

# **TIBCO Hawk® – Container Edition**

## **Microagent Reference Guide**

*Software Release 2.0*  
*March 2019*

## 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.

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE OF THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

This document 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, the TIBCO logo, Two-Second Advantage, TIB, Information Bus, Rendezvous, TIBCO Rendezvous, TIBCO Hawk, TIBCO Hawk Container Edition, TIBCO Hawk Microagent for TIBCO BusinessWorks Container Edition, TIBCO BusinessWorks Container Edition, and TIBCO LogLogic Log Management Intelligence (LMI) are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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. Please see the readme.txt 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.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2018-2019. TIBCO Software Inc. All Rights Reserved.

# Contents

---

<b>TIBCO Documentation and Support Services</b> .....	<b>7</b>
<b>Microagent Overview</b> .....	<b>8</b>
<b>Common Methods</b> .....	<b>9</b>
The getReleaseVersion Method .....	9
The getTraceLevel Method .....	9
The setTraceLevel Method .....	10
The getTraceParameters Method .....	11
The setTraceParameters Method .....	11
The _onUnsolicitedMsg Method .....	12
<b>The COM.TIBCO.hawk.microagent.Self Microagent</b> .....	<b>14</b>
The Self:getMicroAgentInfo Method .....	14
The Self:getAgentTimeZone Method .....	15
The Self:getSecurityInfo Method .....	16
The Self:getComponentInfo Method .....	16
The Self:turnDiagnosticsOn Method .....	16
The Self:turnDiagnosticsOff Method .....	17
The Self:getUptime Method .....	17
The Self:doAMIDiscovery Method .....	18
The Self:onMicroAgentEvent Method .....	18
<b>The COM.TIBCO.hawk.microagent.SysInfo Microagent</b> .....	<b>20</b>
The SysInfo:getOperatingSystem Method .....	20
The SysInfo:getArchitecture Method .....	20
The SysInfo:getHostName Method .....	21
The SysInfo:getNetworkAddress Method .....	21
<b>The COM.TIBCO.hawk.microagent.RuleBaseEngine Microagent</b> .....	<b>22</b>
The RuleBaseEngine:deleteRuleBase Method .....	23
The RuleBaseEngine:getExternalVariables Method .....	23
The RuleBaseEngine:getRuleBaseNames Method .....	24
The RuleBaseEngine:getScheduleNames Method .....	24
The RuleBaseEngine:loadRuleBase Method .....	25
The RuleBaseEngine:loadRuleBaseFromFile Method .....	25
The RuleBaseEngine:unloadRuleBase Method .....	25
The RuleBaseEngine:getConfigInfo Method .....	26
The RuleBaseEngine:sendMail Method .....	26
The RuleBaseEngine:onAlertCount Method .....	27
The RuleBaseEngine:onAlertCountForRulebase Method .....	28

The RuleBaseEngine:suspendAlert Method .....	28
The RuleBaseEngine:suspendAllAlerts Method .....	29
The RuleBaseEngine:resumeSuspendedAlerts Method .....	30
The RuleBaseEngine:updateExternalVariables Method .....	30
<b>The COM.TIBCO.hawk.microagent.tcpdaemon.TcpClusterStatus Microagent .....</b>	<b>31</b>
The TcpClusterStatus:getDaemonStatus Method .....	31
The TcpClusterStatus:getClusterStatus Method .....	31
The TcpClusterStatus:getMemberCount Method .....	32
The TcpClusterStatus:onMemberEvent Method .....	32
<b>The COM.TIBCO.hawk.microagent.TCPMessaging Microagent .....</b>	<b>34</b>
The TcpMessaging:sendMessage Method .....	34
The TcpMessaging:onMessage Method .....	34
The TcpMessaging:onNumber Method .....	35
The TcpMessaging:onTimeout Method .....	35
<b>The COM.TIBCO.hawk.microagent.DockerHostMA Microagent .....</b>	<b>37</b>
The DockerHostMA:getContainerInfo Method .....	38
The DockerHostMA:getContainerStats Method .....	39
The DockerHostMA:onContainerLogs Method .....	39
The DockerHostMA:getContainerProcessInfo Method .....	40
The DockerHostMA:startContainer Method .....	40
The DockerHostMA:stopContainer Method .....	41
The DockerHostMA:restartContainer Method .....	41
The DockerHostMA:killContainer Method .....	42
The DockerHostMA:pauseContainer Method .....	42
The DockerHostMA:unpauseContainer Method .....	42
The DockerHostMA:removeContainer Method .....	43
The DockerHostMA:getDockerImages Method .....	43
The DockerHostMA:getDockerVolumes Method .....	44
<b>The COM.TIBCO.hawk.microagent.uc.UniversalCollectorMicroAgent Microagent .....</b>	<b>45</b>
The UniversalCollectorMicroAgent:addDockerLogCollector Method .....	45
The UniversalCollectorMicroAgent:addForwarder Method .....	46
The UniversalCollectorMicroAgent:addLogFileCollector Method .....	46
The UniversalCollectorMicroAgent:addRulebaseDataCollector Method .....	47
The UniversalCollectorMicroAgent:getCollector Method .....	47
The UniversalCollectorMicroAgent:getForwarder Method .....	48
The UniversalCollectorMicroAgent:removeCollector Method .....	49
The UniversalCollectorMicroAgent:removeForwarder Method .....	49
The UniversalCollectorMicroAgent:sendMessage Method .....	49
<b>The COM.TIBCO.hawk.microagent.Custom Microagent .....</b>	<b>51</b>

The Custom:execute Method ..... 51

The Custom:executeForNumber Method ..... 51

The Custom:executeForString Method ..... 52

# TIBCO Documentation and Support Services

---

## 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. To access the latest documentation, visit <https://docs.tibco.com>.

## Product-Specific Documentation

The following documents for this product is available on the [Hawk<sup>®</sup> Container Edition](#) product documentation page.

- *TIBCO Hawk<sup>®</sup> Container Edition Release Notes*
- *TIBCO Hawk<sup>®</sup> Container Edition Installation and Configuration Guide*
- *TIBCO Hawk<sup>®</sup> Container Edition Microagent Reference Guide*

## How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

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

## How to Join TIBCO Community

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

# Microagent Overview

---

The TIBCO Hawk agent uses processes known as Hawk microagents to collect information and operate using that information. These microagents execute specific tasks known as methods. Methods collect information and perform tasks. Methods can be called directly or from within a TIBCO Hawk rulebase.

This manual describes how various Hawk microagents and respective methods can be used for system and application monitoring. It also provides a complete list of all the available methods for the respective microagents. You can use this guide as a reference to know all the available methods for any microagent.

## Microagent Types

TIBCO Hawk microagents are classified by the functions they perform and can be one of three types:

- `IMPACT_INFO` returns information.
- `IMPACT_ACTION` performs an action on the TIBCO Hawk system.
- `IMPACT_ACTION_INFO` both returns information and takes an action based on it.

## Method Output

Methods can return information by invocation or subscription. Invocation returns information on a one-time basis. Subscription information is returned either synchronously, on a regular time schedule, or asynchronously, when data becomes available.

Method invocation returns a table of results, according to specified arguments (if any).



# Common Methods

Some TIBCO Hawk microagent methods are common across all microagents. These common methods performs release and trace-related functions.

## Methods

- [The getReleaseVersion Method](#) on page 9
- [The getTraceLevel Method](#) on page 9
- [The setTraceLevel Method](#) on page 10
- [The getTraceParameters Method](#) on page 11
- [The setTraceParameters Method](#) on page 11
- [The \\_onUnsolicitedMsg Method](#) on page 12

## The getReleaseVersion Method

The `getReleaseVersion` method returns version information about the current release of this microagent. The version comprises a major, minor, and update number separated by periods (for example, 3.1.1). Both a string representation of the version (Version) and an integer representation of the major, minor, and update components of the version (Major, Minor, and Update, respectively) is returned.

### Type

Synchronous, `IMPACT_INFO`.

### Arguments

None

### Returns

Name	Type	Description
Name	String	Name of the binary
Version	String	Version number
Date	String	Version date
Major	Integer	Release version major number
Minor	Integer	Release version minor number
Update	Integer	Release version update number

## The getTraceLevel Method

The `getTraceLevel` method returns the trace level settings.

### Remark

Invocation will return a table, with the setting for each trace level (Information, Warning, Error, Debug, or Application Management Interface(AMI)) listed as being either on or off.

**Type**

Synchronous, `IMPACT_ACTION`.

**Arguments**

None

**Returns**

Name	Type	Description
Information	String	Information trace setting, on or off value
Warning	String	Warning trace setting, on or off value
Error	String	Error trace setting, on or off value
Debug	String	Debug trace setting, on or off value
AMI	String	AMI debug trace setting, on or off value

**The setTraceLevel Method**

The `setTraceLevel` method sets the trace level bits.

**Remarks**

Each trace level can be set to On (enabled), Off (disabled) or Current (leave setting as is). Invocation will return a table, with the setting for each trace level (Information, Warning, Error, Debug, or AMI) listed as being either On or Off.

**Type**

Synchronous, `IMPACT_ACTION_INFO`.

**Arguments**

Name	Type	Description
Information	String	Information trace setting, on or off value
Warning	String	Warning trace setting, on or off value
Error	String	Error trace setting, on or off value
Debug	String	Debug trace setting, on or off value
AMI	String	AMI debug trace setting, on or off value

**Returns**

Name	Type	Description
Information	String	Current information trace setting, on or off value
Warning	String	Current warning trace setting, on or off value
Error	String	Current error trace setting, on or off value
Debug	String	Current debug trace setting, on or off value
AMI	String	Current AMI debug trace setting, on or off value

**The getTraceParameters Method**

The `getTraceParameters` method returns tracing parameter information.

**Type**

Synchronous, `IMPACT_INFO`.

**Arguments**

None

**Returns**

Name	Type	Description
Directory	String	Current trace file directory
File Name	String	Current trace file name
Max File Size	Integer	Current trace file maximum size (in kilobytes) before rollover
Max Trace File	Integer	Current maximum number of trace files to keep in the trace directory

**The setTraceParameters Method**

The `setTraceParameters` method sets tracing parameters.

**Remarks**

Use string current for keeping the current setting for string parameters and use integer value 0 for keeping the current setting for integer parameters.

**Type**

Synchronous, `IMPACT_ACTION_INFO`.

### Arguments

Name	Type	Description
Directory	String	Trace file directory where the trace log file should be kept.
File Name	String	Trace file name
Max File Size	Integer	Maximum size (in kilobytes) of trace file before rollover (kilobytes)
Max Trace File	Integer	Maximum number of trace files to keep in trace directory

### Returns

Name	Type	Description
Directory	String	Current trace file directory
File Name	String	Current trace file name
Max File Size	Integer	Current trace file maximum size (in kilobytes) before rollover
Max Trace File	Integer	Current maximum number of trace files to keep in the trace directory

## The \_onUnsolicitedMsg Method

The \_onUnsolicitedMsg method returns any unsolicited notifications that are sent from this microagent. Invoking this method returns the last such message that was received (if any).

### Remarks

An unsolicited message is an information, warning, or error message that is sent directly to the manager.

### Type

Asynchronous, IMPACT\_INFO.

### Arguments

None

### Returns

Name	Type	Description
TYPE	String	One of INFO, WARNING, or ERROR.
TEXT	String	The text of the message
ID	Integer	A number used by the application to identify the message

Name	Type	Description
INBOX	String	An inbox address used to send the message

# The COM.TIBCO.hawk.microagent.Self Microagent

The Self microagent (on all platforms) returns information from and takes actions that affect the TIBCO Hawk agent itself. You use its methods to examine release version and auto-configuration information, to examine active microagents, and to start and stop diagnostic tools you might need when speaking to technical support personnel.

## Remarks

Because the `Self:turnDiagnosticsOn` and `Self:turnDiagnosticsOff` methods can affect the agent's performance, you should only use these methods when asked to do so by TIBCO Hawk technical support.

## Proprietary Methods

The following method of the Self microagent is proprietary. The TIBCO Hawk system does not support direct user invocation of this method from the TIBCO Hawk Display or Console API.

However, this method is used indirectly by the TIBCO Hawk Display when interacting with agents and thus has security implications.

Method Name	Impact	Description
describe	Info	This method retrieves microagent descriptors

## Methods

- [The Self:getMicroAgentInfo Method](#)
- [The Self:getAgentTimeZone Method](#)
- [The Self:getSecurityInfo Method](#)
- [The Self:getComponentInfo Method](#)
- [The Self:turnDiagnosticsOn Method](#)
- [The Self:turnDiagnosticsOff Method](#)
- [The Self:getUptime Method](#)
- [The Self:doAMIDiscovery Method](#)
- [The Self:onMicroAgentEvent Method](#)

## The Self:getMicroAgentInfo Method

The `Self:getMicroAgentInfo` method (on all platforms) returns information about the microagents currently active on a TIBCO Hawk agent.

## Remarks

Since applications instrumented with the TIBCO Hawk Application Management Interface (AMI) appear as microagents, this method can also gather information on AMI applications.

## Type

Synchronous, `IMPACT_INFO`.

**Arguments**

Name	Type	Description
Name	String	The microagent name. If no name is specified, the method returns information on all microagents.

**Returns**

Name	Type	Description
Name	String	The microagent name. Table is indexed on the Name field.
Display Name	String	The microagent display name.
Count	Integer	The number of microagent instances with that name on the host.
Help	String	A brief description of the microagent. Retrieved from the microagent.

**The Self:getAgentTimeZone Method**

The `Self:getAgentTimeZone` method returns the Agent Time Zone information.

**Type**

Open, Synchronous, `IMPACT_INFO`.

**Arguments**

None

**Returns**

Name	Type	Description
TimeZone ID	String	The time zone ID of the agent.
TimeZone Name	String	The name of the agent's time zone suitable for presentation to the user in the default locale.
TimeZone offset	Integer	The amount of time (considering daylight saving changes) in milliseconds to add to UTC to get standard time in the current time zone.
TimeZone String	String	The textual representation of the current time zone.

## The Self:getSecurityInfo Method

The `Self:getSecurityInfo` method (on all platforms) returns information about the security policy currently in effect.

### Type

Synchronous, `IMPACT_INFO`.

### Arguments

None

### Returns

Name	Type	Description
Policy Class	String	Name of security policy class
Description	String	Text description of security policy class

## The Self:getComponentInfo Method

The `Self:getComponentInfo` method (on all platforms) returns version information about the components in the current release.

### Type

Synchronous, `IMPACT_INFO`.

### Arguments

Name	Type	Description
Component	String	The component name. If no component is specified, returns information for all components.

### Returns

Name	Type	Description
Name	String	The microagent name. Table is indexed on the Name field
Version	String	The component version
Date	String	The component date

## The Self:turnDiagnosticsOn Method

The `Self:turnDiagnosticsOn` method turns on agent diagnostics.

### Remarks

The `diagnostics` microagent diagnoses problems that appear in the TIBCO Hawk system. Because the `Self:turnDiagnosticsOn` and `Self:turnDiagnosticsOff` methods can affect the agent's



performance, you should only use these methods when asked to do so by TIBCO Hawk technical support.

Diagnostic microagent methods are not described in this documentation set, but could be created as an application. For more information on creating applications, refer to the TIBCO Hawk programming manuals.

### Type

Synchronous, `IMPACT_ACTION`.

### Arguments

None

### Returns

None

## The Self:turnDiagnosticsOff Method

The `Self:turnDiagnosticsOff` method turns off agent diagnostics.

### Remarks

The `diagnostics` microagent is used in diagnosis of problems that appear in the TIBCO Hawk system. Diagnostics microagent methods are not described in this documentation set.

### Type

Synchronous, `IMPACT_ACTION`.

### Arguments

None

### Returns

None

## The Self:getUptime Method

The `Self:getUptime` method returns how long the agent has been running.

### Type

Synchronous, `IMPACT_INFO`.

### Arguments

None

### Returns

Name	Type	Description
Uptime	String	Total days, hours, and minutes the agent has been running.
Total days	String	Total number of days the agent has been running.

Name	Type	Description
Total hours	String	Total number of hours the agent has been running.
Total millisec	Integer	Total number of milliseconds the agent has been running.

## The Self:doAMIDiscovery Method

The `Self:doAMIDiscovery` method re-discovers all AMI instrumented applications.

### Remarks

Since applications instrumented with the TIBCO Hawk Application Management Interface (AMI) appear as microagents, this method can also gather information on AMI applications.

When TIBCO Hawk agent loses contact with an AMI instrumented application due to a network failure, the microagent corresponding to the application will be removed after 90 seconds and will not be re-discovered after network connection is restored. This method when invoked re-discovers all AMI instrumented applications. After network failure, AMI applications will not get rediscovered if using an earlier AMI implementation (prior to TIBCO Hawk 4.5) or if using a custom AMI implementation that does not provide rediscovery.

It is recommended that this method is not invoked frequently, as this will lead to the TIBCO Hawk agent receiving a number of AMI application discovery messages thereby increasing local traffic.

### Type

Synchronous, `IMPACT_ACTION`.

### Arguments

None

### Returns

None

## The Self:onMicroAgentEvent Method

The `Self:onMicroAgentEvent` method reports the change in the status of the microagent.

### Remarks

If the TIBCO Hawk Agent is unable to communicate with the microagent, the status of the microagent is immediately changed to `Unreachable`. After 90 seconds of being inaccessible, the microagent status is changed to `Removed`, after which the TIBCO Hawk agent actually removes the microagent. The status of a newly discovered microagent is `Added`.

### Type

Asynchronous, `IMPACT_INFO`.

### Arguments

Name	Type	Description
Name	String	The name of the microagent.

**Returns**

Name	Type	Description
Name	String	The name of the microagent
Display Name	String	The microagent display name.
Status	String	Can be Unreachable, Removed, Added

# The COM.TIBCO.hawk.microagent.SysInfo Microagent

The `COM.TIBCO.hawk.microagent.SysInfo` microagent identifies a network host. Its methods return name, address, type of computer and operating system.

## Methods

- [The SysInfo:getOperatingSystem Method](#)
- [The SysInfo:getArchitecture Method](#)
- [The SysInfo:getHostName Method](#)
- [The SysInfo:getNetworkAddress Method](#)

## The SysInfo:getOperatingSystem Method

The `SysInfo:getOperatingSystem` method returns operating system information.

### Type

Synchronous, `IMPACT_INFO`.

### Arguments

None.

### Returns

Name	Type	Description
OS Name	String	The operating system name
OS Version	String	The operating system version

## The SysInfo:getArchitecture Method

The `SysInfo:getArchitecture` method returns hardware architecture.

### Type

Synchronous, `IMPACT_INFO`.

### Arguments

None

### Returns

Name	Type	Description
Architecture	String	The architecture name

## The SysInfo:getHostName Method

The `SysInfo:getHostName` method returns the name of the host.

### Type

Synchronous, `IMPACT_INFO`.

### Arguments

None

### Returns

Name	Type	Description
Host Name	String	The host name
Agent Domain	String	The name of the agent domain

## The SysInfo:getNetworkAddress Method

The `SysInfo:getNetworkAddress` method returns the IP address of the host.

### Type

Synchronous, `IMPACT_INFO`.

### Arguments

None.

### Returns

Name	Type	Description
IP Address	String	The IP address of the host. For example: 160.101.26.126.

# The COM.TIBCO.hawk.microagent.RuleBaseEngine Microagent

The RuleBaseEngine microagent returns information from rulebases and takes actions on rulebases that affect the TIBCO Hawk agent itself. RuleBaseEngine methods are used to examine auto-configuration information, to examine active microagents, and to load and unload rulebases (for example in a rulebase-management rulebase).

## Remarks

In prior releases of the TIBCO Hawk software, many of the RuleBaseEngine functions were handled by the Self microagent.

Some methods of the RuleBaseEngine microagent are proprietary. They are listed here for completeness, but cannot be invoked directly.

## Proprietary Methods

The following methods of the RuleBaseEngine microagent are proprietary. The TIBCO Hawk system does not support direct user invocation of these methods from the TIBCO Hawk Display or Console API.

However, these methods are used indirectly by TIBCO Hawk Display when interacting with agents and thus have security implications. Access to proprietary methods with an impact of Action should only be granted to trusted users who are allowed access to all Action methods.

Method Name	Impact	Description
getCommands	Info	Returns the rulebase commands
getRuleBases	Info	Retrieves rulebases from the agent
getSchedules	Info	Returns the Schedules List
getRBMap	Info	Returns the RuleBaseMap file
addRuleBase	Action	Sends rulebases to the agent
updateRuleBase	Action	Updates rulebases on the agent
setSchedules	Action	Sets the schedules list for an agent
setRBMap	Action	This method sets the RuleBaseMap file for an agent
runTest	Action	Runs tests for network queries
onAlertPosted	Info	Provides data for alerts posted by the Rulebase Engine
onAlertCleared	Info	Provides data for alerts cleared by the Rulebase Engine

## Methods

- [The RuleBaseEngine:deleteRuleBase Method](#)
- [The RuleBaseEngine:getExternalVariables Method](#)

- [The RuleBaseEngine:getRuleBaseNames Method](#)
- [The RuleBaseEngine:loadRuleBase Method](#)
- [The RuleBaseEngine:loadRuleBaseFromFile Method](#)
- [The RuleBaseEngine:unloadRuleBase Method](#)
- [The RuleBaseEngine:getConfigInfo Method](#)
- [The RuleBaseEngine:sendMail Method](#)
- [The RuleBaseEngine:onAlertCount Method](#)
- [The RuleBaseEngine:onAlertCountForRulebase Method](#)
- [The RuleBaseEngine:suspendAlert Method](#)
- [The RuleBaseEngine:suspendAllAlerts Method](#)
- [The RuleBaseEngine:resumeSuspendedAlerts Method](#)
- [The RuleBaseEngine:updateExternalVariables Method](#)

## The RuleBaseEngine:deleteRuleBase Method

The `RuleBaseEngine:deleteRuleBase` method deletes the specified rulebase.

### Remarks

The agent will delete the rulebase from its cache. If the agent is configured to run under `auto config` mode it will delete this rulebase from the `auto config` directory as well.

### Type

Synchronous, `IMPACT_ACTION`.

### Arguments

Name	Type	Description
RuleBaseName	String	The name of the rulebase to delete.

### Returns

None

## The RuleBaseEngine:getExternalVariables Method

The `RuleBaseEngine:getExternalVariables` method returns the currently loaded external variables as tabular data in two fields, variable name and value.

### Type

Synchronous, `IMPACT_INFO`.

### Arguments

None

**Returns**

Name	Type	Description
Variable Name	String	The name of the external variable
Value	String	The name of the external variable

**The RuleBaseEngine:getRuleBaseNames Method**

The `RuleBaseEngine:getRuleBaseNames` method returns the names of the currently loaded rulebases.

**Remarks**

Each record of the table identifies one rulebase. Therefore, tests that search for strings within a name (such as NT) will result in separate actions for each rulebase name that matches the criterion.

**Type**

Synchronous, `IMPACT_INFO`.

**Arguments**

None

**Returns**

Name	Type	Description
RuleBaseName	String	The name of each rulebase. Table is indexed on the RuleBaseName field

**The RuleBaseEngine:getScheduleNames Method**

The `RuleBaseEngine:getScheduleNames` method returns the names of the currently loaded schedules.

**Type**

Synchronous, `IMPACT_INFO`.

**Arguments**

None

**Returns**

Name	Type	Description
ScheduleName	String	The name of each schedule. Table is indexed on the ScheduleName field.



## The RuleBaseEngine:loadRuleBase Method

The RuleBaseEngine:loadRuleBase method attempts to find and load the named rulebase.

### Remarks

The agent will search all directories in its manual configuration path for the rulebase file, whether manual or automatic configuration is being used.

### Type

Synchronous, IMPACT\_ACTION.

### Arguments

Name	Type	Description
RuleBaseName	String	The name of the rulebase.

### Returns

None

## The RuleBaseEngine:loadRuleBaseFromFile Method

The RuleBaseEngine:loadRuleBaseFromFile method attempts to find and load the rulebase in the named file. An absolute file path is required.

### Type

Synchronous, IMPACT\_ACTION.

### Arguments

Name	Type	Description
File	String	The absolute path name of the file from which to load rulebases.

### Returns

None

## The RuleBaseEngine:unloadRuleBase Method

The RuleBaseEngine:unloadRuleBase method attempts to unload the named rulebase.

### Remarks

If the rulebase is not currently loaded then no error is returned.

### Type

Synchronous, IMPACT\_ACTION.

**Arguments**

Name	Type	Description
RuleBaseName	String	The name of the rulebase.

**Returns**

None

**The RuleBaseEngine:getConfigInfo Method**

The `RuleBaseEngine:getConfigInfo` method returns information on how the rulebase is currently configured.

**Type**

Synchronous, `IMPACT_INFO`.

**Arguments**

None

**Returns**

Name	Type	Description
Name	String	Name of rulebase
Reason Loaded	String	Reason for loading rulebase, such as auto-config startup
Loaded From	String	Source of rulebase
Load Time	String	Time rulebase was loaded, in format of day of week, month, day, time, year

**The RuleBaseEngine:sendMail Method**

The `RuleBaseEngine:sendMail` method sends an email notification.

**Remarks**

The `To` and `Subject` fields are only mandatory fields and all other fields are optional. If `From` is not specified, the current host ID is used. If the `Content` field is blank, the text of the `Subject` line is used. If the `Mail Server` is not specified, then SMTP server configured in the agent sends email.

Rulebases can send mail upon detecting a specified condition.

**Type**

Synchronous, `IMPACT_ACTION`.

## Arguments

Name	Type	Description
To	String	Address of the receiver
CC	String	CC (carbon copy) recipients of email
BCC	String	BCC (blind carbon copy) recipients of email
Subject	String	Subject of email
Content	String	Content of email
Mail Server	String	SMTP mail server used to send message
From	String	Address of the sender

## Returns

None

## The RuleBaseEngine:onAlertCount Method

The `RuleBaseEngine:onAlertCount` method provides the rulebase engine alert rate statistics every 5 seconds. It provides alert rate averages over the previous 5, 10, 15, 30 and 60 seconds.

## Remarks

When using this method as a data source in a rulebase, the test's clear condition will default to "Clear Timer" with a value of 900 seconds. However, because this async method provides statistics every 5 sec, choosing clear condition "First False" may be more useful.

`RuleBaseEngine:onAlertCount` and `RuleBaseEngine:suspendAllAlerts` can be used together to create an alert-rate circuit breaker that insulates the network from agents that produce alerts at message high rates that are too high. For example, using `onAlertCount` as a data source, if (5sec > 10) then `suspendAllAlerts` for 60,000 msec. This would suspend all alerts for 1min if the average rate of alerts over the last 5sec was greater than 2 alerts per sec.

## Type

Asynchronous, `IMPACT_INFO`

## Arguments

None

## Returns

Name	Type	Description
5sec	Integer	The alert rate averaged over the previous 5 seconds
10sec	Integer	The alert rate averaged over the previous 10 seconds

Name	Type	Description
15sec	Integer	The alert rate averaged over the previous 15 seconds
30sec	Integer	The alert rate averaged over the previous 30 seconds
60sec	Integer	The alert rate averaged over the previous 60 seconds

## The RuleBaseEngine:onAlertCountForRulebase Method

The `RuleBaseEngine:onAlertCountForRulebase` method provides the rulebase engine alert rate statistics for a given rulebase every five seconds. It provides alert rate averages over the previous 5, 10, 15, 30 and 60 seconds.

### Remarks

When using this method as a datasource in a rulebase, the clear condition of the test defaults to "Clear Timer" with a value of 900 seconds. However, because this async method provides statistics every five seconds, choosing clear condition "First False" may be more useful.

### Type

Open, Synchronous, `IMPACT_INFO`.

### Arguments

Name	Type	Description
RuleBaseName	String	The name of the rulebase.

### Returns

Name	Type	Description
5sec	Integer	The alert rate averaged over the previous 5 seconds
10sec	Integer	The alert rate averaged over the previous 10 seconds
15sec	Integer	The alert rate averaged over the previous 15 seconds
30sec	Integer	The alert rate averaged over the previous 30 seconds
60sec	Integer	The alert rate averaged over the previous 60 seconds

## The RuleBaseEngine:suspendAlert Method

The `RuleBaseEngine:suspendAlert` method suspends an alert for the specified number of milliseconds. The default is 10000 milliseconds.

### Remarks

When using this method as a datasource in a rulebase, the clear condition of the test defaults to "Clear Timer" with a value of 900 seconds. However, because this async method provides statistics every five seconds, choosing clear condition "First False" may be more useful.

**Type**

Open, Synchronous, IMPACT\_ACTION.

**Arguments**

Name	Type	Description
AlertID	Long	Alert ID
Time	Long	Time in milliseconds
Reason	String	Reason to suspend alert

**Returns**

None

**The RuleBaseEngine:suspendAllAlerts Method**

The `RuleBaseEngine:suspendAllAlerts` method suspends all current and future alerts in the rulebase engine for the specified number of milliseconds.

**Remarks**

If a specific rulebase is provided as an argument, this method applies only the alerts generated by that rulebase.

This method is generally used as an Action to a Rule using `RBE:onAlertRate` as a data source to create an alert-rate circuit breaker.

`RuleBaseEngine:suspendAllAlerts` takes a rulebase argument. If empty, it suspends all alerts on all rulebases. After suspending all alerts, a high level alert (one per rulebase) is issued to indicate that the rulebase has been suspended. That alert will be cleared when the suspension period expires.

When a rulebase is suspended, all its current and future alerts are suspended for the specified interval. Note that all other monitoring activity continues. For example, actions that send email or invoke scripts or other methods continue to operate normally.

**Type**

Open, Synchronous, IMPACT\_ACTION.

**Arguments**

Name	Type	Description
RuleBaseName	String	The name of the rulebase whose alerts will be suspended. If omitted, this method applies to all rulebases.
Time	Long	The suspension period in milliseconds. Cannot be zero or negative.
Reason	String	The reason for this suspension. It is useful to also provide the time and source of suspension in this string. Cannot be empty string.

**Returns**

None.

**The RuleBaseEngine:resumeSuspendedAlerts Method**

The RuleBaseEngine:resumeSuspendedAlerts method restarts all alerts after alert suspension.

**Type**

Asynchronous, IMPACT\_ACTION.

**Arguments**

None.

**Returns**

None.

**The RuleBaseEngine:updateExternalVariables Method**

The RuleBaseEngine:updateExternalVariables method reads the variables file and re-loads all rulebases that use the variables listed in the variable file.

**Type**

Synchronous, IMPACT\_ACTION.

**Arguments**

None.

**Returns**

None.

# The COM.TIBCO.hawk.microagent.tcpdaemon.TcpClusterStatus Microagent

The `TcpClusterStatus` microagent provides methods to monitor the health of the TCP transport cluster and TCP Daemons.

## Methods

- [The `TcpClusterStatus:getDaemonStatus` Method](#)
- [The `TcpClusterStatus:getClusterStatus` Method](#)
- [The `TcpClusterStatus:getMemberCount` Method](#)
- [The `TcpClusterStatus:onMemberEvent` Method](#)

## The `TcpClusterStatus:getDaemonStatus` Method

The `TcpClusterStatus:getDaemonStatus` method returns list of TCP Daemons that are connected to the cluster.

### Type

Open, Synchronous, `IMPACT_INFO`.

### Arguments

Name	Type	Description
Address	String	(Optional) Socket address of the TCP daemon

### Returns

Name	Type	Description
Daemon Address	String	Socket address of the TCP daemon. The socket address is the combination of IP and port number.
Daemon Status	String	Status of the TCP daemon

## The `TcpClusterStatus:getClusterStatus` Method

The `TcpClusterStatus:getClusterStatus` method returns statistics for the cluster.

### Type

Open, Synchronous, `IMPACT_INFO`.

### Arguments

None.

**Returns**

Name	Type	Description
Daemon URL	String	Comma-separated socket address of the TCP daemon. The socket address is the combination of IP and port number.
Total Daemon Count	Integer	Total number of TCP daemons configured for the cluster.
Strategy	String	The member removal strategy applied for the cluster. The values are Quorum and Majority.
Required Daemon Count	Integer	Minimum number of TCP daemons that are required to keep the cluster alive.
Reachable Daemon Count	Integer	Number of reachable TCP daemons in the cluster.

**The TcpClusterStatus:getMemberCount Method**

The `TcpClusterStatus:getMemberCount` method returns total number of members in the cluster (including TCP daemons, Hawk agents, and console applications).

**Type**

Open, Synchronous, `IMPACT_INFO`.

**Arguments**

None.

**Returns**

Name	Type	Description
Daemon Count	Integer	Number of TCP daemons in the cluster.
Agent Count	Integer	Number of Hawk agents present in the cluster.
Console Count	Integer	Number of Console applications in the cluster.
Total Count	Integer	Total number of members in the cluster.

**The TcpClusterStatus:onMemberEvent Method**

The `TcpClusterStatus:onMemberEvent` method returns list of TCP Daemons that are connected to the cluster.

**Type**

Open, Synchronous, `IMPACT_INFO`.



### Arguments

Name	Type	Description
Event Type	String	Types of event, such as, members joining and leaving cluster.

### Returns

Name	Type	Description
Member Address	String	Socket address of the cluster member. The socket address is the combination of IP and port number.
Member Status	Status	Status of the cluster member
Member Type	String	Type of the cluster member

# The COM.TIBCO.hawk.microagent.TCPMessaging Microagent

---

The TCPMessaging microagent provides methods to send and receive messages by using the TCP Transport for TIBCO Hawk.

## Methods

- [The TcpMessaging:sendMessage Method](#)
- [The TcpMessaging:onMessage Method](#)
- [The TcpMessaging:onNumber Method](#)
- [The TcpMessaging:onTimeout Method](#)

## The TcpMessaging:sendMessage Method

The `TcpMessaging:sendMessage` method sends a simple message as a string by using the TCP Transport for TIBCO Hawk.

### Type

Synchronous, `IMPACT_ACTION`.

### Arguments

Name	Type	Description
Topic	String	The message topic
Message	Long	The content of the message

### Returns

None

## The TcpMessaging:onMessage Method

The `TcpMessaging:onMessage` method receives a message by using the TCP Transport for TIBCO Hawk.

### Type

Asynchronous, `IMPACT_INFO`.

### Arguments

Name	Type	Description
Topic	String	The message topic

**Returns**

Name	Type	Description
Topic	String	The message topic
Message	Long	The numeric data

**The TcpMessaging:onNumber Method**

The `TcpMessaging:onNumber` method receives a message having a numeric value.

**Type**

Asynchronous, `IMPACT_INFO`.

**Arguments**

Name	Type	Description
Topic	String	The message topic

**Returns**

Name	Type	Description
Topic	String	The message topic
Message	Long	The numeric data

**The TcpMessaging:onTimeout Method**

The `TcpMessaging:onTimeout` method returns the number of timeout intervals that have elapsed because of the arrival of the previous message or because of the start of the subscription to this method.

**Type**

Asynchronous, `IMPACT_INFO`.

**Arguments**

Name	Type	Description
Topic	String	The message topic
Timeout Interval	Long	Timeout interval in seconds

**Returns**

Name	Type	Description
Timeout Interval Count	Integer	The count of consecutive timeout intervals that have passed.  The return value is reset to 0 (zero) when a message is received with the specified subject or topic.

# The COM.TIBCO.hawk.microagent.DockerHostMA Microagent

The DockerHostMA microagent is available on each Docker host as a part of the Hawk agent. It monitors and manages all containers running on that particular Docker host. The microagent communicates with DockerHost using Docker client API.

For details, see [Docker Client API Documentation](#).

## Environment Variables

Environment Variable	Description
DOCKER_HOST	<p>The DOCKER_HOST environment variable specifies URL for the Docker host.</p> <p>The DOCKER_HOST URL can either be a UNIX socket or an TCP socket.</p> <p>For using the Unix socket, mount the Docker Unix socket into the hkce_agent container as a volume: <code>/var/run/docker.sock:/var/run/docker.sock</code>. Then set DOCKER_HOST to the Unix socket URL:  <code>unix:///var/run/docker.sock</code></p> <p>For using the TCP socket, enable the TCP socket for Docker. For details, see the <code>dockerd</code> command details at <a href="#">Docker Documentation</a>. Then the DOCKER_HOST should be set to the HTTP endpoint of your TCP socket.</p>
DOCKER_CERT_PATH	The DOCKER_CERT_PATH environment variable specifies the location of client certificates for secure communication with the Docker host.

## Methods

- [The DockerHostMA:getContainerInfo Method](#)
- [The DockerHostMA:getContainerStats Method](#)
- [The DockerHostMA:onContainerLogs Method](#)
- [The DockerHostMA:getContainerProcessInfo Method](#)
- [The DockerHostMA:startContainer Method](#)
- [The DockerHostMA:stopContainer Method](#)
- [The DockerHostMA:restartContainer Method](#)
- [The DockerHostMA:killContainer Method](#)
- [The DockerHostMA:pauseContainer Method](#)
- [The DockerHostMA:unpauseContainer Method](#)
- [The DockerHostMA:removeContainer Method](#)
- [The DockerHostMA:getDockerImages Method](#)
- [The DockerHostMA:getDockerVolumes Method](#)

## The DockerHostMA:getContainerInfo Method

The `DockerHostMA:getContainerInfo` method returns information about the container that matches the given `ContainerName`.

### Type

Synchronous, `IMPACT_INFO`

### Arguments

Name	Type	Description
ContainerName	String	The name of the Docker container. If not specified, the method returns information for all running containers.

### Returns

Name	Type	Description
ContainerId	String	The ID of the Docker container
ContainerName	String	The name of the Docker container
ImageId	String	The ID of the Docker container image.
Image	String	Name of the Docker container image.
Ports	String	Container ports as a string.
Command	String	Container command
State	String	Container state
Status	String	Container status
HostnamePath	String	Container hostname path
Hostpath	String	Container hostpath
LogPath	String	Container log path
RestartCount	Long	Number of times the Docker container is restarted.
IsRunning	Boolean	Is container running
IsPaused	Boolean	Is container paused
IsOOMKilled	Boolean	Is container killed due to Out-of memory (OOM) error
IsRestarting	Boolean	Is container restarting
StartedAt	String	Time at which the Docker container started.

Name	Type	Description
FinishedAt	String	Time at which the Docker container finished running.

## The DockerHostMA:getContainerStats Method

The `DockerHostMA:getContainerStats` method returns the resource usage details for the container that matches the given `ContainerName`.

### Type

Synchronous, `IMPACT_INFO`

### Arguments

Name	Type	Description
ContainerName	String	The name of the Docker container.

### Returns

Name	Type	Description
ContainerName	String	The name of the Docker container
ContainerId	String	The ID of the Docker container
%Memory	Double	Memory utilization percentage for the container
%CPU	Double	CPU utilization percentage for the container
Network I/O	String	Network I/O usage for the container
Disk I/O	String	Disk I/O usage for the container

## The DockerHostMA:onContainerLogs Method

The `DockerHostMA:onContainerLogs` method listens for container logs and returns them.

### Type

Asynchronous, `IMPACT_INFO`

### Arguments

Name	Type	Description
ContainerName	String	The name of the Docker container

**Returns**

Name	Type	Description
ContainerId	String	The ID of the Docker container
PrevLine	String	Previous line in log
NextLine	String	Next line in log

**The DockerHostMA:getContainerProcessInfo Method**

The `DockerHostMA:getContainerProcessInfo` method returns the details of processes running inside the container.

**Type**

Synchronous, `IMPACT_INFO`

**Arguments**

Name	Type	Description
ContainerName	String	The name of the Docker container

**Returns**

Name	Type	Description
UID	String	User ID
PID	String	Process ID
PPID	String	Parent process ID
C	String	CPU usage of the process
STIME	String	Start time of the process
TTY	String	Name of the terminal
TIME	String	CPU Time
CMD	String	Process command

**The DockerHostMA:startContainer Method**

The `DockerHostMA:startContainer` method starts the container that matches the given `ContainerName`.

**Type**

Synchronous, `IMPACT_ACTION`



**Arguments**

Name	Type	Description
ContainerName	String	The name of the Docker container

**Returns**

None

**The DockerHostMA:stopContainer Method**

The `DockerHostMA:stopContainer` method stops the container that matches the given `ContainerName`.

**Type**Synchronous, `IMPACT_ACTION`**Arguments**

Name	Type	Description
ContainerName	String	The name of the Docker container
Interval	Integer	Wait interval in seconds before stopping the container

**Returns**

None

**The DockerHostMA:restartContainer Method**

The `DockerHostMA:restartContainer` method restarts the container that matches the given `ContainerName`.

**Type**Synchronous, `IMPACT_ACTION`**Arguments**

Name	Type	Description
ContainerName	String	The name of the Docker container
Interval	Integer	Wait interval in seconds before restarting the container

**Returns**

None

## The DockerHostMA:killContainer Method

The `DockerHostMA:killContainer` method kills the container that matches the given `ContainerName`.

### Type

Synchronous, `IMPACT_ACTION`

### Arguments

Name	Type	Description
ContainerName	String	The name of the Docker container

### Returns

None

## The DockerHostMA:pauseContainer Method

The `DockerHostMA:pauseContainer` method pauses the container that matches the given `ContainerName`.

### Type

Synchronous, `IMPACT_ACTION`

### Arguments

Name	Type	Description
ContainerName	String	The name of the Docker container

### Returns

None

## The DockerHostMA:unpauseContainer Method

The `DockerHostMA:unpauseContainer` method resumes the paused container that matches the given `ContainerName`.

### Type

Synchronous, `IMPACT_ACTION`

### Arguments

Name	Type	Description
ContainerName	String	The name of the Docker container

### Returns

None

## The DockerHostMA:removeContainer Method

The `DockerHostMA:removeContainer` method removes the container that matches the given `ContainerName`.

### Type

Synchronous, `IMPACT_ACTION`

### Arguments

Name	Type	Description
ContainerName	String	The name of the Docker container

### Returns

None

## The DockerHostMA:getDockerImages Method

The `DockerHostMA:getDockerImages` method returns information of the Docker image.

### Type

Synchronous, `IMPACT_INFO`

### Arguments

Name	Type	Description
ImageId	String	The ID of the Docker image

### Returns

Name	Type	Description
ImageId	String	The ID of the Docker image
ParentId	String	The ID of the parent Docker image
RepoTags	String	List of repository names or tags for the Docker image
Created	String	Date when the Docker image created
Size	Long	Size of the Docker image
VirtualSize	Long	Virtual size of Docker image
OS	String	Base operating system of the Docker image
Architecture	String	Architecture of the base operating system

## The DockerHostMA:getDockerVolumes Method

The `DockerHostMA:getDockerVolumes` method returns information about the Docker volume.

### Type

Synchronous, `IMPACT_INFO`

### Arguments

Name	Type	Description
VolumeName	String	The name of the Docker volume

### Returns

Name	Type	Description
VolumeName	String	The name of the Docker volume
Driver	String	Driver of the volume
Mountpoint	String	Mountpoint of the volume
Scope	String	Scope of the volume

# The COM.TIBCO.hawk.microagent.uc.UniversalCollectorMicroAgent Microagent

---

The Universal Collector microagent enables you to monitor the log files, docker container logs, and Hawk rulebases and forward the logs and rulebase data to TIBCO LogLogic® Log Management Intelligence (LMI) or another Syslog server.

The microagent uses collectors and forwarders to distribute the data from different sources to the specified destinations. To forward the data, the microagent uses ULDP for LogLogic LMI and TCP/TLS Syslog for other Syslog consumers.

## Methods

- [The UniversalCollectorMicroAgent:addDockerLogCollector Method](#)
- [The UniversalCollectorMicroAgent:addForwarder Method](#)
- [The UniversalCollectorMicroAgent:addLogFileCollector Method](#)
- [The UniversalCollectorMicroAgent:addRulebaseDataCollector Method](#)
- [The UniversalCollectorMicroAgent:getCollector Method](#)
- [The UniversalCollectorMicroAgent:getForwarder Method](#)
- [The UniversalCollectorMicroAgent:removeCollector Method](#)
- [The UniversalCollectorMicroAgent:removeForwarder Method](#)
- [The UniversalCollectorMicroAgent:sendMessage Method](#)

## The UniversalCollectorMicroAgent:addDockerLogCollector Method

This method adds a new Docker log collector. The Docker log collector collects the logs from the Docker container specified by the **containerName** argument.

### Type

Synchronous, IMPACT\_ACTION

### Arguments

Name	Type	Description
containerName	String	The name of the container to be used for data collection.
Regex	String	Regular expression to match in the container logs. If this argument is provided then the logs matching the regular expression are published. Otherwise, all the new logs are published.
Prefix	String	Prefix to append to logs before publishing them. If this argument is empty then no prefix is added.

### Returns

None

## The UniversalCollectorMicroAgent:addForwarder Method

The method configures a new forwarder for publishing the logs.

### Type

Synchronous, IMPACT\_ACTION

### Arguments

Name	Type	Description
Type	String	The type of forwarder to add. The types of forwarders available are ULDP and TCPSyslog.
Host	String	The host name of forwarder destination
Port	String	The port of forwarder destination
Use TLS	Boolean	Whether to use TLS to publish data
Keystore Path	String	Keystore path
Keystore Password	String	Keystore password. The password can be plain text or can be encrypted by using the <b>tibhawkpassword</b> utility.
TLS Protocol	String	TLS protocol
Cipher Name	String	Cipher name
Use Octet Counting	Boolean	Whether to use octet counting framing. The octet counting helps syslog receiver to decide where the message ends.  This argument is used only when the <b>Type</b> is TCPSyslog and <b>Use TLS</b> is true.

### Returns

None

## The UniversalCollectorMicroAgent:addLogFileCollector Method

The method adds a new log file collector that sends the collected logs to the configured forwarders.

### Type

Synchronous, IMPACT\_ACTION

**Arguments**

Name	Type	Description
Logfile	String	Path of the log file to be monitored
Regex	String	Regular expression to specify data that must be send to a forwarder. Only the logs that match the regular expression are forwarded. When regex is not defined, all new logs are forwarded.
Prefix	String	Add a prefix to the logs before publishing them

**Returns**

None

**The UniversalCollectorMicroAgent:addRulebaseDataCollector Method**

The purpose of this method is to add a new rulebase data collector to collect the data from the data sources configured in the Hawk rulebases.

**Remarks**

The data sources are the microagent methods specified in the Hawk rulebases. The collector subscribes to the method for the same interval that is specified in the Hawk rule and forwards the subscription data to all the configured forwarders.

**Type**

Synchronous, IMPACT\_ACTION

**Arguments**

Name	Type	Description
Rulebase Name	String	The name of the Hawk rulebase whose datasource must be used for collection. If this argument is empty, all the available Hawk rulebases are considered for collection.

**Returns**

None

**The UniversalCollectorMicroAgent:getCollector Method**

This method returns the details about the collector.

**Type**

Synchronous, IMPACT\_INFO

### Arguments

Name	Type	Description
Collector Key	String	The argument returns the information about all the configured collectors when not unspecified.

### Returns

Name	Type	Description
Collector Key	String	The collector key
Data Source	String	The data source of this collector
Logs Collected	Long	All logs collected by this collector
Total Errors	Long	All errors that occurred during collection

## The UniversalCollectorMicroAgent:getForwarder Method

The method returns the details about the forwarder.

### Type

Synchronous, IMPACT\_INFO

### Arguments

Name	Type	Description
Forwarder Key	String	If this argument is empty, information about all the configured forwarders is returned.

### Returns

Name	Type	Description
Forwarder Key	String	The forwarder key
Host	String	The host name of forwarder destination
Port	String	The port of forwarder destination
Using TLS	Boolean	Whether the forwarder is using TLS



## The UniversalCollectorMicroAgent:removeCollector Method

The purpose of this method is to remove the configured collector.

### Type

Synchronous, IMPACT\_ACTION

### Arguments

Name	Type	Description
Collector Key	String	The collector key. Use <code>getCollector</code> method with no argument to retrieve keys for all configured collectors.

### Returns

None

## The UniversalCollectorMicroAgent:removeForwarder Method

This method removes the configured forwarder.

### Type

Synchronous, IMPACT\_ACTION

### Arguments

Name	Type	Description
Forwarder Key	String	The forwarder key. Use the <code>getForwarder</code> method with no argument to retrieve keys for all configured forwarders.

### Returns

None

## The UniversalCollectorMicroAgent:sendMessage Method

The method sends messages to configured forwarders.

### Type

Synchronous, IMPACT\_ACTION

### Arguments

Name	Type	Description
Message	String	The message to be send to forwarders

**Returns**

None

# The COM.TIBCO.hawk.microagent.Custom Microagent

The custom microagent enables you to run executable programs and scripts from within Hawk agent container.

You can retrieve information from a script or program. Custom methods also allow you to write rules that act on that information as though it came from a microagent method call.

## Methods

- [The Custom:execute Method](#)
- [The Custom:executeForNumber Method](#)
- [The Custom:executeForString Method](#)

## The Custom:execute Method

This method executes a command and ignores the result.

If this method call is performed within an action, you can provide parameters for a script or program by using variable substitution. Variable substitution inserts information into the command from the rule data source.

On the OS/400 platform, the supported commands are:

- commands that start with the system (when using a CL command)
- commands that run on the Qshell (qsh) utility. Use the `-c` option when using the `qsh` command.

For example:

```
qsh -c ls
system DSPDTAARA DTAARA (<LibName>/<Data Area Name>)
```

## Type

Synchronous, IMPACT\_ACTION

## Arguments

Name	Type	Description
command	String	The command to execute

## Returns

None



All shell scripts called from your Hawk environment must define the shell, for example, `#!/bin/csh`.

## The Custom:executeForNumber Method

This method executes a command and parses the output as a number which is then returned as both an integer and a double value. The integer value of a true double truncates the remainder.

The command returns information by sending a number to `stdout`. If the executed command returns more than one number, an error occurs.

On the OS/400 platform, the supported commands are:

- commands that start with the system (when using a CL command)
- commands that run on the Qshell (qsh) utility. Use the -c option when using the qsh command.

For example:

```
qsh -c ls
system DSPDTAARA DTAARA (<LibName>/<Data Area Name>)
```

### Type

Synchronous, IMPACT\_ACTION\_INFO

### Arguments

Name	Type	Description
command	String	The command to execute. External and internal variables can be used in the command string.

### Returns

Name	Type	Description
returnInteger	Integer	The integer value of the output.
returnDouble	Double	The double value of the output.

## The Custom:executeForString Method

This method executes a command and returns the output as string.

The executable can return information by sending strings to stdout. If the executable returns more than one line, the lines are concatenated to produce one string.

On the OS/400 platform, the supported commands are:

- commands that start with the system (when using a CL command)
- commands that run on the Qshell (qsh) utility. Use the -c option when using the qsh command.

For example:

```
qsh -c ls
system DSPDTAARA DTAARA (<LibName>/<Data Area Name>)
```

### Type

Synchronous, IMPACT\_ACTION\_INFO

### Arguments

Name	Type	Description
command	String	The command to execute. External and internal variables can be used in the command string.

**Returns**

Name	Type	Description
returnString	String	The string value of the output.