rotated, you may enter [id] or [date] or both in the file name and configure the File rotation parameters. For example, `c:\temp\logFile[date].log` to obtain file names such as `logFile20170521.log`
File rotation	Click Yes or No to activate or deactivate the option.
(If File rotation is active) Date pattern	Select check box and enter the date format you want to use for the [date] parameter. For example, yyyyMMdd for 20170421.
(If File rotation is active) Max number of digits	Select the check box and enter the maximum number of digits you want for the [id] parameter. Universal Collector microagent can collect any file with an [id] whose number of digits is between 1 and 9 inclusive. For example, If you set 5, the following [id] will be taken into account: 1, 054, 586, 00599, 78945, etc.

Option	Description
File change notification	Click Yes or No to activate or deactivate the option. This option allows you to monitor file changes. If set Yes, a notification will be sent to LogLogic LMI through the uc.log file when the modified date of the specified file changes. The notification includes the changed content and time. A new log is recorded for the notification when Universal Collector microagent internal logs are forwarded to LogLogic LMI. The file changes are not monitored for rotated files. In this case, the File change notification option is disabled. The specified file size must be less than the default size (10MB). If the file size is more than 10MB, the notification does not include changed content. Before activating this monitoring option, ensure that you set the LMI **Connection > Forwarding > Forward UC Internal Logs** option is ON.
Multiline messages	Click Yes or No to activate or deactivate the option to define whether the single message has several lines.
(If Multiline messages is ON) Multiline Header Type	Select the type of multi-line logs. For example, 'jboss', 'tomcat', 'weblogic', 'websphere' or 'custom'.
(If Multiline messages is ON) Custom Header regex	Set a regular expression matching the header of the first line of a log.
(If Multiline messages is ON) Custom Separator	Specify a custom delimiter to use as a separator for multiple lines. The default separator is \r\n. If the field is empty, a space is added in the message.
(If Multiline messages is ON) Send orphaned lines	Indicate whether you want Universal Collector microagent to send messages that do not match the Header Regexp.
(If Multiline messages is ON) Multiline timeout after detected header	Indicate the number of seconds after which the multi-line logs are ready to be sent.

Option	Description
(If Directory is selected) Directory path	If Directory is selected, enter the directory pathname. Ensure that you use the forward slash (/) and not the backward slash in the path.
(If Directory is selected) File(s) Include	Enter the files that must be included in the collection. The field supports the standard common wildcard characters for matching file names (* and ?).
(If Directory is selected) File(s) Exclude	Enter the files that must be excluded from the collection. The field supports the standard common wildcard characters for matching file names (* and ?).
Device type	Select the type of logs to be collected.
Test connection	Click this button to check if the connection to the remote log source is working.
Advanced	
Log Source IP	Select an option: Log Source IP- Remote file server: selected by default. The IP is grabbed from the host IP that you previously entered. This option is not available when the file protocol is selected. UC: IP address of the workstation where Hawk agent is running. You can change it as you want. The IP address will be set as the host IP address when the file protocol is selected.
Delete inactive file	Click Yes or No to activate or deactivate the option. You can purge files that are older than certain time based on the modified time.
[If Delete inactive file is selected] Inactive Days	Enter the number of days after which the inactive file is deleted. The default is set to 7 days.
Schedule	Select the collection period, either per minute, hour, daily or weekly at a specific hour.

Forwarding Logs

Universal Collector microagent collects the information from various types of log sources and forwards them to an LogLogic LMI server or syslog server.

The logs are forwarded to a LogLogic LMI server or syslog server. For the LogLogic LMI server the proprietary ULDP protocol is used and for Syslog server UDP or TCP protocols are used. Universal Collector microagent sends a maximum of 1 MB log message per line using TCP (Syslog) forwarder.

Ensure that the jumbo message support is enabled for TIBCO LogLogic Log Management Intelligence.

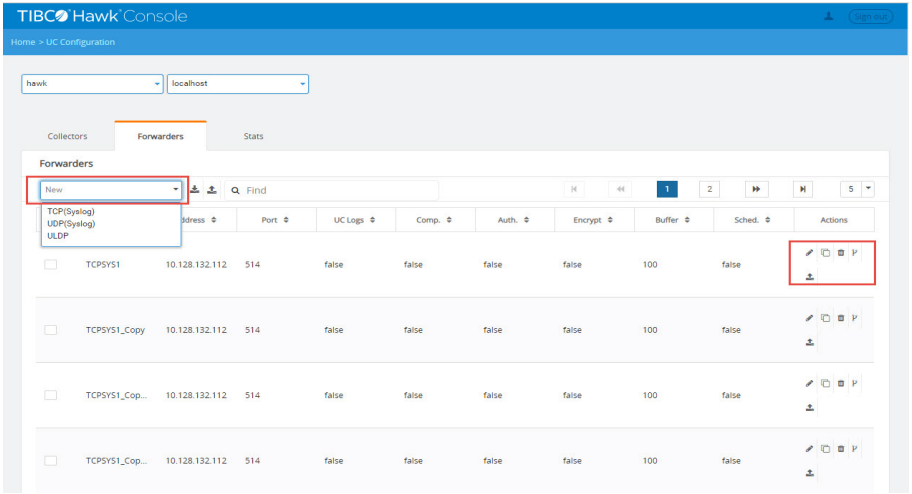
The message length more than 64KB is only supported for LMI 6.2.0 and later versions. For more details, see *TIBCO LogLogic Log Management Intelligence Administration Guide*.

A file is identified by a file identifier usually a string representing the path name of the file in the source device.

Forwarding Logs to LogLogic LMI

For the LogLogic LMI server, the proprietary ULDP protocol is used.You can configure the LogLogic LMI forwarding connection using the following steps:

1. In the Hawk Console, click the **UC Configuration** tab in the upper-right corner.
2. Select **Domain** and **Agent** from the dropdown list.
3. Click **Forwarders** tab.
4. From the **New** dropdown list, select ULDP.



You can modify the following fields in the LogLogic LMI forwarding connection:

Option	Description
LMI Connection Name	Name of the LMI Connection
Security	**NOTE:** If you have Java 1.8.0 update 201 and higher version installed, you must enable both the authentication and encryption settings for Secured ULDP forwarder.
Authentication	Activates the authenticated communication when the button is set as Yes
Encryption	Activates the encrypted communication when the button is set as **Yes**
Initialize secured connection	Click to select secure connection method from supported formats: PEM, PKCS12, JKS **NOTE:** The security certificate must be available at the Hawk Agent. You must specify the path of the security certificate in the following fields.
PEM	
PEM Certificate File	Specify the path of Security Certificate in `*.pem` format.
PEM Private Key File	Specify the path of Import Private Key file in `.pem` format.

Option	Description
Password	Enter the private key password.
Root CA Certificate File	Specify the path of the root CA certificate stored in *.pem format.
PKCS12	
PKCS12 Certificate File	Specify the path of the UC PKCS#12 Certificate in `*.p12` format.
Password	Enter the certificate password
Root CA Certificate File	Specify the path of the root CA certificate stored in *.p12 format.
JKS	
JKS File	Specify the path of the UC JKS Certificate in `*.jks` format.
Password	Enter the certificate password
Message Buffer	
Buffer Size	Enter the buffer size in megabytes. (Default: 100 MB)
Forwarding	
Address	Enter the IPv4 address or host name of LogLogic LMI.
Port	Select the LogLogic LMI port or enter a port. - 5515 for secured connection with LogLogic LMI (configurable in LogLogic LMI) - 5516 for unsecured connection with LogLogic LMI
Test Connection	Test the connection between Universal Collector microagent and LogLogic LMI.
Forward UC Internal Logs	Define whether the Universal Collector microagent internal logs are sent to the remote LogLogic LMI by selecting Yes.

Option	Description
Compress Messages	If the connection is slow, you can configure the logs to be compressed for a more rapid flow of data. Define whether the logs are compressed by selecting Yes.
Advanced	
Reconnection	Enter the reconnection frequency to LogLogic LMI (in seconds)
Session timeout	Enter the session timeout to LogLogic LMI (in seconds)
UC Binding interface	If there are multiple network interfaces, enter the IP address that Universal Collector microagent uses when establishing the connection to LogLogic LMI. Default: 0.0.0.0

Creating a TCP or UDP Syslog Connection

For Syslog server, UDP or TCP protocols are used. You can configure the TCP or UDP syslog connection using the following steps:

1. In the Hawk Console, click the **UC Configuration** tab in the upper-right corner.
2. Select **Domain** and **Agent** from the dropdown list.
3. Click **Forwarders** tab.
4. From the **New** dropdown list, select **TCP(Syslog)** or **UDP(Syslog)**.

You can modify the following fields in the TCP or UDP syslog forwarding connection:

Option	Description
General	
TCP(Syslog) or UDP(Syslog) Connection Name	Name of the TCP(Syslog) or UDP(Syslog) connection
Security	
Authentication	Activates the authenticated communication when the button is set as **Yes**
Encryption	Activates the encrypted communication when the button is set as **Yes**

Option	Description
Initialize secured connection	Click to select secure connection method from supported formats: PEM, PKCS12, JKS **NOTE:** The security certificate must be available at the Hawk Agent. You must specify the path of the security certificate in the following fields.
PEM	
PEM Certificate File	Specify the path of the security certificate in `*.pem` format.
PEM Private Key File	Specify the path of the Private Key file in `.pem` format.
Password	Enter the private key password.
Root CA Certificate File	Specify the path of the root CA certificate stored in *.pem format.
PKCS12	
PKCS12 Certificate File	Specify the path of the UC PKCS#12 certificate in `*.p12` format.
JKS	
JKS File	Specify the path of the UC JKS certificate in `*.jks` format.
Password	Enter the certificate password
Message Buffer	
Buffer Size	Enter the buffer size in megabytes. (Default: 100 MB)
Forwarding	
Address	Enter the IPv4 address or host name of the syslog server.
Port	Enter a port number. (Default: 514)
[TCP Only] Test Connection	Test the connection between Universal Collector microagent and the server.
Message Format	

Option	Description
Facility	Select the facility to be applied to the log: 0 - kernel messages 1 - user-level messages 2 - mail system 3 - system daemons 4 - security/authorization messages (note 1) 5 - messages generated internally by syslog 6 - line printer subsystem 7 - network news subsystem 8 - UUCP subsystem 9 - clock daemon (note 2) 10 - security/authorization messages (note 1) 11 - FTP daemon 12 - NTP subsystem 13 - log audit (note 1) 14 - log alert (note 1) 15 - clock daemon (note 2) 16 - local use 0 (local0) 17 - local use 1 (local1) 18 - local use 2 (local2) 19 - local use 3 (local3) 20 - local use 4 (local4) 21 - local use 5 (local5) 22 - local use 6 (local6) 23 - local use 7 (local7)

Option	Description
Severity	Select the severity to be applied to the log: 0 - Emergency: system is unusable 1 - Alert: action must be taken immediately. 2 - Critical: critical conditions. 3- Error: error conditions. 4 - Warning: warning conditions. 5 - Notice: normal but significant condition. 6 - Informational: informational messages. 7 - Debug: debug-level messages.
Custom Header	Indicate the header of the message.
Advanced	
[TCP only] Session timeout	Enter the session timeout (in seconds)
UC Binding interface	If there are multiple network interfaces, enter the IP address that the Universal Collector microagent uses when establishing the connection. Default: 0.0.0.0

Copying Forwarder

To copy a forwarder, click **Copy** icon in the Actions column of the Forwarder.

The copied Forwarders are displayed below the list of forwarders with the name *Forwarder_name*_copy.

Deleting Forwarder

To delete a Forwarder, click Delete icon in the Actions column of the Forwarder.

Before deleting the Forwarder, ensure that the log sources linked are removed or disabled .

Deploying Forwarder

You can deploy Forwarder to one or more hawk agents.

Procedure

1. Open the Universal Collector microagent Console by clicking the **UC Configuration** tab in the upper-right corner.

2. Click Deploy to icon in the Actions column of the Forwarder you want to deploy.

3. In Deploy Forwarder to window, select Hawk agents to which you want to deploy the Forwarder and click **Yes**.

Importing Forwarders

Procedure

1. Open the Universal Collector microagent Console by clicking the **UC Configuration** tab in the upper-right corner.

2. On the Forwarders tab, click import icon beside New dropdown list.

- To import single Forwarder, select the earlier exported Forwarder (`.xml`) file.
- To import multiple Forwarders at the same time, select the .zip file which contains multiple Forwarder (`.xml`) files.

3. Click **Open**.

Result

If the import is successful, the imported Forwarder is listed on the **Forwarders** tab.

Exporting Forwarders

Procedure

1. Open the Universal Collector microagent Console by clicking the **UC Configuration** tab in the upper-right corner.

2. On the **Forwarders** tab, export the Forwarder by using either of the following ways:

- From the Forwarders list, under the Actions column, select the Export option for the Forwarder that you want to export.
- To export multiple Forwarders, select the check boxes next to Forwarders which you want to export and click the export icon in the top beside new dropdown list. The .zip is downloaded to the Downloads folder of your machine which contains exported Forwarders (.xml) files.

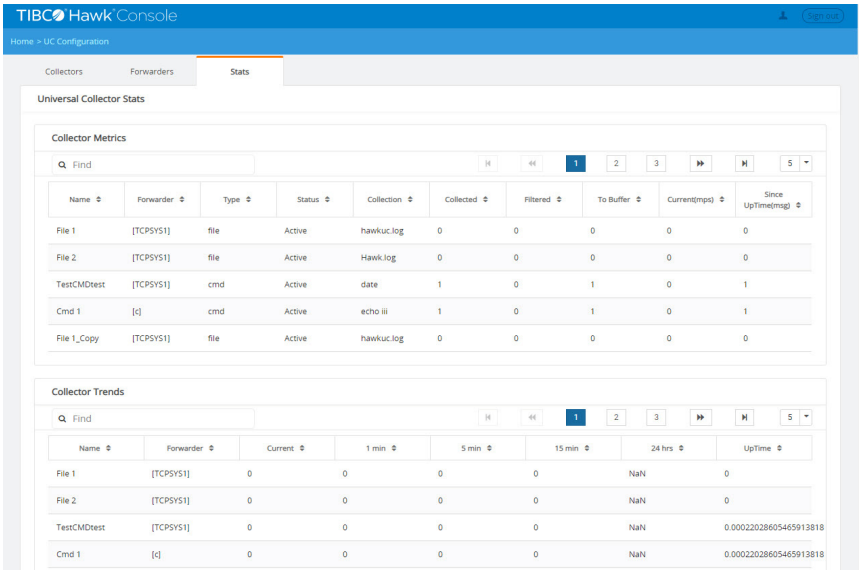
3. In the Save As window, browse to the location where you want to save the .xml or .zip file and click **Save**.

Editing Forwarders

You can edit Forwarder configuration using edit icon in the **Actions** column of Forwarders list.

Monitoring Universal Collector Microagent Activities

You can view log source metrics, trends, forwarders and log sources summary from the **Stats** tab.



Collector Metrics

Column	Description
Name	Name of the Log Source
Forwarder	Define the current Forwarding connection with the Log Source
Type	Type of the Log Source: Real Time File, Remote File, Syslog, Command output, Hawk Rulebase, Hawk Metrics

Column	Description
Status	Status of the Log Source: Active: the connection is OK Err: the connection encountered an error Idle: the connection never received a message from the source or nothing at all for 24 hours Inactive: a Log Source is inactive
Collection	Connection parameters • Syslog: protocol/bound port • RT File: File name (no path) • Remote: File path • Cmd: Command • Hawk Rulebase (Rulebase name) • Hawk Metrics
Collected	Total number of collected message for a given period of time
Filtered	Total number of filtered message for a given period of time
To Buffer	Total number of forwarded message for a given period of time
Current (mps)	Current Log Rate
Since UpTime (msg)	Total number of collected messages since Universal Collector microagent is started

Collector Trends

Column	Description
Name	Name of the log source
Forwarder	Name of the forwarding connection
Current, 1 min, 5 min, 15 min, 24h, since uptime	Log rate over different time periods. • n/a: value not available

Forwarders & Log Sources Summary

Column	Description
All Forwarding Connections	Forwarding connection status: • OK: the forwarding connection works correctly • Error: there is an error on the forwarding connection • Total: total number of enabled forwarding connections
All Log Sources (Syslog, Real-Time File, Remote File, Command Output, Hawk Rulebase, Hawk Metrics)	Log Sources status: • OK: the log sources are answering correctly • Error: there is an error on the log source • Total: total number of log sources