TIBCO® Operational Intelligence Hawk® RedTail

Installation, Configuration, and Administration Enterprise Edition

Version 7.2.0
August 2022

Contents

TIBCO® Operational Intelligence Hawk® RedTail Overview

TIBCO® Operational Intelligence Hawk® RedTail is a sophisticated hybrid monitoring and management application for distributed applications and systems deployed on both enterprise and container-based PaaS platforms by using rulebases and microagents. It allows for remediation actions when specific conditions are detected. TIBCO® OI Hawk® RedTail also enables multidomain monitoring and deeper analysis of metrics with centralized time-series metrics storage, advanced visualization dashboards, and preconfigured Content Packs. TIBCO OI Hawk RedTail allows persona-driven role-based access control to advanced features. TIBCO OI Hawk RedTail can be integrated with TIBCO LogLogic® Log Management Intelligence for forwarding log data and running remote searches on it.

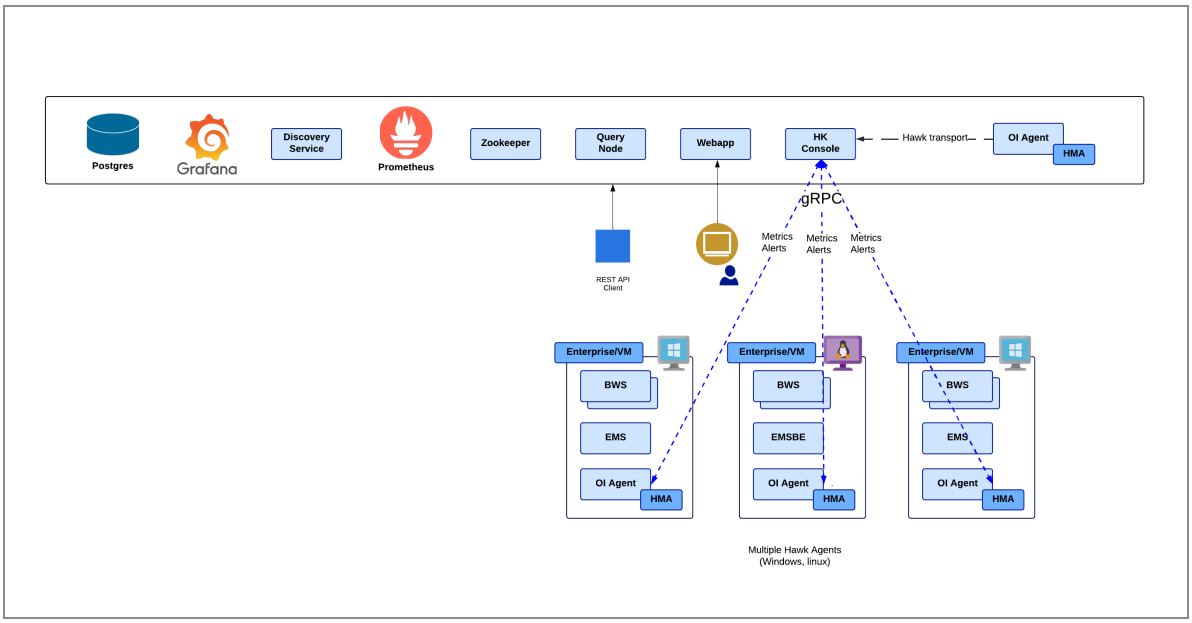
TIBCO OI Hawk RedTail also provides public APIs to develop custom components (using the REST API, Hawk AMI, and Hawk Console API) as required. For more information, see *TIBCO® Operational Intelligence Hawk® RedTail Programmer's Guide*. You can enable additional monitoring capabilities in TIBCO OI Hawk RedTail with the "Standard Edition" license of the application.

System Architecture and Components for Enterprise Deployment

This section describes the architecture for the following types of deployment of TIBCO OI Hawk RedTail:

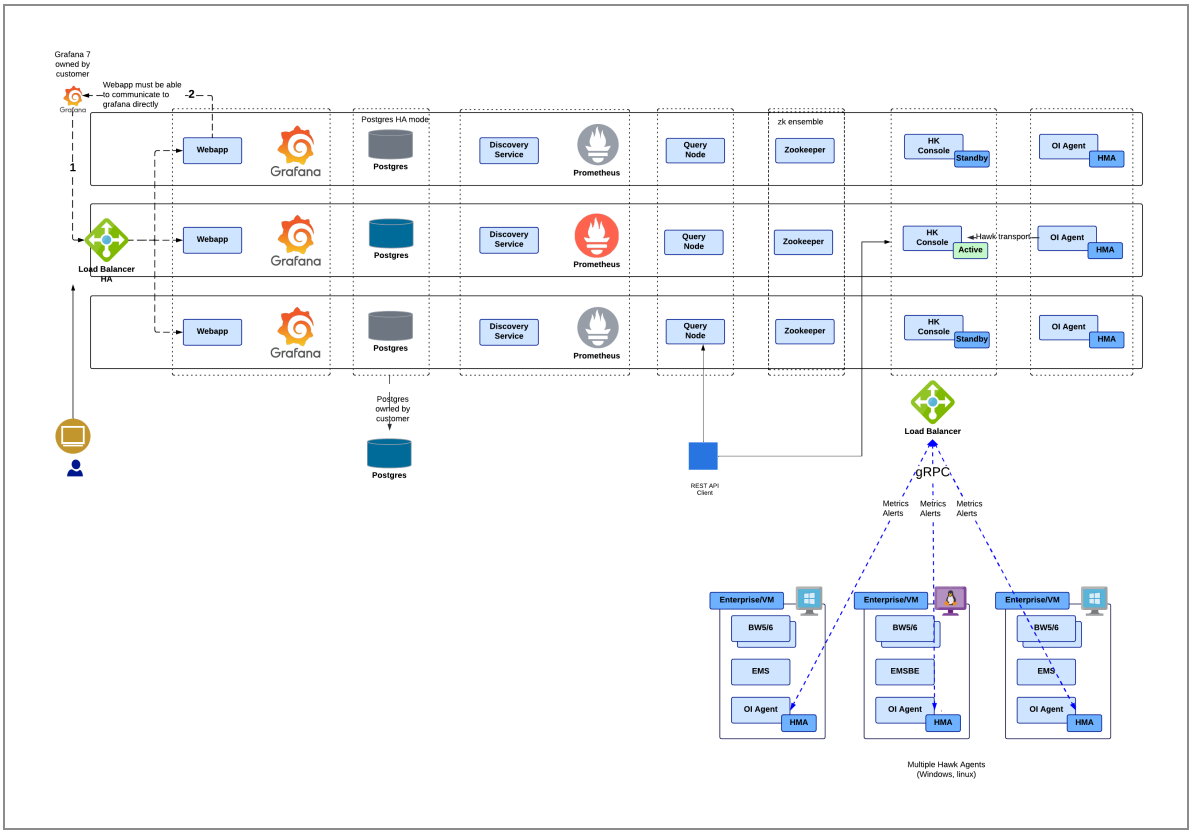
- Architecture for TIBCO OI Hawk RedTail Standalone Enterprise Deployment
- Architecture for TIBCO OI Hawk RedTail High Availability Enterprise Deployment

Architecture for TIBCO OI Hawk RedTail Standalone Enterprise Deployment



In this type of deployment, all the components of TIBCO OI Hawk RedTail are deployed on a single physical or virtual machine. There is only a single instance running of each component, hence no fallback or recovery mechanism occurs if a component stops functioning. Therefore, you must manually restart the component.

Architecture for TIBCO OI Hawk RedTail High Availability Enterprise Deployment



In this type of deployment, you must install all the components of TIBCO OI Hawk RedTail on three separate machines and then configure those components to communicate with each other through ZooKeeper to achieve high availability.

To enable HA, perform the following steps:

- Install TIBCO OI Hawk RedTail on three separate physical or virtual machines.
- Configure ZooKeeper ensemble and other services to communicate with each other.
- Configure the `zookeeper.connectString` parameter in every component's configuration file.

Make note of the following conditions in case of high availability deployment:

- Only one instance of Hawk RedTail Console is in active state and the other two instances are in standby state.

- Only one instance of Prometheus server and Postgres server is in use in the HA deployment and the components deployed on the other machines must be configured to use this server.
- The query is handled for its entire lifecycle by the same instance of the querynode on which it was created.

TIBCO OI Hawk RedTail Components

- Hawk Agent
- Hawk Microagent
- Hawk RedTail Console
- Grafana
- Time Series Storage (Prometheus)
- Prometheus Service Discovery Service
- Apache ZooKeeper
- Query Node
- Webapp

These components run as separate services and can be configured as required. You can configure these components using the configuration files. For more information, see Configuration of TIBCO OI Hawk RedTail Enterprise Components.

Note: Among the aforementioned components, the Hawk agent and the Hawk microagent are the only components that you can install on Microsoft Windows and the other remaining components can only be deployed on Linux.

Hawk Agent

The Hawk Agent is a process that monitors activity on a particular application by using microagents.

In TIBCO OI Hawk RedTail, the Hawk Agent has built-in microagents to monitor the enterprise infrastructure. The Hawk Agent uses rulebases to automate the monitoring using

rules, alerts and actions. The Hawk Agent connects to the Hawk RedTail Console by using the gRPC transport for Hawk.

Hawk Microagent

TIBCO OI Hawk RedTail has built-in microagents for monitoring enterprise infrastructure and you can also configure other microagents to monitor TIBCO and third party applications and services for example, TIBCO BusinessWorks™ Container Edition, TIBCO FTL, TIBCO ActiveMatrix, etc. For more information, refer to *TIBCO® Operational Intelligence Hawk® RedTail Microagent Reference*. Hawk microagents connect to the Hawk agent using the gRPC Transport for Hawk.

Hawk RedTail Console

You can use the REST API to access the TIBCO OI Hawk RedTail features like Hawk microagent methods, alerts, tag based rulebases, content packs, and query. The Hawk RedTail Console and the Query Node expose the other TIBCO OI Hawk RedTail components and external clients/scripts. The Hawk RedTail Console exposes administration and functional APIs. You can access the Hawk RedTail Console REST API by navigating to the Swagger page `https://<redtail_console_IP>:<rtc_port>/hawkconsole/v1/docs`.

Grafana

The Grafana component enables you to create customized dashboards. You can create and maintain multiple dashboards at once and also customize the panels within the dashboards in which multiple queries can be configured.

With the **Grafana RedTail Datasource** Plug-in, each dashboard panel can be visualized as graphical representations such as line charts, tables, and gauges. This is possible by using the **Grafana RedTail Datasource Plug-in**. This is the default plug-in that acts as a translator between Grafana and TIBCO OI Hawk RedTail. The plug-in fetches the query results from TIBCO OI Hawk RedTail then converts those results into Grafana compatible information. Grafana then displays this translated information in the form of visualization specified by the user. For more information about Grafana, see https://grafana.com/docs/.

Time Series Storage (Prometheus)

A time-series database is used to store and retrieve data records that are part of a “time series,” which is a set of data points that are associated with timestamps. The data is collected from a data source over a period of time. A time-series database lets you store

large volumes of time stamped data in a format that allows fast insertion and fast retrieval to support complex analysis on that data. The collection of data is done by using metrics exporter. An exporter converts standard metrics into time series compatible metrics. The Hawk RedTail Console acts as a Prometheus Exporter, that is, the Prometheus server scrapes metrics from Hawk RedTail Console at a regular interval. The Hawk RedTail Console then generates metrics by subscribing to microagent methods of different Hawk Agents. For more information about Prometheus, see https://prometheus.io/docs/.

Prometheus Service Discovery Service

The Prometheus Service Discovery service is a helper to the Prometheus server. This component assists Prometheus to discover metrics path from Hawk RedTail Console by using file-based discovery. The service discovers new targets exposed by the Hawk RedTail Console and then updates these targets in the `/usr/local/tibco_redtail_data/prometheus_discoveryservice/hawktargets.json` file. The service keeps updating the `hawktargets.json` file. The Prometheus server uses this file to discover new targets or update the existing targets and then scrapes metrics from the targets.

In the `prometheus.yml` file, the `job_name` parameter (which has the job name as `prometheusmetrics`) uses the file-based service discovery by listening on the `/etc/hawkprometheus-discovery/hawktargets.json` file. The service continually updates the `hawktargets.json` file. The `prometheus.yml` file contains the following jobs:

- **hawkmetrics:** To pull Hawk exporter metrics
- **prometheusmetrics:** To pull Prometheus exporter metrics

This is the `prometheus.yml` file showing examples of Hawk metrics and Prometheus metrics jobs:

```
# Global config
global:
 scrape_interval: 15s # Set the scrape interval to every 15 seconds.
Default is every 1 minute.
 evaluation_interval: 15s # Evaluate rules every 15 seconds. The default
is every 1 minute.
 # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
 alertmanagers:
 - static_configs:
  - targets:
   # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global
```

```
'evaluation_interval'.
rule_files:
# - "first_rules.yml"
# - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
# The job name is added as a label
`job=<job_name>` to any timeseries scraped from this config.
- job_name: 'redtailmetrics'
 scheme: https
 honor_labels: true
 tls_config:
  insecure_skip_verify: true
  ca_file:   '/usr/local/etc/rt_certs/cacert'
  cert_file: '/usr/local/etc/rt_certs/prometheus-client-certificate'
  key_file:  '/usr/local/etc/rt_certs/prometheus-client-key'
 file_sd_configs:
  - files:
    # Output file from Prometheus-discoveryservice service
    - /usr/local/tibco_redtail_data/prometheus_
discoveryservice/hawktargets.json
```

Apache ZooKeeper

Apache ZooKeeper is a centralized service for maintaining configuration information, applying naming conventions, and providing group services. These services are used in distributed applications.

Query Node

Query node helps in the creation of a search query for collecting the data about the metrics by using the Operational Intelligence Search Query Language. The search query supports Event Query Language (EQL) and a subset of Structured Query Language (SQL). You can run queries using Data Models, which are normalized data sets automatically created for every collected metric/Hawk microagent data. You can also use filters, limited regular expressions and time range filters in the queries. The Query Node exposes APIs to execute a query and other query related operations. You can access the Query Node REST API by navigating to the Swagger page *https://<querynode>:<querynode_port>/docs*. For more information about querying, see *TIBCO® Operational Intelligence Hawk® RedTail User Guide*.

Webapp

Webapp provides a GUI for a centralized view across the distributed components within the TIBCO OI Hawk RedTail environment.

With integrated data from components such as Grafana and Prometheus, the Webapp component provides a user-friendly interface for users to interact and monitor the data of the system.

Additionally, the user is able to configure new artifacts such as metrics, use search tools to query the data, or define the access privileges for different users.

Hardware and Software Requirements for Enterprise Deployment

Hardware and Software Requirements for Linux Platform

Note: You can deploy TIBCO OI Hawk RedTail components in standalone or High Availability mode only on the Linux platform.

Hardware Requirements for Enterprise Deployment

The minimum installation requirements for TIBCO OI Hawk RedTail are a Multi-Core CPU with at least 6 cores, 16 GB RAM and 50 GB of free disk space.

Make sure you have adequate system memory and disk space before proceeding with TIBCO OI Hawk RedTail installation.

The TIBCO Universal Installer requires disk space in the temporary directory before installation, and additional space in the temporary directory for running the installer. Refer to the following table and ensure you have sufficient disk space available in the directory you want to use as the installation environment *TIBCO_HOME* directory.

Directory/Location	Disk Space Requirement
Temporary Directory before installation For example, `/tmp`	Before you start the installation, you need this space to download an installable archive file. For example, file with the name `TIB_oihr_<version>_linux_x86_64.zip` is the installable archive file. This file needs about **3 GB** of disk space.

Directory/Location	Disk Space Requirement
Temporary Directory during installation For example, */tmp/oihr<version>install*	This is the directory where you extract the installable `.zip` so that you can later execute the Universal Installer. This directory requires about **3 GB** of disk space. On Linux, the default temporary directory location is `/tmp`. If your system does not have sufficient free disk space in the default temporary directory, you can use the `is:tempdir` option when starting the installer to run the installer with a different temporary directory. For example: `TIBCOUniversalInstaller -is:tempdir \new_tmp` where `\new_tmp` has sufficient free disk space.
Installation Environment Directories	These directories are *OIHR_HOME*, *CONFIG_FOLDER*, *CONFIG_FOLDER_REDTAIL*, and *DATA_FOLDER*. Together they need at least **1 GB** of disk space.

Software Requirements for Enterprise Deployment

Operating System Requirements

You must install TIBCO OI Hawk RedTail components on one of the following Linux platforms:

- Redhat Enterprise Linux 7
- CentOS 7
- Oracle Linux 7

Refer to the following table for software requirements for a well-functioning TIBCO OI Hawk RedTail system. TIBCO OI Hawk RedTail installation includes some of the components, although they may be optional. For details about the supported versions of various third-party software, see the `Readme`.

Software	Optional?	Bundled with installer?	Description
Java Development Kit	No	Yes	You can use the JDK version bundled with the TIBCO OI Hawk RedTail installation. Alternatively, you can use your own version of JDK (new or previously installed on the same machine). In this case, you must edit the required ".tra" and ".cfg" file to reflect the accurate location.
TIBCO Rendezvous	Yes	No	If you have already installed TIBCO Rendezvous on a network-wide basis and want to use it as a transport for TIBCO Hawk, you do not need additional TIBCO Rendezvous licenses unless you are running TIBCO Rendezvous Routing Daemon (RVRD) processes on a particular machine. In that case, you need a valid RVRD license in the `tibrv.tkt` file for that machine. TIBCO Rendezvous is used for inter-process communication even if TIBCO Enterprise Message Service (EMS) is chosen as the primary transport. TIBCO Rendezvous is used for communication between the Hawk agent and the Hawk microagent (HMA) even if TCP Transport is chosen as the primary transport. **Note:** This mode of transport cannot be used for standalone or HA deployment of TIBCO OI Hawk RedTail.
TIBCO Enterprise Message	Yes	No	To use TIBCO Enterprise Message Service as the primary messaging transport, at least

Software	Optional?	Bundled with installer?	Description
Service			one EMS server must be installed on the network and you must select to install the TIBCO EMS Java client during the TIBCO EMS installation on every machine running Hawk agent and the OI Hawk Console applications. **Note:** This mode of transport cannot be used for standalone or HA deployment of TIBCO OI Hawk RedTail.
Eclipse LGPL Software Assembly	Required on Linux	No	If the Eclipse LGPL software assembly (`product_tibco_eclipse_lgpl`) is not present in the temporary directory where you extracted the product archive file, and the machine on which you plan to run the TIBCO OI Hawk RedTail installer is not connected to the Internet, download the Eclipse LGPL software assembly for your target platform before you install TIBCO OI Hawk RedTail. Save the downloaded assembly in a temporary directory accessible to the machine on which you plan to run the installer. During installation, provide the location of the temporary directory where the downloaded software assembly is available. If the machine is connected to the Internet, you can download the Eclipse LGPL software assembly file for your target platform from the eDelivery during installation. The `product_tibco_eclipse_lgpl_<version>_OSplatform.zip` is downloaded to the

Software	Optional?	Bundled with installer?	Description
			location you select.
Sunec LGPL Software Assembly	Yes	No	The software requires the Sunec LGPL library (Oracle Elliptic Curve Cryptography (ECC) Library) only if you want to use the ECC ciphers with SSL/TLS configurations. Without the ECC library, the SSL/TLS functionality is still available, but without ECC ciphers. If the Sunec LGPL software assembly (`product_tibco_sunec`) is not present in the temporary directory where you extracted the product archive file, and the machine on which you plan to run the installer is not connected to the Internet, download the Sunec LGPL software assembly for your target platform before you install TIBCO OI Hawk RedTail. Save the downloaded assembly in a temporary directory accessible to the machine on which you plan to run the installer. During installation, provide the location of the temporary directory where the downloaded software assembly is available. If the machine is connected to the Internet, you can download the Eclipse LGPL software assembly file for your target platform from the TIBCO Software Product Download Site during installation. The `product_tibco_sunec_<version>_OSplatform.zip` is downloaded to the location you select.

Hardware and Software Requirements for Microsoft Windows Platform

Note: You can only install the following components on the Microsoft Windows platform:

- Hawk Agent
- Hawk Microagent
- OI Hawk Console
- Hawk Event Service
- Hawk Admin Agent
- Hawk Cluster Manager

Hardware Requirements for Installing TIBCO OI Hawk RedTail on Microsoft Windows

TIBCO OI Hawk RedTail installation requires a minimum of 512 MB system memory (RAM) and about 3 GB of free disk space. Make sure you have adequate system memory and disk space before proceeding with TIBCO OI Hawk RedTail installation.

The TIBCO Universal Installer requires disk space in the temporary directory before installation, and additional space in the temporary directory for running the installer. Refer to the following table and ensure you have sufficient disk space available in the directory you want to use as the installation environment *TIBCO_HOME* directory.

Directory/Location	Disk Space Requirement
Temporary Directory before installation For example, `c:\temp`	Before you start the installation, you need this space to download an installable archive file. For example, file with the name `TIB_oihr_<version>_win_x86_64.zip` is the installable archive file. This file needs about **3 GB** of disk space.
Temporary Directory during	This is the directory where you extract the installable

Directory/Location	Disk Space Requirement
installation For example, *c:\temp\oihr<version>install*	`.zip` so that you can later execute the Universal Installer. This directory requires about **3 GB** of disk space. On Microsoft Windows, the default temporary directory location is `%SystemDrive%:\Documents and Settings\user_name\Local Settings\Temp` If your system does not have sufficient free disk space in the default temporary directory, you can use the `is:tempdir` option when starting the installer to run the installer with a different temporary directory. For example: `TIBCOUniversalInstaller -is:tempdir Z:\my_very_own_tmp` where `Z:\my_very_own_tmp` has sufficient free disk space.
Installation Environment Directories	These directories are *OIHR_HOME* and *CONFIG_FOLDER*. Together they need at least **1 GB** of disk space.

Software Requirements for Installing TIBCO OI Hawk RedTail on Microsoft Windows

Refer to the following table for software requirements for a well-functioning TIBCO OI Hawk RedTail system. TIBCO OI Hawk RedTail installation includes some of the components, although they may be optional. For details about the supported versions of various third-party software, see the `Readme`.

Software	Optional?	Bundled with installer?	Description
Java	No	Yes	You can use the JDK version bundled with

Software	Optional?	Bundled with installer?	Description
Development Kit			the TIBCO OI Hawk RedTail installation. Alternatively, you can use your own version of JDK (new or previously installed on the same machine). In this case, you must edit the required ".tra" and ".cfg" file to reflect the accurate location.
TIBCO Rendezvous	Yes	No	If you have already installed TIBCO Rendezvous on a network-wide basis and want to use it as a transport for TIBCO Hawk, you do not need additional TIBCO Rendezvous licenses unless you are running TIBCO Rendezvous Routing Daemon (RVRD) processes on a particular machine. In that case, you need a valid RVRD license in the `tibrv.tkt` file for that machine. TIBCO Rendezvous is used for inter-process communication even if TIBCO Enterprise Message Service (EMS) is chosen as the primary transport. TIBCO Rendezvous is used for communication between the Hawk agent and the Hawk microagent (HMA) even if TCP Transport is chosen as the primary transport.
TIBCO Enterprise Message Service	Yes	No	To use TIBCO Enterprise Message Service as the primary messaging transport, at least one EMS server must be installed on the network and you must select to install the TIBCO EMS Java client during the TIBCO EMS installation on every machine running Hawk agent and the OI Hawk Console applications.

Software	Optional?	Bundled with installer?	Description
Eclipse LGPL Software Assembly	Required on Linux	No	If the Eclipse LGPL software assembly (`product_tibco_eclipse_lgpl`) is not present in the temporary directory where you extracted the product archive file, and the machine on which you plan to run the TIBCO OI Hawk RedTail installer is not connected to the Internet, download the Eclipse LGPL software assembly for your target platform before you install TIBCO OI Hawk RedTail. Save the downloaded assembly in a temporary directory accessible to the machine on which you plan to run the installer. During installation, provide the location of the temporary directory where the downloaded software assembly is available. If the machine is connected to the Internet, you can download the Eclipse LGPL software assembly file for your target platform from the eDelivery during installation. The `product_tibco_eclipse_lgpl_<version>_OSplatform.zip` is downloaded to the location you select.
Sunec LGPL Software Assembly	Yes	No	The software requires the Sunec LGPL library (Oracle Elliptic Curve Cryptography (ECC) Library) only if you want to use the ECC ciphers with SSL/TLS configurations. Without the ECC library, the SSL/TLS functionality is still available, but without ECC ciphers. If the Sunec LGPL software assembly

Software	Optional?	Bundled with installer?	Description
			(`product_tibco_sunec`) is not present in the temporary directory where you extracted the product archive file, and the machine on which you plan to run the installer is not connected to the Internet, download the Sunec LGPL software assembly for your target platform before you install TIBCO OI Hawk RedTail. Save the downloaded assembly in a temporary directory accessible to the machine on which you plan to run the installer. During installation, provide the location of the temporary directory where the downloaded software assembly is available. If the machine is connected to the Internet, you can download the Eclipse LGPL software assembly file for your target platform from the TIBCO Software Product Download Site during installation. The `product_tibco_sunec_<version>_OSplatform.zip` is downloaded to the location you select.

Installation and Configuration Layout for Enterprise Deployment of TIBCO OI Hawk RedTail

Perform the following steps to install, configure, and deploy TIBCO OI Hawk RedTail components:

1. Run TIBCO Universal Installer and select **TIBCO OI Hawk RedTail 7.2.0 Enterprise** installation profile. For more information, see Installation by Using TIBCO Universal Installer.
2. Depending on the type of deployment (Standalone or High Availability), configure TIBCO OI Hawk RedTail components. For more information, see Configuring TIBCO OI Hawk RedTail for High Availability Deployment and Configuration of TIBCO OI Hawk RedTail Enterprise Components.
3. Start TIBCO OI Hawk RedTail components. For more information, see Starting TIBCO OI Hawk RedTail Services in an Enterprise Environment.
4. Access the WebApp URL for TIBCO OI Hawk RedTail at *https://<webapp_IP>:<webapp_port>/redtail* and configure the Grafana data source. For more information about configuring Grafana data source, see Configuring Grafana Data Source.
5. Identify and access the Hawk RedTail Console URL which is listening to the gRPC connection: *redtail_console_IP:redtail_console_port*.
6. (Optional) To monitor a different environment, run TIBCO Universal Installer and select the **TIBCO OI Hawk RedTail 7.2.0 Agent** installation profile to install the Hawk agent and the Hawk microagent components in the target environment. For more information, see Installation by Using TIBCO Universal Installer.
7. Once the Hawk agent and the Hawk microagent components are installed on the target environment, configure gRPC Transport for TIBCO Hawk and `hawk-domain` parameter in `hawkagent.cfg` and `hawkhma.cfg` files and start the Hawk agent and Hawk microagent components. For configuration of on-premise components, see Configuration of TIBCO OI Hawk RedTail Enterprise Components and for running Hawk agent and Hawk microagent in an on-premise environment, see Running

TIBCO Hawk Agent in an Enterprise Environment.

Installation by Using TIBCO Universal Installer

TIBCO Universal Installer provides different modes to install TIBCO OI Hawk RedTail on Linux and Microsoft Windows platforms.

If this is the first TIBCO software product you are installing, you must specify the installation directory where all TIBCO products are installed. This directory is referred in this documentation as *TIBCO_HOME*. On Microsoft Windows platforms, the default *TIBCO_HOME* is *C:\tibco*. On UNIX or Linux, the default *TIBCO_HOME* directory is */opt/tibco*. During installation, you must also specify the installation environment and components to be installed.

- **TIBCO Universal Installer:** You can install TIBCO OI Hawk RedTail by using TIBCO Universal Installer, which provides three installation modes. For more information, see Installation Modes and Procedures.
- ***TIBCO_HOME*:** This is the directory where TIBCO OI Hawk RedTail is installed and is referred in this documentation as *TIBCO_HOME*. For example, the default *TIBCO_HOME* path on Windows is *C:\tibco\hawk\<version>*. On UNIX or Linux, the default path is */opt/tibco/hawk/<version>*.
- ***CONFIG_FOLDER*:** This is the directory where all default configuration for TIBCO OI Hawk RedTail is stored. For example, the default configuration path on Microsoft Windows is *C:\ProgramData\TIBCO_HOME\cfgmgmt\hawk*, whereas on UNIX or Linux, it is */root/tibco/cfgmgmt/hawk*.
- ***CONFIG_FOLDER_REDTAIL*:** This is the directory where all default configuration for TIBCO OI Hawk RedTail advanced feature components is stored. For example, the default configuration path on UNIX or Linux is */usr/local/etc/tibco_redtail_conf*.
- ***DATA_FOLDER*:** This is the directory where all the TIBCO OI Hawk RedTail advanced feature components is stored. The data is categorized into different directories that is named after the related components. For example, the default data path on UNIX or Linux is */usr/local/tibco_redtail_data*.

To familiarize yourself with TIBCO Universal Installer, read the following topics before you begin installation:

- Installation Environment

- Installation Components and Profiles
- Installation Modes and Procedures

Installation Environment

An installation environment isolates product installations. A product installed in one installation environment cannot access components in other installation environments.

An installation environment is the top-level installation directory for TIBCO products. An installation environment consists of the following properties:

Property	Description
Directory	Identifies the directory where the product is installed.
TIBCO_HOME	It is the top-level installation directory for TIBCO products.
OIHR_HOME	Identifies the name of the folder where the product is installed.

Installation Components and Profiles

The components that you can install are grouped into different installation profiles. You can associate an installation profile with one or more components. When a profile is selected, the components that are default to the selected profile are selected for installation. By default, the **TIBCO OI Hawk RedTail 7.2.0 Agent** profile is selected. You can customize the installation by explicitly selecting the components that you want to install.

The following table describes the different installation profiles:

Profile	Description
(Default selection) **TIBCO OI Hawk RedTail 7.2.0**	Installs the Hawk agent and the Hawk microagent for the enterprise platform.

Profile	Description
Agent	**Note:** To install other enterprise TIBCO OI Hawk RedTail components such as OI Hawk Console and Hawk Event Service, customize installation and then manually select the components.
TIBCO OI Hawk RedTail 7.2.0 Container	Installs the TIBCO OI Hawk RedTail container components.
TIBCO OI Hawk RedTail Admin Agent	Installs the Cluster Manager and the Admin Agent component for the enterprise platform.
TIBCO OI Hawk RedTail 7.2.0 Enterprise	Installs Hawk agent, Querynode, Webapp, Hawk RedTail Console, Postgres, Prometheus, and Grafana for Linux platform.
Custom Installation	In this profile, every component can be selected for installation. To enable this, select the **Customize Installation** check box first and then select the profiles that you want in the **Installation Profile Selection** page.

Installation Modes and Procedures

You can run TIBCO Universal Installer in GUI, console, or silent mode.

Mode of installation	Description
GUI	In GUI mode, you can select a product, product location, and so on. To invoke the installer in GUI mode, double-click the executable file. For more information, see Installing in GUI Mode.
Console	In console mode, you can run the installer on a command line. This is useful if your machine does not have a GUI environment. For more information, see

Mode of installation	Description
	Installing in Console Mode.
Silent	In silent mode, the installer installs the product without prompting you for information. The installer uses either default or custom settings that are saved in a response file. For more information, see Installing in Silent Mode.

Installing in GUI Mode

This section describes the procedure to start the GUI mode for installing TIBCO OI Hawk RedTail on Windows and Linux systems.

Before you begin

1. Open the physical media or download the installation package from the TIBCO eDelivery website. To download the installation package, a user name and password are required. If you do not have a user name and password, contact TIBCO Support.
2. Extract the content of the installation package to a temporary directory and then use the extracted installation package to start the installation.
3. Stop all the running applications and services.

Procedure

1. Open the installer in one of the following ways:
    - On Microsoft Windows, double-click the following .exe file: `TIBCOUniversalInstaller-x86-64.exe`
    - On Linux, run the following command: `./TIBCOUniversalInstaller-lnx-x86-64.bin`
2. Follow the wizard instructions till you reach the Installation Profile Selection page.
3. On the Installation Profile Selection page, specify the installation path:
    - To specify an existing installation environment, select **Use an existing TIBCO_HOME** and click **Next**.

- To define a new installation environment, select **Create a new TIBCO_HOME** and click **Next**.
  For more details, see Installation Environment.

4. Select an installation profile on the **Installation Profile Selection** page to specify the components that you want to install.

   Note: To install all components, select the **Customize Installation** check box and manually select the components to be installed.

   For more details, see Installation Components and Profiles.

5. On the Java Home page, perform one of the following steps:
   - To use the JVM bundled with the installer, click **Use JVM Provided By TIBCO** and click **Next**.
   - To use your own JVM, click **Specify Currently Installed Java** and then click **Browse** to specify the path where JVM is installed and click **Next**.

   Note: When you obtain third-party software or services, it is your responsibility to ensure you understand the license terms associated with such third-party software or services and comply with such terms.

6. On the TIBCO OI Hawk RedTail Agent - GRPC Transport page, specify the Hawk Domain name and the Hawk RedTail Console gRPC URL in the `host:port` format and click **Next**.
7. On the TIBCO OI Hawk RedTail Agent - GRPC Transport page, specify AMI TCP Session in the format `Self IP Address:Port` and also specify whether you want to set the AMI RVD Session by clicking the **Set AMI RVD Session** checkbox and click **Next**.
8. On the Oracle Elliptic Curve Cryptography Library LGPL License Agreement page, accept the terms of agreement and click **Next**.
9. On the LGPL Assembly Download page, perform one of the following steps:
   - To download LGPL assembly files, select **Download Oracle Elliptic Curve Cryptography Library Assembly from TIBCO** and click **Next**.

- To use your own LGPL assembly files, specify the path where the assembly file is stored, select **Provide the location for the assembly previously downloaded from TIBCO**, click **Browse**, and then click **Next**.

10. Follow the wizard instructions to complete the installation process and exit the installer by clicking **Finish**.

What to do next

- For errors that occur during the installation process, see the installation log file at `User_Home/.TIBCO`.
  - If you want to configure TIBCO OI Hawk RedTail components for HA deployment, see Configuring TIBCO OI Hawk RedTail for High Availability Deployment.
  - To start TIBCO OI Hawk RedTail, see Starting TIBCO OI Hawk RedTail Services in an Enterprise Environment.

Installing in Console Mode

In console mode, you can run the installer from the command line.

Before you begin

1. Open the physical media or download the installation package from the TIBCO eDelivery website. To download the installation package, a user name and password are required. If you do not have a user name and password, contact TIBCO Support.
2. Extract the content of the installation package to a temporary directory and then use the extracted installation package to start the installation.
3. Stop all the running applications and services.

Procedure

1. On a command line, navigate to the temporary directory to which you extracted the installation package.
2. Run the following command to start the installation:
   - On Microsoft Windows run the following `.exe` on console mode:
     `TIBCOUniversalInstaller-x86-64.exe -console`

- On Linux run the following .bin file on console mode:
  `./TIBCOUniversalInstaller-lnx-x86-64.bin -console`

3. Follow the CLI instructions till you reach the TIBCO HOME Selection panel.
4. In the TIBCO HOME Selection panel, specify the installation path:
   - To define a new installation environment, select **Create a new TIBCO installation environment** option by entering **1** and specify the path to the new installation environment.
   - To specify an existing installation environment, select **Choose an existing environment** option by entering **2**.

   For more details, see Installation Environment.
5. Select an installation profile on the **Installation Profile Selection** panel to specify the components that you want to install.

   Note: To install all components, select the **Customize Installation** check box and manually select the components to be installed.

   Note: If you have already begun the installation by selecting one of the options, you can change the selection by customizing the install feature selections.

   For more details, see Installation Components and Profiles.
6. Enter **0** once you have selected the installation components.
7. Enter **yes** when prompted if you want to customize the install feature selections.
8. Select the features that you want to install in the TIBCO Operational Intelligence Hawk RedTail 7.2.0 - Feature Selection panel and then enter **0** after selecting the desired features.
9. In the JAVA Home Directory panel, perform one of the following steps:
   - To use the JVM bundled with the installer, select **Use JVM Provided By TIBCO** option by entering **1**.
   - To use your own JVM, select **Specify Currently Installed Java** option by entering **2** then specify the path where JVM is installed.

> **Note:** When you obtain third-party software or services, it is your responsibility to ensure you understand the license terms associated with such third-party software or services and comply with such terms.

10. In the TIBCO OI Hawk RedTail Agent - GRPC Transport panel, retain the default configuration values for Hawk Domain and Hawk RedTail Console or specify new values and click **1**.
11. In the TIBCO OI Hawk RedTail Agent - GRPC Transport panel, retain the default configuration values for AMI TCP Session or specify new values and also specify whether you want to set the AMI RVD Session, and then click **1**.
12. In the Oracle Elliptic Curve Cryptography Library LGPL License Agreement panel, accept the terms of agreement by entering **yes**.
13. In the LGPL Assembly Download panel, perform one of the following steps:
    - To download LGPL assembly files, select **Download Oracle Elliptic Curve Cryptography Library assembly from the public TIBCO download site** option by entering **1**.
    - To use your own LGPL assembly files, select **Specify the location of a previously downloaded Oracle Elliptic Curve Cryptography Library assembly** option by entering **2** and specify the path where the assembly file is stored.
14. When the installation is completed, press **Enter** to exit the installer.

What to do next

- For errors that occur during the installation process, see the installation log file at `User_Home/.TIBCO`.
  - If you want to configure TIBCO OI Hawk RedTail components for HA deployment, see Configuring TIBCO OI Hawk RedTail for High Availability Deployment.
  - To start TIBCO OI Hawk RedTail, see Starting TIBCO OI Hawk RedTail Services in an Enterprise Environment.

Installing in Silent Mode

It is good practice to install in silent mode when you decide to rapidly deploy TIBCO OI Hawk RedTail on several machines. In Silent mode, the TIBCO Universal Installer does not prompt for any inputs during installation. Instead, the inputs are read from a configuration file that is provided as a command-line parameter. If no value is specified, the installer uses the default `TIBCOUniversalInstaller_oihr_<version>.silent` file. This default file is included in the directory that contains the TIBCO Universal Installer. In silent mode, you can run the installer without user input by pointing the installer to an existing response file.

Note: TIBCO Recommends that you must only install the Hawk agent by using this mode of installation.

Before you begin

1. Open the physical media or download the installation package from the TIBCO eDelivery website. To download the installation package, a user name and password are required. If you do not have a user name and password, contact TIBCO Support.
2. Extract the content of the installation package to a temporary directory and then use the extracted installation package to start the installation.
3. Stop all the running applications and services.

Procedure

1. Open the `TIBCOUniversalInstaller_oihr_<version>.silent` file.
2. Use text editor to update the information regarding install location, `ENV_NAME`, and features in the following way:
   a. The following elements can be set to true or false in the .silent file for downloading and installing LGPL assemblies. You can also specify the path if you have already downloaded the LGPL assemblies by setting the `LGPLAssemblyDownload` key to false and specifying the path in `LGPLAssemblyPath` key.

```
<entry key="LGPLAssemblyLicenseAccepted">true</entry>
<entry key="LGPLAssemblyDownload">true</entry>
<entry
key="LGPLAssemblyPath">/opt/tibco/thirdpartyDownload</entry>
```

b. Update the install location.
For example, update the directory as follows:

```
<entry key="installationRoot">/opt/tibco</entry>
```

c. Update `ENV_NAME`
For example, update `ENV_NAME` as follows:

```
<entry key="environmentName">TIBCO-HAWK-HOME</entry>
<entry key="environmentDesc">My HAWK Installation</entry>
```

d. Update features to be installed. Set the component features that you want to install to true. For more details, see Installation Components and Profiles.

e. The following elements can be set to true or false in the `.silent` file for installing your choice of components, to simulate the TIBCO OI Hawk RedTail custom installation.

```
<entry key="feature_JRE_Hawk">true</entry>
<entry key="feature_TIBCO Hawk Documentation_hawk">true</entry>
<entry key="feature_Agent, SDK, Examples_oihr">true</entry>
<entry key="feature_Event Service_oihr">true</entry>
<entry key="feature_Console_oihr">true</entry>
<entry key="feature_Agent, Console, QueryNode, WebApp
Containers_oihr">true</entry>
<entry key="feature_Admin Agent Runtime_hawk">true</entry>
```

f. Configure the Hawk Agent as follows:

```
<!-- GRPC session address -->
<entry key="hawk.agent.grpc.session">localhost:9697</entry>
<!-- Domain -->
<entry key="hawk.agent.hawk.domain">classic</entry>
<!-- AMI settings for TCP -->
<entry key="hawk.agent.ami.tcp.port">localhost:2571</entry>
<entry key="hawk.agent.ami.rvd.session">false</entry>
```

g. Configure the `CONFIG_HOME` by configuring the following parameter:

```
<entry key="configDirectoryRoot">/home/user/tibco</entry>
```

h. Configure the JRE Directory by configuring the following parameter:

```
<entry
key="java.home.directory">/opt/tibco/tibcojre64/11</entry>
```

i. If you want to force re-installation of previously installed TIBCO OI Hawk RedTail features, uncomment and configure the following parameter:

```
<entry key="reinstallFeatures">true</entry>
```

3. Enter the following command to start the installation:

   a. On Microsoft Windows:

```
TIBCOUniversalInstaller.cmd -silent -V
responseFile="TIBCOUniversalInstaller_hawk_<version>.silent"
```

   b. On Linux:

```
./TIBCOUniversalInstaller.bin -silent -V
responseFile="TIBCOUniversalInstaller_hawk_<version>.silent"
```

What to do next

- For errors that occur during the installation process, see the installation log file at `User_Home/.TIBCO`.
  - If you want to configure TIBCO OI Hawk RedTail components for HA deployment, see Configuring TIBCO OI Hawk RedTail for High Availability Deployment.
  - To start TIBCO OI Hawk RedTail, see Starting TIBCO OI Hawk RedTail Services in an Enterprise Environment.

Post installation Tasks for Enterprise Deployment

Running install.sh Script

This script installs the advanced features of TIBCO OI Hawk RedTail such as Webapp, Querynode, hawkconsolenode, Prometheus, Prometheus discovery service, Grafana, and Postgres.

Perform the following steps to run the `install.sh` script:

Before you begin

Ensure that you have generated the certificates. For more information, see *TIBCO® Operational Intelligence Hawk® RedTail Security Guidelines Enterprise Edition*.

Procedure

1. Navigate to *OIHR_HOME/redtail/on_prem/node-bin/scripts*.
2. Open the terminal and run the following command as a root user. You can also run the command as a non-root user by prepending sudo to the command.

   ```
   sudo ./install.sh option
   ```

   Where, `option` must be one of the options listed in the following table:

Option	Description
`deploy`	Use this option if you want to install TIBCO OI Hawk RedTail components. It is recommended that you use this option for high availability deployment.
`deployAndStart`	Use this option if you want to install TIBCO OI Hawk RedTail components and to automatically start those components once successfully installed.

3. Enter information for the following parameters when prompted:

> **Note:** You can either specify custom paths or choose to retain the default paths for the following parameters.

Parameter	Description
TIBCO RedTail data path	This is the directory where all the TIBCO OI Hawk RedTail advanced feature components is stored. The data is categorized into different directories that is named after the related components. For example, the default data path on UNIX or Linux is */usr/local/tibco_redtail_data*.
TIBCO RedTail conf path	This is the directory where all default configuration for TIBCO OI Hawk RedTail advanced feature components is stored. For example, the default configuration path on UNIX or Linux is */usr/local/etc/tibco_redtail_conf*.
TIBCO RedTail certificate path	This is the directory where all certificates for TIBCO OI Hawk RedTail components is stored. For example, the default certificate path on UNIX or Linux is */usr/local/etc/rt_certs*.
TIBCO RedTail prometheus backup destination path	This is the path where Prometheus data is periodically backed up. This backup in this path is useful if the Prometheus server stops working or is unresponsive so you can configure Prometheus server on a separate machine to use this data. **Note:** This path must be accessible from all three machines in the case of HA deployment. **Note:** The Query node must be reconfigured if a new Prometheus server has been set up in the case of HA deployment.

4. (Optional) If you have ran the script with the deploy option, then you need to manually start TIBCO OI Hawk RedTail components. For more information, see Starting TIBCO OI Hawk RedTail Services in an Enterprise Environment.

External JRE

For JVM microagents: If you have specified external JRE when installing TIBCO OI Hawk RedTail, `tools.jar` in the `.hma` file must point to a JDK installation location.

If you plan to use a JRE version other than the one supplied with TIBCO Hawk, make sure that the correct values are set for `JVM_LIB_PATH`, `JVM_LIB_DIR`, `JVM_LIB_SERVER_DIR`, `JAVA_HOME`, `JRE_HOME`, `JRE_ROOT` in the `.cfg` and `.tra` files in *CONFIG_FOLDER**bin*.

Setting Permissions for Executing HMA on UNIX/Linux

TIBCO Hawk MicroAgent (HMA) process must execute under “root” privileges, on UNIX/Linux platforms.

This process internally gathers various system level information through different system artifacts such as files, folders, scripts and so on. Access failure to such system-guarded items results in incorrect results of some of the microagent methods.

If the installation is done using root user, then the installation process, accordingly creates “`setuid`” permissions with root ownership for the TIBCO HMA executable.

If the installation is done using a non-root user, then after installation is complete, the root user must change the ownership of the following files to root and set the setuid permission as follows:

```
chown root tibhawkhma
```

```
chown root starthma
```

```
chmod u+s tibhawkhma
```

```
chmow u+s starthma
```

Then, a normal user with executable permissions can execute HMA with effective "root" permissions.

Configuring TIBCO OI Hawk RedTail for High Availability Deployment

To achieve High Availability following the deployment of TIBCO OI Hawk RedTail components, perform the following procedures:

- Configure ZooKeeper Ensemble for TIBCO OI Hawk RedTail
- Configure Nginx Load Balancer for TIBCO OI Hawk RedTail
- Configure Postgres for TIBCO OI Hawk RedTail

Configure ZooKeeper Ensemble for TIBCO OI Hawk RedTail

By configuring a ZooKeeper ensemble, the ZooKeeper service on all machines can communicate with each other for service discovery, leader election, and fallback mechanism.

Example: The machines that are to be part of the HA deployment have the following IP addresses:

- 192.0.2.1
- 192.0.2.2
- 192.0.2.3

To configure ZooKeeper ensemble, perform the following steps:

Before you begin

- Install TIBCO OI Hawk RedTail on at least three separate machines.
- Specify IP addresses as the value of configuration parameters instead of `localhost` for every TIBCO OI Hawk RedTail component. For more information, see Configuration of TIBCO OI Hawk RedTail Enterprise Components.

- Stop all running TIBCO OI Hawk RedTail services. For more information, see Stopping and Restarting TIBCO OI Hawk RedTail Enterprise Components.

Note: Perform these steps on all the machines which are to be a part of High Availability deployment.

Procedure

1. Go to *DATA_FOLDER/zookeeper*.
2. Create a file named `myid` and insert 1 as the value. The value within the file is treated as the machine ID. For example, to create a file named `myid`, and enter 1 as the id for the 192.0.2.1 machine, you can run the following commands:

```
sudo echo 1 > myid
sudo chown redtail:redtail myid
```

Note: You must create the file with the name `myid`.

3. Verify the value of the file and then save the file.

Note: Each machine that is a part of HA deployment must have incremental IDs. For example, if the file present on the ZooKeeper data directory on 192.0.2.1 has its value as 1, then the value of the file present in the ZooKeeper data directory on 192.0.2.2 must be 2.

4. Modify the `zoo.cfg` file in the *CONFIG_FOLDER_REDTAIL* and append the following configuration parameters:

```
tickTime=2000
initLimit=10
syncLimit=5

server.1=192.0.2.1:2788:3788
server.2=192.0.2.2:2788:3788
server.3=192.0.2.3:2788:3788
```

5. Comment the `clientPortAddress` parameter in the `zoo.cfg` file. By commenting this, you enable ZooKeeper member on different machines to communicate with each

other.

6. Open the ports for communication within ZooKeeper servers for leader election and data synchronization and then restart the firewall. For example, if `2788` and `3788` are the ports that we want to use for communication, use the following command to open the ports for communication:

   ```
   sudo firewall-cmd --zone=public --add-port=2788/tcp --
   permanent

   sudo firewall-cmd --zone=public --add-port=3788/tcp --
   permanent

   sudo firewall-cmd --reload
   ```

   You can use the following command to verify whether the ports are actually open:

   ```
   sudo firewall-cmd --list-all
   ```

7. Start the ZooKeeper service and verify the ZooKeeper logs.
8. Configure the `zookeeper.connectString` parameter for every component in TIBCO OI Hawk RedTail.

   For TLS connection:
   `"192.0.2.1:9600,192.0.2.2:9600,192.0.2.3:9600"`

   For example, you must configure the value of `zookeeper.connectString` parameter in the `rt_querynode_vars.json` as follows:
   `"zookeeper.connectString":"192.0.2.1:9600,192.0.2.2:9600,192.0.2.3:9600"`

   For non-TLS connection:
   `"192.0.2.1:9601,192.0.2.2:9601,192.0.2.3:9601"`

   For example, you must configure the value of `zookeeper.connectString` parameter in the `rt_webapp_vars.json` as follows:
   `"ZOOKEEPER_CONNECT_STRING":"192.0.2.1:9601,192.0.2.2:9601,192.0.2.3:9601"`

   For more information, see Configuration of TIBCO OI Hawk RedTail Enterprise Components.

Configure Nginx Load Balancer for TIBCO OI Hawk RedTail

This section describes the process of configuring the Nginx load balancer for the Webapp component.

It also discusses the procedure for configuring Nginx for load balancing the communication between Hawk RedTail Console and Hawk Agents through gRPC transport.

Configure Nginx Load Balancer for the Webapp Component

The Nginx load balancer is used to manage the workload of the Webapp components. The Webapp component is a Web application (Web app) program that is delivered over the Internet through a browser interface, where the user can interact with the TIBCO OI Hawk RedTail system. In short, the load balancer is responsible for maintaining an equal workload between the instances of Webapp and automatically redirecting the user to an instance of Webapp in case one of the instances is not functioning.

Before you begin

- Install TIBCO OI Hawk RedTail on at least three separate machines.
- Install Nginx. For more information about installing Nginx, see Nginx documentation.

  Note: When you obtain third-party software or services, it is your responsibility to ensure you understand the license terms associated with such third-party software or services and comply with such terms.

- Ensure that you have configured ZooKeeper ensemble on the machines which are a part of HA deployment. For more information, see Configure ZooKeeper Ensemble for TIBCO OI Hawk RedTail.

Procedure

1. Start Webapp on the machines on which you have installed TIBCO OI Hawk RedTail.
2. Modify the contents of *nginx.conf* file as follows:

```
worker_processes 2;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;

    map_hash_bucket_size 128;

    upstream redtailwebapp {
        server <Webapp_IP_1>:<Webapp_PORT_1> weight=1;
        server <Webapp_IP_2>:<Webapp_PORT_2>;
        server <Webapp_IP_3>:<Webapp_PORT_3>;
    }

    map $request_uri $redtailwebappMap {
       # UI pages:
       "~^(/redtail/content-pack/base/remote)
(search|config)*$" '<Webapp_IP_1>:<Webapp_PORT_1>';

       # APIs:
      /redtail/v1/content-pack/base/invalidateRemoteLMI
'<Webapp_IP_1>:<Webapp_PORT_1>';
      "~^(/redtail/v1/content-pack/base/remoteLMI)
(/validate|/login)*$" '<Webapp_IP_1>:<Webapp_PORT_1>';

      # LMI requests (through proxy):
      "~^(/redtail/webapi/v(1|2))(\s|\S)*$" '<Webapp_IP_
1>:<Webapp_PORT_1>';
      # Any other request, goes to webapp as usual:
      default redtailwebapp;
    }

    server {
        listen <nginx_port>;
        server_name <nginx_IP> # you can put the IP of the
machine where the NGINX is installed.
        underscores_in_headers on;
```

```
        location /redtail {
            proxy_pass https://$redtailwebappMap;
            proxy_set_header Host $host;
            proxy_ssl_verify off;
            proxy_ssl_server_name on;
            proxy_ssl_protocols TLSv1.3;
        }

        error_page 500 502 503 504 /50x.html;

        location = /50x.html {
            root html;
        }
    }
    include servers/*;
}
```

Modify the `upstream redtailwebapp` method and specify the `server` parameters with the Webapp instances which are installed on three separate machines and needs to be load balanced.

```
upstream redtailwebapp {
server <Webapp_IP_1>:<Webapp_port_1> weight=1;
server <Webapp_IP_2>:<Webapp_port_2>;
server <Webapp_IP_3>:<Webapp_port_3>;
}
```

Modify the `server` method and specify the `server_name` and `listen` parameters with the IP address and port of the machine on which Nginx service is running.

Modify the `listen` parameter and specify a port for that parameter. The Nginx service starts on the port specified on this parameter.

Note: The `/redtail` prefix is used for load balancing Webapp instances. You must not modify this prefix.

```
server {
listen <nginx_port>;
server_name <nginx_IP>;
underscores_in_headers on;
location /redtail {
```

```
proxy_pass https://redtailwebappMap;
proxy_ssl_verify off;
proxy_ssl_server_name on;
proxy_ssl_protocols TLSv1.3;
proxy_set_header Host $host;
}
error_page 500 502 503 504 /50x.html;
location = /50x.html {
root html;
}
```

Modify the map `$request_uri $redtailwebappMap` method so that the user is automatically redirected to the Webapp which is currently authenticated with LogLogic LMI for accessing LogLogic LMI logs remotely.

> **Note:** If the Webapp instance which is communicating with LogLogic LMI stops functioning, then you must reconfigure this to another available Webapp instance's IP address and port.

For more information about configuring Nginx, see Nginx documentation.

3. Start the Nginx service after the configuration is complete. For more information about starting Nginx service, see Nginx documentation.
4. Access the Webapp by opening a browser and then enter the following address: `http://<nginx_IP>:<nginx_port>/redtail`

Configuring Nginx to allow gRPC Connection to Hawk RedTail Console

The Nginx load balancer is used to manage the workload of the Hawk RedTail Console. We are configuring Nginx to maintain communication between Hawk RedTail Console and Hawk Agents through the gRPC transport as only a single instance of Hawk RedTail Console is active at a time. The load balancer switches the agents to the active instance of Hawk RedTail Console if the existing instance of Hawk RedTail Console stops functioning.

Before you begin

- Install TIBCO OI Hawk RedTail on at least three separate machines.

- Install Nginx. For more information about installing Nginx, see Nginx documentation.

> **Note:** When you obtain third-party software or services, it is your responsibility to ensure you understand the license terms associated with such third-party software or services and comply with such terms.

- Ensure that you have configured ZooKeeper ensemble on the machines that are a part of HA deployment. For more information, see Configure ZooKeeper Ensemble for TIBCO OI Hawk RedTail.

Procedure

1. Open the *nginx.conf* file.
2. Add the Hawk RedTail Console upstream server information:

```
#HawkConsole GRPC server info
upstream grpcservers {
       #host-port_of_hawkconsole 1
       server <IP of machine-01>:9697 weight=1;

       #host-port_of_hawkconsole 2
       server <IP of machine-02>:9697;

       #host-port_of_hawkconsole 3
       server <IP of machine-03>:9697;

       }
```

   Modify the `upstream grpcservers` method with the Hawk RedTail Console IP addresses and gRPC ports.

3. Add the nginx server information for all Hawk RedTail Consoles:

```
server {
       #This will be port hawk agents will communicate with nginx
over http2
       listen          <listening-port> http2;

       location
/COM.TIBCO.hawk.console.nest.shared.grpc.HawkConsoleService     {
               grpc_pass grpc://grpcservers;
               grpc_socket_keepalive on;
               error_page 502 = /error502grpc;
```

```
                #set a value for below to properties otherwise nignx
will close
                #connection between hawkconsole and hawkagent if there
is no data
                #transfer between hawk console and hawkagent
                grpc_read_timeout 9000s;
        }

        location = /error502grpc
        {
                internal;
                default_type application/grpc;
                add_header grpc-status 14;
                add_header grpc-message "unavailable";
                return 503;
        }

}
```

Configure Postgres for TIBCO OI Hawk RedTail

You can enable TIBCO OI Hawk RedTail to connect to an external PostgreSQL server by performing the following steps:

Enabling Remote Connectivity on Postgres

You must enable remote connectivity on Postgres for HA deployment. To enable remote connectivity, perform the following:

Procedure

1. Go to *DATA_FOLDER/postgresql* and open the `pg_hba.conf` file.
2. Append the following line after "# IPv4 local connections":

   ```
   host all all <IP_address>/32 trust
   ```

   Where, `IP_address` is the IP address of the machine on which Postgres is running.

Configuring a Single Postgres Server for TIBCO OI Hawk RedTail HA Deployment

If you do not want to create a database cluster and want to use only a single Postgres server to serve all nodes for incoming requests, perform the following steps:

Procedure

1. Go to *DATA_FOLDER/postgresql* and open the `pg_hba.conf` file.
2. Append the following after "# IPv4 local connections":

```
host all all 127.0.0.1/32 trust
host all all <IP_address_1>/32 trust
host all all <IP_address_2>/32 trust
host all all <IP_address_3>/32 trust
```

   Where, `IP_address` denotes the IP addresses of the machines which are a part of HA deployment.

Configuring Streaming Replication for Postgres for HA Deployment

Streaming replication allows the updated information on the primary server to be transferred to the standby server in real time, so that the databases of the primary server and standby server can be kept in sync. Configuring streaming replication for Postgres is a four step process. The steps are:

1. Configuring the Primary Database for Replication
2. Configuring the Standby Database for Replication
3. Verifying the Replication Setup
4. Updating Configuration Parameters

Configuring the Primary Database for Replication

Procedure

1. Go to *DATA_FOLDER/postgresql* and open the `postgresql.conf` file.
2. Modify the value of `listen_addresses` parameter as `'*'`.
3. Log in to primary database and create a replication user in one of the following ways:

   Without password: `postgres=# CREATE USER hawkreplicationuser REPLICATION;`

   With password: `postgres=# CREATE USER hawkreplicationuser REPLICATION PASSWORD '${Password}';`
4. Enable remote access for the standby database on the primary database by going to *DATA_FOLDER/postgresql* and appending the following in the `pg_hba.conf` file:

   ```
   host replication hawkreplicationuser ${STANDBY IP}/32 trust
   ```
5. Restart the primary database by running the following command in the terminal:

   ```
   sudo systemctl restart rt_postgresql-13.service
   ```

Configuring the Standby Database for Replication

Procedure

1. Stop the standby database by running the following command in terminal:

   ```
   sudo systemctl stop rt_postgresql-13.service
   ```
2. Ensure that the standby data directory *DATA_FOLDER/postgresql* is empty and if not, then run the following command in terminal:

   ```
   sudo rm -rfv /postgresql/*
   ```
3. Set the existing database as replica by performing the following:

   a. Log in as a root user.

b. Open the terminal and enter the following command:

```
su postgres
```

c. Run the following command to copy the data stored in the primary server to the secondary (standby) server:

Without password: `#pg_basebackup -h ${Primary DB IP} -U hawkreplicationuser --checkpoint=fast -D /usr/local/tibco_redtail_data/postgresql -R --slot=${some hawk slotname} -C`

With password: `#pg_basebackup -h ${Primary DP IP} -U hawkreplicationuser --password --checkpoint=fast -D /usr/local/tibco_redtail_data/postgresql -R --slot=${some hawkslotname} -C`

Note: Do not use special characters for the `hawkslotname` parameter.

Note: If you have specified a custom data path, then you must replace the `/usr/local/tibco_redtail_data/postgresql` path with your custom path in the aforementioned commands.

d. The `Standby.signal` is created along with the required Postgres files in the *DATA_FOLDER/postgresql* directory and `Postgresql.auto.conf` is configured for the standby database replication. The `standby.signal` file is used to differentiate the primary and secondary (standby) server from each other and this file indicates that the server running on the same machine is a standby server.

4. Start the standby database by running the following command in terminal:

```
sudo systemctl start rt_postgresql-13.service
```

Verifying the Replication Setup

Procedure

1. Run the the following query on the primary database in psql shell:

```
postgres=# SELECT * FROM pg_stat_replication ;
```

2. Run the the following query on the standby database in psql shell:

```
postgres=# SELECT * FROM pg_stat_wal_receiver ;
```

   If the records present on the `replication` table are also present on the `wal_receiver` table, the the replication is successful.

Updating Configuration Parameters

Make the changes to the configuration parameters of the following components:

Webapp

Open the `rt_webapp_vars.json` file in a text editor and update the `DATABASE_URL` configuration parameter as following in all three machines that are a part of HA deployment:

```
"DATABASE_URL" : <protocol>://<DB username>:<DB
password>@<host1>:<port1>,<host2>:<port2>/<DB name to access>
```

Where,

- The DB user name and password are set only once, and they are provided at the beginning of the URL. The user name and password should be the same for all hosts. These parameters are optional. They can be empty.
- The hosts and ports are a set of keys and values for the different IP addresses where the databases are present.
- The DB name is the name of the database that must be accessed.

Example: `DATABASE_URL=postgres://postgres:mypassword@192.0.2.1:5432,192.0.2.2:5432,192.0.2.3:5432/logapplogu`

Hawk RedTail Console

Open the `rt_hawkconsolenode_vars.json` file in a text editor and update the `datasource_url` configuration parameter as following in all three machines that are a part of HA deployment:

```
"datasource_url" :
"jdbc:postgresql://<host1>:<port1>,<host2>:<port2>/logumon"
```

Grafana

Open the `rt_grafana_server_vars.json` file in a text editor and update the `datasource_url` configuration parameter as following in all three machines that are a part of HA deployment:

```
"GF_DATABASE_URL":
"postgres://postgres:mypassword@<host1>:<port1>,postgres:mypassword@<host2
>:<port2>/grafana"
```

Now, remove the following parameters:

- `GF_DATABASE_TYPE`
- `GF_DATABASE_HOST`
- `GF_DATABASE_NAME`
- `GF_DATABASE_USER`
- `GF_DATABASE_PASSWORD`

Switching Control to a Standby Database

Note: Once you promote a standby server database as active, the previous 'active' server cannot be changed to become active again. Use this command carefully.

Procedure

1. Log in as a root user on the machine where the standby server is to be changed as active server.
2. Run the following commands in a terminal:

```
su postgres
cd /usr/pgsql-13/bin
./pg_ctl promote -D <path_of_secondary_data_folder>
```

Configuring Grafana Data Source

In TIBCO OI Hawk RedTail, Grafana comes pre-installed with two data source plug-ins, namely `Hawk RedTail` data source and `Loglogic` data source. These data sources are responsible for fetching metrics and logs from TIBCO OI Hawk RedTail and LogLogic® Log Management Intelligence (LMI) respectively, and then transform the results as per Grafana requirement.

Configuring Hawk RedTail Data Source

TIBCO OI Hawk RedTail provides preinstalled data source named `Hawk RedTail`. This is the default plug-in which acts as a translator between Grafana and TIBCO OI Hawk RedTail. The plug-in fetches query results from TIBCO OI Hawk RedTail and then transforms those results into Grafana compatible information. Grafana then displays this translated information in the form of visualization specified by the user. To access `Hawk RedTail` data source settings, hover over the Configuration () icon, click Data Sources, and then click the `Hawk RedTail` data source.

Name	Default Value	Description
Name	Hawk RedTail	The data source name. This is how you refer to the data source in panels and queries.
HTTP		
Default	Yes	Default data source means that it is pre-selected for new panels.
URL	*https://localhost:9680*	URL that needs to be accessible from the Grafana server. **Note:** You must change the URL based on the actual IP address and port configured for Webapp component.

Name	Default Value	Description
Whitelisted Cookies	`jwtBearerToken`	Cookies by name that should be forwarded to the data source.
Auth		
Basic Auth	No	Specifies whether basic authentication to the Hawk RedTail data source is enabled.
TLS Client Auth	No	Specifies whether TLS authentication to the Hawk RedTail data source is enabled.
Skip TLS Verify	Yes	Specifies whether the certificate verification must be skipped.
Forward OAuth Identity	No	Specifies whether to forward the user's upstream OAuth identity to the data source.
With Credentials	No	Specifies whether credentials such as cookies or auth headers should be sent with cross-site requests.
With CA Cert	No	Specifies whether self-signed TLS certificates must be verified.

Configuring LogLogic Data Source

TIBCO OI Hawk RedTail provides the data source named `LogLogic`. This data source must be manually configured. When configured, this datasource is capable of fetching logs from a remote TIBCO LogLogic appliance and translating the logs to Grafana format. Grafana then displays this translated information in a tabular format. The datasource fully supports the TIBCO LogLogic EQL. To access the `LogLogic` data source settings, hover over the Configuration () icon, click Data Sources, and then click the `LogLogic` data source.

Name	Default Value	Description
Name	LogLogic	The data source name. This is how you refer to the

Name	Default Value	Description
		data source in panels and queries.
HTTP		
Default	No	Default data source means that it is pre-selected for new panels.
URL	*https://<remoteLMI>:9681*	IP address of the LogLogic server.
Whitelisted Cookies	null	Cookies by name that should be forwarded to the data source.
Auth		
Basic Auth	Yes	Specifies whether basic authentication to the LogLogic data source is enabled.
TLS Client Auth	No	Specifies whether TLS authentication to the LogLogic data source is enabled.
Skip TLS Verify	Yes	Specifies whether the certificate verification must be skipped.
Forward OAuth Identity	No	Specifies whether to forward the user's upstream OAuth identity to the data source.
With Credentials	No	Specifies whether credentials such as cookies or auth headers should be sent with cross-site requests.
With CA Cert	No	Specifies whether self-signed TLS certificates must be verified.
Basic Auth Details		
User	user name	User name for logging in to LogLogic server.

Name	Default Value	Description
Password	password	Password for logging in to LogLogic server.

Connecting an External Grafana Server to TIBCO OI Hawk RedTail

You can install the RedTail Grafana plug-in on an external Grafana server. This enables the external Grafana server to connect to TIBCO OI Hawk RedTail.

Note: The RedTail Grafana plug-in only supports the Grafana 7.x version.

Before you begin

Ensure that you have installed the Grafana server. For more information about installing the Grafana server, see Grafana documentation.

Procedure

1. To install the RedTail Grafana plug-in, copy the RedTail Grafana plug-in (dist folder) to the plug-in directory of the external Grafana server. By default, the path of the dist directory is *OIHR_HOME/on_prem/redtail-grafana-plugin/dist* and the path of Grafana plug-in directory is */usr/local/var/lib/grafana/plugins*. For more information, see https://grafana.com/docs/grafana/v7.0/administration/configuration/#fs-plugins.
2. Copy the `datasource.yaml` file to the provisioning folder. By default, the path of the `datasource.yaml` is `OIHR_HOME/on_prem/redtail-grafana-plugin/datasources_external_plugin.yaml` and the path of the provisioning folder is *provisioning/datasources*. For more information, see https://grafana.com/docs/grafana/v7.0/administration/provisioning/#data-sources.
3. Modify the `Hawk RedTail` data source and change the **server URL** as follows:
   - **HA deployment:** `http://<nginx_IP>:<nginx_port>`
   - **Standalone deployment:** `https://<webapp_IP>:<webapp_port>`
4. Reset the value of the `Authorization header` in the **Custom HTTP Headers** section and specify the type as `Bearer <JWT token>`, where the JWT token is a valid token from the Webapp component.

Starting TIBCO OI Hawk RedTail Services in an Enterprise Environment

You can start TIBCO OI Hawk RedTail services in an enterprise environment in one of the following ways:

- Using a Script to Start TIBCO OI Hawk RedTail Services
- Manually Start TIBCO OI Hawk RedTail Services

Before you begin

- Ensure that TIBCO OI Hawk RedTail components have been installed and Hawk RedTail Console is accessible.
- Ensure that a domain for the Hawk agent is created in TIBCO OI Hawk RedTail and the `hawk_domain` parameter is specified in the `hawkagent.cfg` file.
- Configure the gRPC Transport for TIBCO Hawk parameters in the `hawkagent.cfg` file.

Using a Script to Start TIBCO OI Hawk RedTail Services

Procedure

1. Go to */usr/local/bin*.
2. Open the terminal and run the following command as a root user. You can also run the command as a non-root user by prepending sudo to the command.

```
sudo ./redtail_start.sh
```

Manually Start TIBCO OI Hawk RedTail Services

Procedure

1. Open the terminal and enter the following commands to start the TIBCO OI Hawk RedTail services:

   Note: You must start the TIBCO OI Hawk RedTail services in the specified order.

   ```
   sudo systemctl start rt_zookeeper.service
   sudo systemctl start rt_postgresql-13.service
   sudo systemctl start rt_machinenode.service
   sudo systemctl start rt_hawkconsolenode.service
   sudo systemctl start rt_querynode.service
   sudo systemctl start rt_prometheus.service
   sudo systemctl start rt_prometheus_ds.service
   sudo systemctl start rt_grafana_server.service
   sudo systemctl start rt_webapp.service
   sudo systemctl start rt_prometheus_backup.service
   ```

2. Start the Hawk agent and the Hawk microagent. For more information, see Running TIBCO Hawk Agent in an Enterprise Environment.

Running TIBCO Hawk Agent in an Enterprise Environment

Note: If the Hawk RedTail Console is not deployed or is inaccessible at the time of installing the Hawk agent, then you must edit the Hawk agent configuration from the `hawkagent.cfg` file once Hawk RedTail Console is deployed. For more information about configuring `hawkagent.cfg`, see Enterprise Hawk Agent Configurations.

On Windows

Procedure

1. Open *TIBCO_HOME* and ensure that the following folders exist under *TIBCO _HOME*:
    - *TIBCO_HOME/hawk/<version>*
    - *TIBCO_HOME/tibcojre64*
2. Start the Hawk agent by using one of the following methods:
    - Click **Start** > **All Programs** > **TIBCO** > *OIHR_HOME* > **TIBCO Hawk** <*version*> > **Start Hawk Agent**.
    - Double-click `tibhawkagent` from *CONFIG_FOLDER**bin*.
3. Start the Hawk microagent by using one of the following methods:
    - Click **Start** > **All Programs** > **TIBCO** > *OIHR_HOME* > **TIBCO Hawk** <*version*> > **Start Hawk Microagent**.
    - Double-click `tibhawkhma` from *CONFIG_FOLDER**bin*.
4. Verify whether the agent is available in the specified domain along with the microagent in TIBCO OI Hawk RedTail UI.

On Linux/Unix

Procedure

1. Open *TIBCO_HOME* and ensure that the following folders exist under *TIBCO _HOME*:
    - *TIBCO_HOME/hawk/<version>*
    - *TIBCO_HOME/tibcojre64*
2. Start the Hawk agent by running the `tibhawkagent` from *CONFIG_FOLDER**bin*.
3. Start the Hawk microagent by running the `starthma`. The `starthma` must be run as root.
4. Verify whether the agent is available in the specified domain along with the microagent in TIBCO OI Hawk RedTail UI.

Performing Prometheus Disaster Recovery for High Availability Deployment

This section describes how to recover data for the Prometheus component if the currently running instance of Prometheus stops working.

Note: The time series data that is collected while you are setting up a new instance of Prometheus server is lost.

Before you begin

- The configured Prometheus service must be running on one of the machines that are a part of HA deployment.
- Ensure that the path specified for **TIBCO RedTail prometheus backup destination path** parameter while running `install.sh` script is a shared location.

Procedure

1. Mount the shared location on another machine where you want to start the Prometheus service.
2. Update the `zookeeper.connectString` parameter in the following configuration files in *CONFIG_FOLDER_REDTAIL* as we need to start a new instance of Prometheus server:
   - `rt_prometheus_vars.json`
   - `rt_prometheus_discoveryservice_vars.json`
   - `rt_prometheus_backup_vars.json`

   For more information, see Configuration of TIBCO OI Hawk RedTail Enterprise Components.
3. Restore the Prometheus data by using the `prom-snapshot-startup.sh` script. This script is used to restore the backed up data for the Prometheus instance which you plan to run as the active server. The backup data is collected from the previously active Prometheus backup service in the cluster and is stored as a `.zip` file. You can

use this script in one of the following ways:

- Run the following command without any options:

```
sudo /usr/local/bin/prom-snapshot-startup.sh
```

  This script captures the `backup_file_path` from the `rt_prometheus_backup_vars.json` file.

  > **Note:** The path specified as a value for the `backup_file_path` configuration parameter must be a shared location.

- Run the following command:

```
sudo /usr/local/bin/prom-snapshot-startup.sh <option>
```

  where, `option` is the shared file path that contains the backed up `.zip` file. Example:

```
sudo /usr/local/bin/prom-snapshot-startup.sh <zip-file-of-prometheus-backup>
```

  Running any of these commands starts the Prometheus and the Prometheus Discovery services.

4. Update the value of the `backup_file_path` configuration parameter in the `rt_prometheus_backup_vars.json` file on the machine where you have set up another instance of Prometheus server. This is because the service then starts to create a backup of the data that is generated by the Prometheus service.
5. Start the `rt_prometheus_backup` service. For example, you can run the following command to start the Prometheus backup service:

```
sudo systemctl start rt_prometheus_backup.service
```

6. Update the value of `prometheus_node_url` configuration parameter in the `rt_querynode_vars.json` file for all the machines which are a part of HA deployment.
7. Restart the `rt_querynode` service on all the machines that are a part of the HA deployment. For example, you can run the following command to restart the Query

node service:

```
sudo systemctl restart rt_querynode.service
```

Administration

Administration feature enables Role Based Access Control (RBAC) in TIBCO OI Hawk RedTail by granting and revoking privileges to content packs. The administrator can create, modify, and delete roles and users through the role management and user management page in TIBCO OI Hawk RedTail.

Administration is management of users and roles to define and control access privileges within the TIBCO OI Hawk RedTail environment. The Administration tab enables you to set up users and give them access to resources so that the operations in TIBCO OI Hawk RedTail can be conducted in a secure manner.

The **Administration** tab consists of the **Users** tab, the **Roles** tab, the **Configure Remote LDAP** tab, and **About** window for choosing a license. You can decide the columns that must be displayed or hidden associated with the user or roles by clicking the Select Columns ( ) icon on the right side of the **Users** page and **Roles** page.

Actions

The following actions are available on the Administration tab for administration:

- **Add new user** - Go to the Users tab and then click the Add ( ) icon to create a new user for the Hawk agent. For details, see Users Tab.
- **Delete a user** - Go to the Users tab, select a user and then click the delete ( ) icon to delete the user. For details, see To delete a user.
- **Duplicate a user** - Go to the Users tab, select a user and then click the duplicate ( ) icon to duplicate the details of the selected user. For details, see Duplicating a User.
- **Add new role** - Go to the Roles tab and then click the Add ( ) icon to create a new user for the Hawk agent. For details, see Roles Tab.
- **Delete a role** - Go to the Roles tab, select a user and then click the delete ( ) icon to delete the user. For details, see To delete a role.
- **Duplicate a role** - Go to the Users tab, select a user and then click the duplicate ( ) icon to duplicate the details of the selected role. For details, see Duplicating a Role.

- **Choose a license** - Go to the About window, choose the appropriate license and then click yes for changing the license in the **Change License** dialog box. For details, see Choosing a License.

For more information, see the following sections:

- Users Tab
- Roles Tab
- Deleting a User or a Role
- Configuring a Remote LDAP Server
- Choosing a License

Users Tab

A user can be an administrator, an expert programmer or standard user. Use the following procedure to create a user in TIBCO OI Hawk RedTail.

Note: You cannot add or duplicate a user if you have configured LDAP for remote authentication.

Adding a New User

1. Navigate to **Administration > Users**.
2. On the Users tab, click the Add ( ) icon.
3. In the Create User dialog box, enter the new **Name**, **Email**, and **Password** and assign a **Role** to the user.
4. Click **Create**.

Result

A new user with the specified **user name** is listed on the page.

Viewing Details about a User

You can view details about an existing user. To view the details, perform the following steps:

Procedure

1. Navigate to **Administration>Users**.
2. Click on the name of the user for which you want to view details.
3. A new panel appears on the right showing the following details about the selected user:
   - **Name:** Name of the user
   - **Email:** Email ID of the user
   - **Role:** User role

Duplicating a User

To duplicate an existing user, perform the following steps:

Procedure

1. Select the user name that you want to duplicate.
2. Click the duplicate ( ) icon.
3. In the Duplicate User window, enter the new **Password**, specify other parameters and then assign a **Role** to the user.
4. Click **Create**.

Result

A duplicated user with the specified parameters is listed on the page.

Roles Tab

You can use the administration feature to centrally manage what the users can do in TIBCO OI Hawk RedTail. For example, you can control who can log in or have access to specific

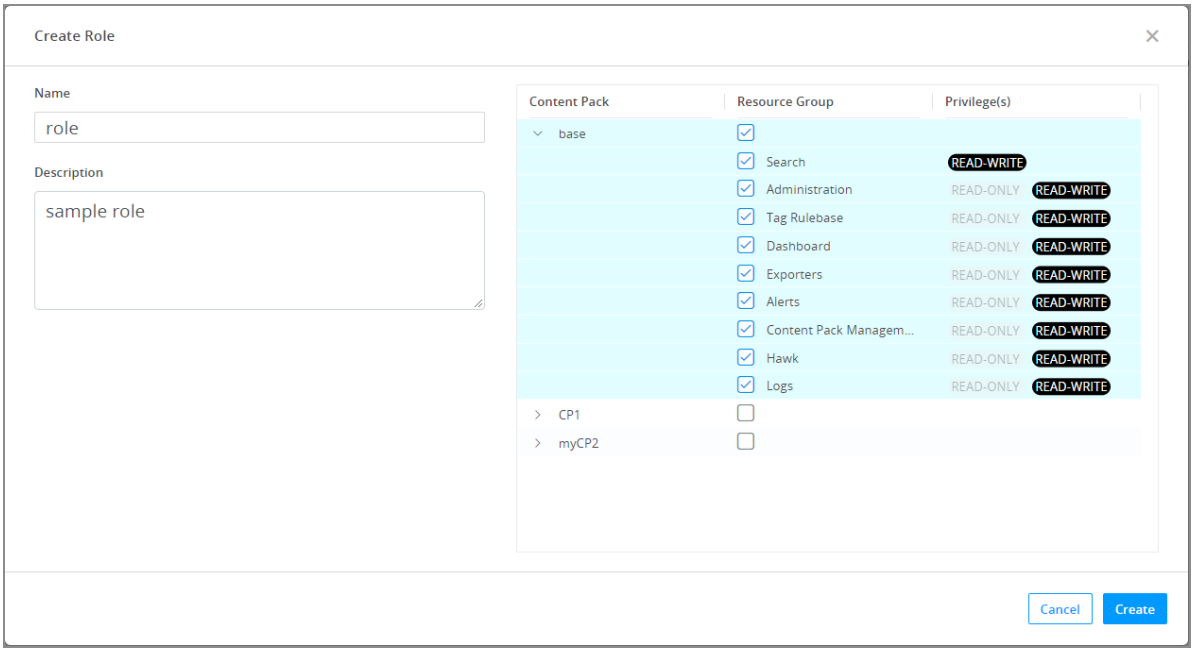
information and activities through the assignment of roles. You can only provide access and modification privileges to the features that are available with your license.

> **Note:** If you have configured LDAP for remote authentication, create a user group for the users configured via LDAP and assign a role to that user group.

For a better understanding about the available resource groups (features) for providing license based access privileges, see *TIBCO® Operational Intelligence Hawk® RedTail Concepts*.

In the following image, a role is being created by giving `READ-WRITE` privileges to every resource group of the base content pack.

Create Role Window



Adding a Role

1. Navigate to **Administration > Roles**.
2. On the Roles tab, click the Add ( ) icon.
3. In the Create Role window, enter the following information and click **Create**:
   - **Name** and **Description** for the new role.
   - **Content Pack** for which access to the resource group is to be provided.

- **Resource Group** of the selected content pack.
- **Privileges** assigned to the resource group.

Note: While creating a role, you must specify the privilege for at least a single resource group of a content pack.

Result

A new role with the specified **Role Name** is listed on the Roles page.

Viewing Details about a Role

You can view details about an existing role. To view the details, perform the following steps:

Procedure

1. Navigate to **Administration>Roles**.
2. Click on the name of the role for which you want to view details.
3. A new panel appears on the right showing the following details about the selected role:
   - **Name:** Name of the role
   - **Description:** Description for the role
   - **Content Pack:** Name of the content pack
   - **Resource Group:** Specifies the resource group belonging to the content pack
   - **Privileges:** Privileges of the users for the resource group.

Duplicating a Role

If you want to duplicate an existing role, then perform the following steps:

Procedure

1. Select the role that you want to duplicate.
2. Click the duplicate ( ) icon.
3. In the Duplicate Role window, specify the required information and click **Create**.

Result

A duplicated role with the specified parameters is listed on the Roles page.

Deleting a User or a Role

If you no longer require a user or a role in TIBCO OI Hawk RedTail, then you can delete the specified user or a role.

> **Warning:** You cannot recover a user or a role after it is deleted.

To delete a user

Procedure

1. Navigate to **Administration > Users**.
2. On the Users tab, select the user that you want to delete.
3. Click the delete (🗑) icon and confirm deletion when prompted.

To delete a role

Follow these steps to delete an existing role in TIBCO OI Hawk RedTail.

> **Note:** You cannot delete a role if you have configured LDAP for remote authentication.

Procedure

1. Navigate to **Administration > Roles**.
2. On the Roles tab, select the role that you want to delete.
3. Click the delete (🗑) icon and confirm deletion when prompted.

Configuring a Remote LDAP Server

LDAP user authentication is the process of validating a user name and password combination with a directory server. Follow these steps to configure a remote LDAP server in TIBCO OI Hawk RedTail:

Note: For using Secure LDAP, you must configure the certificate keys as environment variables in the Hawk RedTail Console.

1. Navigate to **Administration > Remote LDAP Server** and then specify the following parameters:

Parameter	Description
LDAP Server IP	IP address of the LDAP server
LDAP Server Port	Port of the LDAP server
Base DN	Base DN for the users to search
Admin DN	LDAP manager user DN for accessing the server (prevent anonymous access to the server)
Admin Password	LDAP admin password for accessing the server
SSL Enabled	Enable this button to connect to the LDAP server over SSL
Group Membership Attribute	The relationship of user with the group based on which the user role is derived

2. Click **Test Connection** to validate the entered values and then click **Apply** to establish the connection with the LDAP server with the specified values.

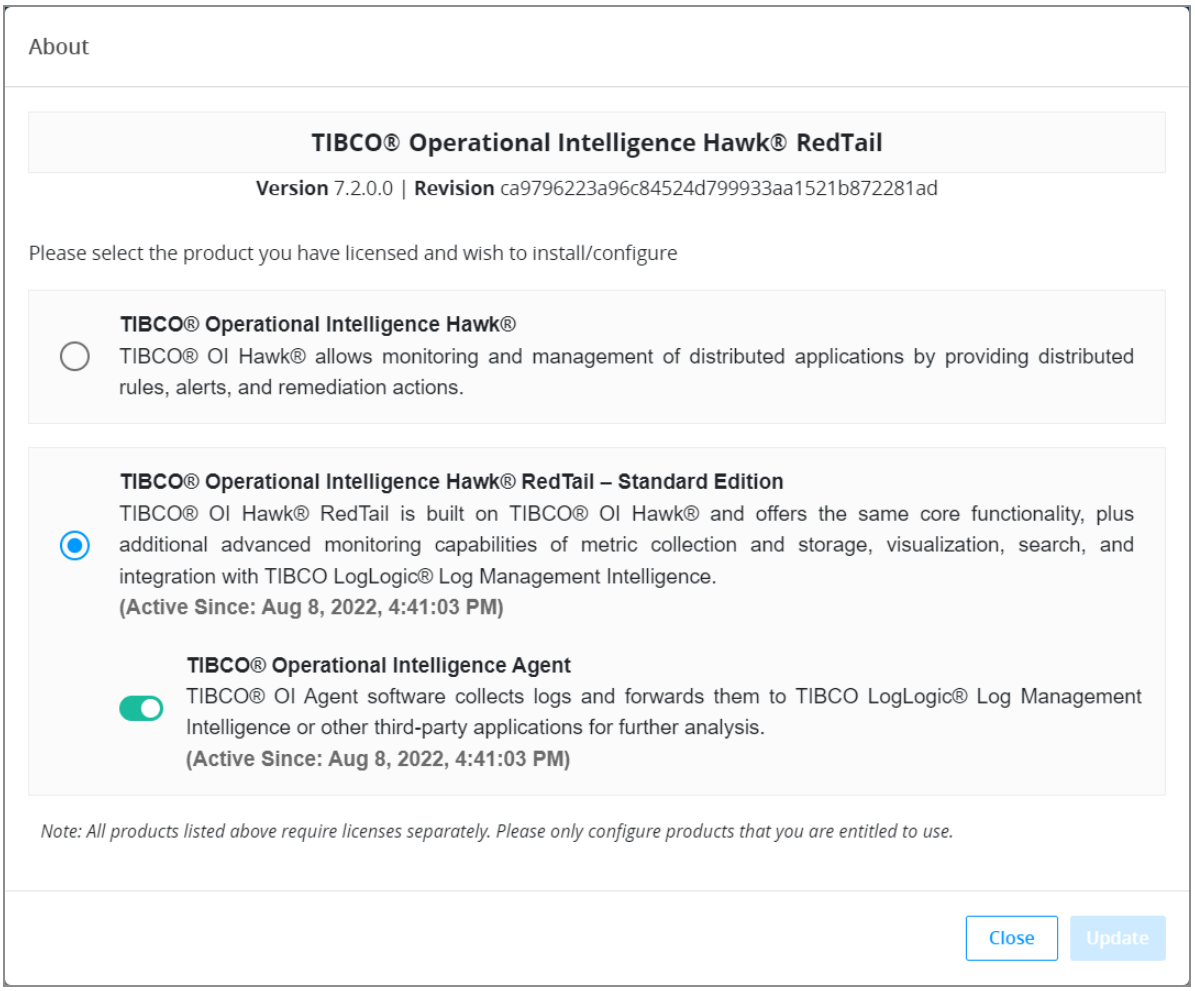
Choosing a License

Based on your requirement, you can choose the license of TIBCO OI Hawk RedTail and enable or disable additional monitoring capabilities of TIBCO OI Hawk RedTail. To choose a

license, perform the following steps:

Note: For information about the license based features, see "Overview of TIBCO® Operational Intelligence Hawk® RedTail" in *TIBCO® Operational Intelligence Hawk® RedTail User Guide*.

1. Navigate to **Administration > About**.
2. Select the product in the **About** dialog box that you are licensed for and click on the **Update** button.



3. In the **Change License** confirmation window, click **Yes**.

Note: Contact your account executive to ensure that you have the correct TIBCO OI Hawk RedTail license for your needs.

The session restarts and features based on the selected license are loaded.

Configuration of TIBCO OI Hawk RedTail Enterprise Components

Transport Mode Configuration

Different transport modes are available to be configured as a mean of communication between Hawk Agent and HMA, and Hawk Agent and OI Hawk Console, and they are:

- TIBCO Rendezvous (RV)
- TCP Transport for TIBCO Hawk
- TIBCO Enterprise Message Service (EMS)

Note: At least one transport mode must be configured in the configuration files of Hawk Agent, Hawk Microagent, Hawk Event, and OI Hawk Console to enable message or event communication among various TIBCO OI Hawk RedTail components.

TIBCO OI Hawk RedTail installation has TCP Transport for TIBCO Hawk configured as the default mode of message and event transport between Hawk Agent and AMI-based applications, as well as between Hawk Agent and OI Hawk Console.

If you choose TCP Transport for TIBCO Hawk as the transport between Hawk Agent and OI Hawk Console, you can use the same transport between Hawk agent and AMI based application except HMA. For HMA, you can use the TIBCO Rendezvous as transport and use the TCP Transport for TIBCO Hawk and TIBCO Rendezvous Bridge for communicating with Hawk Agent.

If TIBCO Rendezvous is chosen as a transport between the Hawk Agent and the OI Hawk Console, the same is used as a transport between the Hawk Agent and the HMA or other AMI based applications. You cannot configure it to use a different transport.

If you choose to use the TIBCO Enterprise Message Service as the transport between the Hawk Agent and the OI Hawk Console, you can only use TIBCO Rendezvous as the transport between the Hawk Agent and the HMA or other AMI based applications.

The above combinations can be configured using various configuration files, as described in the next few sections.

TIBCO Rendezvous and TIBCO EMS are two independent products that need to be installed separately. Additional configurations need to be performed manually based on whether they are installed before or after installing TIBCO OI Hawk RedTail, and whether any of them share the same *TIBCO_HOME* installation folder.

gRPC Transport for TIBCO Hawk

Before running TIBCO OI Hawk RedTail, you must ensure that you have configured the communication between the Hawk agent and the Hawk RedTail Console by using the gRPC transport. For more information about architecture of the gRPC Transport for TIBCO Hawk, refer to the *TIBCO® Operational Intelligence Hawk® RedTail Concepts*.

TLS Configurations for gRPC Transport

To create a secure communication channel between TIBCO OI Hawk RedTail components, you can configure gRPC transport for TIBCO Hawk to use TLS authentication.

You must configure the following TLS parameters for gRPC transport in each component of gRPC transport for secure communication:

- `grpc_enable_tls`
- `grpc_server_ca`
- `grpc_server_key`
- `grpc_server_hostname`
- `grpc_client_ca`
- `grpc_client_certificate`
- `grpc_client_key`

For details on these parameters for each component, see Configuration of TIBCO OI Hawk RedTail Enterprise Components and Enterprise Hawk Agent Configurations.

Hawk Domain

A Hawk domain is a logical grouping of TIBCO OI Hawk RedTail components. The Hawk agent, the Hawk Console API, Hawk RedTail Console and the AMI instrumented applications can all communicate with each other only if they all belong to the same Hawk domain. A Hawk domain consists of a transport and a domain name.

Some components may have additional requirements in order to communicate with the Hawk agent such as to specify the Hawk agent name to connect to.

> **Note:** The Hawk domain must be the same for all TIBCO OI Hawk RedTail components to be part of the cluster.

TIBCO Rendezvous Transport

TIBCO Rendezvous can be used as the transport between the Hawk microagent and Hawk agent and also between the Hawk agent and the OI Hawk Console.

Configure the `-rvd_session` parameter in the configuration files to enable the TIBCO Rendezvous as transport.

Comment this option, or let it be commented in the configuration file, if you are using TIBCO EMS or TCP Transport for TIBCO Hawk as the primary transport.

TIBCO OI Hawk RedTail connects to the TIBCO Rendezvous daemon by creating a session. In the configuration files, ensure that the `-tcp_session` and `-ems_transport` parameters are commented out, and then configure the `-rvd_session` parameter. TIBCO Rendezvous transport creation calls accept three parameters that govern the behavior of the transport: `service`, `network` and `daemon`.

```
-rvd_session <service><network> <daemon>
```

where,

- **service** instructs the Rendezvous daemon to use this service whenever it conveys messages on this transport. You can specify the port number as the service to be used, for example, "`7474`".
- **network** instructs the Rendezvous daemon to use a particular network for all communications involving this transport. The network parameter consists of up to three parts, separated by semicolons: network, multicast groups, and send address.
- **daemon** instructs the transport creation function about how and where to find the Rendezvous daemon and establish communication. For remote daemons, specify two parts (introducing the remote host name as the first part), for example, `tcp_host:7474`:
  - Remote host name
  - Port number

The default value in the configuration file for the Rendezvous session is

```
-rvd_session 7474 ; tcp:7474
```

For more details on TIBCO Rendezvous, refer to the TIBCO Rendezvous documentation.

TCP Transport for TIBCO Hawk

TCP Transport for TIBCO Hawk is a TCP based transport for TIBCO OI Hawk RedTail components using the Akka clustering designs.

For more information about architecture of the TCP Transport for TIBCO Hawk, refer to the *TIBCO® Operational Intelligence Hawk® RedTail Concepts*.

To setup the TCP Transport for TIBCO Hawk for TIBCO OI Hawk RedTail components, you must configure the TIBCO OI Hawk RedTail components to use the TCP session and join the cluster. The TCP session parameter in the TIBCO OI Hawk RedTail components requires a unique self socket address and address of the OI Hawk Console acting as the seed node to join the cluster.

TCP Transport for TIBCO Hawk Configuration

Hawk Component	TCP Transport Details
Hawk agent	**Configuration file:** *CONFIG_FOLDER*\bin\hawkagent.cfg **Parameters to configure:** Configuration for setting up TCP Transport for TIBCO Hawk: • Specify the Hawk domain name same as specified in the DomainTransportConfig.yml. • In case of connection to OI Hawk Console, specify the tcp_session parameter. • In case of connection to TIBCO Hawk microagent (HMA), also uncomment the -ami_rvd_session parameter (in addition to -M AMIService and -ami_tcp_session parameters) for connection using the Hawk TCP-RV Bridge for the TCP Transport for TIBCO Hawk. For Hawk microagent to use the TCP Transport for TIBCO Hawk no configuration is required in hawkhma.cfg. For more information about -tcp_session, -ami_rvd_session, and -ami_tcp_session parameters for Hawk agent, see Enterprise Hawk Agent Configurations.
OI Hawk Console	**Configuration file:** *CONFIG_FOLDER*\bin\DomainTransportConfig.yml **Parameters to configure:** You can configure TCP Transport for TIBCO Hawk by creating a domain through the OI Hawk Console Web UI and then configuring the hawkagent.cfg file for enabling the Hawk agent to

Hawk Component	TCP Transport Details
	communicate with the OI Hawk Console over TCP. You can also manually specify the domain details and `tibtcp` as transport. Specify the following parameters for the TCP Transport for TIBCO Hawk in the `DomainTransportConfig.yml` file: • `domainName` - Specify the Hawk domain name. • `transport` - Specify the value as `tibtcp` for TCP Transport for TIBCO Hawk. • `tcpSelfUrl` - Specify the socket address of the OI Hawk Console. Specify the following parameter in the `hawkconsole.cfg` file: • `domain_config_file` - Path to the file where the domain configurations are specified. For more details, see OI Hawk Console Configurations.
Hawk Admin Agent	**Configuration file:** *CONFIG_FOLDER*`\hawkteaagent\config\hawk-domain-transport-cfg.xml` **Parameters to configure:** Specify the following parameters for the TCP Transport for TIBCO Hawk: • `<hk:HawkDomainName>` - Specify the Hawk domain name same as specified in the OI Hawk Console (`hawkconsole.cfg`). • `<hk:selfUrl>` - Specify the socket address of the Hawk Admin Agent for joining the cluster. • `<hk:daemonUrl>` - Specify the socket address of the Cluster Manager acting as the seed node for the cluster. This socket address is same as `<cluster_manager_IP>:<port>` specified for the `-tcp_session` parameter in the OI Hawk Console (`hawkconsole.cfg`).
Hawk Event Service	**Configuration file:** *CONFIG_FOLDER*`\hawk\tibco\cfgmgmt\hawk\bin\hawkevent.cfg`

Hawk Component	TCP Transport Details
	Parameters to configure: Perform the following configuration for setting up TCP Transport for TIBCO Hawk for Hawk Event service: • Specify the Hawk domain name same as specified in the Hawk agent (`hawkagent.cfg`). • Uncomment the `-ami_tcp_session` parameter for connecting to the Hawk agent. For more information about `-tcp_session` and `-ami_tcp_session` parameters for Hawk Event Service, see Hawk Event Service Configurations.

SSL Configurations for TCP Transport for TIBCO Hawk

To create a secure communication channel between TIBCO OI Hawk RedTail components, you can configure TCP transport for TIBCO Hawk to use two-way SSL authentication.

You must configure the following SSL parameters for TCP Transport for TIBCO Hawk in each component of TCP Transport Cluster for the secure communication:

- `tcp_key_store`
- `tcp_trust_store`
- `tcp_key_store_password`
- `tcp_key_password`
- `tcp_trust_store_password`
- `tcp_ssl_protocol`
- `tcp_enabled_algorithms`

For details on these parameters for each component, see Configuration of TIBCO OI Hawk RedTail Enterprise Components.

TIBCO Enterprise Message Service (EMS) Transport

This section describes configuration options for connecting to TIBCO EMS server as transport for TIBCO OI Hawk components.

Comment this option if you are using TCP Transport for TIBCO Hawk or TIBCO Rendezvous as the primary transport.

The two ways to specify the TIBCO EMS transport parameters are:

1. Specify only the location of the EMS server.

   For example,

   ```
   -ems_transport tcp://server1:7222
   ```

   If communicating with the EMS server using SSL, specify the location of the EMS server as follows for the above example

   ```
   -ems_transport ssl://server1:7222
   ```

   also specify the additional options as outlined below.

2. Specify the location of the EMS server and a valid user name and password for the EMS server.

   These parameters are separated by a space and can be an empty string to indicate a null value.

   For example,

   ```
   -ems_transport tcp://server1:7222 admin "#!NhAD1NBC"
   ```

   For instructions to modify the password which was specified during installation, see Handling Passwords for TIBCO EMS Transport.

   If communicating with the EMS server using SSL, specify the location of the EMS server as follows for the above example

   ```
   -ems_transport ssl://server1:7222 admin "#!NhAD1NBC"
   ```

   and also specify the additional options as outlined in TIBCO Enterprise Message Service (EMS) Transport Using SSL.

Re-Connection Setup

To ensure the TIBCO EMS client attempts to reconnect after losing connection to the EMS server, repeat the server URL in the URL list. For example,

```
-ems_transport tcp://H1:7222,tcp://H1:7222
```

Fault Tolerance Setup

You can specify backup servers to connect to in the event of the failure of the primary server. The server URLs for the primary and backup server are specified as a comma-separated list of URLs.

For example,

```
-ems_transport tcp://server1:7222,tcp://server2:7344
```

If a connection to the first URL fails, the next URL in the list is used to attempt a reconnection. The connections in the list are attempted in sequence (wrapping to the start of the list, if the first connection was not the failed connection) until all URLs have been tried. If no connection is established after all URLs have been tried, the connection fails.

In addition to specifying the `ems_transport` options, the following parameters in the EMS server configuration file, `tibemsd.conf`, should be considered:

- `ft_active`—the name of the active server.
- `ft_reconnect_timeout`—the amount of time a backup server waits for clients to reconnect.
- `store`—the directory to store TIBCO EMS data.

For more information, see TIBCO Enterprise Message Service documentation.

TIBCO Enterprise Message Service (EMS) Transport Using SSL

Specifies the SSL parameters used by OI Hawk Console when connecting to the EMS server.

If the `ems_transport` parameter is not used, the following options are ignored.

```
-ssl_vendor <name of the vendor>
```

The name of the vendor of the SSL implementation. The valid choices are

- `j2se`—Use this option when you want to use the default Java Cryptography Extension (JCE) bundled with the Java JRE.

  On IBM platforms (such as AIX), this option defaults to `ibm`.

- `entrust61`—Use this option when you want to use the Entrust libraries.
- `ibm`—On non-IBM platforms, this option can be used only if the IBM version of JCE is installed.
- `-ssl_ciphers <suite-name>`—When specifying this option to specify the cipher suites that can be used, use the ^ qualifier instead of a - qualifier. For more information about specifying cipher suites, refer to the TIBCO Enterprise Message Service documentation.

In addition, the following sets of options are used:

For TIBCO OI Hawk components to verify the EMS server

- `-ssl_no_verify_host`—If this option is present, this indicates that the TIBCO OI Hawk component should not verify the server. Conversely, if this option is not included in the configuration file, it indicates that the TIBCO OI Hawk component should verify the server.
- `-ssl_trusted`—The option specifies the file name of the server certificates. This option can be repeated if more than one certificate file is used.
- `-ssl_no_verify_hostname`—This option specifies that the client should not verify the name in the CN field of the server certificate. Conversely, if this option is not included in the configuration file, it indicates that TIBCO OI Hawk component should verify the name in the CN field of the server certificate.
- `-ssl_expected_hostname`—The name that is expected in the name of the CN field of the server certificates is specified by this option. The value of this option is used when the -ssl_no_verify_hostname option is absent from the configuration file.

Note: If the `-ssl_no_verify_host` is not specified, the option `-ssl_trusted` has to be used. Along with the option `-ssl_trusted`, specify either `-ssl_no_verify_hostname` or `-ssl_expected_hostname`.

For the EMS server to verify TIBCO OI Hawk components

- `-ssl_identity`—This option specifies the digital certificate of the TIBCO OI Hawk components.
- `-ssl_private_key`—This option indicates the private key of the TIBCO OI Hawk component. If the key is included in the digital certificate in -ssl_identity, then you may comment out this parameter.

- `-ssl_password`—The password to decrypt the identity file of the TIBCO OI Hawk component.

Handling Passwords for TIBCO EMS Transport

On Microsoft Windows, the password is obfuscated before it is stored in the Microsoft Windows registry. In order to use the EMS password encrypt/decrypt functionality, all TIBCO OI Hawk components (including the `tibhawkpassword` wrapper) have to use JRE 1.8 or above.

If you need to change the user name and password information for the EMS server after installation, a utility is provided to encrypt your password.

Procedure

1. Invoke the command line using the syntax

   ```
   tibhawkpassword -encrypt
   ```

2. Enter the password you want to encrypt when prompted.
3. Copy and paste the output of the utility within quotes ("") into the configuration file.

   For example,

   ```
    -ems_transport tcp://emsServer:7222 username
   "#!FrHOG/QbvQMdVk4/wMv/1DA0"
   ```

4. Re-start the TIBCO OI Hawk component whose configuration file you updated in step above.

Enterprise Hawk Agent Configurations

You can configure the Hawk agent for the enterprise platforms such as Linux or Microsoft Windows. All the required configuration parameters are stored in *CONFIG_FOLDER*`/bin/hawkagent.cfg`.

Each of the parameters are explained in more detail in the following table:

Hawk Agent Configuration Options

Property	Description
`cluster`	The name of the container in which the agent appears in the display by default. The display creates the container if it does not already exist. Allows for grouping of multiple agents. The cluster name must be enclosed within quotes, if the name contains spaces. **Mandatory:** No **Suggested value:** IP subnet address
`agent_name`	Each agent being managed must have a unique combination of `agent_name`, `agent_domain`, and `hawk_domain` values. To use the host name as the agent name, comment this option. **Mandatory:** No **Suggested value:** Host Name of the computer **Note:** Agent names with multiple words separated by dots are not supported.
`agent_domain`	An agent domain must be specified when two computers within the same TIBCO OI Hawk RedTail domain have the same name but reside in different network domains. For example, you might specify this option as: `agent_domain pa.tibco.com`. **Mandatory:** No **Suggested value:** "none"
`hawk_domain`	As explained in Hawk Domain **Mandatory:** No

Property	Description
	Suggested value: classic
`transport_ timeout`	The default timeout used by transport for internal invocations **Mandatory:** No **Suggested value:** 30000
TIBCO Rendezvous Transport	
`rvd_session`	Comment this option if you are using TCP Transport for TIBCO Hawk or TIBCO EMS as the primary transport. The format is `rvd_session <service> <network> <daemon>`. If you use this option, all three parameters must be present and separated by white space. Use a semicolon (;) to indicate a null value, or use an empty string, for example: `rvd_session 7474; tcp:7474` **Mandatory:** No **Suggested value:** `7474; tcp:7474`
gRPC Transport for TIBCO Hawk (Default)	
`grpc_session`	Specifies that the TIBCO Hawk agent must use gRPC transport. The syntax of the property is: `grpc_session` *<host>:<port>* where, • *<host>:<port>* - Unique socket address for the Hawk RedTail Console component for connecting to the gRPC session.

Property	Description
	eg. `grpc_session localhost:9697` **Note:** You must specify the load balancer gRPC URL as the value for this parameter if TIBCO OI Hawk RedTail is configured in HA mode. **Mandatory:** yes **Suggested value:** `localhost:9697`
`grpc_max_reconnect_attempts`	Specifies the maximum number or reconnection attempts to be made by Hawk agent if the Hawk RedTail Console component is not available **Mandatory:** yes **Suggested value:** 100
`grpc_reconnect_interval`	Specifies the interval (in milliseconds) between reconnection attempts **Mandatory:** yes **Suggested value:** 5000
gRPC Transport for TIBCO Hawk SSL Parameters The following TLS/SSL parameters are applicable to `-grpc_session`	
`grpc_enable_tls`	Set to true, if communication needs to happen over TLS protocol **Mandatory:** No **Suggested Value:** true
`grpc_server_ca`	Path of the Certificate Authority of the Hawk RedTail Console server

Property	Description
	Mandatory: No **Suggested Value:** <Path to Certificate Authority of Hawk RedTail Console Server>
`grpc_server_hostname`	Host name of the Hawk RedTail Console **Mandatory:** No **Suggested Value:** Hawk RedTail Console host name
`grpc_client_certificate`	Path of the client certificate for the Hawk agent **Mandatory:** No **Suggested Value:** <Path to client certificate for the Hawk agent>
`grpc_client_key`	Path of the client private key for the Hawk agent in PKCS8 format **Mandatory:** No **Suggested Value:** <Path to client private key for Hawk agent in PKCS8 format>
TCP Transport for TIBCO Hawk	
`tcp_session`	Specifies that the TIBCO Hawk agent must use TCP Transport for TIBCO Hawk. The syntax of the property is: `tcp_session` *<self_IP>:<port> <HAWKCONSOLE_IP_ADDRESS>:<port>* where, • *<self_IP>:<port>* - Unique socket address of the Hawk agent for connecting to the cluster.

Property	Description
	• *<HAWKCONSOLE_IP_ADDRESS>:<port>* - The IP address of instance running OI Hawk Console. **Note:** Multiple agents/OI Hawk Console running on the same instance must be bound to separate ports. For example, if hawkagent1 binds to port 2551, then hawkagent2 can use port 2552 or any port other than 2551. **Mandatory:** Yes **Suggested Value:** `localhost:2551, localhost:2561`
TCP Transport for TIBCO Hawk SSL Parameters The following TLS/SSL parameters are applicable to `-tcp_session` and `-ami_tcp_session`.	
`tcp_key_store`	Path of the key store file **Mandatory:** No **Suggested Value:** <file-name>
`tcp_trust_store`	Path of the trust store file **Mandatory:** No **Suggested Value:** <file-name>
`tcp_key_store_password`	Password for the key store file **Mandatory:** No **Suggested Value:** <password_string>
`tcp_key_`	Encrypted key password

Property	Description
password	**Mandatory:** No **Suggested Value:** <password_string>
tcp_trust_store_password	Password for the trust store file **Mandatory:** No **Suggested Value:** <password_string>
tcp_ssl_protocol	Protocol for a secure connection **Mandatory:** No **Suggested Value:** TLSv1.2
tcp_enabled_algorithms	Algorithm to be used for the security protocol. You can specify multiple algorithms as a comma-separated list without space **Mandatory:** No **Suggested Value:** TLS_RSA_WITH_AES_128_CBC_SHA
max_reconnect_attempts_after_restart	Specifies the number of reconnect attempts to be made when the agent gets disconnected from the Daemon **Mandatory:** No **Suggested Value:** 1000
max_reconnect_attempts_during_	Specifies the number of reconnect attempts made when the connection is disconnected from the Daemon after it has been established

Property	Description
connect	**Mandatory:** No **Suggested Value:** 20
TIBCO EMS Transport	
ems_transport	Specifies that the TIBCO Hawk Agent should use TIBCO EMS transport for communication with the agents. Either one of the following format can be used: • ems_transport <serverURL> • ems_transport <serverURL> <username> <password> For example: -ems_transport tcp://server1:7222 admin "" **Note:** If EMS is configured as transport, then the ami_rvd_session parameter must be configured. **Note:** When using encrypted password generated using tibhawkpassword, the password must be placed within double quotation marks ("). Comment this option if you are using TCP Transport for TIBCO Hawk or TIBCO Rendezvous as the primary transport. **Mandatory:** No **Suggested Value:** -
TIBCO EMS SSL Parameters (In case EMS Server is configured for SSL communication)	
ssl_vendor	The name of the vendor of the SSL implementation. The valid choices are • j2se-default: Use this option when you want to use the default JCE bundled with the Java JRE. On IBM platforms (such as AIX), this option defaults to ibm.

Property	Description
	• j2se • entrust61: Use this option when you want to use the Entrust libraries. • ibm: On non-IBM platforms, this option can be used only if the IBM version of JCE is installed. **Mandatory:** No **Suggested Value:** j2se
`ssl_ciphers`	Cipher suite name. Use circumflex (^) instead of hyphen (-) when specifying `ssl_ciphers` **Mandatory:** No **Suggested Value:** -
`ssl_no_verify_host`	Indicate not to verify the EMS server **Mandatory:** No **Suggested Value:** -
`ssl_trusted`	File name of the server certificates. The file should be accessible locally/ shared drive. You can specify more than one `ssl_trusted`. **Mandatory:** No **Suggested Value:** -
`ssl_no_verify_hostname`	Indicates not to verify the name in CN field of the server certificate **Mandatory:** No

Property	Description
	Suggested Value: -
`ssl_expected_hostname`	If the -ssl_no_verify_host is not specified, the option `ssl_trusted` has to be used. Along with the option `ssl_trusted` specify either `ssl_no_verify_hostname` or `ssl_expected_hostname`. **Mandatory:** No **Suggested Value:** -
`ssl_identity`	Digital certificate **Mandatory:** No **Suggested Value:** -
`ssl_password`	Password **Mandatory:** No **Suggested Value:** -
`ssl_private_key`	Private key **Mandatory:** No **Suggested Value:** -
`use_thread_pool`	Optimizes the number of threads the agent creates for every microagent it discovers. It is advisable to turn this option **On** if the agent is going to discover over 100 microagents. This value is OS dependent and should be set to the maximum number of threads allowable per process.

Property	Description
	Mandatory: No **Suggested Value:** 256
`character_encoding`	Character encoding to be used for strings sent all TIBCO Rendezvous transport **Mandatory:** No **Suggested Value:** UTF-8
`hma_plugin_dir`	Specify the directory used for Hawk microagent plug-in configuration **Mandatory:** No **Suggested Value:** *CONFIG_FOLDER*`/plugin`
`rulebases`	List of rulebases to be loaded at the start up. This is used in manual configuration mode. This might not be used with the `auto_config_dir` option. **Note:** The rulebase names must be separated by a blank space. **Warning:** You must not specify the file extension (`.hrb`) along with the rulebase name. **Mandatory:** No **Suggested Value:** -
`config_path`	The list of directories to use as configuration sources. Used in the case of manual configuration. The delimiter for path entries is a colon (:) on Linux OS and a semicolon (;) for Microsoft Windows. This might not be used with the `auto_config_dir` option.

Property	Description
	Mandatory: No **Suggested Value:** *CONFIG_FOLDER*/config
auto_config_dir	The directory to auto-load Rulebases at the startup. If this option is present, the agent runs in an automatic configuration mode. Specify the directory from which the Rulebase and schedule configuration objects are loaded at the startup. The default directory, *CONFIG_FOLDER*/autoconfig, is used if a value is commented. If you use automatic configuration, comment the following options: config_path, repository_path, repository_cache_dir, rulebases **Mandatory:** No **Suggested Value:** *CONFIG_FOLDER*/autoconfig
repository_path	List of repositories to use as configuration sources. If repository configuration mode is used, specify the path to be searched for repositories. The delimiter for path entries is a colon (:) on Linux OS and a semicolon (;) for Microsoft Windows. This might not be used with the auto_config_dir and config_path options. **Mandatory:** No **Suggested Value:** -
repository_cache_dir	If repository configuration mode is used, all configuration objects loaded from the repository may be cached in a local directory, specified in this option. This cache is used if a repository fails, and also to minimize network traffic. If repository_cache_dir is used, comment the auto_config_dir and config_path options.

Property	Description
	Mandatory: No **Suggested Value:** *CONFIG_FOLDER*/cache
variables	Properties file to specify variables file. The variables file can pass data to define external variables to be passed to rules for use in rulebase configurations. The format of the file is that used by the standard Java class `java.util.Properties`. **Mandatory:** No **Suggested Value:** -
Email Configurations Email configuration options are used to send the notification email.	
email_smtp_server	Specifies the host name of the SMTP server **Mandatory:** No **Suggested Value:** <SMTP HostName>
email_smtp_port	Specifies the port at which the SMTP server is listening **Mandatory:** No **Suggested Value:** 25
email_smtp_auth_required	Specifies whether authentication is required for the SMTP server. The default value is `false`. If the value is `true`, you need to provide the user name (`email_smtp_user`) and password (`email_smtp_password`) for authentication.

Property	Description
	Mandatory: No **Suggested Value:** false
email_smtp_tls_required	Specifies whether TLS encryption is required for the SMTP server. If set to true, use of the STARTTLS command is required (if supported by the server) to switch the connection to a TLS-protected connection before issuing any login commands. **Note:** Some providers such as gmail have started enforcing TLS. The user must review blocked sign-in attempts and change the security policy for the email that is being used. **Mandatory:** No **Suggested Value:** false
email_smtp_tls_trust	Skips certificate validation of SMTP server. If set to "*", all hosts are trusted hosts. If set to a whitespace separated list of hosts, those hosts are trusted. Otherwise, trust depends on the certificate the server presents. **Mandatory:** No **Suggested Value:** -
email_smtp_socket_factory_port	Specifies the port to connect to when using TLS. If not set, the default port is used. **Mandatory:** No **Suggested Value:** 25
email_smtp_user	Specifies the sender's user name for the SMTP server authentication. The field is mandatory if the authentication option (email_smtp_auth_required) is set to true.

Property	Description
	Mandatory: No **Suggested Value:** -
`email_smtp_password`	Specifies the sender's password for the SMTP server authentication. The field is mandatory if the authentication option (`email_smtp_auth_required`) is set to `true`. **Mandatory:** No **Suggested Value:** -
`email_from`	Specifies the sender's email address for sending the email. The default is the current system user, for example, `"HawkAdministrator"<admin@abc.com>` **Mandatory:** No **Suggested Value:** -
`email_content_type`	Specifies the content type of email. The email application interprets the text characters in the body of the email based on this property. **Mandatory:** No **Suggested Value:** text/html
Configuration for AMI communication	
`ami_rvd_session`	Configures the agent with a RVD session to be used to communicate with applications implementing the TIBCO Hawk Application Management Interface. Multiple `ami_rvd_session` parameters may be specified. If none are specified, the RV session used for AMI is the primary session of the `Self` Module.

Property	Description
	Note: *127.0.0.1* must be used as the network parameter when you are using TIBCO EMS transport for communication. For example: *ami_rvd_session 7474 127.0.0.1 tcp:7474* If you are using TCP Transport for TIBCO Hawk then uncomment this option, so that the Hawk agent can communicate with Hawk microagent using the TCP-RV bridge. Same value must be used in the Hawk microagent configuration (`hawkhma.cfg`) for the `rvd_session` parameter. **Mandatory:** No **Suggested Value:** `tcp:7474`
`ami_tcp_session`	Configures the Hawk agent with a TCP session to be used to communicate with applications implementing the TIBCO Hawk Application Management Interface. If this parameter is not specified while using TCP Transport for TIBCO Hawk, the default value (`localhost:2571`) is used. The syntax of the property is: `-ami_tcp_session` *<self_IP>:<port>* where, *<self_IP>:<port>* is the unique socket address for communication with TIBCO Hawk Application Management Interface. **Mandatory:** No **Suggested Value:** `ami_tcp_session localhost:2571`
Logging	
`log_dir`	The directory in which to store log files generated by the Hawk agent

Property	Description
	Mandatory: No **Suggested Value:** *CONFIG_FOLDER*/logs
log_max_size	The maximum size of a rotating log file in kilobytes. You can also apply a suffix 'M' or 'm' for indicating values in megabytes. **Mandatory:** No **Suggested Value:** 10M
log_max_num	The maximum number of rotating log files **Mandatory:** No **Suggested Value:** 10
log_level	Specifies the level of diagnostic information stored in the logs. The following are the logging levels: 4 - Indicates error level trace messages should be enabled. 6- Indicates warning level trace messages should be enabled. 7 - Indicates information level trace messages should be enabled. 8 - Indicates debug level trace messages should be enabled. 16 - Indicates AMI level trace messages should be enabled. A value of zero turns all tracing off. A value of -1 turns all tracing on. **Mandatory:** No **Suggested Value:** 7
log_format	The format for trace log messages

Property	Description
	Mandatory: No **Suggested Value:** "default"
TIBCO Protocol Adapter	
`interval`	The heartbeat interval in seconds **Mandatory:** No **Suggested Value:** 30
`security_policy`	The fully qualified name of the Java class which implements the security policy **Mandatory:** No **Suggested Value:** -
Rulebase Repository	
`repository_name`	The name of the rulebase repository **Mandatory:** No **Suggested Value:** -
`repository_dir`	The location of the repository **Mandatory:** No **Suggested Value:** -

Property	Description
Logfile MicroAgent	
`scan_rate`	The interval (in seconds) after which the log files are scanned **Mandatory:** No **Suggested Value:** 10
`block_size`	The maximum number of kilobytes to read on each scan **Mandatory:** No **Suggested Value:** 16
`eval_rate`	The interval (in seconds) after which all the log files being monitored are re-evaluated **Mandatory:** No **Suggested Value:** 300

Logging for the Hawk Agent

TIBCO OI Hawk RedTail provides two different modes of logging: trace mode and log4j mode.

Trace Logging Mode

By default, the Hawk agent uses the trace mode for logging requirements. TIBCO OI Hawk RedTail uses the trace mode logging mechanism to match parity with different versions of Hawk, bundled in different TIBCO products.

The logging parameters for Hawk agent, microagent, and the Hawk Event Service are configured using the logging parameters defined in their respective configuration files. For more details on these logging parameters, refer to the following sections:

- Enterprise Hawk Agent Configurations
- Enterprise Hawk Microagent Configurations
- Hawk Event Service Configurations

Log4j Logging Mode

You can enable the log4j mode for the Hawk agent logging requirements. By default, the log4j mode is disabled for the Hawk agent. Modify the log4j configuration in the respective .tra files to enable the log4j for logging.

The configuration for Hawk agent is included in `tibhawkagent.tra` at:

```
java.extended.properties=-Duse_log4j=false -Dlog4j.configuration=file:///%HAWK_
CONFIG_HOME%/bin/log4j_agent.properties
```

To enable the log4j for logging, update the value of the `-Duse_log4j` parameter to `true` in the configuration mentioned above.

Specify "`-Dlog4j.configuration`" as command-line parameter to override the log file configuration.

A default log4j properties file exists at *CONFIG_FOLDER/bin/log4j_agent.properties*. You can modify this configuration file or use your own properties file.

The logging properties specified in `hawkagent.cfg` for –M LogService viz. log_dir, log_max_size, log_max_num and `log_level` override those in log4j RootLogger's RollingFileAppender.

Use of -log_level Parameter in Agent Configuration

If `–log_level` <*int_value*> is specified in –M `LogService` section of agent configuration and the value of `log_level` is greater than seven, the log4j root logger level is set to `DEBUG`.

Note:

- Ensure that the log4j configuration file specified above must have at least one RollingFileAppender applied to the Root logger.
- Agent logging configuration properties given in hawkagent.cfg only apply to RollingFileAppender for the Root category (Logger). If there are sub-categories with different file appenders, the agent properties do not override them.
- Log4j.xml configuration file is not supported.

Mapping of TIBCO Hawk default Util Logging Levels with Log4j:

Util Logging (in Hawk Agent)

Util Logging Level	Log4j Logging Level
Log.INFO (value 0)	INFO
Log.WARNING (value 1)	WARN
Log.DEBUG (value 2)	DEBUG
Log.ERROR (value 3)	ERROR
Log.EVENT (value 4)	INFO
Log.EXCEPTION	EXCEPTION

Trace Logging (in AMI)

Util Logging Level	Log4j Logging Level
Log.ALWAYS (value 0)	DEBUG
Log.INFO (value 1)	INFO
Log.WARNING (value 2)	WARN
Log.ERROR (value 4)	ERROR
Log.DEBUG (value 8)	DEBUG
Log.AMI (value 16)	INFO

Enterprise Hawk Microagent Configurations

You can configure the Hawk agent for the enterprise platforms such as Linux or Microsoft Windows. All the required configuration parameters are stored in *CONFIG_FOLDER*`/bin/hawkhma.cfg`.

Each of the parameters are explained in more detail in the following table:

Hawk Microagent Configuration Details

Property	Description
TIBCO HMA Common	
`hawk_domain`	Used to set the Hawk domain name. **Mandatory:** Yes **Suggested Value:** "default"
`agent_name`	Sets the name of Hawk agent. The microagent is monitored by the agent having the name specified in this parameter. **Mandatory:** Yes **Suggested Value:** Name of the agent
`agent_domain`	Sets the Agent domain name. Each agent must have a unique combination of `agent_name`, `agent_domain`, and `hawk_domain` values. **Mandatory:** Yes **Suggested Value:** "none"
TIBCO Rendezvous Transport	
`rvd_session`	Comment this option if you are using TCP Transport for TIBCO Hawk or TIBCO EMS as the primary transport. However, if you are using TCP Transport for TIBCO Hawk then uncomment this option, so that the Hawk agent can communicate with the Hawk microagent using the TCP-RV bridge. The value of this parameter must be the same as the value of the `ami_rvd_session` parameter in the Hawk agent configuration file (`hawkagent.cfg`). The format is `rvd_session <service> <network> <daemon>`.

Property	Description
	If you use this option, all three parameters must be present and separated by white space. Use a semicolon (;) to indicate a null value, or use an empty string, for example: `rvd_session 7474 "" tcp:7474` **Mandatory:** No **Suggested Value:** `7474 "" tcp:7474`
Logging	
`logdir`	The directory in which to store log files generated by the TIBCO Hawk HMA **Mandatory:** No **Suggested Value:** *CONFIG_FOLDER/logs*
`logmaxsize`	The maximum size of a rotating log files in kilobytes **Mandatory:** No **Suggested Value:** 1024
`logmaxnum`	The maximum number of rotating log files **Mandatory:** No **Suggested Value:** 5
`log_format`	The format for trace log messages **Mandatory:** No

Property	Description
	Suggested Value: "default"
Timeout	
`transport_timeout`	The timeout used by transport for internal invocations. Timeout value is in milliseconds. **Mandatory:** No **Suggested Value:** 30000
`timeout`	The method invocation timeout period to be used by all AMI methods. Timeout value is in milliseconds. **Mandatory:** No **Suggested Value:** 10000
Trace Level	
`tracelevel`	Specifies the level of diagnostic trace output. The desired trace level is specified by adding the following values together: 1 - Indicates information level trace messages should be enabled. 2 - Indicates warning level trace messages should be enabled. 4 - Indicates error level trace messages should be enabled. 8 - Indicates debug level trace messages should be enabled. 16 - Indicates AMI level trace messages should be enabled. 32 - Adds source file name and line number to all messages. A value of zero turns all tracing off A value of -1 turns all tracing on.

Property	Description
	Mandatory: No **Suggested Value:** 7
Encoding	
codepage	The desired code-page for multi-byte/Unicode character sets **Mandatory:** No **Suggested Value:** 65001

Logging for HMA

The TIBCO Hawk HMA process creates log files for each microagent, such as `Hawk_Process.log`. The HMA process also creates a `Hawk_HMA.log` file for microagent errors.

To see console logs on command console, add "`-console`" argument as one of the application arguments in the file `tibhawkhma.tra`. Otherwise, the logs get logged as Windows Events. If the logging is enabled, the logs appear in the related files.

You control the size and level of detail in HMA log files at the start using the `hawkhma.cfg` file or during runtime using the `setTraceLevel`() and `setTraceParameters`() methods. These standard methods are included for default platform-specific microagents, and can be added when instrumenting an application using the AMI protocol.

Following are some representative lines in an HMA log file for the Services microagent:

```
INFO 01/15/2013 11:14:39
OPTIONS: Transport: RV
RV Session : Service : 7474 -- Network : ; -- Daemon : tcp:7474
Timeout : 10000
CodePage : 65001
TraceLevel : -1
Logdir : C:/ProgramData/hawkv16/tibco/cfgmgmt/hawk/log --
LogMaxSize : 1024 -- Max Log Files : 5 -- Log Format : default
INFO 01/15/2013 11:14:53 TIBCO Hawk HMA initialization completed
successfully.
```

Enterprise Webapp Configurations

You can configure the Webapp for enterprise deployment. All the required configuration parameters are stored in *CONFIG_FOLDER_REDTAIL*/`rt_webapp_vars.json`

Each of the parameters is explained in more detail in the following table:

Webapp Configuration Options

Parameter	Description
`certs_conf_path`	The directory where the certificates required for communicating with other nodes is stored **Mandatory:** Yes **Suggested Value:** `/usr/local/etc/rt_certs`
`data_path`	The base directory for storing TIBCO OI Hawk RedTail services data. All the services store the data in the path categorized by service name. **Mandatory:** Yes **Suggested Value:** *DATA_FOLDER*
`zookeeper_node_url`	TLS IP of the zookeeper service which is required to connect to Zookeeper **Mandatory:** Yes **Suggested Value:** localhost
`grafana_node_url`	IP address of the Grafana service **Mandatory:** Yes **Suggested Value:** localhost

Parameter	Description
querynode_node_url	IP address of the query node service **Mandatory:** Yes **Suggested Value:** localhost
hawkconsole_node_url	hawkconsolenode URL **Mandatory:** **Suggested Value:** localhost
postgres_node_url	IP address of the database server **Mandatory:** Yes **Suggested Value:** localhost
ZK_CLIENT_KEY_FILE	Path to the ZooKeeper client private key **Mandatory:** Yes **Suggested Value:** {certs_conf_path}/key
ZK_CLIENT_KEY_PASSWORD	Password of the ZooKeeper client key **Mandatory:** Yes **Suggested Value:** <password>
ZK_CLIENT_CACERT_FILE	Path to CA certificate file used for generating the key **Mandatory:** Yes **Suggested Value:** {certs_conf_path}/cacert

Parameter	Description
ZK_CLIENT_CERT_FILE	Path to the ZooKeeper client certificate **Mandatory:** Yes **Suggested Value:** {certs_conf_path}/certificate
machineId	Internal component ID **Mandatory:** Yes **Suggested Value:** machine-0000000000
LOAD_CONFIG_FROM_ENV	The flag that indicates whether to use variables or predefined default values **Mandatory:** Yes **Suggested Value:** TRUE
unity.services.rest.host	Host IP for Webapp REST communication **Mandatory:** Yes **Suggested Value:** 0.0.0.0
unity.services.rest.port	Host port for Webapp REST communication **Mandatory:** Yes **Suggested Value:** 9680
ETAG_HEADER_ENABLED	Flag to indicate if the ETAG is enabled in our network traffic between the UI (client side) and the webapp service.

Parameter	Description
	The ETAG works as a caching layer of the network traffic between the client and the server. It refers to transmission of the response data that is sent over HTTP(S) to the client, and not to internal caching of server-to-database or a similar action. The value of this parameter is blank by default. To disable caching in the network traffic, set it to "false"." **Mandatory:** No **Suggested Value:** false
REST_TLS_CIPHERS	Supported Cipher suites **Mandatory:** Yes **Suggested Value:** AES128-GCM-SHA256:AES128-SHA256:AES256-GCM-SHA384:AES256-SHA256:ECDH-ECDSA-AES128-GCM-SHA256:ECDH-ECDSA-AES128-SHA:ECDH-ECDSA-AES128-SHA256:ECDH-ECDSA-AES256-GCM-SHA384:ECDH-ECDSA-AES256-SHA:ECDH-ECDSA-AES256-SHA384:ECDH-RSA-AES128-GCM-SHA256:ECDH-RSA-AES128-SHA:ECDH-RSA-AES128-SHA256:ECDH-RSA-AES256-GCM-SHA384:ECDH-RSA-AES256-SHA:ECDH-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-RSA-AES256-SHA384:TLS_AES_256_GCM_SHA384:TLS_AES_128_GCM_SHA256
NODE_CLIENT_AUTH_ENABLED	Flag that indicates whether a server should request a certificate from a connecting client. **Mandatory:** No

Parameter	Description
	Suggested Value: false
NODE_ENV	Flag that indicates whether the NodeJS engine (webapp) should run in production or development mode. **Mandatory:** No **Suggested Value:** production
REST_TLS_PROTOCOL	Supported TLS protocol **Mandatory:** Yes **Suggested Value:** TLSv1.3
RATE_LIMITER_BLOCK_DURATION	Used to limit the time frame (in seconds) and the number of requests performed within this time frame in the event of a DDos or brute force attack. **Mandatory:** No **Suggested Value:** 60
RATE_LIMITER_DURATION	Used to limit the maximum number of requests that can be processed in seconds in the event of a DDoS or a brute force attack. **Mandatory:** Yes **Suggested Value:** 30
RATE_LIMITER_ENABLED	Flag to indicate if Webapp service must be secured

Parameter	Description
	Mandatory: No **Suggested Value:** false
RATE_LIMITER_MAX_REQUESTS	Maximum number of requests the webapp component must accept in the timeframe delimited by the parmeter RATE_LIMITER_DURATION. **Note:** You must configure this parameter if you have enabled the RATE_LIMITER_ENABLED parameter. **Mandatory:** No **Suggested Value:** 2000
RATE_LIMITER_DURATION	Used to limit the maximum number of requests can be processed in seconds in the event of a DDoS or a brute force attack. **Note:** You must configure this parameter if you have enabled the RATE_LIMITER_ENABLED parameter. **Mandatory:** No **Suggested Value:** 30
LOG_LEVEL	Specifies the level of diagnostic information stored in the logs. The logging levels are as follows: • ERROR - Indicates that error level trace messages should be enabled. • WARNING - Indicates that warning level trace messages should be enabled. • INFO - Indicates that information level trace messages

Parameter	Description
	should be enabled. • DEBUG - Indicates that debug level trace messages should be enabled. • TRACE - Indicates that trace level messages should be enabled. **Mandatory:** Yes **Suggested Value:** info
DATABASE_URL	`<protocol>://<DB username>:<DB password>@<host1>:<port1>,<host2>:<port2>/<DB name to access>` Where, • The `DB user name` and `password` are set only once, and they are provided at the beginning of the URL. The user name and password should be the same for all hosts. These parameters are optional and can be empty. • The hosts and ports are a set of keys and values for different IP addresses where the databases are present. • The DB name is the name of the database that must be accessed. **Mandatory:** Yes **Suggested Value:** `postgres://postgres:mypassword@{postgres_node_url}:5432/logapplogu`
DATABASE_TLS_ENABLED	True, if communication with database is over TLS

Parameter	Description
	Mandatory: Yes **Suggested Value:** true
DATABASE_TLS_CACERT_FILE	Path to the database CA certificate **Mandatory:** No **Suggested Value:** {certs_conf_path}/rt_dbcacert
DATABASE_POOL_CONNECTION_LIMIT	The maximum number of connections that can be created at once **Mandatory:** Yes **Suggested Value:** 5
DATABASE_POOL_IDLE_TIMEOUT_MILLIS	Number of milliseconds a client must be idle for in the pool and is not checked out before the client is disconnected from the backend and discarded. The default value is 10000 milliseconds (10 seconds) - set to 0 to disable auto-disconnection of idle clients. **Mandatory:** Yes **Suggested Value:** 10000
REST_TLS_KEY_FILE	Key pair used for setting up REST TLS communication **Mandatory:** Yes **Suggested Value:** {certs_conf_path}/key
REST_TLS_CERT_FILE	Certificate used for REST TLS communication

Parameter	Description
	Mandatory: Yes **Suggested Value:** `{certs_conf_path}/certificate`
`REST_TLS_CACERT_FILE`	Certificate of the CA used to sign the REST TLS certificate **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/cacert`
`QUERYNODE_TLS_CACERT_FILE`	Query node CA certificate. It is required for TLS communication with the Query node. **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/cacert`
`HAWKCONSOLE_TLS_CACERT_FILE`	The `hawkconsolenode` CA certificate. It is required for TLS communication with Hawk RedTail Console. **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/cacert`
`GRAFANA_TLS_CACERT_FILE`	Grafana CA certificate. It is required for TLS communication with Grafana. **Mandatory:** No **Suggested Value:** `{certs_conf_path}/cacert`
`TLS_CLIENT_KEY_FILE`	Path for webapp client key. It is required for mutual authentication with any other component. For example, if

Parameter	Description
	Grafana is configured with a reverse proxy using TLS via mutual authentication. **Mandatory:** No **Suggested Value:** `{certs_conf_path}/webapp-client-key`
`TLS_CLIENT_CERT_FILE`	Path of Webapp client certificate **Mandatory:** No **Suggested Value:** `{certs_conf_path}/webapp-client-certificate`
`TLS_CLIENT_KEY_PASSWORD`	Password for the WebApp client key **Mandatory:** No **Suggested Value:** <password>
`REST_TLS_KEY_PASSWORD`	Password to key pair used for REST TLS communication **Mandatory:** Yes **Suggested Value:** <password>
`TLS_SKIP_CERTIFICATE_VERIFICATION`	Specifies whether the webapp must skip certificate verification while communicating with other TIBCO OI Hawk RedTail nodes **Mandatory:** Yes **Suggested Value:** false

Parameter	Description
TLS_SKIP_HOSTNAME_VERIFICATION	Specifies whether the Webapp must skip host name verification while communicating with other TIBCO OI Hawk RedTail nodes **Mandatory:** Yes **Suggested Value:** true
zookeeper.connectString	This parameter is used to set the TLS connection string of the Zookeeper. It is used by the installer when the webapp service is started. **Mandatory:** Yes **Suggested Value:** `localhost:9680`
ZOOKEEPER_CONNECT_STRING	This parameter is used to set the non-TLS connection string of the Zookeeper. It is used by the webapp service. **Mandatory:** Yes **Suggested Value:** `localhost:9681`
CONNECT_TO_ZOOKEEPER	Variable indicates when the Zookeeper connection is to be loaded in the webapp service When set to false, the webapp is not be able to communicate with other services. Therefore, this must be set to true. **Mandatory:** Yes **Suggested Value:** true
INSECURE_ZOOKEEPER_CONNECTION	Indicates to webapp that a non-TLS connection is being provided by the Zookeeper service.

Parameter	Description
	Mandatory: Yes **Suggested Value:** true

Enterprise Hawk RedTail Console Configurations

You can configure the Hawk RedTail Console for the enterprise deployment. All the required configuration parameters are stored in *CONFIG_FOLDER_REDTAIL*/`rt_hawkconsolenode_vars.json`

Each of the parameters is explained in more detail in the following table:
Hawk RedTail Console Configuration Options

Parameter	Description
`certs_conf_path`	The directory where the certificates required for communicating with other nodes is stored **Mandatory:** Yes **Suggested Value:** `/usr/local/etc/rt_certs`
`data_path`	The base directory for storing TIBCO OI Hawk RedTail services data. All the services store the data in the path categorized by service name. **Mandatory:** Yes **Suggested Value:** *DATA_FOLDER*
`conf_path`	The base directory for storing TIBCO OI Hawk RedTail

Parameter	Description
	services configuration. **Mandatory:** **Suggested Value:** *CONFIG_FOLDER_REDTAIL*
`zookeeper_node_url`	ZooKeeper URL **Mandatory:** Yes **Suggested Value:** localhost
`grafana_node_url`	Grafana URL **Mandatory:** Yes **Suggested Value:** localhost
`postgres_node_url`	Postgres URL **Mandatory:** **Suggested Value:** localhost
`zookeeper.connectString`	Host and port of ZooKeeper **Mandatory:** Yes **Suggested Value:** `zookeeper:9600`
`zookeeper.config.path`	ZooKeeper namespace where the configuration is stored **Mandatory:** No

Parameter	Description
	Suggested Value: `/unity/system/config`
`ZK_CLIENT_KEY_FILE`	Path to the ZooKeeper client private key **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/key`
`ZK_CLIENT_KEY_PASSWORD`	Password of the ZooKeeper client key **Mandatory:** Yes **Suggested Value:**`<password>`
`ZK_CLIENT_CACERT_FILE`	Path to CA certificate file used for generating the key **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/cacert`
`ZK_CLIENT_CERT_FILE`	Path to the ZooKeeper client certificate **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/certificate`
`ZK_CLIENT_TRUSTSTORE_FILE`	Path to the ZooKeeper client truststore. This truststore must contain the certificate of the CA that issued the certificate to the ZooKeeper server. The supported truststore types are PEM, PKCS12, and JKS. To create a PKCS12 trust store without a key and to use it as zookeeper, use Java's keytool utility so that a java-based application can understand them. Example:

Parameter	Description
	`keytool -import -alias mycert -file certificate.pem -keystore truststore.p12 -storetype PKCS12 -storepass password` **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/zookeeper-client-truststore.pem`
`LOAD_CONFIG_FROM_ENV`	The flag that indicates whether to use variables or predefined default values **Mandatory:** Yes **Suggested Value:** TRUE
`grpc_session_port`	Port that the Hawk RedTail Console component service uses to listen to incoming transport request from Hawk agents **Mandatory:** No **Suggested Value:** 9697
`hawk_domain`	The Hawk domain name. **Mandatory:** Yes **Suggested Value:** `redtail`
`hawk_domain_platform`	Platform of the domain mentioned in property `hawk_domain`. Possible values are kubernetes, docker, or enterprise **Mandatory:** No

Parameter	Description
	Suggested Value: `enterprise`
`machineId`	Internal component ID **Mandatory:** Yes **Suggested Value:** machine-0000000000
`unity.services.rest.host`	Host IP for Hawk RedTail Console REST communication **Mandatory:** Yes **Suggested Value:** 0.0.0.0
`hawk_console_server_port`	Listen port for Hawk RedTail Console REST communication **Mandatory:** Yes **Suggested Value:** 9687
`GRAFANA_URL`	Grafana URL **Mandatory:** Yes **Suggested Value:** *http://grafana:3000*
`REST_TLS_PROTOCOL`	Supported TLS protocols **Mandatory:** Yes **Suggested Value:** `TLSv1.2, TLSv1.3`
`REST_TLS_CIPHERS`	Supported Cipher Suites

Parameter	Description
	Mandatory: Yes **Suggested Value:**`TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, TLS_AES_128_GCM_SHA256`
`REST_TLS_KEY_FILE`	Key pair used for setting up REST TLS communication **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/key`
`REST_TLS_KEY_PASSWORD`	Password to key pair used for REST TLS communication **Mandatory:** Yes **Suggested Value:** `<password>`
`REST_TLS_CERT_FILE`	Certificate used for REST TLS communication **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/certificate`
`REST_TLS_CACERT_FILE`	Certificate of the CA used to sign the REST TLS certificate

Parameter	Description
	Mandatory: Yes **Suggested Value:** `{certs_conf_path}/cacert`
`TLS_SKIP_CERTIFICATE_VERIFICATION`	Skip certificate verification **Mandatory:** No **Suggested Value:** false
`TLS_SKIP_HOSTNAME_VERIFICATION`	Skip host name verification **Mandatory:** No **Suggested Value:** true
`datasource_url`	Connection URL to PostgreSQL server **Mandatory:** Yes **Suggested Value:** `jdbc:postgresql://{postgres_node_url}:5432/logumon`
`datasource_drivername`	JDBC class name **Mandatory:** Yes **Suggested Value:** `org.postgresql.Driver`
`datasource_username`	Database user name **Mandatory:** Yes **Suggested Value:** postgres

Parameter	Description
`datasource_password`	Database password **Mandatory:** Yes **Suggested Value:** mypassword
`datasource_connection_pool_initial_size`	Database Connection pool size at start up **Mandatory:** No **Suggested Value:** "10"
`datasource_connection_pool_max_idle`	Maximum number of idle connections allowed in the database connection pool **Mandatory:** No **Suggested Value:** "20"
`datasource_connection_pool_max_active`	Maximum number of active connections allowed in the database connection pool **Mandatory:** No **Suggested Value:** 100
`datasource_tls_skip_hostname_verification`	Skip host name verification while communicating with database over TLS **Mandatory:** No **Suggested Value:** true
`datasource_tls_skip_`	Skip certificate verification while communicating with

Parameter	Description
`certificate_verification`	database over TLS **Mandatory:** No **Suggested Value:** true
`datasource_tls_cacert_file`	Path to the database CA certificate. This is valid if PostgreSQL is secured with TLS. **Mandatory:** No **Suggested Value:** `{certs_conf_path}/rt_dbcacert`
`user_store_type`	Type of store where the users are stored **Mandatory:** Yes **Suggested Value:** database
`hawk_console_retention_count_notification`	Alert limit for notifications **Mandatory:** Yes **Suggested Value:** 100000
`hawk_console_retention_count_high_alerts`	Alert limit for high level alerts **Mandatory:** Yes **Suggested Value:** 100000
`hawk_console_retention_count_medium_alerts`	Alert limit for medium level alerts

Parameter	Description
	Mandatory: Yes **Suggested Value:** 100000
`hawk_console_retention_count_low_alerts`	Alert limit for low level alerts **Mandatory:** Yes **Suggested Value:** 100000
`JAVA_OPTS`	JVM properties which can be configured to tune the JVM process For example, `-Xms512m-Xmx2g` **Mandatory:** No **Suggested Value:** `-Djava.io.tmpdir=/tmp -DContainer-Hawk=true --add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/java.math=ALL-UNNAMED --add-opens java.base/java.util=ALL-UNNAMED --add-opens java.base/java.net=ALL-UNNAMED --add-opens java.base/java.text=ALL-UNNAMED --add-opens java.base/java.nio=ALL-UNNAMED --add-opens java.base/java.util.concurrent=ALL-UNNAMED -Xms2g -Xmx4g`

Enterprise Machine Node Configurations

You can configure the Machine node for the enterprise deployment. All the required configuration parameters are stored in *CONFIG_FOLDER_REDTAIL*`/rt_machinenode_vars.json`

Each of the parameters is explained in more detail in the following table:

Machine Node Configuration Options

Parameter	Description
`certs_conf_path`	The directory where the certificates required for communicating with other nodes is stored **Mandatory:** Yes **Suggested Value:**`/usr/local/etc/rt_certs`
`data_path`	The base directory for storing TIBCO OI Hawk RedTail services data. All the services store the data in the path categorized by service name. **Mandatory:** Yes **Suggested Value:***DATA_FOLDER*
`zookeeper_node_url`	ZooKeeper URL **Mandatory:** Yes **Suggested Value:** localhost
`prometheus_node_url`	Prometheus URL **Mandatory:** Yes **Suggested Value:** localhost
`hawkconsole_node_url`	`hawkconsole`node URL **Mandatory:** Yes **Suggested Value:** localhost

Parameter	Description
`zookeeper.connectString`	Host and port of ZooKeeper **Mandatory:** Yes **Suggested Value:** `zookeeper:9600`
`zookeeper.config.path`	ZooKeeper namespace where the configuration is stored **Mandatory:** No **Suggested Value:**`/unity/system/config`
`ZK_CLIENT_KEY_FILE`	Path to the ZooKeeper client private key **Mandatory:** Yes **Suggested Value:**`{certs_conf_path}/querynode-client-key`
`ZK_CLIENT_KEY_PASSWORD`	Password of the ZooKeeper client key **Mandatory:** Yes **Suggested Value:**`<password>`
`ZK_CLIENT_CACERT_FILE`	Path to CA certificate file used for generating the key **Mandatory:** Yes **Suggested Value:**`{certs_conf_path}/cacert`
`ZK_CLIENT_CERT_FILE`	Path to the ZooKeeper client certificate **Mandatory:** Yes

Parameter	Description
	Suggested Value: `{certs_conf_path}/querynode-client-certificate`
`ZK_CLIENT_TRUSTSTORE_FILE`	Path to the ZooKeeper client truststore. This truststore must contain the certificate of the CA which issued the certificate to the ZooKeeper server. The supported truststore types are PEM, PKCS12, and JKS. To create a PKCS12 trust store without a key and to use it as zookeeper, use Java's keytool utility so that a java-based application can understand them. Example: `keytool -import -alias mycert -file certificate.pem -keystore truststore.p12 -storetype PKCS12 -storepass password` **Mandatory:** Yes **Suggested Value:**`{certs_conf_path}/zookeeper-client-truststore.pem`
`machineId`	Internal component ID **Mandatory:** Yes **Suggested Value:** machine-0000000000

Enterprise Query Node Configurations

You can configure the Query Node for the enterprise deployment. All the required configuration parameters are stored in *CONFIG_FOLDER_REDTAIL*`/rt_querynode_vars.json`

Each of the parameters is explained in more detail in the following table:
Query Node Configuration Options

Parameter	Description
`certs_conf_path`	The directory where the certificates required for communicating with other nodes is stored **Mandatory:** Yes **Suggested Value:**`/usr/local/etc/rt_certs`
`data_path`	The base directory for storing TIBCO OI Hawk RedTail services data. All the services store the data in the path categorized by service name. **Mandatory:** Yes **Suggested Value:***DATA_FOLDER*
`zookeeper_node_url`	ZooKeeper URL **Mandatory:** Yes **Suggested Value:** localhost
`prometheus_node_url`	Prometheus URL **Mandatory:** Yes **Suggested Value:** localhost
`hawkconsole_node_url`	`hawkconsole`node URL **Mandatory:** Yes

Parameter	Description
	Suggested Value: localhost
`zookeeper.connectString`	Host and port of ZooKeeper **Mandatory:** Yes **Suggested Value:** `zookeeper:9600`
`zookeeper.config.path`	ZooKeeper namespace where the configuration is stored **Mandatory:** No **Suggested Value:** `/unity/system/config`
`unity.services.rest.host`	Host address of `querynode` **Mandatory:** Yes **Suggested Value:** 0.0.0.0
`unity.services.rest.port`	Port on which query listen to HTTP request **Mandatory:** Yes **Suggested Value:** 9681
`unity.services.rest.options.results.maxpage`	Maximum number of search results shown on the search page **Mandatory:** Yes

Parameter	Description
	Suggested Value: 1000000
`unity.services.query.host`	Query Node self host IP which gets registered with ZooKeeper **Mandatory:** Yes **Suggested Value:** localhost
`unity.services.query.port`	Query Node self port which gets registered with ZooKeeper **Mandatory:** Yes **Suggested Value:** 9620
`unity.storage.cache`	Internal cache for storing query results for each cached query **Mandatory:** Yes **Suggested Value:** `{data_path}/querynode/.query/qcache`
`unity.storage.maxSplitH2fileSize`	Defines maximum size of file for H2 file splitting • 0 - defines no file splitting. • 31 - 2 GB file size. **Mandatory:** Yes **Suggested Value:** 31
`unity.maxConcurrentQuery`	Maximum number of queries that can

Parameter	Description
	be run concurrently **Mandatory:** Yes **Suggested Value:** 25
ZK_CLIENT_KEY_FILE	Path to the ZooKeeper client private key **Mandatory:** Yes **Suggested Value:** {certs_conf_path}/querynode-client-key
ZK_CLIENT_KEY_PASSWORD	Password of the ZooKeeper client key **Mandatory:** Yes **Suggested Value:** <password>
ZK_CLIENT_CACERT_FILE	Path to CA certificate file used for generating the key **Mandatory:** Yes **Suggested Value:**{certs_conf_path}/cacert
ZK_CLIENT_CERT_FILE	Path to the ZooKeeper client certificate **Mandatory:** Yes **Suggested Value:** {certs_conf_path}/querynode-client-certificate

Parameter	Description
ZK_CLIENT_TRUSTSTORE_FILE	Path to the ZooKeeper client truststore. This truststore must contain the certificate of the CA which issued the certificate to the ZooKeeper server. The supported truststore types are PEM, PKCS12, and JKS. To create a PKCS12 trust store without a key and to use it as zookeeper, use Java's keytool utility so that a java-based application can understand them. Example: `keytool -import -alias mycert -file certificate.pem -keystore truststore.p12 -storetype PKCS12 -storepass password` **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/zookeeper-client-truststore.pem`
machineId	Internal component ID **Mandatory:** Yes **Suggested Value:** machine-0000000000
LOAD_CONFIG_FROM_ENV	Flag to indicate whether to load variables or to use predefined default values

Parameter	Description
	Mandatory: Yes **Suggested Value:** TRUE
LOG_LEVEL	Specifies the level of diagnostic information stored in the logs. The logging levels are as follows: • ERROR - Indicates that error level trace messages should be enabled. • WARNING - Indicates that warning level trace messages should be enabled. • INFO - Indicates that information level trace messages should be enabled. • DEBUG - Indicates that debug level trace messages should be enabled. • TRACE - Indicates that trace level messages should be enabled. **Mandatory:** Yes **Suggested Value:** info
REST_TLS_PROTOCOL	Supported TLS protocols **Mandatory:** Yes **Suggested Value:** `TLSv1.2, TLSv1.3`

Parameter	Description
REST_TLS_CIPHERS	Supported Cipher Suites **Mandatory:** Yes **Suggested Value:**TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, TLS_AES_128_GCM_SHA256
REST_TLS_KEY_FILE	Key used for setting up REST TLS communication **Mandatory:** Yes **Suggested Value:** {certs_conf_path}/key
REST_TLS_KEY_PASSWORD	Password to the key used for REST TLS communication **Mandatory:** Yes **Suggested Value:** <password>

Parameter	Description
`REST_TLS_CERT_FILE`	Certificate used for REST TLS communication **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/certificate`
`REST_TLS_CACERT_FILE`	Certificate of the CA used to sign the REST TLS certificate **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/cacert`
`TLS_SKIP_CERTIFICATE_VERIFICATION`	Skip certificate verification **Mandatory:** No **Suggested Value:** false
`TLS_SKIP_HOSTNAME_VERIFICATION`	Skip host name verification **Mandatory:** No **Suggested Value:** true
`TLS_CLIENT_KEY_FILE`	Path to the `querynode` client key to communicate with `hawkconsolenode` **Mandatory:** Yes **Suggested Value:** `{certs_conf_`

Parameter	Description
	path}/querynode-client-key
TLS_CLIENT_CERT_FILE	Path to querynode client certificate to communicate with hawkconsolenode **Mandatory:** Yes **Suggested Value:** {certs_conf_path}/querynode-client-certificate
PROMETHEUS_SERVER_HOST	Host name of the Prometheus server **Mandatory:** Yes **Suggested Value:** {prometheus_node_url}
PROMETHEUS_SERVER_PORT	Port of the Prometheus server **Mandatory:** Yes **Suggested Value:** 9090
PROMETHEUS_SERVER_TLS_ENABLED	Set to true, if communication with Prometheus server needs to happen over TLS protocol **Mandatory:** No **Suggested Value:** false
HAWKCONSOLE_HOST	hawkconsolenode host

Parameter	Description
	Note: You must delete this parameter from the configuration file when configuring TIBCO OI Hawk RedTail in HA mode. **Mandatory:** Yes **Suggested Value:** `{hawkconsole_node_url}`
`HAWKCONSOLE_PORT`	`hawkconsolenode` port **Note:** You must delete this parameter from the configuration file when configuring TIBCO OI Hawk RedTail in HA mode. **Mandatory:** Yes **Suggested Value:** 9687
`HAWKCONSOLE_TLS_ENABLED`	True, if communication with Prometheus server needs to happen over TLS protocol **Mandatory:** Yes **Suggested Value:** true
`JAVA_OPTS`	JVM properties which can be configured to tune the JVM process. For example, `-Xms512m-Xmx2g`

Parameter	Description
	Mandatory: No **Suggested Value:** `-Xms512m-Xmx2g`

Enterprise Prometheus Configurations

You can configure the Prometheus for enterprise deployment. All the required configuration parameters are stored in *CONFIG_FOLDER_REDTAIL*`/rt_prometheus_vars.json`

Each of the parameters is explained in more detail in the following table:
Prometheus Configuration Options

Parameter	Description
`certs_conf_path`	The directory where the certificates required for communicating with other nodes is stored **Mandatory:** Yes **Suggested Value:** `/usr/local/etc/rt_certs`
`data_path`	The base directory for storing TIBCO OI Hawk RedTail services data. All the services store the data in the path categorized by service name. **Mandatory:** Yes **Suggested Value:** *DATA_FOLDER*
`prometheus_node_url`	Prometheus URL **Mandatory:** Yes **Suggested Value:** localhost

Parameter	Description
`prometheus_port`	Prometheus port **Mandatory:** Yes **Suggested Value:** 9090
`zookeeper_node_url`	ZooKeeper URL **Mandatory:** Yes **Suggested Value:** localhost
`zookeeper.connectString`	Host and port of ZooKeeper **Mandatory:** Yes **Suggested Value:** `zookeeper:9600`
`zookeeper.config.path`	ZooKeeper namespace where the configuration is stored **Mandatory:** **Suggested Value:** `/unity/system/config`
`ZK_CLIENT_KEY_FILE`	Path to the ZooKeeper client private key **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/key`
`ZK_CLIENT_KEY_PASSWORD`	Password of the ZooKeeper client key **Mandatory:** Yes

Parameter	Description
	Suggested Value: `<password>`
`ZK_CLIENT_CACERT_FILE`	Path to CA certificate file used for generating the key **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/cacert`
`ZK_CLIENT_CERT_FILE`	Path to the ZooKeeper client certificate **Mandatory:** Yes **Suggested Value:**`{certs_conf_path}/certificate`

Enterprise Prometheus Service Discovery Configurations

You can configure the Prometheus service discovery for the on-premises deployment. All the required configuration parameters are stored in *CONFIG_FOLDER_REDTAIL*`/rt_prometheus_discoveryservice_vars.json`

Each of the parameters is explained in more detail in the following table:
Prometheus Service Discovery Configuration Options

Parameter	Description
`certs_conf_path`	The directory where the certificates required for communicating with other nodes is stored **Mandatory:** Yes **Suggested Value:**`/usr/local/etc/rt_certs`

Parameter	Description
`data_path`	The base directory for storing TIBCO OI Hawk RedTail services data. All the services store the data in the path categorized by service name. **Mandatory:** Yes **Suggested Value:** *DATA_FOLDER*
`zookeeper_node_url`	ZooKeeper URL **Mandatory:** Yes **Suggested Value:** localhost
`zookeeper.connectString`	Host and port of ZooKeeper **Mandatory:** Yes **Suggested Value:** `zookeeper:9600`
`zookeeper.config.path`	ZooKeeper namespace where the configuration is stored **Mandatory:** **Suggested Value:**`/unity/system/config`
`ZK_CLIENT_KEY_FILE`	Path to the ZooKeeper client private key **Mandatory:** Yes **Suggested Value:**`{certs_conf_path}/key`
`ZK_CLIENT_KEY_PASSWORD`	Password of the ZooKeeper client key

Parameter	Description
	Mandatory: Yes **Suggested Value:**<password>
ZK_CLIENT_CACERT_FILE	Path to CA certificate file used for generating the key **Mandatory:** Yes **Suggested Value:**{certs_conf_path}/cacert
ZK_CLIENT_CERT_FILE	Path to the ZooKeeper client certificate **Mandatory:** Yes **Suggested Value:**{certs_conf_path}/certificate
client_certificate	The certificate which is going to be used for TLS communication with the Hawk RedTail Console **Mandatory:** Yes **Suggested Value:** /usr/local/etc/rt_certs/prometheus-client-certificate
client_key	The key which is going to be used for TLS communication with the Hawk RedTail Console **Mandatory:** Yes **Suggested Value:** /usr/local/etc/rt_certs/prometheus-client-key
hawkconsole_ca	The CA certificate of the Hawk RedTail Console

Parameter	Description
	Mandatory: Yes **Suggested Value:** `/usr/local/etc/rt_certs/cacert`
`target_output_file`	Path to the file where the discovered Prometheus targets are stored **Mandatory:** Yes **Suggested Value:** `/usr/local/tibco_redtail_data/prometheus_discoveryservice/hawktargets.json`
`log_level`	Specifies the level of diagnostic information stored in the logs. The logging level are as follows: • ERROR - Indicates that error level trace messages should be enabled. • WARNING - Indicates that warning level trace messages should be enabled. • INFO - Indicates that information level trace messages should be enabled. • DEBUG - Indicates that debug level trace messages should be enabled. • TRACE - Indicates that trace level messages should be enabled. **Mandatory:** Yes **Suggested Value:** INFO

Enterprise Prometheus Backup Service Configurations

You can configure the Prometheus backup service for enterprise deployment. All the required configuration parameters are stored in *CONFIG_FOLDER_REDTAIL*`/rt_prometheus_backup_vars.json`

Each of the parameters is explained in more detail in the following table:

Prometheus Backup Configuration Options

Parameter	Description
`certs_conf_path`	The directory where the certificates required for communicating with other nodes is stored **Mandatory:** Yes **Suggested Value:**`/usr/local/etc/rt_certs`
`zookeeper_node_url`	ZooKeeper URL **Mandatory:** Yes **Suggested Value:** localhost
`zookeeper.connectString`	Host and port of ZooKeeper **Mandatory:** Yes **Suggested Value:** `zookeeper:9600`
`zookeeper.config.path`	ZooKeeper namespace where the configuration is stored **Mandatory:** No **Suggested Value:** */unity/system/config*

Parameter	Description
ZK_CLIENT_KEY_FILE	Path to the ZooKeeper client private key **Mandatory:** Yes **Suggested Value:** {certs_conf_path}/key
ZK_CLIENT_KEY_PASSWORD	Password of the ZooKeeper client key **Mandatory:** Yes **Suggested Value:** <password>
ZK_CLIENT_CACERT_FILE	Path to CA certificate file used for generating the key **Mandatory:** Yes **Suggested Value:** {certs_conf_path}/cacert
ZK_CLIENT_CERT_FILE	Path to the ZooKeeper client certificate **Mandatory:** Yes **Suggested Value:** {certs_conf_path}/certificate
ZK_CLIENT_TRUSTSTORE_FILE	Path to the ZooKeeper client truststore. This truststore must contain the certificate of the CA that issued the certificate to the ZooKeeper server. The supported truststore types are PEM, PKCS12, and JKS. To create a PKCS12 trust store without a key and to use it as zookeeper, use Java's keytool utility so that a java-based application can understand them. Example:

Parameter	Description
	`keytool -import -alias mycert -file certificate.pem -keystore truststore.p12 -storetype PKCS12 -storepass password` **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/zookeeper-client-truststore.pem`
`data_path`	The base directory for storing TIBCO OI Hawk RedTail services data. All the services store the data in the path categorized by service name. **Mandatory:** Yes **Suggested Value:***DATA_FOLDER*
`backup_file_path`	The directory where the Prometheus backup service stores the data **Mandatory:** Yes **Suggested Value:** */usr/local/tibco_redtail_data/prometheus_backup*
`prometheus_backup_time_interval`	The time (in seconds) after which the backup process is started **Mandatory:** Yes **Suggested Value:** 900
`prometheus_max_backup`	The number of copies of the backup data

Parameter	Description
	Mandatory: Yes **Suggested Value:** 5

Enterprise Postgresql Configurations

You can configure the Prometheus for the enterprise deployment. All the required configuration parameters are stored in *CONFIG_FOLDER_REDTAIL*/`rt_postgresql_vars.json`

Each of the parameters is explained in more detail in the following table:
Postgresql Configuration Options

Parameter	Description
`certs_conf_path`	The directory where the certificates required for communicating with other nodes is stored **Mandatory:** Yes **Suggested Value:** `/usr/local/etc/rt_certs`
`data_path`	The base directory for storing TIBCO OI Hawk RedTail services data. All the services store the data in the path categorized by service name. **Mandatory:** Yes **Suggested Value:** *DATA_FOLDER*
`zookeeper_node_url`	ZooKeeper URL **Mandatory:** Yes **Suggested Value:** localhost

Parameter	Description
`zookeeper.connectString`	Host and port of ZooKeeper **Mandatory:** Yes **Suggested Value:** `zookeeper:9600`
`zookeeper.config.path`	ZooKeeper namespace where the configuration is stored **Mandatory:** **Suggested Value:** `/unity/system/config`
`ZK_CLIENT_KEY_FILE`	Path to the ZooKeeper client private key **Mandatory:** Yes **Suggested Value:**`{certs_conf_path}/key`
`ZK_CLIENT_KEY_PASSWORD`	Password of the ZooKeeper client key **Mandatory:** Yes **Suggested Value:**`<password>`
`ZK_CLIENT_CACERT_FILE`	Path to CA certificate file used for generating the key **Mandatory:** Yes **Suggested Value:**`{certs_conf_path}/cacert`
`ZK_CLIENT_CERT_FILE`	Path to the ZooKeeper client certificate **Mandatory:** Yes

Parameter	Description
	Suggested Value: `{certs_conf_path}/certificate`
`POSTGRES_HOST_AUTH_METHOD`	Authentication mechanism to be used with the PostgreSQL server **Mandatory:** Yes **Suggested Value:** `password`
`POSTGRES_PASSWORD`	PostgreSQL server root user's password **Mandatory:** Yes **Suggested Value:** mypassword

Enterprise Grafana Configurations

You can configure the Grafana for the enterprise deployment. All the required configuration parameters are stored in *CONFIG_FOLDER_REDTAIL*`/rt_grafana_vars.json`

Each of the parameters is explained in more detail in the following table:

Grafana Configuration Options

Parameter	Description
`certs_conf_path`	The directory where the certificates required for communicating with other nodes is stored **Mandatory:** Yes **Suggested Value:** `/usr/local/etc/rt_certs`
`data_path`	The base directory for storing TIBCO OI Hawk RedTail services

Parameter	Description
	data. All the services store the data in the path categorized by service name. **Mandatory:** Yes **Suggested Value:** *DATA_FOLDER*
`conf_path`	The base directory for storing TIBCO OI Hawk RedTail services configuration. **Mandatory:** **Suggested Value:** *CONFIG_FOLDER_REDTAIL*
`zookeeper_node_url`	ZooKeeper URL **Mandatory:** Yes **Suggested Value:** localhost
`grafana_node_url`	Grafana URL **Mandatory:** Yes **Suggested Value:** localhost
`postgres_node_url`	Postgres URL **Mandatory:** **Suggested Value:** localhost
`zookeeper.connectString`	Host and port of ZooKeeper

Parameter	Description
	Mandatory: Yes **Suggested Value:** `zookeeper:9600`
`ZK_CLIENT_KEY_FILE`	Path to the ZooKeeper client private key **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/key`
`ZK_CLIENT_KEY_PASSWORD`	Password of the ZooKeeper client key **Mandatory:** Yes **Suggested Value:** `<password>`
`ZK_CLIENT_CACERT_FILE`	Path to the CA certificate file used for generating the key **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/cacert`
`ZK_CLIENT_CERT_FILE`	Path to the ZooKeeper client certificate **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/certificate`
`GF_USERS_ALLOW_SIGN_UP`	When set to false, this parameter prohibits users from being able to sign up or create user accounts. The admin user can still create users from the Grafana Admin Pages. **Default:**`false`.

Parameter	Description
	Mandatory: Yes **Suggested Value:** false
GF_USERS_AUTO_ASSIGN_ORG	**When set to `true`:** Automatically adds new users to the main organization (ID 1). **When set to `false`:** A new organization is created for the new user automatically. **Default:**`true` **Mandatory:** Yes **Suggested Value:** true
GF_USERS_AUTO_ASSIGN_ORG_ROLE	The role new users are assigned for the main organization (if `GF_USERS_AUTO_ASSIGN_ORG` is set to true). For TIBCO OI Hawk RedTail, this value must be `Editor`. **Default Value:** Viewer **Mandatory:** Yes **Suggested Value:** Editor
GF_USERS_DEFAULT_THEME	Set the default UI theme: `dark` or `light`. Default is `dark`. For TIBCO OI Hawk RedTail, the suggested value is light. **Mandatory:** Yes **Suggested Value:** light
GF_AUTH_PROXY_ENABLED	Set to true, for Grafana to let a HTTP reverse proxy handle authentication. For TIBCO OI Hawk RedTail, this value must be

Parameter	Description
	true. **Mandatory:** Yes **Suggested Value:** true
GF_AUTH_PROXY_HEADER_NAME	HTTP Header name that contains the user name **Mandatory:** Yes **Suggested Value:** X-WEBAUTH-USER
GF_AUTH_PROXY_HEADER_PROPERTY	HTTP Header property, defaults to `username` **Mandatory:** Yes **Suggested Value:** `username`
GF_AUTH_PROXY_AUTO_SIGN_UP	Set to `true` to enable auto sign up of users who do not exist in the Grafana database. Default is `true`. **Mandatory:** Yes **Suggested Value:** true
GF_SERVER_DOMAIN	This setting is only used in as a part of the `root_url` setting **Mandatory:** Yes **Suggested Value:** `{grafana_node_url}`
GF_SERVER_HTTP_PORT	The port to bind to

Parameter	Description
	Default Value: 3000 **Mandatory:** Yes **Suggested Value:** 3000
GF_SERVER_ROOT_URL	This is the full URL used to access Grafana from a web browser **Mandatory:** Yes **Suggested Value:** `%(protocol)s://%(domain)s:%(http_port)s/redtail/grafana`
GF_AUTH_BASIC_ENABLED	Basic authentication is enabled by default and works with built-in Grafana. Do not disable basic auth for TIBCO OI Hawk RedTail. **Mandatory:** Yes **Suggested Value:** true
GF_SECURITY_ALLOW_EMBEDDING	**Default:**`false`. When false, the X-Frame-Options deny HTTP header is set in the Grafana HTTP responses. Thus, browsers do not allow rendering Grafana in <frame>, <iframe>, <embed>, or <object>. For TIBCO OI Hawk RedTail, set this value to `true`. **Mandatory:** Yes **Suggested Value:** true
GF_PATHS_PLUGINS	Directory where Grafana automatically scans and looks for plug-ins. Manually or automatically install any plug-ins here.

Parameter	Description
	Mandatory: Yes **Suggested Value:** `{conf_path}/grafana/plugins`
`GF_DATABASE_TYPE`	Type of database where Grafana stores all the data **Mandatory:** Yes **Suggested Value:** postgres
`GF_DATABASE_HOST`	Host and IP port of the database **Mandatory:** Yes **Suggested Value:** *{postgres_node_url}:5432*
`GF_DATABASE_NAME`	Name of the database **Mandatory:** Yes **Suggested Value:** grafana
`GF_DATABASE_USER`	Database user name **Mandatory:** Yes **Suggested Value:** postgres
`GF_DATABASE_PASSWORD`	Database user password **Mandatory:** Yes **Suggested Value:** `mypassword`

Parameter	Description
GF_DATABASE_SSL_MODE	Skips verification of the certificate chain and `hostname` when making the connection **Mandatory:** Yes **Suggested Value:** require
GF_DATABASE_CA_CERT_PATH	Database CA certificate **Mandatory:** Yes **Suggested Value:** `{certs_conf_path}/gf_dbcacert`
GF_LOG_LEVEL	Specifies the level of diagnostic information stored in the logs. The logging levels are as follows: • ERROR - Indicates that error level trace messages should be enabled. • WARNING - Indicates that warning level trace messages should be enabled. • INFO - Indicates that information level trace messages should be enabled. • DEBUG - Indicates that debug level trace messages should be enabled. • TRACE - Indicates that trace level messages should be enabled. **Mandatory:** No **Suggested Value:** DEBUG

OI Hawk Console Configurations

You can install the OI Hawk Console in an enterprise environment by selecting the Custom Installation profile and then selecting the **Console** feature while installing TIBCO OI Hawk RedTail. The OI Hawk Console does not support any of the TIBCO OI Hawk RedTail Advanced features and must be configured by using either of TIBCO EMS, TIBCO Rendezvous or TCP transport for TIBCO Hawk. You must also configure the Hawk agent (`hawkagent.cfg`) to communicate over TIBCO EMS, TIBCO Rendezvous or TCP transport for TIBCO Hawk so that the Hawk agent is able to communicate with the OI Hawk Console.

All the required configuration parameters for OI Hawk Console are stored in the `hawkconsole.cfg` configuration file located at *CONFIG_FOLDER/bin*.

> **Note:** Ensure that the `RV_HOME` or `EMS_HOME` parameters are configured correctly in the `tibhawkagent.tra` file when configuring the Hawk agent to communicate with the OI Hawk Console TIBCO Rendezvous transport or TIBCO EMS transport.

For more information about the configurations that you can perform in OI Hawk Console, see the following topics:

- Domain and Transport Configuration for OI Hawk Console
- User Authentication in OI Hawk Console
- Secure Communication over OI Hawk Console
- Configuring an External Database
- Configuring OI Hawk Console Database Schema
- OI Hawk Console Configuration Options

Domain and Transport Configuration for OI Hawk Console

You can register a Hawk domain to the OI Hawk Console and specify the transport type for communication. You can either use the web interface of OI Hawk Console or configure the domain and transport configuration file (`DomainTransportConfig.yml`).

Domain Registration by Using Configuration File

The domain and transport configuration file (`DomainTransportConfig.yml`) for the OI Hawk Console contains the parameters to connect to regular and proxy domains.

You can specify the location of the `DomainTransportConfig.yml` file by using the `domain_config_file` option in the OI Hawk Console configuration file (`hawkconsole.cfg`). For details on options present in the `hawkconsole.cfg` file, see OI Hawk Console Configuration Options.

- In the `DomainTransportConfig.yml` file you can specify the following elements for the connection:
- `domainConfiguration` - The parent tag for the domain and transport configurations for OI Hawk Console.
- Domain type - Specify whether the Hawk domain to be registered is a regular domain or a proxy domain. Based on the domain type, specify additional configuration parameters. The tags used for the domain type are:
  - `regular` - For details about fields for the regular domain type, see Configuration Options for DomainTransportConfig.yml for the Regular Domain Type.
  - `proxy` - For details about fields for the proxy domain type, see Configuration Options for DomainTransportConfig.yml for the Proxy Domain Type.
- `domainName` - Name of the domain that is to be registered.

Configuration Options for DomainTransportConfig.yml for the Regular Domain Type

Property	Description
`transport`	The type of transport over which the communication between the OI Hawk Console and the Hawk agent takes place. You can specify this parameter as one of the following: • **tibtcp:** TCP Transport for TIBCO Hawk • **tibrv:** TIBCO Rendezvous • **tibems:** TIBCO Enterprise Message Service Transport You must specify the transport configuration based on the selected transport type.

Property	Description
	Mandatory: Yes **Suggested Value:** tibtcp
`securityPolicy`	The security policy currently in effect. **Mandatory:** No **Suggested Value:** -
TCP Transport for TIBCO Hawk Parameters (Mandatory if you have specified the `transport` parameter as tibtcp)	
`tcpSelfUrl`	Unique socket address of the OI Hawk Console for connecting to the TCP Transport for TIBCO Hawk cluster. The syntax is *<self IP>:<self port>*. **Mandatory:** No **Suggested Value:** `localhost:2561`
`tcpDaemonUrl`	The socket address of the Cluster Manager acting as the seed node for the TCP Transport for TIBCO Hawk cluster. **Mandatory:** No **Suggested Value:** `localhost:2561`
TCP Transport for TIBCO Hawk SSL Parameters (Mandatory if you want to configure TCP Transport for TIBCO Hawk over SSL)	
`tcpSslKeyStore`	Path of the keystore file

Property	Description
	Mandatory: No **Suggested Value:** Path of the keystore file
`tcpSslTrustStore`	Path of the trust store file **Mandatory:** No **Suggested Value:** Path of the trust store file
`tcpSslKeyStorePassword`	Password to access the keystore **Mandatory:** No **Suggested Value:** Password to access the keystore
`tcpSslKeyPassword`	Password to access the private key **Mandatory:** No **Suggested Value:** Password to access the private key
`tcpSslTrustStorePassword`	Password to access the trust store **Mandatory:** No **Suggested Value:** Password to access the trust store
`tcpSslProtocol`	Protocol for a secure connection **Mandatory:** No **Suggested Value:** TLSv1.2

Property	Description
`tcpSslEnabledAlgorithms`	Algorithm to be used for the security protocol. You can specify multiple algorithms as a comma-separated list without space. **Mandatory:** No **Suggested Value:** `TLS_RSA_WITH_AES_128_CBC_SHA`
TIBCO Rendezvous Transport Parameters (Mandatory if you have specified the `transport` parameter as tibrv)	
`rvService`	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". **Mandatory:** No **Suggested Value:** `<RV service name>`
`rvNetwork`	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. **Mandatory:** No **Suggested Value:** `<RV network name>`
`rvDaemon`	Specify the socket address of the Rendezvous daemon. **Mandatory:** No **Suggested Value:** `<RV daemon name>`

Property	Description
TIBCO Enterprise Message Service Transport Parameters (Mandatory if you have specified the `transport` parameter as tibems)	
`emsServerUrl`	Specify the location of the EMS server **Mandatory:** No **Suggested Value:** *<server url>*
`emsUserName`	Specify the user name to login to the EMS server **Mandatory:** No **Suggested Value:** `<user name>`
`emsPassword`	Specify the password for the `emsUserName` **Mandatory:** No **Suggested Value:** `<password_string>`
TIBCO Enterprise Message Service SSL Parameters (Mandatory if you want to configure TIBCO EMS over SSL)	
`emsSslVendor`	The name of the vendor of the SSL implementation. The valid choices are: • j2se (default) - Use this option when you want to use the default Java Cryptography Extension (JCE) bundled with the Java JRE. • entrust61 - Use this option when you want to use the Entrust libraries. • ibm - On non-IBM platforms, this option can be used only if the IBM version of JCE is installed.

Property	Description
	Mandatory: No **Suggested Value:** `<SSL vendor>`
`emsSslTrace`	The option enables the SSL tracing **Mandatory:** No **Suggested Value:** `<SSL trace>`
`emsSslTrusted`	The option specifies the file name of the server certificates. This option can be repeated if more than one certificate file is used. **Mandatory:** No **Suggested Value:** `<SSL trusted>`
`emsSslPrivateKey`	This option indicates the private key of the TIBCO OI Hawk component **Mandatory:** No **Suggested Value:** `<SSL private key>`
`emsSslExpectedHostname`	The name that is expected in the name of the CN field of the server certificates is specified by this option **Mandatory:** No **Suggested Value:** `<SSL expected>`
`emsSslPassword`	The password to decrypt the identity file of the

Property	Description
	TIBCO OI Hawk component **Mandatory:** No **Suggested Value:** `<password_string>`
`emsSslIdentity`	Digital certificate **Mandatory:** No **Suggested Value:** `<identity store>`
`emsSslNoVerifyHost`	Specifies whether the EMS server should be verified or not **Mandatory:** No **Suggested Value:** true
`emsSslNoVerifyHostname`	Specifies whether the host name must not be verified **Mandatory:** No **Suggested Value:** true
`emsSslCiphers`	Supported cipher suites **Mandatory:** No **Suggested Value:** `<cipher suite-names>`

Configuration Options for `DomainTransportConfig.yml` for the Proxy Domain Type

Property	Description
`host`	URL of the domain that needs to be registered. The syntax is *<domain IP>:<domain port>* **Mandatory:** Yes **Suggested Value:** `<domain IP>:<domain port>`
`credentials`	User name and password required to log in to the domain. The syntax is `<user name>:<encrypted password>` **Mandatory:** Yes **Suggested Value:** `<user name>:<encrypted password>`
`securedChannel`	Specifies whether the domain should be connected over a secured channel **Mandatory:** Yes **Suggested Value:** true

User Authentication in OI Hawk Console

The OI Hawk Console supports user authentication based on a file, a database, and LDAP-based authentications. You can set the authentication mode by using the OI Hawk Console configuration file (`hawkconsole.cfg`).

File-Based Authentication

For the file-based authentication, the user details are stored in the `hawkconsole-user.cfg` file. By default, the configuration file is located at *CONFIG_FOLDER/bin*. If required, you can configure its location by using the `user_file_store` option in the `hawkconsole.cfg` file.

The syntax for a user entry in the `hawkconsole-user.cfg` file is:

```
<user_name>:<encrypted_password>
```

For example,

```
admin:##########
```

You can use the `tibhawkpassword` utility at *OIHR_HOME/bin* to encrypt the password. For more details on user authentication properties, see OI Hawk Console Configuration Options.

Database-Based Authentication

In the database based authentication, the user names and passwords are stored in the database. The OI Hawk Console supports both in memory database and external database to store authentication details. For more information about configuring an external database in the OI Hawk Console, see Configuring an External Database.

You can add a new user in the external database using the following steps:

1. Add the new user in the `users` table.

For example:

```
insert into users (name, password, email, role_id) values('new_user',
'#!SXcfN3U19IiH/Eai55LWvV4XNKV/eQIDfri6+J+rho4=', 'newUser@xyz.com',1);
```

2. Create a mapping in the table `user_privilege_mapping`.

For example:

```
insert into user_privilege_mapping (user_id, privilege_id) values((select id
from users where name = 'new_user'), 1);
```

LDAP-Based Authentication

For the LDAP-based authentication, the user details are stored in the `hawkconsole.cfg` file. By default, the configuration file is located at *CONFIG_FOLDER/bin*.

For selecting LDAP as the user store, modify the `hawkconsole.cfg` file as follows:

1. Under `-M UserAuth`, specify LDAP as the user store type:

`-user_store_type ldap`

2. Under `-user_store_type ldap`, specify the LDAP-based user authentication properties.

For more details about user authentication properties that can be specified, see OI Hawk Console Configuration Options.

Secure Communication over OI Hawk Console

You can access the OI Hawk Console over a secure channel by using SSL or TLS security protocols. To enable the secure communication, uncomment and configure the following fields in the OI Hawk Console configuration file (`hawkconsole.cfg`):

- `-key_alias`
- `-key_password`
- `-key_store`
- `-key_store_password`
- `-protocol`
- `-ciphers`

For more details on these properties, see OI Hawk Console Configuration Options.

Configuring an External Database

Hawk alerts can be persisted by configuring an external database to store the alerts. If the OI Hawk Console is restarted, then also you can view the previous alerts since when the Hawk agent is active. Following databases are supported in this release:

- MySQL
- Apache Ignite
- H2 database in server and disc mode

Before you begin

- You must set up the database and perform database-specific configuration depending on the database vendor.

- Add the appropriate `.jar` file of the JDBC Driver classes, from the database vendor, to the folder `OIHR_HOME/<version>/lib/ext/console-ext`.

Procedure

To configure the external database, uncomment and configure the following fields in the OI Hawk Console configuration file (`hawkconsole.cfg`). By default the configuration file is located at *CONFIG_FOLDER/bin*.

- `-datasource_url`
- `-datasource_drivername`
- `-datasource_username`
- `-datasource_password`
- `-datasource_connection_pool_initial_size`
- `-datasource_connection_pool_max_idle`
- `-datasource_connection_pool_max_active`

For more details on these properties, see OI Hawk Console Configuration Options.

Configuring OI Hawk Console Database Schema

You can configure the attributes of the table for the database which stores the OI Hawk Console data. To do so, you must first uncomment the `sql_schema_path` configuration option in the `hawkconsole.cfg` file and then provide the folder path as a parameter. The path must contain the scripts that enable you to configure the database schema. By default, this parameter is configured with the `CONFIG_FOLDER/hawk/sql` or `OIHR_HOME/7.1/sql`.

The path specified for the `sql_schema_path` configuration option contains the following files:

Note: The OI Hawk Console uses the in memory H2 database as the default database for storing the OI Hawk Console data.

- **Schema files:** These files contain the Data Definition Language (DDL) which is used to define data structures for the database. You can configure the following files depending on the database management software currently is use:

    - `schema.sql`: Modify this file if you are using MySQL or H2 database software.
    - `schema-ignite.sql`: Modify this file if you are using the Apache Ignite database software.
- **Data files:** These files contain the Data Manipulation Language (DML) which is used to manipulate data for the database.You can configure the following files depending on the database management software currently is use:
    - `data.sql`: Modify this file if you are using MySQL or H2 database software.
    - `data-ignite.sql`: Modify this file if you are using the Apache Ignite database software.

For example, if you have configured the alerts to provide detailed information about the events, then you can configure the `sql_schema_path` option so that you can increase the size of the `alert_text` column in the alert table. This enables the OI Hawk Console to store more data for the `alert_text` than it was previously allowed to.

OI Hawk Console Configuration Options

You can configure the Hawk agent for the enterprise platforms such as Linux or Microsoft Windows. All the required configuration parameters are stored in `CONFIG_FOLDER/bin/hawkconsole.cfg`.

Each of the parameters are explained in more detail in the following table:

Hawk Console Component Configuration Options

Property	Description
`domain_config_file`	Path of file that contains the domain and transport configurations for OI Hawk Console. For details on the domain and transport configuration file, see Domain and Transport Configuration for OI Hawk Console. **Mandatory:** No **Suggested Value:** `CONFIG_FOLDER/bin/DomainTransportConfig.yml`
`server_port`	The server port to access OI Hawk Console

Property	Description
	Mandatory: No **Suggested Value:** 8083
Secure Communication (SSL Authentication) Options	
`key_alias`	Key alias **Mandatory:** No **Suggested Value:** -
`key_password`	Encrypted key password **Mandatory:** No **Suggested Value:** -
`key_store`	The path of the key store file **Mandatory:** No **Suggested Value:** -
`key_store_password`	The password for the key store file **Mandatory:** No **Suggested Value:** -
`protocol`	The security protocol for a secure communication

Property	Description
	Mandatory: No **Suggested Value:** TLSv1.2
`ciphers`	The ciphers to be used for the specified security protocol. You can specify multiple ciphers as a comma-separated list. **Mandatory:** No **Suggested Value:** `TLS_RSA_WITH_AES_128_CBC_SHA`
`repository_path`	The path to the rulebase repository **Mandatory:** No **Suggested Value:** `CONFIG_FOLDER/hawk/repository`
`subscription_queue_size`	A bounded circular queue is maintained for each subscription for storing its results. This parameter defines the maximum size of the queue. If the maximum size of the queue is reached then old results are overridden by the new ones. **Mandatory:** No **Suggested Value:** 128
`subscription_expiry_time`	Time (in milliseconds) after which a subscription expires if the results of subscription are accessed **Mandatory:** No **Suggested Value:** 90000
Proxy Domain Options	

Property	Description
`proxy_alert_count_pull_interval`	Time interval (in milliseconds) in which the alert count is fetched from proxy domains **Mandatory:** No **Suggested Value:** 15000
`proxy_domain_reachability_check_interval`	Time interval (in milliseconds) in which proxy domains are checked for reachability **Mandatory:** No **Suggested Value:** 15000
Alert Configurations	
`retention_count_for_notification`	Alert limit for notifications **Mandatory:** Yes **Suggested Value:** 100000
`retention_count_for_high_alerts`	Alert limit for high level alerts **Mandatory:** Yes **Suggested Value:** 100000
`retention_count_for_medium_alerts`	Alert limit for medium level alerts **Mandatory:** Yes **Suggested Value:** 100000

Property	Description
retention_count_for_low_alerts	Alert limit for low level alerts **Mandatory:** Yes **Suggested Value:** 100000
alert_manager_activity_interval	Time interval (in milliseconds), after which the alert manager starts to store alerts in the database and purge extra alerts in the database **Mandatory:** Yes **Suggested Value:** 20000
max_reconnect_attempts_after_restart	Specifies the number of reconnect attempts to be made when the agent gets disconnected from the Daemon **Mandatory:** No **Suggested Value:** 1000
max_reconnect_attempts_during_connect	Specifies the number of reconnect attempts made when the connection is disconnected from the Daemon after it has been established **Mandatory:** No **Suggested Value:** 20
External Database Configuration	
datasource_url	URL which identifies the database connection **Mandatory:** No **Suggested Value:** "jdbc:h2:mem:test;DB_CLOSE_DELAY=-1"

Property	Description
`datasource_drivername`	Name of the JDBC driver **Mandatory:** No **Suggested Value:** "`org.h2.Driver`"
`datasource_username`	User name to connect to the database **Mandatory:** No **Suggested Value:** "`sa`"
`datasource_password`	User's password to connect to the database **Mandatory:** No **Suggested Value:** ""
`datasource_connection_pool_initial_size`	Initial number of database connections to be allocated **Mandatory:** No **Suggested Value:** "10"
`datasource_connection_pool_max_idle`	Maximum number of idle connections allowed in the database connection pool **Mandatory:** No **Suggested Value:** "20"
`datasource_connection_pool_max_active`	Maximum number of active connections allowed in the database connection pool

Property	Description
	Mandatory: No **Suggested Value:** "100"
Logging	
`log_dir`	The directory in which to store log files generated by the Hawk agent **Mandatory:** No **Suggested Value:** *CONFIG_FOLDER/logs*
`log_max_size`	The maximum size of a rotating log files in KB. You can apply a suffix '`m`' or '`M`' for indicating MB values **Mandatory:** No **Suggested Value:** 10M
`log_max_num`	The maximum number of rotating log files **Mandatory:** No **Suggested Value:** 10
`log_level`	Specifies the level of diagnostic information stored in the logs. The following are the logging levels: • `4` - Indicates error level trace messages should be enabled. • `6`- Indicates warning level trace messages should be enabled. • `7` - Indicates information level trace messages should be enabled. • `8` - Indicates debug level trace messages should be enabled. • `16` - Indicates AMI level trace messages should be enabled.

Property	Description
	• A value of zero turns all tracing off. • A value of `-1` turns all tracing on. **Mandatory:** No **Suggested Value:** 7
`log_format`	The format for trace log messages **Mandatory:** No **Suggested Value:** ae4
User Authentication	
`user_store_type`	Specify whether OI Hawk Console uses an inbuilt database or a file for user authentication. The values are: • `database` - In the `database` based configuration, the user names and passwords are stored in the database. • `file` - In the `file` based configuration, the user names and passwords are stored in a file in the disk. Specify the location of the user authentication file in the `-user_file_store` property. • `ldap` - In the `ldap` based configuration, the user names and passwords are validated with a LDAP directory server. **Mandatory:** No **Suggested Value:** `file`
`user_file_store`	If `-user_store_type` is `file`, specify the path of the file which stores user details for authentication

Property	Description
	Mandatory: No **Suggested Value:** `CONFIG_FOLDER/bin/hawkconsole-users.cfg`
User Authentication: LDAP-Based	
`ldap_host`	Host name for the LDAP server **Mandatory:** Yes **Suggested Value:** -
`ldap_port`	Port of the LDAP server **Mandatory:** No **Suggested Value:** 389
`ldap_adminDN`	LDAP manager user DN for accessing the server, to avoid anonymous access to the server **Mandatory:** Yes **Suggested Value:** -
`ldap_admin_password`	LDAP admin password for accessing the server. The password is obfuscated. **Mandatory:** Yes **Suggested Value:** -
`ldap_baseDN`	Base DN for the users to search

Property	Description
	Mandatory: Yes **Suggested Value:** -
`ldap_uid_attr`	UID attribute to perform the user search **Mandatory:** No **Suggested Value:** UID
`ldap_groupDN`	Group DN for the users to log in who belong to a particular group. Multiple groups can be specified by separating the string with the pipe(|) operator. **Mandatory:** No **Suggested Value:** -
`ldap_group_ search_filter`	Query attribute to search in group **Mandatory:** No **Suggested Value:** `'memberOf'`
`ldap_ssl_ enabled`	Specifies whether to connect to LDAP over SSL or not **Mandatory:** No **Suggested Value:** false
`ldap_disable_ hostname_ verification`	Specifies whether the host name verification is enabled or disabled **Mandatory:** No

Property	Description
	Suggested Value: false
`hawkconsole_user_access_list`	Specifies the file path for user based agent/node filtering **Mandatory:** No **Suggested Value:** `CONFIG_FOLDER/bin/hawkconsole_user_access_list.cfg`

Hawk Event Service Configurations

You can install the TIBCO Hawk Event Service in an enterprise environment by selecting the Custom Installation profile and then selecting the **Event Service** feature while installing TIBCO OI Hawk RedTail. The TIBCO Hawk Event Service is a separate process that collects information about Hawk agents. Because it runs independently from other Hawk processes, the Event Service can detect and report the event even when an agent process fails.

The Hawk Event Service can communicate with the Hawk agent only if the Hawk agent is running on an enterprise platform such as Linux or Microsoft Windows. The Hawk Event Service must be configured by using TIBCO EMS, TIBCO Rendezvous or TCP transport for TIBCO Hawk. You must also configure the Hawk agent (`hawkagent.cfg`) to communicate over TIBCO EMS, TIBCO Rendezvous or TCP transport for TIBCO Hawk so that the Hawk agent can communicate with the Hawk Event Service.

The main tasks of the Event Service are:

- Record events reported by agents in text files or relational databases using JDBC
- Detect and respond to agent termination
- Asynchronous notifications using AMI

The Hawk Event Service records:

- All alerts raised and cleared by the Hawk agents across the network, as well as the changes in Agent’s alert level
- Record events reported by agents in text files or relational databases using JDBC
- Asynchronous notifications using AMI

Persistence of TIBCO Hawk Events using JDBC

All alerts generated and cleared by Hawk agents across the network, as well as agent activation and expiration events, are written to a relational database using JDBC. Data is stored in two separate tables, created automatically at startup (if they are not already present in the specified database):

- **HawkAgentInfo:** The events `onAgentAlive`, `onAgentExpired`, `onMicroAgentAdded`, `onMicroAgentRemoved`, `onRulebaseAdded`, and `onRulebaseRemoved` add rows to this table.
- **HawkAlertClearInfo:** Events `onAlert` and `onClear` add rows to this table.

Fault Tolerance

To enable fault tolerance, uncomment the `ft` parameter.

This instance joins a fault tolerant group named `HawkEventService:hawkdomain`, where *hawkdomain* is the domain of the agent.

Note: Separate instances of TIBCO OI Hawk RedTail must be running on at least two machines in order to use fault tolerance. Fault tolerance must be enabled on each instance.

Weight

Assign the weight of this instance using a positive integer. The member with the highest weight receives rank 1 (so it outranks all other members). When an instance fails, the next-highest instance is activated and the member with the next highest weight receives rank 2; and so on.

Be careful not to confuse TIBCO Hawk Event Service data files (`Event.dat`) with Event Service log files (`Event.log`).

- `Event.dat` data files contain the data produced by the Event Service.
- `Event.log log` files record the state of the Event Service itself.

You can configure the Hawk Event Service for the enterprise platforms such as Linux or Microsoft Windows. All the required configuration parameters are stored in *CONFIG_FOLDER*`/bin/hawkevent.cfg`.

Each of the parameters are explained in more detail in the following table:

Hawk Event Service Configuration Details

Property	Description
`hawk_domain`	The Hawk domain name. The agents and the event service must have the same hawk domain value in order to communicate. **Mandatory:** Yes **Suggested Value:** "default"
`agent_name`	The name of the agent to communicate **Mandatory:** Yes **Suggested Value:** Host name
TIBCO Rendezvous Transport	
`rvd_session`	Comment this option if you are using TCP Transport for TIBCO Hawk or TIBCO EMS as the primary transport. The value of this parameter must be the same as the value of the `ami_rvd_session` parameter in the Hawk Agent configuration file (`hawkagent.cfg`). The format is `rvd_session <service> <network> <daemon>`. If you use this option, all three parameters must be present and separated by white space. Use a semicolon (;) to indicate a null value, or use an empty string, for example: `-rvd_session 7474 "" tcp:7474` **Mandatory:** No **Suggested Value:** `7474 "" tcp:7474`

Property	Description
TCP Transport for TIBCO Hawk	
`tcp_session`	Set this option to configure the TCP Transport for TIBCO Hawk as the primary transport for the communication. The syntax of the property is: `-tcp_session` *<self_IP>:<port> <HAWKCONSOLE_IP_ADDRESS>:<port>* where, • *<self_IP>:<port>* - The unique socket address of the Hawk Event service for joining the cluster. • *<HAWKCONSOLE_IP_ADDRESS>:<port> - The IP address of instance running OI Hawk Console.* **Note:** Multiple agents/OI Hawk Console running on the same instance must be bound to separate ports. For example, if hawkagent1 binds to port 2551, then hawkagent2 can use port 2552 or any port other than 2551. **Mandatory:** No **Suggested Value:** `localhost:2582 localhost:2561`
TCP Transport for TIBCO Hawk SSL Parameters The following TLS/SSL parameters are applicable to `tcp_session` and `ami_tcp_session`.	
`tcp_key_store`	Path of the key store file **Mandatory:** No **Suggested Value:** -
`tcp_trust_store`	Path of the trust store file

Property	Description
	Mandatory: No **Suggested Value:** -
`tcp_key_store_password`	Password for the key store file **Mandatory:** No **Suggested Value:** -
`tcp_key_password`	Encrypted key password **Mandatory:** No **Suggested Value:** -
`tcp_trust_store_password`	Password for the trust store file **Mandatory:** No **Suggested Value:** -
`tcp_ssl_protocol`	Protocol for a secure connection **Mandatory:** No **Suggested Value:** TLSv1.2
`tcp_enabled_algorithms`	Algorithm to be used for the security protocol. You can specify multiple algorithms as a comma-separated list without space. **Mandatory:** No

Property	Description
	Suggested Value: `TLS_RSA_WITH_AES_128_CBC_SHA`
`max_reconnect_attempts_after_restart`	Specifies the number of reconnect attempts to be made when the agent gets disconnected from the Daemon **Mandatory:** No **Suggested Value:** 1000
`max_reconnect_attempts_during_connect`	Specifies the number of reconnect attempts made when the connection is disconnected from the Daemon after it has been established **Mandatory:** No **Suggested Value:** 20
`transport_timeout`	The default timeout used by transport for internal invocations **Mandatory:** No **Suggested Value:** 30000
TIBCO EMS Transport	
`ems_transport`	Comment this option if you are using TCP Transport for TIBCO Hawk or TIBCO Rendezvous as the primary transport. Specifies location of EMS server. For example, `ems_transport tcp://server1:7222`. **Note:** If EMS is configured as Transport, the `ami_rvd_session` parameter must be configured.

Property	Description
	Note: When using encrypted password generated using `tibhawkpassword`, the password must be placed within double quotation marks ("). **Mandatory:** No **Suggested Value:** -
TIBCO EMS SSL Parameters (In case EMS Server is configured for SSL communication)	
`ssl_vendor`	The name of the vendor of the SSL implementation. On IBM platforms (such as AIX), this option defaults to ibm. The valid choices are • j2se-default: Use this option when you want to use the default JCE bundled with the Java JRE. • entrust61: Use this option when you want to use the Entrust libraries. • ibm: On non-IBM platforms, this option can be used only if the IBM version of JCE is installed. **Mandatory:** No **Suggested Value:** j2se
`ssl_ciphers`	Cipher suite name. Use circumflex (^) instead of hyphen (-) when specifying `-ssl_ciphers` **Mandatory:** No **Suggested Value:** -
`ssl_no_verify_host`	Indicates not to verify the EMS server

Property	Description
	Mandatory: No **Suggested Value:** -
`ssl_trusted`	File name of the server certificates. The file must be accessible locally/ shared drive **Mandatory:** No **Suggested Value:** -
`ssl_no_verify_hostname`	Indicates not to verify the name in CN field of the server certificate **Mandatory:** No **Suggested Value:** -
`ssl_expected_hostname`	If the `ssl_no_verify_host` is not specified, the option `ssl_trusted` has to be used. Along with the option `ssl_trusted`, specify either `ssl_no_verify_hostname` or `ssl_expected_hostname`. **Mandatory:** No **Suggested Value:** -
`ssl_identity`	Digital certificate **Mandatory:** No **Suggested Value:** -

Property	Description
`ssl_private_key`	Private key **Mandatory:** No **Suggested Value:** -
`ssl_password`	Password **Mandatory:** No **Suggested Value:** -
`character_encoding`	Character encoding to be used across the configured transport **Mandatory:** No **Suggested Value:** UTF-8
AMI Session Configurations	
`ami_rvd_session`	Specifies the TIBCO Rendezvous session used by the Hawk Event Service for AMI communications. If this option is used, all three parameters must be present and separated by whitespace but any of them can be an empty string to indicate a null value. **Note:** The default `ami_rvd_session` uses `rvd_session` values. For example: `-rvd_session 7474 127.0.0.1 tcp:7474` **Mandatory:** No **Suggested Value:** `7474 "" tcp:7474`

Property	Description
`ami_tcp_session`	The AMI TCP session needs self address (different from `tcp_session`) and the AMI address of the agent that is supposed to detect `hawkEventService` microagent. The syntax for the property is: `-ami_tcp_session` *<self_IP>:<port> <hawk_agent_IP>:<AMI_session_port>* where, • *<self_IP>:<port>* - The unique socket address of the Hawk Event service for AMI communication. The socket address should be different from the *<self_IP>:<port>* specified for the `-tcp_session` property of Hawk Event Service. • *<hawk_agent_IP>:<AMI_session_port>* - The socket address of the Hawk agent for AMI communication. This socket address is the same as *<self_IP>:<port>* specified for the `-ami_tcp_session` parameter in `hawkagent.cfg`. **Mandatory:** No **Suggested Value:** `localhost:2575 localhost:2571`
`max_reconnect_attempts_after_restart_for_ami`	Specifies the number of reconnect attempts to be made when the AMI gets disconnected from the Hawk agent **Mandatory:** No **Suggested Value:** 1000
`max_reconnect_attempts_during_connect_for_ami`	Specifies the number of reconnect attempts made when the connection is disconnected from the Hawk agent after it has been established **Mandatory:** No

Property	Description
	Suggested Value: 20
Logging	
`logdir`	The directory in which to store log files generated by the Hawk Event Service **Mandatory:** No **Suggested Value:** *CONFIG_FOLDER*`/logs`
`logmaxsize`	The maximum size of a rotating log files in kilobytes **Mandatory:** No **Suggested Value:** 10M
`logmaxnum`	The maximum number of rotating log files **Mandatory:** No **Suggested Value:** 10
`log_level`	Specifies the level of diagnostic information stored in the logs. The following are the logging levels: 4 - Indicates error level trace messages should be enabled 6- Indicates warning level trace messages should be enabled 7 - Indicates information level trace messages should be enabled 8 - Indicates debug level trace messages should be enabled 16 - Indicates AMI level trace messages should be enabled A value of zero turns all tracing off.

Property	Description
	A value of -1 turns all tracing on. **Mandatory:** No **Suggested Value:** 7
`log_format`	The format for trace log messages **Mandatory:** No **Suggested Value:** "default"
Data event	
`script`	Specifies the fully-qualified name of an executable file to be executed when an agent is lost **Mandatory:** No **Suggested Value:** -
`datamaxnum`	The maximum number of rotating data files **Mandatory:** Yes **Suggested Value:** 5
`datamaxsize`	The maximum size of a rotating data file in KB. You can apply a suffix 'm' or 'M' for indicating MB values **Mandatory:** Yes **Suggested Value:** 1024

Property	Description
`datadir`	The directory in which the data files generated by the TIBCO Hawk Event are stored **Mandatory:** Yes **Suggested Value:** *CONFIG_FOLDER*`/data`
Fault Tolerance	
`ft`	Fault tolerance weight for `TibrvFtMember` **Mandatory:** No **Suggested Value:** no fault tolerance
`ft_rvd_session`	TIBCO Rendezvous session used for fault tolerance. This option is ignored if the `-ft` option is not specified. **Mandatory:** No **Suggested Value:** `7474 "" tcp:7474`
`character_encoding`	Specifies the character encoding to be used for strings sent over all TIBCO Rendezvous transport **Mandatory:** No **Suggested Value:** UTF-8
Database Based Event Store	
`JDBCdriverClassName`	Class name for the vendor's JDBC driver. For example, `com.microsoft.jdbc.sqlserver.SQLServerDriver`

Property	Description
	Mandatory: No **Suggested Value:** -
`JDBCuserName`	User name to connect to the database **Mandatory:** No **Suggested Value:** -
`JDBCpassword`	User's password to connect to the database **Mandatory:** No **Suggested Value:** -
`JDBCurl`	URL which identifies the database connection **Mandatory:** No **Suggested Value:** -
`JDBCdbType`	Database vendor type. Supported values are ORACLE, SQLSERVER, DB2 or SYBASE. **Mandatory:** No **Suggested Value:** -
`JDBCalertTableFields`	User defined alert action property fields that must be created as additional columns in the `HawkAlertClearInfo` table

Property	Description
	Mandatory: No **Suggested Value:** -

Database Configuration

To setup the database, add the appropriate `.jar` file of the JDBC driver classes, from the database vendor, to the `OIHR_HOME/lib` based on the value of the `-JDBCdbType` parameter.

-JDBCdbType Value	Required .jar Files
ORACLE	ojdbc6.jar
SQLSERVER	sqljdbc.jar sqljdbc4.jar
DB2	jconn3.jar
SYBASE	db2jcc4.jar

Hawk Cluster Manager Configurations

You can optionally install and configure the Admin Agent as a console to monitor the Hawk agents. If you plan to use TCP transport for Admin Agent to Hawk Agent communication, then you must configure the Hawk Cluster Manager. All the required configuration parameters are stored in the *CONFIG_FOLDER*/hawktcpdaemon.cfg file.

Each of the parameters are explained in more detail in the following table:

Hawk Cluster Manager Configuration Details

Property	Description
TCP Transport for TIBCO Hawk	

Property	Description
`tcp_session`	Setups the TCP session for the Hawk components. Hawk Cluster Manager joins the TCP Transport for TIBCO Hawk cluster as seed node using this property. The syntax of the property is: `-tcp_session` *<self_IP>:<port> <cluster_manager_IP>:<port>* where, • *<self_IP>:<port>* - Unique socket address of the Hawk agent for connecting to the cluster. • *<cluster_manager_IP>:<port>* - The socket address of the Cluster Manager acting as the seed node for the cluster. In case there is only one daemon in the cluster then both the socket addresses (self and daemon) are the same. **Note:** Multiple agents/console/clustermanagers running on the same instance must be bound to separate ports. For fault tolerance, you can specify multiple seed daemon's socket addresses in a comma-separated list. `-tcp_session` *<self_IP>:<port> <daemon1_IP>:<port>, <daemon2_IP>:<port>* **Mandatory:** Yes **Suggested value:** `localhost:2561 localhost:2561`
TCP Transport for TIBCO Hawk SSL Parameters	
`tcp_key_store`	Path of the key store file **Mandatory:** No

Property	Description
	Suggested value: -
`tcp_trust_store`	Path of the trust store file **Mandatory:** No **Suggested value:** -
`tcp_key_store_password`	Password for the key store file **Mandatory:** No **Suggested value:** -
`tcp_key_password`	Encrypted key password **Mandatory:** No **Suggested value:** -
`tcp_trust_store_password`	Password for the trust store file **Mandatory:** No **Suggested value:** -
`tcp_ssl_protocol`	Protocol for a secure connection **Mandatory:** No **Suggested value:** TLSv1.2
`tcp_enabled_algorithms`	Algorithm to be used for the security protocol. You can specify multiple algorithms as a comma-separated list without space

Property	Description
	Mandatory: No **Suggested value:** `TLS_RSA_WITH_AES_128_CBC_SHA`
`strategy`	Specifies the strategy for resolving network partitions of the cluster members from the TCP Transport for TIBCO Hawk cluster. The two strategies that you can choose are: • **Quorum** - This strategy defines the minimum number of daemons required for a cluster to be operational (`quorum size`). In case of network partition, the partition with the required quorum size remains operational while the other partition is shut down. • **Majority** - If the network partition occurs then the partition that has the majority of nodes remains operational while the other partition is shut down. **Mandatory:** No **Suggested value:** Quorum
`hawk_domain`	Specifies the Hawk domain name. The Hawk agents, Hawk cluster managers, and the console applications must have the same hawk domain value in order to communicate. **Mandatory:** No **Suggested value:** default
Logging	
`log_dir`	The directory to store log files generated by the TIBCO Hawk Cluster Manager **Mandatory:** No

Property	Description
	Suggested value: *CONFIG_FOLDER*/logs
log_max_size	The maximum size of a rotating log files in kilobytes. You can apply the suffix 'm' or 'M' for indicating MB values **Mandatory:** No **Suggested value:** 10M
log_max_num	The maximum number of rotating log files **Mandatory:** No **Suggested value:** 10
log_level	Specifies the level of diagnostic information stored in the logs. The following are the logging levels: • **4** - Indicates error level trace messages should be enabled. • **6** - Indicates warning level trace messages should be enabled. • **7** - Indicates information level trace messages should be enabled. • **8** - Indicates debug level trace messages should be enabled. • **16** - Indicates AMI level trace messages should be enabled. • **0** - A value of zero turns OFF all tracing. • **-1** - A value of -1 turns ON all tracing. **Mandatory:** No **Suggested value:** 7
log_format	The format for trace log messages

Property	Description
	Mandatory: No **Suggested value:** "default"

For more information about the Admin Agent and its configurations, see *TIBCO® Operational Intelligence Hawk® RedTail Admin Agent*.

Configuring TIBCO OI Hawk RedTail in Compatibility Mode

After the successful installation of TIBCO OI Hawk RedTail, perform the following steps:

Before you begin

- Ensure that you have installed and configured the OI Hawk Console component.
- (Optional) Ensure that you have installed and configured the Hawk Event Service.

On Windows

1. Open *TIBCO_HOME* and ensure that the following folders exist under *TIBCO_HOME*:
   - *TIBCO_HOME/hawk/<version>*
   - *TIBCO_HOME/tibcojre64*
2. Start the following TIBCO OI Hawk RedTail components:
   Start the enterprise Hawk Agent using one of the following methods:
   - Click **Start** > **All Programs** > **TIBCO** > *OIHR_HOME* > **TIBCO Hawk** <*version*> > **Start Hawk Agent**.
   - Double-click `tibhawkagent` from *CONFIG_FOLDER\bin*.

   Start the enterprise Hawk microagent using one of the following methods:
   - Click **Start** > **All Programs** > **TIBCO** > *OIHR_HOME* > **TIBCO Hawk** <*version*> > **Start Hawk Microagent**.
   - Double-click `tibhawkhma` from *CONFIG_FOLDER\bin*.

   Start the Hawk Event Service (if needed) using one of the following methods:
   - Click **Start** > **All Programs** > **TIBCO** > *OIHR_HOME* > **TIBCO Hawk** <*version*> > **Start Hawk Event**.
   - Start Hawk Event by double clicking `tibhawkevent` from *CONFIG_FOLDER\bin*.

3. Start OI Hawk Console by using either of the following methods:
   - Click **Start** > **All Programs** > **TIBCO** > OIHR_HOME > **TIBCO Hawk** <*version*> > **Start Hawk Console**.
   - Double-click `tibhawkconsole` from *CONFIG_FOLDER\bin*.
4. Access the OI Hawk Console by using the following URL:

   http://<address>:<port_number>/HawkConsole

   where the default <*port_number*> is 8083.

   For example, `http://localhost:8083/HawkConsole`

   In the login window, enter a valid user name and password. The default credentials are:

   - Username: **admin**
   - Password: **admin**

On UNIX/ Linux

Procedure

1. Open *TIBCO_HOME* and ensure that the following folders exist under *TIBCO _HOME*:
   - *TIBCO_HOME/hawk/<version>*
   - *TIBCO_HOME/tibcojre64*
2. Start the following TIBCO OI Hawk RedTail components:

   Start the enterprise Hawk agent by executing `tibhawkagent` from *CONFIG_FOLDER\bin*.

   Start the enterprise Hawk microagent by executing `starthma`. The `starthma` must be run as root.

   Start the Hawk Event Service (if needed) by executing `tibhawkevent` from *CONFIG_FOLDER\bin*.

   Start the OI Hawk Console by executing `tibhawkconsole` from *CONFIG_FOLDER\bin*.
3. Access the OI Hawk Console by using the following URL:

   http://<address>:<port_number>/HawkConsole

   where the default <*port_number*> is 8083.

For example, `http://localhost:8083/HawkConsole`

In the login window, enter a valid user name and password. The default credentials are:

- Username: **`admin`**
- Password: **`admin`**

TIBCO Hawk Security Model

This section discusses the security models for OI Hawk Console.

Note: This section is applicable only when the Hawk agents are being monitored with the OI Hawk Console.

Trusted Security Model

OI Hawk Console uses the Trusted Security model to guarantee that only authorized users can perform restricted actions.

The Trusted model uses an ASCII file as a simple yet effective entitlement server. This has the benefit of being easily distributed to all nodes, making it a very scalable mechanism. A scan of the ASCII file for information about the user determines if the request is granted.

Users are explicitly granted or denied access through the access control file. A user who is not in this file is not allowed to perform any operations on the TIBCO Hawk system. Access control information is in a plain ASCII file located in the *OIHR_HOME*\hawk\<version>\examples\security folder.

Copy this file to *CONFIG_FOLDER**security* manually. See Access Control File, for more details.

Trusted Model

The Trusted model provides a simple yet effective mechanism for addressing authorization concerns. It addresses security issues as follows:

- **Authentication**: The Trusted security model does not guarantee the authenticity of the request.
- **Integrity**: The Trusted security model does not guarantee the integrity of the request.
- **Authorization**: The Trusted security model guarantees that only authorized users can perform restricted actions.
- **Privacy**: The Trusted security model does not address the privacy of the request. All requests are sent using plain text.

Authorization at the Console API Layer

The Trusted model uses an ASCII file as a simple yet effective entitlement server. This has the benefit of being easily distributed to all nodes, making it a very scalable mechanism. A scan of the ASCII file for information about the user determines if the request is granted.

Users are explicitly granted or denied access through the access control file. A user who is not in this file is not allowed to perform any operations on the OI Hawk Console. Access control information is in a plain ASCII file located in the *OIHR_HOME*`/examples/security` folder.

Copy this file to *CONFIG_FOLDER/security* manually. See Access Control File, for more details.

Authorization at the REST API Layer

You can configure agent level authorization in OI Hawk Console. The user can only view the agents and the domains for which he has required privileges. This access control is also applicable over the statistics displayed on domain cards, i.e. the statistics (alert counts) displayed in domain card view displays the summation of agents based on the user privileges.

The authorization is implemented with the help of the configuration file `hawkconsole_user_access_list.cfg`.

Make sure that you have performed the following tasks before implementing this authorization mechanism:

- Uncomment the `hawkconsole_user_access_list` option in the `hawkconsole.cfg` file. This enables the OI Hawk Console to identify the users and their privileges based on the file path provided for that configuration option.
- Ensure that the agent for which you want to implement this authorization belongs to a domain with a trusted security model.

Example of `hawkconsole_user_access_list.cfg` file

```
#
# This file is used by agent running with
"COM.TIBCO.hawk.security.trusted.Trusted" security model.
#
#
# Explanation of Settings:
#
# This file provides authorization or filter level for node/agent for
WebConsole application.
# This doesn't include microagent and methods level authorization, even
if the microagent and methods are defined, then those will be ignored.
# The node column can have a node/agent name or "agent dns domain" or
"agent:dns:domain" format.
# Wild card * is supported in both user and node column.
# Access restrictions can be defined for an user by starting record with
!
# In case of any conflict in grant and restriction, the restriction will
have the precedence.
#      Examples
#             1: admin will have access to all agents/nodes and domains
#              2. user1 will have access to agent1 under domain1 with dns dns1.
#              3. user2 will have access to all agents/nodes under domain domain2
and any dns.
#              4. user3 will have access to agent3 if agent3 doesn't belong to
domain3
#
# File format:
#
# user       node
#            access
#             &
#          restrictions
#
# admin               *
admin          *
user1          "agent1 dns1 domain1"
user2          "* * domain2"
```

```
user3       agent3
!user3         "* * domain3"
```

Logging

All trusted requests (both Trusted and `TrustedWithDomain`) can be logged to rolling log files in a directory of your choice.

The current log file is named `Trusted.log`. When it reaches the maximum size (`size`), it is closed and renamed `Trusted`*`1`*`.log`, and a new `Trusted.log` is started. When the number of logs exceeds the maximum (`n`), log entries roll over to reuse the oldest log file.

To activate logging, add the following line to the access control file:

```
<LogService> -log_dir <logDir> -log_max_size <size> -log_max_num <n>
```

where:

Option	Description
`logDir`	The directory where the log file is saved. Make sure this directory exists before you activate logging.
`size`	The maximum size of the rolling log file in KB. The suffix `m` or `M` can be used for indicating MB.
`n`	The maximum number of rolling log files.

Example Log File Entries

This is an example log entry for an authorized request:

```
Tue Dec 31 11:14:13 EST 2002: Trusted operation: userID=HAWK-
TRUSTDMN\hawkuser, node=hawkuser-DT:none:default,
microagent=COM.TIBCO.hawk.microagent.SysInfo, method=
getOperatingSystem.
```

This is an example of an entry for an unauthorized request:

```
Tue Dec 31 11:19:54 EST 2002: Trusted operation: userID= HAWK-
TRUSTDMN\hawkuser, node=hawkuser-DT:none:default, microagent=
COM.TIBCO.hawk.microagent.Repository, method= getRBMap - permission
denied.
```

Using both Trusted and TrustedWithDomain

An agent using the `Trusted` or `TrustedWithDomain` security model allows users with either `Trusted` or `TrustedWithDomain` to access the agent.

- To allow access to a user who starts OI Hawk Console (or the Console API application) in Trusted security mode, the entry for *<user>* specified in the agent's `Trusted.txt` or `TrustedWithDomain.txt` should not include the domain of the user who actually starts OI Hawk Console (or the Console API application).
- To allow access to a user who starts OI Hawk Console in `TrustedWithDomain` security mode, the entry for *<user>* specified in the agent's `Trusted.txt` or `TrustedWithDomain.txt` should include the domain of the user who actually starts OI Hawk Console.

To Use the Trusted Model

Two sample access control files are included with TIBCO OI Hawk RedTail.

- `Trusted.txt` can be used on UNIX or Microsoft Windows. It is used when the command line specifies `Trusted`.
- `TrustedWithDomain.txt` is for use on Microsoft Windows only, and is used when the command line specifies `TrustedWithDomain`.

The access control files, `Trusted.txt` and `TrustedWithDomain.txt`, are described in the next section.

To use the Trusted model:

If you have multiple Hawk agents running on a machine and these Hawk agents, in turn, belong to different Hawk domains, you can specify separate access control files for each domain.

1. For each Hawk domain, create a directory:

CONFIG_FOLDER/hawk/domain/<domain-name>/security where <*domain-name*> is the name of the Hawk domain.

2. Provide a remote `Trusted.txt` file to configure a security URL on Agent,
   — add/append the following system parameter to `java.extended.properties` in `tibhawkagent.tra`

   ```
   -Dhawk.security_file_url=file:///D:/temp/Trusted.txt
   ```

   Or

   ```
   -Dhawk.security_file_url=http://<hostname:port>/Trusted.txt
   ```

   The Agent always gives precedence to the local file, if found in the *hawk/domain* folder.
3. Modify the appropriate sample access control file, `Trusted.txt` or `TrustedWithDomain.txt`, according to the requirements of your system.
4. Save the modified file in the directory you created, without changing the file name. The program automatically searches for the access control file in this directory.
5. Ensure that the `security_policy` parameter in Hawk agent configuration is set to one of the following, before starting TIBCO Hawk Agent and OI Hawk Console:

   ```
   COM.TIBCO.hawk.security.trusted.Trusted
   or
   COM.TIBCO.hawk.security.trusted.TrustedWithDomain
   ```

The Trusted model is now in effect. The security policy stays in force as long as the process is running.

Access Control File

To store access control information, the Trusted model uses an ASCII file. Two sample access control files are included with TIBCO OI Hawk RedTail: `Trusted.txt` and `TrustedWithDomain.txt`.

Sample access control files are shipped with the TIBCO OI Hawk RedTail software, in the directory *OIHR_HOME/examples/security/*.

Trusted.txt

This access control file can be used with UNIX or with Microsoft Windows XP. The user for authorization is the login ID of the OI Hawk Console owner.

TrustedWithDomain.txt

This file can only be used with Microsoft Windows XP, and only when specified in the command used to start TIBCO Hawk agent and OI Hawk Console, as in `-security_policyCOM.TIBCO.hawk.security.trusted.TrustedWithDomain`.

The user is the login ID and the domain where the user is logged on. For example, for user1 in domainX, the user is `<domainX>\user1`.

Group Operations

A group operation effectively performs a method invocation simultaneously on all of the specified target microagents. It is useful for affecting a group of microagents in a single operation. There are two kinds of group operation: network query and network action.

Wildcards characters + and * affect permissions on group operations and point-to-point invocations as shown in the following section.

- Use + in node access to allow access to group operations.
- Use * in node access to allow access to point-to-point invocations.
- Use + in method access to allow access to all INFO and ACTION methods.
- Use * in method access to allow access.

Access Control File Conventions

The access control file uses the following conventions to grant or deny access.

- Explicit access for a particular resource implicitly denies access to all other resources in the same class. The defined classes are nodes, microagents, and methods.
- Explicit restriction for a particular resource implicitly allows access to all other resources in the same class, provided they have been explicitly granted. The defined classes are nodes, microagent, and methods.
- Permissions always default to the most restrictive case.

File Settings for the Trusted Model

This table presents how individual restrictions and permissions are defined for nodes, microagents, and methods. Individual node, microagent, and method names can be specified. In addition, wildcard characters can be used as shown in the table.

Each individual setting is represented by one line in the access control file. Complex permissions and restrictions can be defined using sets of related lines. For example, you can give a user access to all methods on a node in one line, then in the following line, restrict that user's access to one of those methods. See Disable Custom Microagent, for further details.

Permissions are granted to a user using the user name. Restrictions are defined by prefixing a bang (`!`) character to the user name, as shown in the table.

Access Control File Settings

Effect	User	Node	Microagent	Method
Full Access Grants full access to all methods on all microagents on all nodes, including group operations.	*<user>*			
Full Restriction Denies access to all methods on all microagents on all nodes, including group operations	! *<user>*			
Node Access: All Nodes Grants point-to-point and group operation invocation access to all methods on all microagents.	*<user>*	+		
Node Access: All Nodes Grants point-to-point invocation access to all methods on all microagents. Does not grant group operation invocation access.	*<user>*	*		

Effect	User	Node	Microagent	Method
Node Access: Named node Grants invocation access to all methods on all microagents on the named node. You can add several lines for one user to provide access to a set of nodes.	<*user*>	<*node*>		
Node Restriction: All Nodes Denies point-to-point and group operation invocation access to all methods on all microagents.	! <*user*>	*		
Node Restriction: All Nodes Denies group operation invocation access to all methods on all microagents. (Does not deny point-to-point operation invocations.)	! <*user*>	+		
Node Restriction: Named node Denies invocation access to all methods on all microagents on the named node. You can add several lines for one user to provide access to a set of nodes.	! <*user*>	<*node*>		
Microagent Access Grants access to all methods on the specified microagent. Wildcard characters can be used in place of a specific node name. See *Node Access*.	<*user*>	<*node*>	<*microagent*>	
Microagent Restriction Denies access to all methods on the specified microagent.	! <*user*>	<*node*>	<*microagent*>	

Effect	User	Node	Microagent	Method
Wildcard characters can be used in the Node columns. See *Node Restriction* above.				
Method Access Grants access to all `ACTION` and `INFO` methods on the specified microagent (but not `ACTIONINFO` methods). Wildcard characters can be used in the Node and Microagent columns.	<*user*>	<*node*>	<*microagent*>	+
Method Access Grants access to all `INFO` methods on the specified microagent (but not `ACTION` or `ACTIONINFO` methods). Wildcard characters can be used in the Node and Microagent columns.	<*user*>	<*node*>	<*microagent*>	*
Method Access Grants access to the specified method on the specified microagent. Wildcard characters can be used in the Node and Microagent columns.	<*user*>	<*node*>	<*microagent*>	<*method*>
Method Restriction Denies access to all methods on the specified microagent. Wildcard characters can be used in the Node and Microagent columns.	! <*user*>	<*node*>	<*microagent*>	*
Method Restriction Denies access to all `ACTION` and `ACTION_`	! <*user*>	<*node*>	<*microagent*>	+

Effect	User	Node	Microagent	Method
`INFO` methods on the specified microagent. Wildcard characters can be used in the Node and Microagent columns.				
Method Restriction Denies access to the specified method on the specified microagent. Wildcard characters can be used in the Node and Microagent columns.	! <*user*>	<*node*>	<*microagent*>	<*method*>

Disable Custom Microagent

The Custom microagent can be disabled by leveraging the Security TrustModel supported by OI Hawk Console. Users are explicitly granted or denied access through the access control file.

The following steps describe how to disable Custom microagent execution.

1. If multiple Hawk agents are running on a machine and these Hawk agents in turn belong to different Hawk domains, specify separate access control files for each domain.

   For each Hawk domain create a directory `OIHR_HOME/domain/<domain-name>/security where <domain-name>` is the name of the Hawk domain.

2. According to the requirements of your system, copy `OIHR_HOME/examples/security/Trusted.txt` or `OIHR_HOME/examples/security/TrustedWithDomain.txt` to `OIHR_HOME/security/`.

3. Modify the file to add the following lines:

   ```
   * * * *
   none * COM.TIBCO.hawk.microagent.Custom +
   ```

   The first line grants access to all users, on all nodes, and for all microagent methods.

The second line grants access only to the user `none`, on all nodes for the Custom microagent, where `none` is a non-existent user. This effectively prevents anyone from executing the Custom microagent.

4. Ensure that the `security_policy` parameter in Hawk agent configuration is set to one of the following, before starting the Hawk agent and OI Hawk Console:

   `COM.TIBCO.hawk.security.trusted.Trusted` or
   `COM.TIBCO.hawk.security.trusted.TrustedWithDomain`

Trusted.txt and TrustedWithDomain File Examples

The following example files demonstrate how a `Trusted.txt` and `TrustedWithDomain.txt` access control file might be constructed. The permissions and restrictions defined in this file are explained in the previous section.

Explanation of Settings

The settings in the example files below provide access to the following users as shown here:

- Grant `user1` point-to-point access to all methods on all microagents, except:
  - All `ACTION` methods on the `Custom` microagent on all nodes.
  - The specified methods on the `Repository` microagent on all nodes.
  - The specified methods on the `RuleBaseEngine` microagent on `nodeA`.
- Grant `user2` point-to-point and group operation invocation access to all methods on all microagents, except:
  - All `ACTION` methods on the `Custom` microagent on all nodes.
  - All `ACTION` methods on the `Repository` microagent on all nodes.
  - All `ACTION` methods on the `RuleBase` microagent on all nodes.
- Grant `user3` point-to-point and group operation invocation access to all methods on all microagents on all nodes, except:
  - group operation invocation access to all `ACTION` methods on the `RuleBase` microagent.
- Grant `user4` full access to all methods on all microagents on `nodeB`.

- Grant `user5` point-to-point access to all `INFO` methods on all microagents on all nodes.

Trusted.txt Example File

```
#
# This file is used by agent running with
COM.TIBCO.hawk.security.trusted.Trusted
# security model.
#
#
# Explanation of Settings:
#
# Grant "user1" point-to-point access to all methods on all Microagents,
EXCEPT
#       - all ACTION methods on the Custom microagent on all nodes.
#       - the specified methods on the Repository microagent on all
nodes.
#       - the specified methods on the RuleBaseEngine microagent on
"nodeA".
#
# Grant "user2" point-to-point and network access to all methods on all
# Microagents, EXCEPT
#       - all ACTION methods on the Custom microagent on all nodes.
#       - all ACTION methods on the Repository microagent on all nodes.
#       - all ACTION methods on the RuleBase microagent on all nodes.
#
# Grant "user3" point-to-point and network access to all methods on all
# Microagents on all nodes, EXCEPT
#       - network access to all ACTION methods on the RuleBase
microagent.
#
# Grant "user4" full access to all methods on all microagents on nodeB.
#
# Grant "user5" point-to-point access to all INFO methods on all
microagents
# on all nodes.
#
#
# Wildcard characters + and * usage:
#
# - Use + in node access for allowing access to group operations.
# - Use * in node access for allowing access to  point-to-point
invocations.
# - Use + in method access for allowing access to all INFO and ACTION
```

```
methods.
# - Use * in method access for allowing access to all INFO methods only.
#
#
# File format:
#
# user     node                  microagent                    method
#         access                   access                     access
#           &                        &                          &
#       restrictions            restrictions                restrictions
#user1        *
 !user1       *     COM.TIBCO.hawk.microagent.Custom             +
 !user1       *     COM.TIBCO.hawk.microagent.Repository     addRuleBase
 !user1       *     COM.TIBCO.hawk.microagent.Repository
updateRuleBase
 !user1       *     COM.TIBCO.hawk.microagent.Repository
deleteRuleBase
 !user1       *     COM.TIBCO.hawk.microagent.Repository
setSchedules
 !user1       *     COM.TIBCO.hawk.microagent.Repository     setRBMap
 !user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine addRuleBase
 !user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine
updateRuleBase
 !user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine
deleteRuleBase
 !user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine
loadRuleBase
 !user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine
unloadRuleBase
 !user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine
loadRuleBaseFromFile
 !user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine
setSchedules
 !user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine setRBMap
 user2        +                        *                         +
 !user2       *     COM.TIBCO.hawk.microagent.Custom             +
 !user2       *     COM.TIBCO.hawk.microagent.Repository         +
 !user2       *     COM.TIBCO.hawk.microagent.RuleBaseEngine     +
 user3
 !user3       +     COM.TIBCO.hawk.microagent.RuleBaseEngine     +
 user4      nodeB
 user5        *                        *                         *
 #
 # To activate logging, uncomment the following:
 # <LogService>  -log_dir logDir  -log_max_size size  -log_max_num n
 #
 # where: logDir is the directory where the log file is stored
 #        size is the maximum size of a rotating log file in KB.
```

```
 #          A suffix m or M  can be used for indicating MB .
 #        n is the maximum number of rotating log files.
```

TrustedWithDomain.txt Example File

```
#
# This file is used by agent running with
# COM.TIBCO.hawk.security.trusted.TrustedWithDomain security model.
#
# To allow a user running with
COM.TIBCO.hawk.security.trusted.TrustedWithDomain
# security model on Windows platform to access this agent, the user
# specified should include the domain of the user.
# For example, for user1 in domainX, the user should be specified as
# "domainX\user1".
#
# Note that agents using the TrustedWithDomain security model also allow
# users running with COM.TIBCO.hawk.security.trusted.Trusted security
model
# to access this agent.  For these users, the domain should not be
# included in the user.
#
#
# Explanation of Settings:
#
# Grant "user1" point-to-point access to all methods on all Microagents,
EXCEPT
#       - all ACTION methods on the Custom microagent on all nodes.
#       - the specified methods on the Repository microagent on all nodes.
#       - the specified methods on the RuleBaseEngine microagent on "nodeA".
#
# Grant "user2" point-to-point and network access to all methods on all
# Microagents, EXCEPT
#       - all ACTION methods on the Custom microagent on all nodes.
#       - all ACTION methods on the Repository microagent on all nodes.
#       - all ACTION methods on the RuleBase microagent on all nodes.
#
# Grant "user3" point-to-point and network access to all methods on all
# Microagents on all nodes, EXCEPT
#       - network access to all ACTION methods on the RuleBase microagent.
#
# Grant "user4" full access to all methods on all microagents on nodeB.
#
# Grant "user5" point-to-point access to all INFO methods on all
```

```
microagents
# on all nodes.
#
#
# Wildcard characters + and * usage:
#
# - Use + in node access for allowing access to group operations.
# - Use * in node access for allowing access to  point-to-point
invocations.
# - Use + in method access for allowing access to all INFO and ACTION
methods.
# - Use * in method access for allowing access to all INFO methods only.
#
#
# File format:
#
# user     node                  microagent                    method
#         access                   access                     access
#           &                        &                          &
#       restrictions            restrictions                restrictions
#
user1        *
!user1       *     COM.TIBCO.hawk.microagent.Custom             +
!user1       *     COM.TIBCO.hawk.microagent.Repository     addRuleBase
!user1       *     COM.TIBCO.hawk.microagent.Repository
updateRuleBase
!user1       *     COM.TIBCO.hawk.microagent.Repository
deleteRuleBase
!user1       *     COM.TIBCO.hawk.microagent.Repository     setSchedules
!user1       *     COM.TIBCO.hawk.microagent.Repository     setRBMap
!user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine addRuleBase
!user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine
updateRuleBase
!user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine
deleteRuleBase
!user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine loadRuleBase
!user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine
unloadRuleBase
!user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine
loadRuleBaseFromFile
!user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine setSchedules
!user1     nodeA   COM.TIBCO.hawk.microagent.RuleBaseEngine setRBMap
user2        +                        *                          +
!user2       *     COM.TIBCO.hawk.microagent.Custom             +
!user2       *     COM.TIBCO.hawk.microagent.Repository         +
!user2       *     COM.TIBCO.hawk.microagent.RuleBaseEngine     +
user3
!user3       +     COM.TIBCO.hawk.microagent.RuleBaseEngine     +
```

```
user4      nodeB
user5        *                        *                          *
## To activate logging, uncomment the following:
# <LogService>  -log_dir logDir  -log_max_size size  -log_max_num n
#
# where: logDir is the directory where the log file is stored
#        size is the maximum size of a rotating log file in KB.
#          A suffix m or M  can be used for indicating MB .
#        n is the maximum number of rotating log files.
```

Running with a localhost rvd

As a further precaution, AMI applications are required to specify `localhost` as part of the TIBCO Rendezvous daemon parameter in order to prevent remote connections to its `rvd` daemon. Instructions to do this for UNIX and Microsoft Windows platforms are given below.

UNIX Procedure

1. Add a command to start the localhost `rvd` prior to starting any TIBCO Hawk processes, as follows:

   `rvd -listen tcp:127.0.0.1:<daemon>`

2. Modify `hawkagent.cfg` and `hawkhma.cfg` and, in the `-rvd_session` parameter, specify the following:

   `tcp:127.0.0.1:<daemon>`

Microsoft Windows Procedure

Use `rvntsreg.exe` to install a localhost `rvd` as a Microsoft Windows service.

Procedure

1. Create an `rvd` service using `rvntsreg.exe`. Use the following parameter:

   `-listen tcp:127.0.0.1:<daemon>`

2. Make all TIBCO Hawk services dependent upon this new `rvd` service.

3. In the Configuration Utility, modify the daemon parameter to `tcp:127.0.0.1:<daemon>`

Trusted Security Sample Implementation

The sample implements the Trusted model described in Trusted Model. This implementation is similar to the default security model provided by OI Hawk Console.

Code

The sample implementation for Trusted Security is provided in the `/examples/security` directory.

Compile

While compiling the security sample, your `CLASSPATH` must include `console.jar` from TIBCO Hawk `lib` folder.

Run

To enable the security for the Hawk Agent and OI Hawk Console, use `-security_policy`.

To use a specific security policy, specify the name of the security policy class on each machine where you want to use the policy. Do not enter the file extension. For example, if your Java class file is named `ASecurityPolicy.class` you would specify `ASecurityPolicy`.

Ensure that this class file is bundled in a jar and placed in *OIHR_HOME/lib/ext*.

Using Trusted Security Model in OI Hawk Console

You can use Hawk Trusted Security Model in OI Hawk Console to ensure that only authorized users can perform restricted actions. You can specify security policy to be applied to a domain when registering a domain using OI Hawk Console web interface or using a configuration file.

To store access control information, the Trusted model uses an ASCII file. For more information about access control files and other configuration, see To Use the Trusted Model.

For more information about how to apply security policy when registering a domain using OI Hawk Console web interface, see TIBCO® Operational Intelligence Hawk® RedTail User Guide.

To apply security policy when registering a domain using a configuration file, set the `securityPolicy` parameter in the file `DomainTransportConfig.yml` to one of the following:

```
COM.TIBCO.hawk.security.trusted.Trusted
or
COM.TIBCO.hawk.security.trusted.TrustedWithDomain
```

TIBCO OI Hawk RedTail Programming

TIBCO OI Hawk RedTail provides APIs to interact with Hawk® applications. You can use the following APIs in TIBCO OI Hawk RedTail:

Hawk Console API

The Hawk Console API is a comprehensive set of Java interfaces that allow you to manage and interact with Hawk agents and monitor alerts generated by these agents. Both the Hawk RedTail Console and TIBCO Hawk® Event Service implement the Console API to monitor and manage agent behavior. Programmers can use the Console API to write custom applications similar to these applications to monitor agent behavior, subscribe to alert messages, and invoke microagent methods.

For more information, see the *TIBCO® Operational Intelligence Hawk® RedTail Programmer's Guide*.

Configuration Object API

The Configuration Object API is a Java interface for writing custom rulebases. Rulebases are used by Hawk agents to monitor and manage systems and applications. The Configuration Object API provides classes to define rules, tests and actions. Instances of these classes are put together to define a new rulebase.

For more information, see the *TIBCO® Operational Intelligence Hawk® RedTail Programmer's Guide*.

AMI API

The AMI API allows you to monitor application statistics with the Hawk API and make them manageable using Hawk Agent.

For more information, see the *TIBCO® Operational Intelligence Hawk® RedTail Programmer's Guide*.

REST API

You can use the REST API to access the TIBCO OI Hawk RedTail features such as Hawk microagent methods, alerts, tag based rulebases, content packs, and query. Hawk RedTail Console exposes the other TIBCO OI Hawk RedTail components, external clients, and external scripts. For more information, see the *TIBCO® Operational Intelligence Hawk® RedTail Programmer's Guide* and the "REST API Reference" section in *TIBCO® Operational Intelligence Hawk® RedTail User Guide*.

Uninstalling TIBCO OI Hawk RedTail

Perform the following procedure to uninstall TIBCO OI Hawk RedTail:

Note: This process removes all files that were installed as a part of TIBCO OI Hawk RedTail, even if those files were modified by the user or the application. Make sure you have a backup of user-modified files before proceeding with the uninstallation.

Note: Installing any TIBCO OI Hawk RedTail Adapter product creates the `OIHR_HOME/adapters` folder by default. Uninstallation of TIBCO OI Hawk RedTail does not remove the adapter folder. However, if you remove that folder manually, the adapters' uninstaller and the entire installation become non-functional.

Before you begin

Stop all running TIBCO OI Hawk RedTail services. For more information, see Stopping and Restarting TIBCO OI Hawk RedTail Enterprise Components.

Running uninstall.sh Script

This script uninstalls the advanced features of TIBCO OI Hawk RedTail such as Webapp, Querynode, hawkconsolenode, Prometheus, Prometheus discovery service, Grafana, and Postgres.

Perform the following steps to run the `uninstall.sh` script:

Procedure

1. Navigate to *OIHR_HOME/redtail/on_prem/node-bin/scripts*.
2. Open the terminal and run the following command as a root user. You can also run the command as a non-root user by prepending sudo to the command.

```
sudo ./uninstall.sh
```

Uninstalling in GUI Mode

Note: It is recommended that you run the `uninstall.sh` script for uninstallation of components before uninstalling TIBCO OI Hawk RedTail in GUI mode.

Procedure

1. Go to *TIBCO_HOME/tools/universal_installer* and run `TIBCOUniversalInstaller`.
2. Select **Uninstall Products from a TIBCO Home Location** and specify or select the TIBCO_HOME location from the drop down menu and click **Next**.
3. Select one of the following options on the **Uninstallation Type** page:
    - **Custom Uninstall:** Choose this option to select the products that you want to uninstall from the specified *TIBCO_HOME*.
    - **Typical Uninstall:** Choose this option to uninstall all products in the specified *TIBCO_HOME*.
4. If you choose the **Custom Uninstall** option, select the products that you want to uninstall, and then click **Next**.
5. Review the **Pre-Uninstall Summary** and click the **Uninstall** button to start the uninstallation process.
6. Review the **Post-Uninstall Summary** and click the **Finish** button to exit the uninstall wizard.

Uninstalling in Console Mode

Note: It is recommended that you run the `uninstall.sh` script for uninstallation of components before uninstalling TIBCO OI Hawk RedTail in console mode.

Procedure

1. Using a command window, navigate to the *TIBCO_HOME/tools/universal_installer* directory.
2. Enter the following command in the terminal or command prompt:

```
TIBCOUniversalInstaller.exe -console
```

3. Complete the uninstallation process by responding to the console window prompts.

Troubleshooting Enterprise Components

This section lists possible installation errors in an enterprise environment and gives the resolutions for the following issues:

Packet Fragmentation Errors with Multicast

You may encounter packet fragmentation errors when using multicast on Microsoft Windows XP. This is due to a known issue in Microsoft Windows.

You may need to apply a Microsoft hotfix. Information about obtaining the fix is in Microsoft Knowledge Base Article Q319627.

Error Message

A message similar to the following appears in the Microsoft Windows Event Log:

```
2002 Sep 13 09:01:31:035 GMT -8 HawkHMA Info [Application] HWKHMA-007012
PdhGetFormattedCounterValue for object PhysicalDisk and instance _Total and
counter Split IO/Sec failed with error 0x800007D8.
```

Code	Text
0x8000078D8	A counter with a negative value was detected.
0x8000078D6	A counter with a negative denominator was detected.

The Microsoft Windows Performance API is driven by a set of Microsoft Windows and third-party extension DLLs, which implement the various performance objects and associated counters. These extension DLLs may occasionally return counter values that cause mathematical errors in performance statistics calculations. These messages are reported by the Microsoft Windows Performance API.

These messages are reported by HMA for information purposes and do not have any adverse effect on functionality. They are not caused by HMA. They are caused by bugs or design flaws in the associated extension DLL.

TIBCO Hawk Services Fail to Start After Installation

After you have completed TIBCO OI Hawk installation on Microsoft Windows, if none of the TIBCO Hawk services start, use the Event Viewer to check for error messages related to the TIBCO Hawk services in the Application Log.

Contact TIBCO Support if the problem still persists after performing the troubleshooting activities or if you encounter any new problem.

Stopping and Restarting TIBCO OI Hawk RedTail Enterprise Components

You can stop TIBCO OI Hawk RedTail services in an enterprise environment in one of the following ways:

- Using a Script for Stopping TIBCO OI Hawk RedTail Services
- Manually stopping TIBCO OI Hawk RedTail Services

Using a Script for Stopping TIBCO OI Hawk RedTail Services

Procedure

1. Navigate to */usr/local/bin*.
2. Open the terminal and run the following command as a root user. You can also run the command as a non-root user by prepending sudo to the command.

```
sudo ./redtail_stop.sh
```

Manually stopping TIBCO OI Hawk RedTail Services

Procedure

1. (Optional) Open the terminal and enter the following command to get a list of all running services:

```
systemctl list-units rt_*
```

2. Open the terminal and enter the following commands to stop TIBCO OI Hawk RedTail components:

```
sudo systemctl stop rt_zookeeper.service
sudo systemctl stop rt_postgresql-13.service
sudo systemctl stop rt_machinenode.service
sudo systemctl stop rt_hawkconsolenode.service
sudo systemctl stop rt_querynode.service
sudo systemctl stop rt_prometheus.service
sudo systemctl stop rt_prometheus_ds.service
sudo systemctl stop rt_grafana_server.service
sudo systemctl stop rt_webapp.service
sudo systemctl stop rt_prometheus_backup.service
```

3. Open the terminal and enter the following commands to restart TIBCO OI Hawk RedTail components:

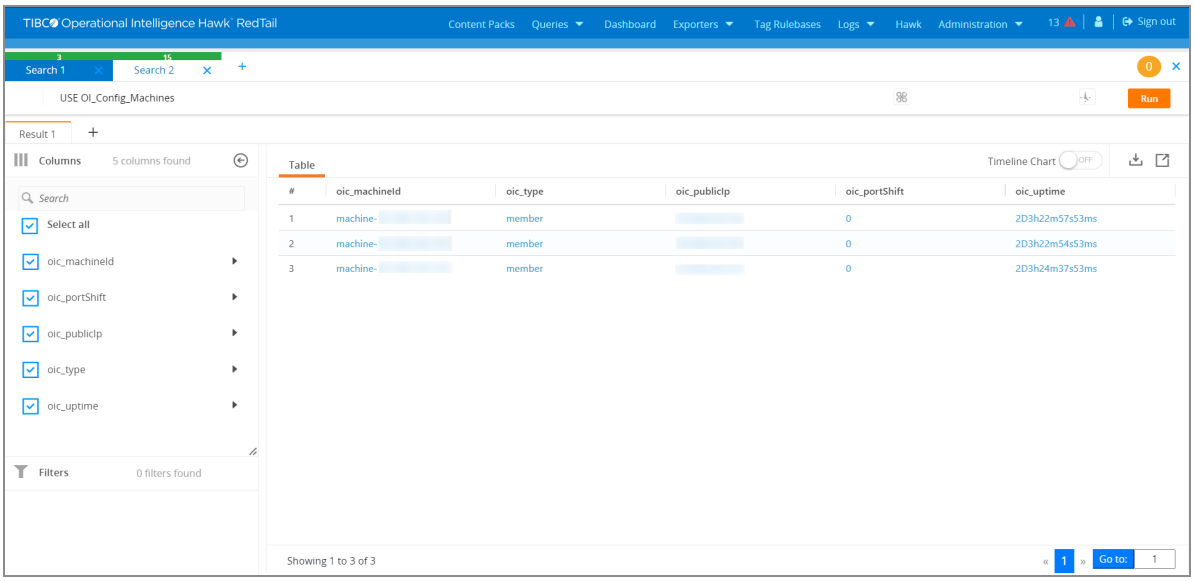
```
sudo systemctl restart rt_zookeeper.service
sudo systemctl restart rt_postgresql-13.service
sudo systemctl restart rt_machinenode.service
sudo systemctl restart rt_hawkconsolenode.service
sudo systemctl restart rt_querynode.service
sudo systemctl restart rt_prometheus.service
sudo systemctl restart rt_prometheus_ds.service
sudo systemctl restart rt_grafana_server.service
sudo systemctl restart rt_webapp.service
sudo systemctl restart rt_prometheus_backup.service
```

Running Infrastructure Queries for Monitoring TIBCO OI Hawk RedTail Components

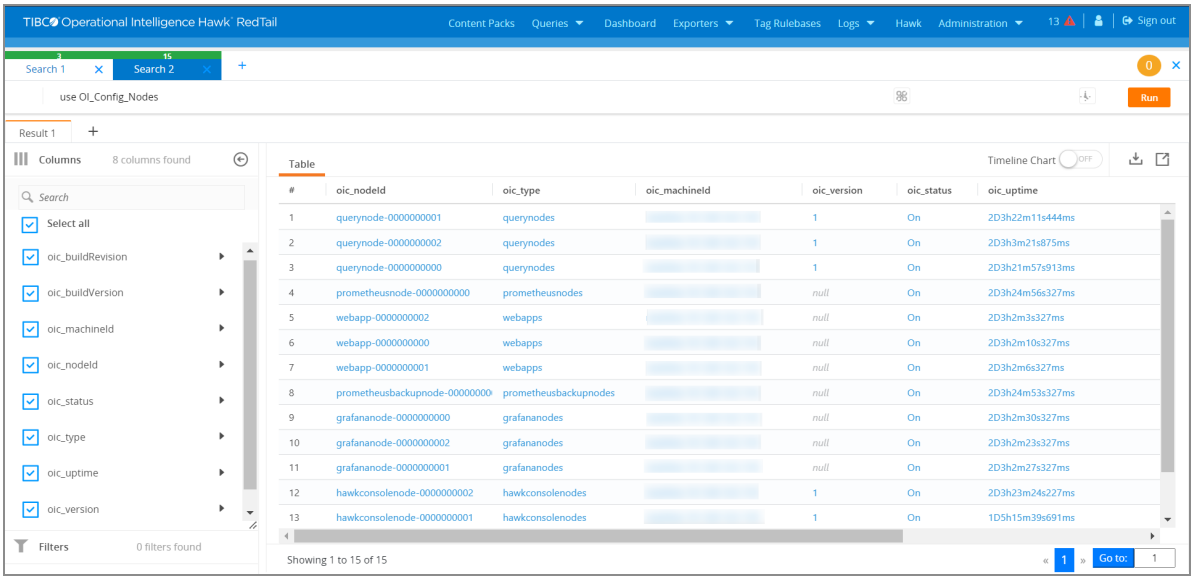
To get information about TIBCO OI Hawk RedTail components, run the following queries from the querying section:

Note: You must have the TIBCO® Operational Intelligence Hawk® RedTail - Standard Edition license to run the queries.

- **USE OI_Config_Machines:** This query returns the details about the machines which are a part of HA deployment and returns general status details about them. For example, running this query on TIBCO OI Hawk RedTail produces the following output:



- **USE OI_Config_Nodes:** This query returns the details about the status of TIBCO OI Hawk RedTail components. For example, running this query on TIBCO OI Hawk RedTail produces the following output:



Viewing the Logs for TIBCO OI Hawk RedTail Components

Perform the following steps to view the logs of TIBCO OI Hawk RedTail components:

Procedure

1. Open the terminal and enter the following command to view the logs for TIBCO OI Hawk RedTail components:

   ```
   journalctl -u rt_<servicename>.service
   ```

   For example, you must execute the following command to view the logs for hawkconsolenode:

   ```
   journalctl -u rt_hawkconsolenode.service
   ```

Error Codes

For information about error codes, see *TIBCO® Operational Intelligence Hawk® RedTail Error Codes*.

TIBCO 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 HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for this product is available on the TIBCO® Operational Intelligence Hawk® RedTail page:

- *TIBCO® Operational Intelligence Hawk® RedTail Release Notes*
- *TIBCO® Operational Intelligence Hawk® RedTail Concepts*
- *TIBCO® Operational Intelligence Hawk® RedTail Installation, Configuration, and Administration Enterprise Edition*
- *TIBCO® Operational Intelligence Hawk® RedTail Installation, Configuration, and Administration Container Edition*
- *TIBCO® Operational Intelligence Hawk® RedTail User Guide*
- *TIBCO® Operational Intelligence Hawk® RedTail Programmer's Guide*
- *TIBCO® Operational Intelligence Hawk® RedTail Admin Agent*
- *TIBCO® Operational Intelligence Hawk® RedTail Plug-in Reference for TIBCO Administrator*
- *TIBCO® Operational Intelligence Hawk® RedTail Microagent Reference*
- *TIBCO® Operational Intelligence Hawk® RedTail Plug-in Reference*
- *TIBCO® Operational Intelligence Hawk® RedTail Error Codes*

- *TIBCO® Operational Intelligence Hawk® RedTail Security Guidelines Enterprise Edition*
- *TIBCO® Operational Intelligence Hawk® RedTail Security Guidelines Container Edition*

How to Contact TIBCO Support

Get an overview of TIBCO Support. You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support website.
- 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 TIBCO Support website. 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 TIBCO Community.

Legal and Third-Party Notices