

TIBCO ActiveMatrix® BPM Hawk Plug-in User's Guide

*Software Release 4.2
August 2017*

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, Two-Second Advantage, TIBCO ActiveMatrix BPM, TIBCO Administrator, TIBCO Business Studio, TIBCO Enterprise Message Service, TIBCO General Interface, TIBCO Hawk, TIBCO iProcess, TIBCO JasperReports, TIBCO Spotfire, TIBCO Spotfire Server, and TIBCO Spotfire Web Player are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2005-2017 TIBCO Software Inc. All rights reserved.

TIBCO Software Inc. Confidential Information

Contents

Figures	4
TIBCO Documentation and Support Services	5
TIBCO Hawk ActiveMatrix Plug-in Overview	7
ActiveMatrix Host Microagent and Service Microagent	8
Monitoring Architecture	8
ActiveMatrix 3.2 Monitoring Service Application	9
ActiveMatrix Host Microagent	11
isHostRunning	11
getHostInfo	12
getNodeInfo	14
onHostEvent	16
onNodeEvent	18
startNode	19
stopNode	19
Common Logging Event Publisher Microagent	21
startPublisher	21
stopPublisher	21
startAllPublishers	22
stopAllPublishers	22
getEventPublisherInfo	22
getEventDataSourceDetails	23
reloadEventDefinitions	24
Sample Event Definitions Configuration to CLEvents from Hawk	27
ActiveMatrix Service Microagent	31
getConfig	31
changeMonitoringConfig	32
getComponentInfo	32
getBindingInfo	34
onComponentEvent	35
onBindingEvent	37
getBindingOperationStats	39
getComponentOperationStats	40
onBindingOperationStats	42
onComponentOperationStats	43
Service Execution Stats	44

Figures

Overview of ActiveMatrix Host and Service Microagents 7

Monitoring Points for Available Execution Statistics8

Monitoring Architecture of ActiveMatrix Host and Service Microagents 9

5 Minute Rolling Window (10 Buckets)10

TIBCO Documentation and Support Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, visit:

<https://docs.tibco.com>

Product-Specific Documentation

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site. To directly access documentation for this product, double-click the following file:

`TIBCO_HOME/release_notes/TIB_amx-bpm_version_docinfo.html`

where *TIBCO_HOME* is the top-level directory in which TIBCO products are installed. On Windows, the default *TIBCO_HOME* is `C:\tibco`. On UNIX systems, the default *TIBCO_HOME* is `/opt/tibco`.

The following documents for this product can be found on the TIBCO Documentation site:

- TIBCO ActiveMatrix BPM SOA Concepts
- TIBCO ActiveMatrix BPM Concepts
- TIBCO ActiveMatrix BPM Developer's Guide
- TIBCO ActiveMatrix BPM Web Client Developer's Guide
- TIBCO ActiveMatrix BPM Tutorials
- TIBCO ActiveMatrix BPM Business Data Services Developer Guide
- TIBCO ActiveMatrix BPM Case Data User Guide
- TIBCO ActiveMatrix BPM Event Collector Schema Reference
- TIBCO ActiveMatrix BPM - Integration with Content Management Systems
- TIBCO ActiveMatrix BPM SOA Composite Development
- TIBCO ActiveMatrix BPM Java Component Development
- TIBCO ActiveMatrix BPM Mediation Component Development
- TIBCO ActiveMatrix BPM Mediation API Reference
- TIBCO ActiveMatrix BPM WebApp Component Development
- TIBCO ActiveMatrix BPM Administration
- TIBCO ActiveMatrix BPM Performance Tuning Guide
- TIBCO ActiveMatrix BPM SOA Administration
- TIBCO ActiveMatrix BPM SOA Administration Tutorials
- TIBCO ActiveMatrix BPM SOA Development Tutorials
- TIBCO ActiveMatrix BPM Client Application Management Guide
- TIBCO ActiveMatrix BPM Client Application Developer's Guide
- TIBCO Openspace User's Guide
- TIBCO Openspace Customization Guide
- TIBCO ActiveMatrix BPM Organization Browser User's Guide (Openspace)
- TIBCO ActiveMatrix BPM Organization Browser User's Guide (Workspace)

- TIBCO ActiveMatrix BPM Spotfire Visualizations
- TIBCO Workspace User's Guide
- TIBCO Workspace Configuration and Customization
- TIBCO Workspace Components Developer Guide
- TIBCO ActiveMatrix BPM Troubleshooting Guide
- TIBCO ActiveMatrix BPM Deployment
- TIBCO ActiveMatrix BPM Hawk Plug-in User's Guide
- TIBCO ActiveMatrix BPM Installation: Developer Server
- TIBCO ActiveMatrix BPM Installation and Configuration
- TIBCO ActiveMatrix BPM Log Viewer
- TIBCO ActiveMatrix BPM Single Sign-On
- Using TIBCO JasperReports for ActiveMatrix BPM

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

How to Join TIBCO Community

TIBCO Community is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCO Community offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

<https://community.tibco.com>

TIBCO Hawk ActiveMatrix Plug-in Overview

TIBCO Hawk ActiveMatrix Plug-in provides features to monitor and manage TIBCO ActiveMatrix host and node processes, as well as TIBCO ActiveMatrix services and components.

These features can be realized by several Hawk microagents:

- The ActiveMatrix Host Microagent
- The ActiveMatrix Services Microagent
- Common Logging Event Publisher Microagent

The Host Microagent enables simple monitoring and management of the TIBCO Host process, by providing basic status information about the host and nodes managed by the host.

The Service Microagent allows for monitoring of status and performance statistics of service components and bindings running on a node. The performance statistics provided are of short duration (in the last five minutes) and include metrics such as hits, faults, and average response time.

The Common Logging Event Publisher Microagent enables Hawk microagents to connect to the ActiveMatrix Common Logging Service, to provide events for logging purposes.

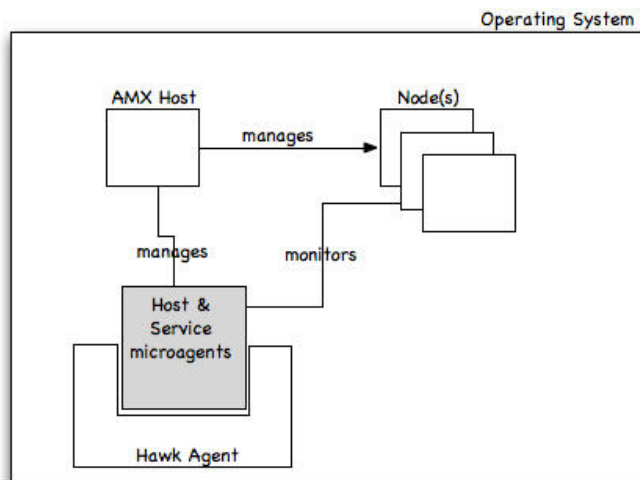
[Overview of ActiveMatrix Host and Service Microagents](#), illustrates how the Host and Service Microagents interact with the ActiveMatrix host and nodes.

The Host and Service Microagents are TIBCO Hawk Plug-in microagents and they are started as part of the Hawk Agent process.



These microagents require the Hawk HMA to be running to function correctly.

Overview of ActiveMatrix Host and Service Microagents



ActiveMatrix Host Microagent and Service Microagent

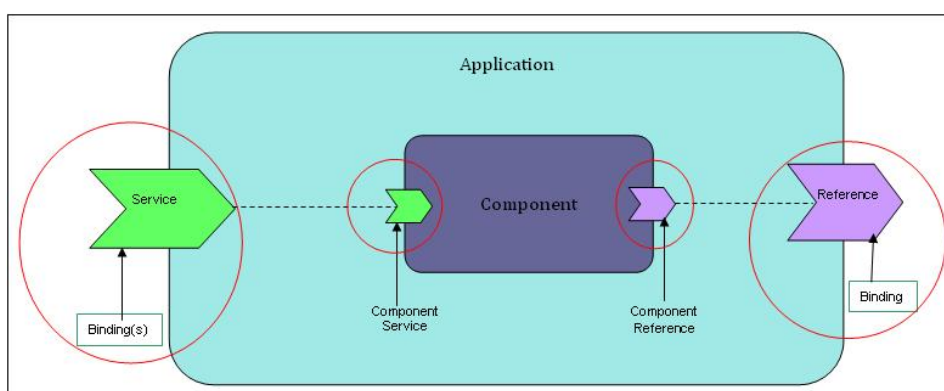
The ActiveMatrix Host Microagent enables the monitoring and management of TIBCO ActiveMatrix host and nodes running on each local machine. It provides methods to retrieve node status and host status, as well as control methods to start or stop nodes.

The ActiveMatrix Service Microagent provides the following monitoring capabilities:

- Monitor status of Component and Binding instances deployed in a node.
- Monitor execution statistics of service binding(s), reference binding, component services and component references of a deployed ActiveMatrix 3 Application.

[Monitoring Points for Available Execution Statistics](#), describes the various monitoring points for which execution statistics are available.

Monitoring Points for Available Execution Statistics



This diagram is representative of a typical ActiveMatrix 3.x Application (an Open SOA SCA Assembly). The indicated monitoring points are the parts of the deployed application for which the ActiveMatrix Service Microagent will provide statistics data.

For details on the metrics provided at each monitoring point, refer to [Service Execution Stats](#).

Monitoring Architecture

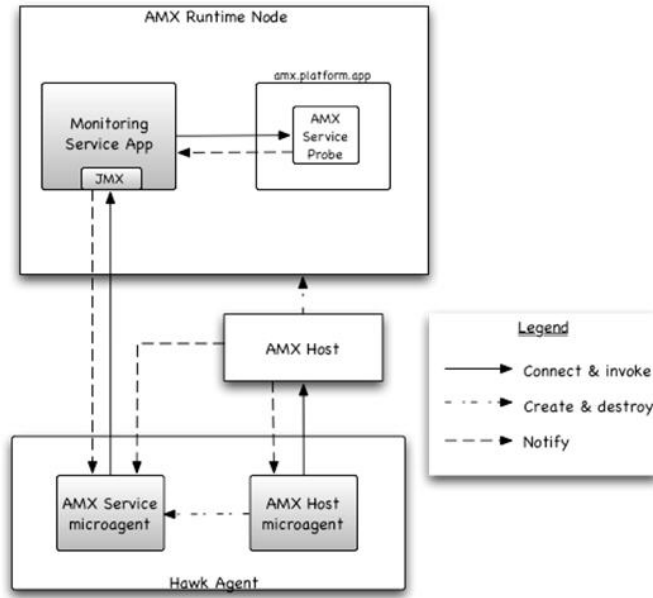
The ActiveMatrix Service microagent is designed to monitor services deployed in the ActiveMatrix node and it is integrated with ActiveMatrix Host microagent.

The ActiveMatrix Host microagent creates instances of the Service Microagent for each node (on node start) associated with ActiveMatrix host, and destroys these instances on Node stop.

In order to collect service operation statistics on each node, the Service microagent requires the deployment of an ActiveMatrix 3.2 Monitoring Application to each node that this microagent is required to monitor.

The architecture of the Host and Service microagent and the interaction with the partnering Monitoring Application is shown in [Monitoring Architecture of ActiveMatrix Host and Service Microagents](#).

Monitoring Architecture of ActiveMatrix Host and Service Microagents



The Service microagent connects to a node process using Java Attach API. Statistics data can then be retrieved using either synchronous or asynchronous method invocations.

In either approach, the Service microagent connects to the Monitoring Application's JMX MBean to retrieve the service operation statistics data.

ActiveMatrix 3.2 Monitoring Service Application

The ActiveMatrix 3.2 Monitoring Service application provides the Service Microagent with access to, or notification of service execution statistics.

The Monitoring Service application provides some configuration capabilities, which can be set in either the TIBCO ActiveMatrix Administrator, or the ActiveMatrix Service Microagent.

Monitoring Configuration

The monitoring configuration offered by the Monitoring Service application enables the configuration of a rolling window for statistics collection.

The maximum length of a rolling window is five minutes, but this is configurable.

The configuration properties are as follows:

Rolling Window Length

Length of rolling window (in seconds, up to 5 minutes)

Statistics Gathering Interval

Length of a bucket (in seconds)

Notification Interval

Rate at which statistics data is published (via notification)

Is Partial Window

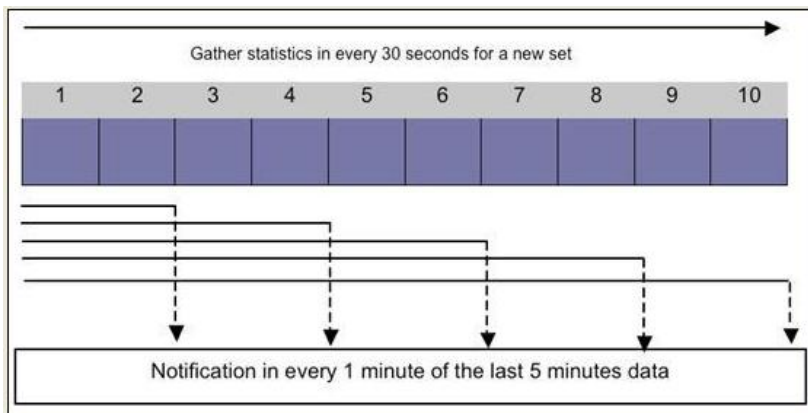
Whether to include the current statistics data (incomplete bucket) in notifications.

For example, the configuration settings to produce the following diagram would be:

- Rolling Window Length = 300
- Statistics Gathering Interval = 30

- Notification Interval = 60
- Is Partial Window = false

5 Minute Rolling Window (10 Buckets)



Common Logging Event Publisher Microagent

This microagent publishes events from TIBCO Hawk using the TIBCO Common Logging in the Base Event Format (BEF)/Common Base Event (CBE).



This version replaces the earlier microagent first released with Hawk 4.9.0.

ActiveMatrix Host Microagent

The ActiveMatrix Host Microagent (com.tibco.hawk.amx.AMXHostMicroAgent) allows monitoring and managing of TIBCO ActiveMatrix Host Instance running in the local machine.

AMXHost Microagent Configuration File

File name: AMXHostPluginConfig.xml

Content:

```
<?xml version="1.0" encoding="UTF-8"?>
<AMXHostConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="AMXHostPluginConfig.xsd">
  <HPAClientConfigPropFile>%TIBCO_HAWK_HOME_ESC%/plugin/amx/
hpaclientconfig.properties</HPAClientConfigPropFile>
  <MethodTimeout>10</MethodTimeout>
  <HostProcessPrefix>tibcohost*</HostProcessPrefix>
  <NodeProcessPrefix>tibamx_*</NodeProcessPrefix>
</AMXHostConfiguration>
```

XSD Configuration file: AMXHostPluginConfig.xsd

Content:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xsd:element name="AMXHostConfiguration" type="AMXHostConfigurationType" />
  <xsd:complexType name="AMXHostConfigurationType">
    <xsd:sequence>
      <xsd:element ref="MicroAgentName" />
      <xsd:element ref="MicroAgentDisplayName" />
      <xsd:element ref="MicroAgentDesc" />
      <xsd:element ref="MethodTimeout" />
      <xsd:element ref="HPAClientConfigPropFile" />
      <xsd:element ref="HostProcessPrefix" />
      <xsd:element ref="NodeProcessPrefix" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="MicroAgentName" type="xsd:string" default="" />
  <xsd:element name="MicroAgentDisplayName" type="xsd:string" default="" />
  <xsd:element name="MicroAgentDesc" type="xsd:string" default="" />
  <xsd:element name="HPAClientConfigPropFile" type="xsd:string"
  default="hpaclientconfig.properties" />
  <xsd:element name="MethodTimeout" type="xsd:string" default="10" />
  <xsd:element name="HostProcessPrefix" type="xsd:string" default="tibcohost*" />
  <xsd:element name="NodeProcessPrefix" type="xsd:string" default="tibamx_*" />
</xsd:schema>
```

AMX Host Microagent Configuration

For configuration of AMX Host Microagent, refer to the `readme.txt` available in `plugin/amx`.

Methods

The methods in the microagent provides the information for the host instance running on the machine.

isHostRunning

The `isHostRunning` method pings the TIBCO ActiveMatrix Host and returns the value as `true` or `false`.

Type

Synchronous, `IMPACT_INFO`.

Arguments

None.

Returns

Name	Type	Description
Running: true/false	Boolean	Pings the TIBCO ActiveMatrix Host and returns true if host is running, or else false .

getHostInfo

The getHostInfo method returns the details of TIBCO ActiveMatrix Host instance running in the local machine.

Type

Synchronous, IMPACT_INFO.

Arguments

None.

Returns

Name	Type	Description
Process ID	String	The Process ID of the TIBCO Host Instance (OS Process ID). Note: ProcessID return "-1", if Host is not running.
Name	String	The name of the ActiveMatrix host (SystemHost).
Type	String	Type of the TIBCO ActiveMatrix host: (TIBCO Host).

Name	Type	Description
Runtime State	String	The state of ActiveMatrix host. • UNKNOWN • NOT_RUNNING • INITIALIZING • INITIALIZING_FAILED • INITIALIZED • STARTING - Starting of the host. The starting state should move to the running state BEFORE the nodes are started. In other words, Host state is independent of Node State. Nodes cannot start unless the Host starts successfully. • STARTING_FAILED - The process of starting failed, the reason on the status will be set. • STOPPING • STOPPED • RUNNING - The host is running successfully. • UNBOUND - The host is not bound to the enterprise. • LOOSING_CONTACT • LOST_CONTACT - Lost contact with a host. Reasons being the host was killed unexpectedly, heartbeats are lost in a flood of traffic, or the network is no longer operational.
Version	String	Version of the TIBCO ActiveMatrix Host.
Time Stamp	String	The time stamp (date and time) of the state changed (For example, from the NOT_RUNNING state to the RUNNING State, the time stamp of the host started and running successfully). Time Stamp returns long value, which is available in Hawk in a readable format of date and time.
Binding Status	String	The binding status of TIBCO Host (for example, bound).
Host Platform Version	String	Platform version of the Host.
Internet Host Name	String	Name of the Internet Host.
OS Name	String	The Operating System name of the machine running the host.

Name	Type	Description
OS Version	String	The Operating System version of the machine running the host.
System Arch	String	System Architecture of the machine running the host.
Stop Reason	String	The reason of the state host is in. It may be null during the normal operation.

getNodeInfo

The getNodeInfo method returns the Node information by the Node Name.

If the Node Name argument is blank, then all ActiveMatrix nodes are returned. On providing the Node Name argument, it serves as a regular expression used to filter the nodes returned.

Type

Synchronous, IMPACT_INFO.

Arguments

Name	Type	Description
Node Name	String	The node name.

Returns

Name	Type	Description
Process ID	String	The Process ID of the ActiveMatrix Node (OS Process ID.) Note: Process ID return "-1", if Node is not running.
Name	String	The name of the Node.
Host	String	The name of the host associated with the node.

Name	Type	Description
Runtime State	String	The actual state of the node as reported by the host. • NOT_RUNNING • INITIALIZING • INITIALIZED • INITIALIZATION_FAILED • STARTING - Starting of the node. The starting state should move to the running state BEFORE the nodes are started. • RUNNING - The node is running successfully. • START_FAILED • STOPPING • STOPPED • LOST_CONTACT - Lost contact with the node. Reasons being the node was killed unexpectedly, or the network is no longer operational. • UNKNOWN
Stop Mode	String	In which mode the node is stopped. Node can be stopped in abort, immediate or normal mode through the ActiveMatrix Admin UI or end the process from the OS or using the API to the Stop Node. • abort - The node is killed immediately. This mode is however not supported on all the HPAs. If you attempt to abort a node on a HPA that does not support node abort, the status block for that node shows an "operation-not-supported" error. • immediate - It implies that the node's infrastructure and applications may not have been able to shut down in a clean manner. This will potentially leave the node in an inconsistent state that will have to be recovered prior to the subsequent successful startup. • normal - Implies that all infrastructure and applications running on the node would have completed all or any task(s) normally associated with their shutdown

Name	Type	Description
Stop Reason	String	Stop reason text message. If the end process is from the OS, then reason returned is "OSGI Framework is stopping".
Stop Reason Code	String	Stop reason error code. Note: Most of the time it is Reason null and reason code -1 .
Time Stamp	String	The time stamp (date and time) of the state changed (For example, from the NOT_RUNNING state to the RUNNING State, the time stamp of the node started and running successfully). Time Stamp returns long value, which is available in Hawk in a readable format of date and time. Note: If an ActiveMatrix node was stopped while starting the Hawk agent, then getNodeInfo() would return 0 in the timestamp column for that particular node.



During the microagent startup, if the node state appears as `Starting` in ActiveMatrix Administrator UI, the `getNodeInfo()` method does not return the correct node state unless the node is restarted.

When the node is restarted (that is, when the microagent receives a `Start` notification from ActiveMatrix Administrator), the microagent returns the correct state of the node.



On Windows, when a node is stopped using "Terminate Node Process" option in the TIBCO ActiveMatrix Administrator, then the last state reported is `LOST_CONTACT` instead of `NOT_RUNNING`.

onHostEvent

The `onHostEvent` method provides notification of the changed run-time state of a TIBCO Host. This method supports only the subscribe mode.

Type

Asynchronous, `IMPACT_INFO`.

Arguments

None.

Returns

Name	Type	Description
Process ID	String	The Process ID of the TIBCO Host (OS Process Id)
Name	String	The name of the ActiveMatrix host (AMXAdminHost).

Name	Type	Description
Type	String	The type of the ActiveMatrix host (AMXAdminHost).
Runtime State	String	The state of the ActiveMatrix host. - UNKNOWN - NOT_RUNNING - INITIALIZING - INITIALIZING_FAILED - INITIALIZED - STARTING: The starting of the host. The starting state should move to the running state BEFORE starting the nodes. In other words, Host state is independent of Node State. Nodes can start unless the Host starts successfully. - STARTING_FAILED: The starting process failed and the reason on the status will be set. - STOPPING - STOPPED - RUNNING: The host is running normally. - UNBOUND: The host is not bound to an enterprise. - LOSING_CONTACT - LOST_CONTACT: Lost contact with a host. Reason being the host was killed unexpectedly, heartbeats are lost in a flood of traffic, or the network is no longer operational.
Version	String	Version of the ActiveMatrix Host.
Time Stamp	String	The time stamp (date and time) of the state changed (For example, from the NOT_RUNNING state to the RUNNING State, the time stamp of the Host started and running successfully). Time Stamp returns long value, which is available in Hawk in a readable format of date and time.
Binding Status	String	Status of Host binding.
Host Platform Version	String	Version of Host Platform.
Internet Host Name	String	Internet Host name.
OS Name	String	Operating System name of the host.

Name	Type	Description
OS Version	String	Operating System version of the host.
System Arc	String	System Architecture of the Host.
Stop Reason	String	The reason of the state the host is in. It may be null during the normal operation.

onNodeEvent

The onNodeEvent method provides notification of the run-time state of the node when changed of a particular node or all nodes associated with the TIBCO Host.

If the Node Name argument is blank then it provides notification for all nodes associated with the TIBCO Host.

This method supports only subscribe mode.

Type

Asynchronous, IMPACT_INFO.

Arguments

Name	Type	Description
Node Name	String	The node name.

Returns

Name	Type	
Name	String	The name of the node.
Runtime State	String	The state of the node.
Host Name	String	The name of the host associated with the node.
Stop Mode	String	In which mode the node is stopped. Node can be stopped in abort , immediate or normal mode through the ActiveMatrix Admin UI or end the process from the OS or using the API to the Stop Node.
Stop Reason	String	Stop reason text message.
Stop Reason Code	String	Stop reason error code.

Name	Type	
Time Stamp	String	The time stamp (date and time) of the state changed (For example, from the NOT_RUNNING state to the RUNNING State, the time stamp of the node started and running successfully). Time Stamp returns long value, which is available in TIBCO Hawk in a readable format of date and time.

startNode

The startNode method starts the TIBCO ActiveMatrix node, if not running.

Type

Synchronous, `IMPACT_ACTION`.

Arguments

Name	Type	Description
Node Name	String	The node name.

Exceptions

Condition	Result
If node does not exist.	Throws a <code>MicroAgentException</code> with Message "TIBCO-AMX-HPA-012106: node TestNode does not exist".
If node is already started/running.	Throws a <code>MicroAgentException</code> with Message "Provided node name '<nodename>' already started for host '<hostname>'.

Returns

None.

stopNode

The stopNode method stops the TIBCO ActiveMatrix node.

Type

Synchronous, `IMPACT_ACTION`.

Arguments

Name	Type	Description
Node Name	String	The node name.

Name	Type	Description
Reason	String	The stop reason text message.
Reason Code	String	The stop reason error code.

Exceptions

Condition	Result
If node does not exist.	Throws a <code>MicroAgentException</code> with Message "Provided node name '<nodename>' not found for host '<hostname>'".
If node is already started/running.	Throws a <code>MicroAgentException</code> with Message "Provided node name '<nodename>' already stopped for host '<hostname>'".

Returns

None.

Common Logging Event Publisher Microagent

The Common Logging Event Publisher Microagent (com.tibco.hawk.commonlogging.event.CLEventPublisher) publishes events from TIBCO Hawk using the TIBCO Common Logging in the Base Event Format (BEF) or Common Base Event (CBE).

Create the Event Definition configuration file with the following definitions:

- What events have to be published (Event Output).
- Event Data Source (from where to get and collect the events).
- Where to publish events (Common Logging Configuration)
- How to publish (Event Subscribe).

Sample event definition is available in the folder <HAWK_HOME>/plugin/commonlogging.

CLEvent Publisher Microagent Configuration

For configuration of CLEventPublisher Microagent, refer to the readme.txt available in <HAWK_HOME>/plugin/commonlogging.

Methods

The methods in the microagent allows to start or stop event publisher and returns event publisher's information.

startPublisher

The startPublisher method starts the event publisher to start publishing events from the TIBCO Hawk based on the Event Publisher Description defined in the Event Definitions file.

Type

Synchronous, IMPACT_ACTION.

Arguments

Name	Type	Description
Event Publisher Name	String	The Event Publisher Name.

Returns

None.

stopPublisher

The stopPublisher method stops the Event Publisher.

Type

Synchronous, IMPACT_ACTION.

Arguments

Name	Type	Description
Event PublisherName	String	The Event Publisher Name.

Returns

None.

startAllPublishers

The startAllPublishers method starts all valid and not started Event Publishers.

Type

Synchronous, `IMPACT_ACTION`.

Arguments

None.

Returns

None.

stopAllPublishers

The stopAllPublishers method stops all started Event Publishers.

Type

Synchronous, `IMPACT_ACTION`.

Arguments

None.

Returns

None.

getEventPublisherInfo

The getEventPublisherInfo method returns the Event Publisher information by Event Publisher name.

If the event publisher name argument is blank then all event publishers are returned. On specifying the event publisher name argument, it serves as a regular expression used to filter the event publisher returned.

Type

Synchronous, `IMPACT_INFO`.

Arguments

Name	Type	Description
Event Publisher Name	String	The Event Publisher Name.

Returns

Name	Type	Description
Name	String	Name of the Event Publisher.
Status	String	The current status of the event publisher (started/stop).
Auto Start	String	The flag to indicate event publisher is auto start configured.
Interval Time	String	The interval time to publish events.
Subscribe On Event Source Name	String	The name of the Event Source on the Event Publisher Subscribe.
CL Logger Name	String	The TIBCO Common Logging logger name.
CL Extended Model Name	String	The TIBCO Common Logging Extended Model name.
CL Extended Model Package Name	String	The TIBCO Common Logging Extended Model package name.

getEventDataSourceDetails

The getEventDataSourceDetails method returns the event data source details.

Type

Synchronous, IMPACT_INFO.

Arguments

None.

Returns

Name	Type	Description
EventSourceId	String	ID of the Event Source.
Microagent	String	The name of the TIBCO Hawk microagent.
Method	String	The name of the microagent method.

Name	Type	Description
Status	String	The status of the event source. - Valid - Whether the microagent available method name is valid. - Invalid - Whether the microagent is available and/or not available and/or method is not valid.
Description/Help	String	Description/Help of the microagent method or reason for its invalid status.

reloadEventDefinitions

The reloadEventDefinitions method stops all started event publishers, reloads event definitions from the configuration file and starts all valid event publishers with an auto start.

You can make changes in the configuration file and reflect changes without restarting the TIBCO Hawk Agent.

Type

Synchronous, `IMPACT_ACTION`.

Arguments

None.

Returns

None.

Event Definitions XSD Details

File: `<HAWK_HOME>/plugin/commonlogging/eventdefs.xsd`

Content:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://xsd.tns.tibco.com/hawk/cleventpublisher/
eventdef" elementFormDefault="qualified"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:eventdef="http://
xsd.tns.tibco.com/hawk/cleventpublisher/eventdef">
  <xsd:annotation>
    <xsd:documentation>Provide details of Event Publisher</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType name="EventSourceType">
    <xsd:attribute name="id" type="xsd:string" use="required"></xsd:attribute>
    <xsd:attribute name="methodName" type="xsd:string" use="required"></
xsd:attribute>
    <xsd:attribute name="microAgentName" type="xsd:string" use="required"></
xsd:attribute>
  </xsd:complexType>
  <xsd:complexType name="EventDataSourceType">
    <xsd:sequence>
      <xsd:element name="EventSource" type="eventdef:EventSourceType"
minOccurs="1" maxOccurs="unbounded"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="EventDefinitionsType">
    <xsd:sequence>
      <xsd:element name="EventPublishers" type="eventdef:EventPublishersType"
```

```

minOccurs="1" maxOccurs="1">
    </xsd:element>
    <xsd:element name="EventDataSource" type="eventdef:EventDataSourceType"
minOccurs="1" maxOccurs="1"></xsd:element>
    <xsd:element name="CommonLoggingConfigs"
type="eventdef:CommonLoggingConfigsType" minOccurs="1" maxOccurs="1">
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="EventPublisherType">
    <xsd:sequence>
        <xsd:element name="CommonLoggingConfigRef"
            type="eventdef:CommonLoggingConfigRefType" minOccurs="1"
            maxOccurs="1">
        </xsd:element>
        <xsd:element name="EventDataSourceRef"
            type="eventdef:EventDataSourceRefType" minOccurs="1"
            maxOccurs="1">
        </xsd:element>
        <xsd:element name="EventOutput"
            type="eventdef:EventOutputType" minOccurs="1" maxOccurs="1">
        </xsd:element>
        <xsd:element name="EventSubscribe"
            type="eventdef:EventSubscribeType" minOccurs="1" maxOccurs="1">
        </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"></xsd:attribute>
    <xsd:attribute name="validateAtStartup" type="xsd:boolean" use="optional"
default="false"></xsd:attribute>
</xsd:complexType>
<xsd:complexType name="CommonLoggingConfigType">
    <xsd:annotation>
        <xsd:documentation>Common Logging details.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="ExtendedModel" type="eventdef:ExtendedModelType"
minOccurs="1" maxOccurs="1"></xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required"></xsd:attribute>
    <xsd:attribute name="loggerName" type="xsd:string" use="required"></
xsd:attribute>
</xsd:complexType>
<xsd:element name="EventDefinitions" type="eventdef:EventDefinitionsType"></
xsd:element>
    <xsd:complexType name="EventDataSourceRefType">
        <xsd:sequence>
            <xsd:element name="EventSourceRef" type="eventdef:EventSourceRefType"
minOccurs="1" maxOccurs="unbounded"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="EventSourceRefType">
        <xsd:sequence>
            <xsd:element name="InputParameters" type="eventdef:InputParametersType"
minOccurs="0" maxOccurs="1"></xsd:element>
            <xsd:element name="EventCorrelation"
type="eventdef:EventCorrelationType" minOccurs="0" maxOccurs="1"></xsd:element>
        </xsd:sequence>
        <xsd:attribute name="ref" type="xsd:string" use="required"></xsd:attribute>
    </xsd:complexType>
    <xsd:complexType name="InputParametersType">
        <xsd:sequence>
            <xsd:element name="InputParameter" type="eventdef:InputParameterType"
minOccurs="1" maxOccurs="unbounded"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="InputParameterType">
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
                <xsd:attribute name="name" type="xsd:string" use="required"></
xsd:attribute>
            </xsd:extension>
        </xsd:simpleContent>

```

```

</xsd:complexType>
<xsd:complexType name="EventOutputType">
  <xsd:sequence>
    <xsd:element name="EventElements" type="eventdef:EventElementsType"
minOccurs="1" maxOccurs="1"></xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="EventElementsType">
  <xsd:choice>
    <xsd:element name="EventElement"
      type="eventdef:EventElementType" minOccurs="0"
      maxOccurs="unbounded">
    </xsd:element>
    <xsd:element name="eventSourceRefs" type="xsd:string" minOccurs="1"
maxOccurs="1"></xsd:element>
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="EventElementType">
  <xsd:attribute name="source" type="xsd:string" use="required"></
xsd:attribute>
  <xsd:attribute name="output" type="xsd:string" use="optional"></
xsd:attribute>
</xsd:complexType>
<xsd:complexType name="EventSubscribeType">
  <xsd:sequence>
    <xsd:element name="IntervalTime" type="xsd:int"
      minOccurs="0" maxOccurs="1" nillable="false">
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="sourceEventId" type="xsd:string" use="required"></
xsd:attribute>
  <xsd:attribute name="autoStart" type="xsd:boolean" use="optional"
default="false"></xsd:attribute>
</xsd:complexType>
<xsd:complexType name="FilterType">
  <xsd:attribute name="identifier" type="xsd:string" use="required"></
xsd:attribute>
  <xsd:attribute name="operator" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="="></xsd:enumeration>
        <xsd:enumeration value=">"></xsd:enumeration>
        <xsd:enumeration value="<"></xsd:enumeration>
        <xsd:enumeration value="=>"></xsd:enumeration>
        <xsd:enumeration value="<="></xsd:enumeration>
        <xsd:enumeration value="!="></xsd:enumeration>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>
<xsd:complexType name="LoggerType">
  <xsd:attribute name="name" type="xsd:string" use="required"></xsd:attribute>
  <xsd:attribute name="configFile" type="xsd:string" use="optional"></
xsd:attribute>
</xsd:complexType>
<xsd:complexType name="ExtendedModelType">
  <xsd:sequence>
    <xsd:element name="Classpath" type="eventdef:ClasspathType"
minOccurs="0" maxOccurs="1"></xsd:element>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required"></xsd:attribute>
  <xsd:attribute name="packageName" type="xsd:string"
    use="required">
  </xsd:attribute>
</xsd:complexType>
<xsd:complexType name="EventPublishersType">
  <xsd:sequence>
    <xsd:element name="EventPublisher" type="eventdef:EventPublisherType"
minOccurs="1" maxOccurs="unbounded"></xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CommonLoggingConfigsType">

```

```

        <xsd:sequence>
          <xsd:element name="CommonLoggingConfig"
type="eventdef:CommonLoggingConfigType" minOccurs="1" maxOccurs="unbounded">
            </xsd:element>
          </xsd:sequence>
          <xsd:attribute name="logConfigFile" type="xsd:string" use="required"></
xsd:attribute>
        </xsd:complexType>
        <xsd:complexType name="CommonLoggingConfigRefType">
          <xsd:attribute name="ref" type="xsd:string" use="required"></xsd:attribute>
        </xsd:complexType>
        <xsd:complexType name="EventCorrelationType">
          <xsd:annotation>
            <xsd:documentation>To correlate with other Event Data Source
</xsd:documentation>
          </xsd:annotation>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="returnElement" type="xsd:string"
use="required">
                <xsd:annotation>
                  <xsd:documentation>Return Element of Hawk MicroAgent Method
</xsd:documentation>
                </xsd:annotation></xsd:attribute>
              </xsd:extension>
            </xsd:simpleContent>
          </xsd:complexType>
          <xsd:complexType name="ClasspathType">
            <xsd:sequence>
              <xsd:element name="path" type="xsd:string" minOccurs="1"
maxOccurs="unbounded"></xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:schema>

```

Sample Event Definitions Configuration to CLEvents from Hawk

The JVMAndProcessInfoEventdefs.xml is the sample Event Definitions configuration to publish the JVM Info and Process Info event of the AMX 3 TIBCO Host instance, and all nodes associated with this Host Instance.

File: <HAWK_HOME>\plugin\commonlogging\JVMAndProcessInfoEventdefs.xml.

Content:

```

<?xml version="1.0" encoding="UTF-8"?>
<def:EventDefinitions xmlns:def="http://xsd.tns.tibco.com/hawk/cleventpublisher/
eventdef"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xsd.tns.tibco.com/hawk/cleventpublisher/eventdef
eventdefs.xsd">
  <!-- Event Publishers Definition for
    1. TIBCO Host JVM Process Info
    2. Process Events for TIBCO Host and AMX Nodes running in local Machine. --
>
  <!-- Event Publishers Details -->
  <def:EventPublishers>
    <!--Sample Event Publisher to Publish Process Info Event for TIBCO AMX Host
-->
    <def:EventPublisher name="AMXHostProcessInfoEventPublisher">
      <!-- Common Logging Configuration Details -->
      <def:CommonLoggingConfigRef ref="ProcessCLConfig" />
      <!--Event Data Source Ref -->
      <def:EventDataSourceRef>
        <def:EventSourceRef ref="AMXHost"/>
        <def:EventSourceRef ref="Process">
          <def:EventCorrelation returnElement="ID Process">${
AMXHost.Process Id}</def:EventCorrelation>
        </def:EventSourceRef>
      </def:EventDataSourceRef>
      <!-- Event Ouput -->

```

```

        <def:EventOutput>
            <def:EventElements>
                <def:EventElement source="${Process.ID Process}"
output="processId" />
                <def:EventElement source="${Process.Process Name}"
output="processName" />
                <def:EventElement source="${Process.User Name}"
output="userName" />
                <def:EventElement source="${Process.CPU Time}"
output="processCpuTime" />
                <!-- On Windows platform Virtual KBytes,Stack KBytes,Heap
KBytes, % CPU and % Memory not available-->
                <def:EventElement source="${Process.Virtual KBytes}"
output="virtualMemory" />
                <def:EventElement source="${Process.Stack KBytes}"
output="stackSize" />
                <def:EventElement source="${Process.Heap KBytes}"
output="heapSize" />
                <def:EventElement source="${Process.% CPU}"
output="cpuUsageInPercent" />
                <def:EventElement source="${Process.% Memory}"
output="memoryUsageInPercent" />
                <def:EventElement source="${Process.Start time}"
output="startTime" />
                <def:EventElement source="${AMXHost.Name}"
output="physicalCompId.matrix.host" />
            </def:EventElements>
        </def:EventOutput>
        <!-- Event Subscription Details -->
        <def:EventSubscribe sourceEventId="AMXHost" autoStart="false">
            <def:IntervalTime>10000</def:IntervalTime>
        </def:EventSubscribe>
    </def:EventPublisher>
    <!--TIBCO Host JVM InfoEvent Publisher -->
    <def:EventPublisher name="AMXHostJVMInfoEventPublisher">
        <def:CommonLoggingConfigRef ref="JVMnfoEventCLConfig" />
        <def:EventDataSourceRef>
            <def:EventSourceRef ref="AMXHost"/>
            <def:EventSourceRef ref="JVMInfo">
                <def:EventCorrelation returnElement="Process Id">${
AMXHost.Process Id}</def:EventCorrelation>
            </def:EventSourceRef>
        </def:EventDataSourceRef>
        <def:EventOutput>
            <def:EventElements>
                <def:EventElement source="${AMXHost.Name}"
output="physicalCompId.matrix.host" />
                <def:EventElement source="${JVMInfo.Maximum Heap Size}"
output="maxMemory" />
                <def:EventElement source="${JVMInfo.Committed Memory}"
output="totalMemory" />
                <def:EventElement source="${JVMInfo.Free Heap Size}"
output="freeMemory" />
                <def:EventElement source="${JVMInfo.Non Heap Memory Used}"
output="nonHeapMemoryInUse" />
                <def:EventElement source="${JVMInfo.Start Time}"
output="startTime" />
                <def:EventElement source="${JVMInfo.Up Time}" output="uptime" />
                <def:EventElement source="${JVMInfo.Live Threads}"
output="threadCount" />
                <def:EventElement source="${JVMInfo.Daemon Threads}"
output="daemonThreadCount" />
                <def:EventElement source="${JVMInfo.Peak Threads}"
output="peakThreadCount" />
            </def:EventElements>
        </def:EventOutput>
        <def:EventSubscribe sourceEventId="AMXHost" autoStart="false">
            <def:IntervalTime>10000</def:IntervalTime>
        </def:EventSubscribe>
    </def:EventPublisher>
    <!--AMX Node(s) ProcessEventInfo Publisher -->
    <def:EventPublisher name="AMXNodeProcessInfoEventPublisher">

```

```

        <def:CommonLoggingConfigRef ref="ProcessCLConfig" />
        <def:EventDataSourceRef>
            <def:EventSourceRef ref="AMXNodeInfo">
                <def:InputParameters>
                    <def:InputParameter name="Node Name"></def:InputParameter>
                </def:InputParameters>
            </def:EventSourceRef>
            <def:EventSourceRef ref="Process">
                <def:EventCorrelation returnElement="ID Process">${
{AMXNodeInfo.Process Id}</def:EventCorrelation>
            </def:EventSourceRef>
        </def:EventDataSourceRef>
        <def:EventOutput>
            <def:EventElements>
                <def:EventElement source="${Process.ID Process}"
output="processId" />
                <def:EventElement source="${Process.Process Name}"
output="processName" />
                <def:EventElement source="${Process.User Name}"
output="userName" />
                <def:EventElement source="${Process.CPU Time}"
output="processCpuTime" />
                <!-- On Windows platform Virtual KBytes,Stack KBytes,Heap
KBytes, % CPU and % Memory not available-->
                <def:EventElement source="${Process.Virtual KBytes}"
output="virtualMemory" />
                <def:EventElement source="${Process.Stack KBytes}"
output="stackSize" />
                <def:EventElement source="${Process.Heap KBytes}"
output="heapSize" />
                <def:EventElement source="${Process.% CPU}"
output="cpuUsageInPercent" />
                <def:EventElement source="${Process.% Memory}"
output="memoryUsageInPercent" />
                <def:EventElement source="${Process.Start time}"
output="startTime" />
                <def:EventElement source="${AMXNodeInfo.Name}"
output="physicalCompId.matrix.node" />
            </def:EventElements>
        </def:EventOutput>
        <def:EventSubscribe sourceEventId="AMXNodeInfo" autoStart="false">
            <def:IntervalTime>10000</def:IntervalTime>
        </def:EventSubscribe>
    </def:EventPublisher>
</def:EventPublishers>
<!-- Event Data Source Details -->
<def:EventDataSource>
    <def:EventSource id="Process" methodName="getProcess"
microAgentName="COM.TIBCO.hawk.hma.Process" />
    <def:EventSource id="AMXHost" methodName="getHostInfo"
microAgentName="com.tibco.hawk.amx.AMXHost" />
    <def:EventSource id="AMXNodeInfo" methodName="getNodeInfo"
microAgentName="com.tibco.hawk.amx.AMXHost" />
    <def:EventSource id="JVMInfo" methodName="getVirtualMachineInfo"
microAgentName="com.tibco.hawk.jvm.JavaVirtualMachine" />
</def:EventDataSource>
<!-- Common Logging Configuration -->
<def:CommonLoggingConfigs logConfigFile="%TIBCO_HAWK_HOME_ESC%/plugin/
commonlogging/sample_log4j.xml">
    <def:CommonLoggingConfig id="ProcessCLConfig"
loggerName="hawk.clevent.logger">
        <def:ExtendedModel
packageName="com.tibco.governance.cl.extension.model.process.impl.ProcessInfoEventPa
ckageImpl"
name="com.tibco.governance.cl.extension.model.process.impl.ProcessInfoEventImpl" />
    </def:CommonLoggingConfig>
    <def:CommonLoggingConfig id="JVMnfoEventCLConfig"
loggerName="hawk.clevent.logger">
        <def:ExtendedModel
packageName="com.tibco.governance.jvmnfo.model.jvmnfoevent.impl.JvmnfoeventPackag
eImpl"

```

```

name="com.tibco.governance.jvminfo.model.jvminfoevent.impl.JVMInfoEventImpl">
    <def:Classpath>
        <def:path>%TIBCO_COMPONENT_STORE_PLUGINS%/
com.tibco.governance.jvminfo.model_1.0.0.004.jar</def:path>
        <def:path>%TIBCO_COMPONENT_STORE_PLUGINS%/
com.tibco.governance.commonlogging.extension.jvminfo.client_1.0.0.004.jar</def:path>
    </def:Classpath>
    </def:ExtendedModel>
    </def:CommonLoggingConfig>
    </def:CommonLoggingConfigs>
</def:EventDefinitions>

```

Sample Common Logging Configuration file

File: <HAWK_HOME>/plugin/commonlogging/sample_log4j.xml

Content:

```

<?xml version="1.0" encoding="UTF-8"?>
<main_configuration>
    <base_hierarchy>
        <log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
            <appender name="CommonLoggingJMSAppender"
class="com.tibco.commonlogging.appender.BEFJMSAppender">
                <param name="serverUrl" value="tcp://localhost:7222" />
                <param name="queueName" value="amx.governance.stats" />
                <param name="userName" value="admin" />
                <param name="password" value="" />
                <param name="type" value="direct" />
            </appender>
            <logger name="hawk.clevent.logger" additivity="false">
                <appender-ref ref="CommonLoggingJMSAppender"/>
            </logger>
            <logger name="root" additivity="false">
                <level value="INFO" />
                <appender-ref ref="CommonLoggingJMSAppender" />
            </logger>
        </log4j:configuration>
    </base_hierarchy>
</main_configuration>

```

Refer to the <HAWK_HOME>/plugin/commonlogging folder for configuration details.

ActiveMatrix Service Microagent

The ActiveMatrix Service Microagent (com.tibco.hawk.amx.AMXService) allows runtime monitoring of the ActiveMatrix components and services on each local node.

Methods

The methods in the microagent allows to change monitoring configurations and returns components and bindings information.

getConfig

The getConfig method provides monitoring configuration details.

Type

Synchronous, IMPACT_INFO.

Arguments

None.

Returns

Name	Type	Description
Node	String	The name of the node.
AMXHost	String	The name of the ActiveMatrix host associated with the node.
Environment	String	The name of the environment.
Monitoring Application State	String	Runtime state of Monitoring Application.  To get service execution statistics, monitoring application must be in RUNNING state.
Notification Enabled	Long	Returns true or false for statistics notification enable or disable
Notification Interval	Long	Notification Interval in milliseconds.
Rolling Window Length	Long	Rolling window length in milliseconds.
Number of Buckets	Long	Number of time intervals in rolling window.
Statistics Gathering Interval	Long	Statistics gathering interval time in milliseconds.
Is Partial Window	Boolean	Indicates whether the sliding window statistics includes the current working bucket.

changeMonitoringConfig

The changeMonitoringConfig method provides option to change monitoring configuration.

User can change rolling window length, statistics gathering interval (bucket size), notification interval and isPartialWindow. This modification overrides configuration details provided by monitoring application and reset after monitoring application restart.

Type

Synchronous, IMPACT_ACTION.

Arguments

Name	Type	Description
Rolling Window Length (in seconds)	Long	Select rolling window length in seconds. Possible values are: 60, 120, 180, 240, 300.
Statistics Gathering Interval (in seconds)	Long	Selects statistics gathering interval time in seconds. The content of list box are: 5, 10, 15, 20, 30, 40, 50, 60.
Notification Interval (in seconds)	Long	Selects notification interval in seconds. The content of list box are: 10, 15, 20, 30, 40, 50, 60.
Is Partial Window	Boolean	Select true/false for the sliding window statistics include the current working bucket? Possible values are: true, false.

Returns

None.

getComponentInfo

The getComponentInfo method returns Component information by Component and Application name. It serves as a regular expression used to filter the components returned.

Type

Synchronous, IMPACT_INFO.

Arguments

Name	Type	Description
Application Name	String	A substring that matches the application name. An empty string (the default) matches all names. The substring provided can also be a regular expression.

Name	Type	Description
Component Name	String	A substring that matches the component name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Runtime State	String	Select Runtime Possible values are: • Blank: All • RUNNING • NOT_RUNNING • STOPPED • LOST_CONTACT • INSTALLED • UNINSTALLED

Returns

Type: `COM.TIBCO.hawk.talon.TabularData`

Element Name	Type	Description
AMX Host	String	The name of the ActiveMatrix host associated with the node.
Node	String	The name of the node.
Environment	String	The name of the environment.
Application	String	The name of the application.
Component	String	The name of the component.
Runtime State	String	Runtime state of the component. • RUNNING • NOT_RUNNING • STOPPED • LOST_CONTACT • INSTALLED • UNINSTALLED • START_FAILED • WAITING_FOR_DEPENDENCIES
Version	String	Version of the component.

Element Name	Type	Description
URI	String	URI of the component
Time Stamp	String	Time stamp of the changed runtime state of the component.



Application Key generated from Application Name, if application deployed only in one environment without any space in words of Application, then Application Key is same of Application Name. But if application contains spaces then spaces replaced with “_” underscore. If same application deployed across the environment with same name then Application name postfix with incremental of one.

getBindingInfo

The getBindingInfo method returns service and reference binding information by binding and application name. It serves as the regular expression used to filter the bindings returned.

Type

Synchronous, IMPACT_INFO.

Arguments

Name	Type	Description
Application Name	String	A substring that matches the application name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
BindingName	String	A substring that matches the name of the binding. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Runtime State	String	Select Runtime Possible values are: • Blank: All • RUNNING • NOT_RUNNING • STOPPED • LOST_CONTACT • INSTALLED • UNINSTALLED

Returns

Type: `COM.TIBCO.hawk.talon.TabularData`

Element Name	Type	Description
AMX Host	String	ActiveMatrix host Name
Node	String	ActiveMatrix Node Name
Environment	String	Environment Name
Application	String	Application Name
Binding	String	Binding Name
Binding Type	String	Binding Type (Service or Reference)
Contract	String	Contract name (that is, Service or Reference name)
Runtime State	String	Runtime state of binding • RUNNING • NOT_RUNNING • STOPPED • LOST_CONTACT • INSTALLED • UNINSTALLED • START_FAILED • WAITING_FOR_DEPENDENCIES
Binding URI	String	Binding URI
Time Stamp	String	Time stamp of runtime state change of binding.

onComponentEvent

The onComponentEvent method provides notification when the runtime state of component is changed.

Notification can be filtered by component and application name and it serves as the regular expression used to filter the components returned.

Type

Asynchronous, IMPACT_INFO.

Arguments

Name	Type	Description
Application Name	String	A substring that matches the application name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Component Name	String	A substring that matches the component name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Runtime State	String	Select Runtime Name Possible values : • Blank: All • RUNNING • NOT_RUNNING • STOPPED • LOST_CONTACT • INSTALLED • UNINSTALLED

Return

Type: `COM.TIBCO.hawk.talon.CompositeData`

Element Name	Type	Description
AMX Host	String	The name of the ActiveMatrix host.
Node	String	Name of the node.
Environment	String	The name of the environment.
Application	String	A substring that matches the application name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Component	String	A substring that matches the component name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Binding Type	String	Binding type (Service or Reference)

Element Name	Type	Description
Runtime State	String	Runtime state of binding • RUNNING • NOT_RUNNING • STOPPED • LOST_CONTACT • INSTALLED • UNINSTALLED • START_FAILED • WAITING_FOR_DEPENDENCIES
Component URI	String	Component URI.
Time Stamp	String	Time stamp of runtime state change of the binding.

onBindingEvent

The onBindingEvent method provides notification when the runtime state of service or reference binding is changed.

Notification can be filtered by binding and application name and it serves as the regular expression used to filter the bindings returned.

Type

Asynchronous, IMPACT_INFO

Arguments

Name	Type	Description
Application Name	String	A substring that matches the application name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Binding Name	String	A substring that matches the name of the service or reference binding. An empty string (the default) matches all names. The substring provided can also be a regular expression.

Name	Type	Description
Runtime State	String	Select Runtime Name Possible values : • Blank : All • RUNNING • NOT_RUNNING • STOPPED • LOST_CONTACT • INSTALLED • UNINSTALLED

Return

Type: `COM.TIBCO.hawk.talon.CompositeData`

Element Name	Type	Description
AMX Host	String	The name of the ActiveMatrix host.
Node	String	Name of the Node.
Environment	String	The name of the environment.
Application	String	A substring that matches the application name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Binding	String	A substring that matches the name of the binding. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Binding Type	String	Binding type (Service or Reference)
Contract	String	Contract name (that is, Service or Reference name)

Element Name	Type	Description
Runtime State	String	Runtime state of binding • RUNNING • NOT_RUNNING • STOPPED • LOST_CONTACT • INSTALLED • UNINSTALLED • START_FAILED • WAITING_FOR_DEPENDENCIES
Binding URI	String	Binding URI.
Time Stamp	String	Time stamp of runtime state change of the binding.

getBindingOperationStats

The `getBindingOperationStats` method provides statistics of service and reference binding operation execution by operation name, service or reference binding name, application name and binding type. It serves as the regular expression used to filter the operations statistics returned.

Type

Synchronous, `IMPACT_INFO`.

Arguments

Name	Type	Description
Application Name	String	A substring that matches the application name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Binding Name	String	A substring that matches the name of the binding. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Operation Name	String	A substring that matches the operation name. An empty string (the default) matches all names. The substring provided can also be a regular expression.

Name	Type	Description
Binding Type	String	Select type. Possible values are : • <none>: All service and reference bindings. • SERVICE: Only service binding. • REFERENCE: Only reference bindings.

Return

Type: `COM.TIBCO.hawk.talon.TabularData`

Element Name	Type	Description
AMX Host	String	ActiveMatrix host name.
Node	String	Node name.
Environment	String	Environment name.
Application	String	Application name.
Binding	String	Service or Reference binding name.
Binding Type	String	Binding type (that is, Component Service or Reference name).
Contract	String	Contract name.
Operation	String	Service operation name
Binding URI	String	Service or Reference binding URI.
Endpoint Direction	String	Endpoint direction (that is, Inbound or Outbound)
Service Execution Stats		Stats. Refer Service Execution Stats .

getComponentOperationStats

The `getComponentOperationStats` method provides statistics of component service and reference operation execution by operation name, component service or reference name, application name and type. It serves as the regular expression used to filter the operations statistics returned.

Type

Synchronous, `IMPACT_INFO`

Arguments

Name	Type	Description
Operation Name	String	A substring that matches the operation name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Contract Name	String	A substring that matches the component service or reference name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Component Name	String	A substring that matches the component name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Application Name	String	A substring that matches the application name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Contract Type	String	Select type. Possible values are: • <none>: All service and reference. • SERVICE: Only Component Services. • REFERENCE: Only Component References.

Return

Type: `COM.TIBCO.hawk.talon.TabularData`

Element Name	Type	Description
AMX Host	String	ActiveMatrix host name.
Node	String	Node name.
Environment	String	Environment name.
Application	String	Application name.
Component	String	Component name.
Contract	String	Contract name.
Contract Type	String	Contract type (that is, Service or Reference)

Element Name	Type	Description
Operation	String	Operation name
URI	String	Component URI.
Service Execution Stats		Stats. Refer Service Execution Stats .

onBindingOperationStats

The onBindingOperationStats method provides notification in each interval period defined in monitoring configuration with statistics of service and reference binding operation for last <n> by operation name, binding name, application name and binding type. It serves as the regular expression used to filter the operations statistics returned.

Type

Synchronous, IMPACT_INFO

Arguments

Name	Type	Description
Operation Name	String	A substring that matches the operation name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Binding Name	String	A substring that matches the name of the service or reference binding. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Application Name	String	A substring that matches the application name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Binding Type	String	Select type. Possible values are: • <none>: All service and reference bindings. • SERVICE: All Service bindings. • REFERENCE: All Reference bindings.

Return

Type: COM.TIBCO.hawk.talon.TabularData

Element Name	Type	Description
AMX Host	String	ActiveMatrix host name.

Element Name	Type	Description
Node	String	Node name.
Environment	String	Environment name.
Application	String	Application name.
Binding	String	Service or Reference binding name.
Binding Type	String	Binding type (that is, Service or Reference
Contract	String	Contract Name.
Operation	String	Service operation name
Binding URI	String	Service or Reference binding URI.
Endpoint Direction	String	Endpoint direction (that is, Inbound or Outbound)
Service Execution Stats		Stats. Refer Service Execution Stats .

onComponentOperationStats

The onComponentOperationStats method provides notification in each interval period defined in monitoring configuration with statistics.

The statistics include component service and reference operation for the last <n> by operation name, component service or reference name, component name, application name and type and it serves as the regular expression used to filter the operations statistics returned.

Type

Synchronous, IMPACT_INFO

Arguments

Name	Type	Description
Operation Name	String	A substring that matches the operation name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Contract Name	String	A substring that matches the component service or reference name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Component Name	String	A substring that matches the component name. An empty string (the default) matches all names. The substring provided can also be a regular expression.

Name	Type	Description
Application Name	String	A substring that matches the application name. An empty string (the default) matches all names. The substring provided can also be a regular expression.
Contract Type	String	Select type. Possible values are: • <none>: All Component Services and References. • SERVICE: All Component Services. • REFERENCE: All Component References.

Return

Type: `COM.TIBCO.hawk.talon.TabularData`

Element Name	Type	Description
AMX Host	String	ActiveMatrix host name.
Node	String	Node name
Environment	String	Environment name.
Application	String	Application name
Component	String	Component name
Contract	String	Contract name
Contract Type	String	Contract type (that is, Service or Reference)
Operation Name	String	Service operation name
URI	String	Component URI.
Service Execution Stats		Stats. Refer to Service Execution Stats .

Service Execution Stats

The statistics-fetching method of ActiveMatrix Service Microagent returns different statistics for the service executed.

Column Name	Type	Description
Start Time	String	The start time of the collection period ISO-8601 format.

Column Name	Type	Description
End Time	String	The end time of the collection period ISO-8601 format.
Total Time	Long	Collection period in milliseconds.
Hits	Double	Number of execution responses for last period.
Faults	Double	Number of execution faults for last period.
Pending Responses	Double	Difference between count of requests and responses at end of the window.
Avg. Response Time	Double	Average response time for collection period.
Min Response Time	Double	Minimum response time over collection period.
Max Response Time	Double	Maximum response time over collection period.
Hits Per Minute	Double	Average number of executions per minute over collection period.
Faults Per Minute	Double	Average number of faults per minute over collection period.
Fault Percent	Double	Ratio of faults to total hits between 0.0 and 100.0