

webMethods EntireX

Administration with Command Central

Version 10.9

April 2023

This document applies to webMethods EntireX Version 10.9 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1997-2023 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: EXX-COMMANDCENTRAL-109-20230403

Table of Contents

1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
2 Introduction to Administering EntireX Broker with Command Central (Linux and Windows)	5
Scope	6
Monitoring EntireX Broker KPIs	7
Configuration Types that EntireX Broker Supports	7
3 Administering EntireX Broker using the Command Central GUI	9
Logging in to Command Central	10
Creating a Broker Instance	11
Configuring a Broker Instance	16
Viewing the Runtime Status	23
Starting a Broker Instance	23
Stopping a Broker Instance	25
Inspecting the Log Files	26
Displaying the Statistics	27
Displaying Services	29
Stopping a Service	30
Displaying Server Instances of a Service	30
Changing the Trace Level Temporarily	32
Changing the Application Monitoring Settings Dynamically	33
Deleting a Broker Instance	34
Troubleshooting	35
4 Administering EntireX Broker using the Command Central Command Line	37
Creating a Broker Instance	38
Configuring a Broker Instance	39
Displaying the EntireX Inventory	61
Viewing the Runtime Status	62
Starting a Broker Instance	63
Stopping a Broker Instance	63
Restarting a Broker Instance	64
Inspecting the Log Files	65
Displaying the Statistics	66
Administering Services	68
Changing the Trace Level Temporarily	71
Changing the Application Monitoring Settings Dynamically	73
Deleting a Broker Instance	74
Troubleshooting	76

1

About this Documentation

■ Document Conventions	2
■ Online Information and Support	2
■ Data Protection	3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

In addition, you can also access the cloud product documentation via <https://www.software-ag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at <https://knowledge.softwareag.com>.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://tech-community.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/publishers/softwareag> and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Introduction to Administering EntireX Broker with Command Central (Linux and Windows)

■ Scope	6
■ Monitoring EntireX Broker KPIs	7
■ Configuration Types that EntireX Broker Supports	7

See also *Administering EntireX Broker using the Command Central GUI* | Command Line.



Note: Command Central functionality that is not EntireX-specific is described in the separate Command Central documentation or the online help provided with Command Central. On Empower, the documentation is provided under webMethods > EntireX > EntireX 10.9 > Additional Documentation.

Scope

This section applies to Broker instances running on Linux and Windows platforms. Since EntireX 10.3 you can also administer EntireX Brokers running on mainframe platforms. See *Introduction to EntireX Mainframe Broker Monitoring*.

For EntireX, you can use Command Central to perform the following operations on EntireX Broker:

- View the EntireX Brokers running in each environment of your IT landscape
- View the versions of EntireX Brokers
- Monitor EntireX Broker installations
- Monitor runtime status, KPIs (key performance indicators), and alerts of EntireX Broker instances
- Start, stop, and restart EntireX Broker
- Configure the following parameters of EntireX Broker:
 - Application Monitoring
 - Attribute File
 - Autostart
 - Monitoring KPIs
 - Persistent Store
 - SSL Ports
 - TCP Ports
 - Security
 - Trace
- Enable and specify EntireX Broker Trace Level temporarily
- Display Services
- Stop Service
- Display Server Instances of a Service
- Stop Server Instance
- Create new EntireX Brokers

■ Delete existing EntireX Brokers

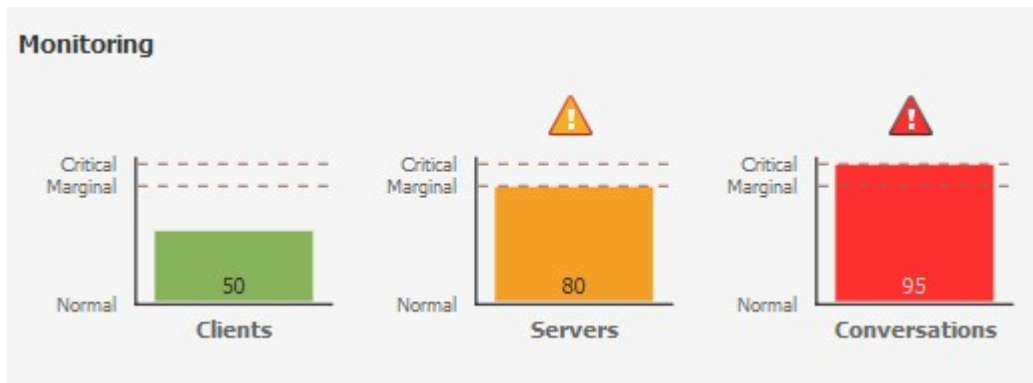


Note: Software AG Platform Manager (installed by default with Command Central) uses EntireX Broker Admin to obtain information about EntireX Brokers, so EntireX Broker Admin must be running if you want to administer EntireX Brokers through Command Central.

For more information see the separate Command Central documentation or the online help provided with Command Central.

Monitoring EntireX Broker KPIs

The visual key performance indicators (KPIs) and alerts enable you to monitor webMethods EntireX Broker's health. The following KPIs help you administer, troubleshoot, and resolve performance issues in EntireX Broker:



KPI	Description
Clients	Number of active clients.
Servers	Number of active servers.
Conversations	Number of active conversations.

Configuration Types that EntireX Broker Supports

The EntireX Broker component supports the configuration instances listed in the following table.

Instance	Type	Use to...
APPLICATION-MONITORING	APPLICATION-MONITORING	Configure Application Monitoring.
ATTRIBUTE-FILE	ATTRIBUTE-FILE	Show and edit the EntireX Broker Attribute file.
BROKER-AUTOSTART	BROKER-AUTOSTART	The autostart value of an EntireX Broker instance determines whether it will be started when the computer is restarted.
SSL	BROKER-PORTS	Show and edit the EntireX Broker SSL Transport parameters.
TCP	BROKER-PORTS	Show and edit the EntireX Broker TCP Transport parameters.
BROKER-PSTORE	BROKER-PSTORE	Define how unit-of-work messages are stored to disk.
BROKER-SECURITY	BROKER-SECURITY	Enable or disable EntireX Broker security.
BROKER-TRACELEVEL	BROKER-TRACELEVEL	Show and edit the EntireX Broker Trace Level.
EXX-MONITORING-KPIS	EXX-MONITORING-KPIS	Show and edit the Monitoring KPI settings, like the marginal and critical bounds, etc.



Note: The default SSL Port configuration uses sample certificates provided with the installation, but we strongly recommend you create your own. This means, however, that you may provide an incorrect certificate in the Command Central graphical user interface. This has the effect that the Monitoring KPIs are not presented correctly and the original communication exception cannot be shown in the GUI. However, the original exception is available in the wrapper.log of the SPM profile. Look for the string pattern "cannot get KPIs! RpcException 0013" in the log file and check the *EntireX Error Messages and Codes* for the original RpcException.

3 Administering EntireX Broker using the Command Central

GUI

■ Logging in to Command Central	10
■ Creating a Broker Instance	11
■ Configuring a Broker Instance	16
■ Viewing the Runtime Status	23
■ Starting a Broker Instance	23
■ Stopping a Broker Instance	25
■ Inspecting the Log Files	26
■ Displaying the Statistics	27
■ Displaying Services	29
■ Stopping a Service	30
■ Displaying Server Instances of a Service	30
■ Changing the Trace Level Temporarily	32
■ Changing the Application Monitoring Settings Dynamically	33
■ Deleting a Broker Instance	34
■ Troubleshooting	35

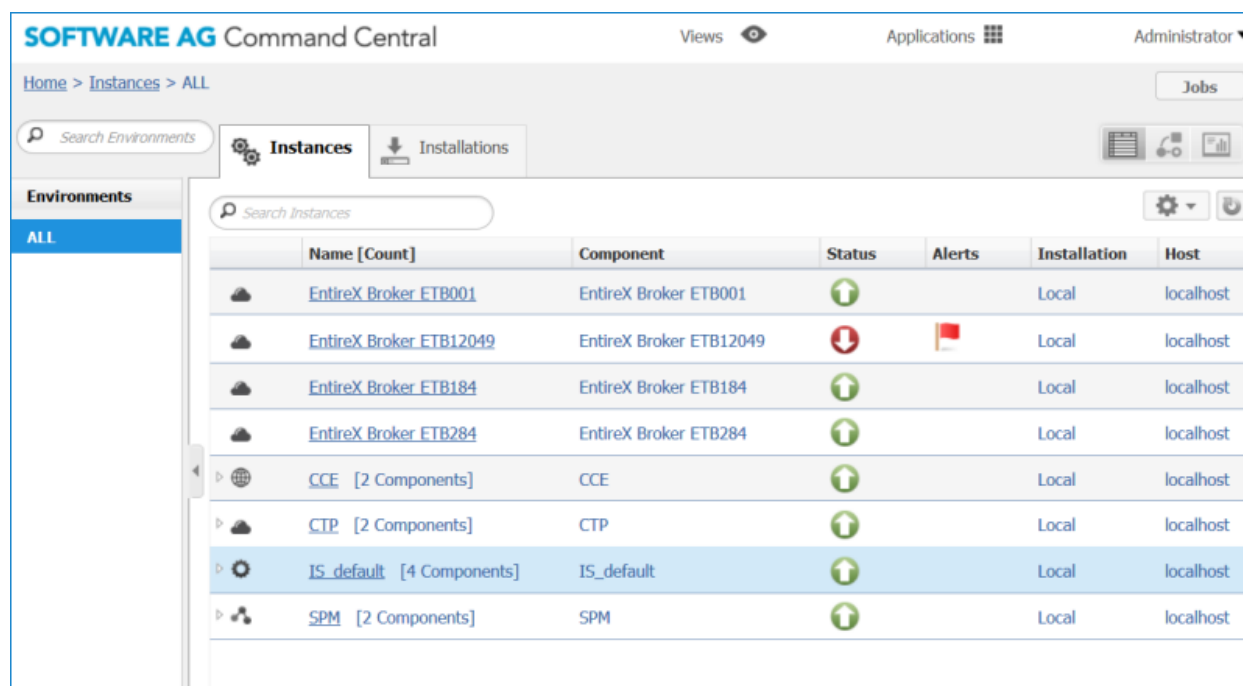
See also [Introduction](#) and [Administering EntireX Broker using the Command Central Command Line](#).

Logging in to Command Central

Open an Internet browser and specify the URL of the Command Central Server as follows: `http://<Command_Central_host>:<Command_Central_port>`. This takes you to the Command Central **Login** page.

On Windows you can also get to the **Login** page from the Command Central **Start Menu** entry.

Provide your user credentials in the **Login** page and click **Log In**. This takes you to the page **Home > Instances**:



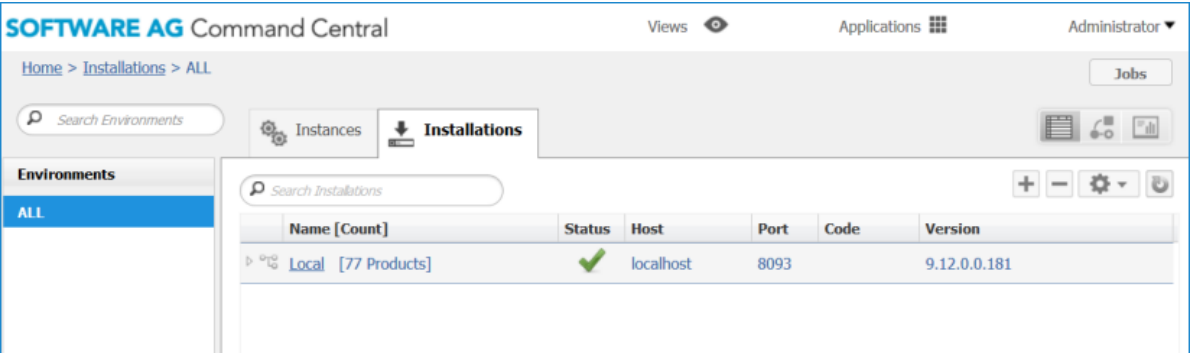
The screenshot shows the 'SOFTWARE AG Command Central' interface. The breadcrumb navigation is 'Home > Instances > ALL'. The left sidebar shows 'Environments' with 'ALL' selected. The main content area has a search bar and a table of instances.

Name [Count]	Component	Status	Alerts	Installation	Host
EntireX Broker ETB001	EntireX Broker ETB001	Up		Local	localhost
EntireX Broker ETB12049	EntireX Broker ETB12049	Down	Alert	Local	localhost
EntireX Broker ETB184	EntireX Broker ETB184	Up		Local	localhost
EntireX Broker ETB284	EntireX Broker ETB284	Up		Local	localhost
CCE [2 Components]	CCE	Up		Local	localhost
CTP [2 Components]	CTP	Up		Local	localhost
IS_default [4 Components]	IS_default	Up		Local	localhost
SPM [2 Components]	SPM	Up		Local	localhost

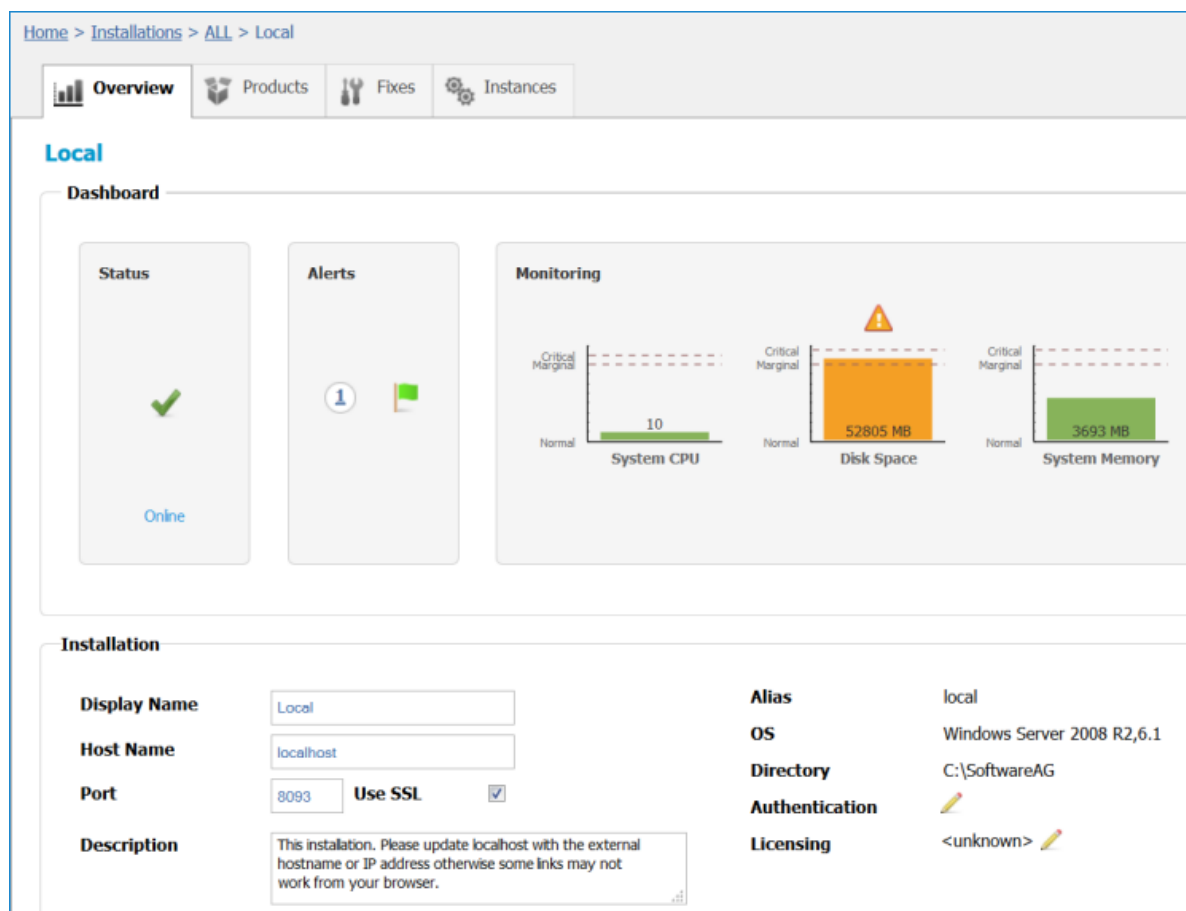
Creating a Broker Instance

> To create a new broker

- 1 In the Command Central **Home** page, click the **Installations** tab.



- 2 Click on the desired installation, for example Local, for which you want to create a new broker instance:



- Click the **Instances** tab.

Home > Installations > ALL > Local

Overview Products Fixes **Instances**

Local

Search Instances

	Name [Count]	Component	Status	Alerts	Installation	Host
	EntireX Broker ETB001	EntireX Broker ETB001	↑		Local	localhost
	EntireX Broker ETB12031	EntireX Broker ETB12031	↓		Local	localhost
	EntireX Broker ETB184	EntireX Broker ETB184	↑		Local	localhost
	EntireX Broker ETB284	EntireX Broker ETB284	↑		Local	localhost
▶	CCE [2 Components]	CCE	↑		Local	localhost
▶	CTP [2 Components]	CTP	↑		Local	localhost
▶	IS_default [4 Components]	IS_default	↑		Local	localhost
▶	SPM [2 Components]	SPM	↑		Local	localhost

- 4 Click the  button in the upper right corner above the list and choose **EntireX Broker**.




Home > Installations > ALL > Local

Overview Products Fixes **Instances**

Local

Search Instances

	Name [Count]	Component	Status	Alerts	Installation	Host
	TCP1971	EntireX Broker ETB001	↑		Local	localhost
	SSL1972	EntireX Broker SSL	↑		Local	localhost
▶	CCE [1 Components]	CCE	↑		Local	localhost
▶	SPM [1 Components]	SPM	↑		Local	localhost



- EntireX Broker
- EntireX RPC Server for C...
- EntireX RPC Server for IBM MQ
- EntireX RPC Server for IMS Con...
- EntireX RPC Server for Java
- EntireX RPC Server for .NET
- EntireX RPC Server for XML SOA...

- 5 Specify the desired properties and press **Next**.

Create Instance - EntireX Broker

1 **Specify Properties** 2 Summary

Instance name *

Transport ☒ TCP ☐ SSL ☐ TCP and SSL ?

TCP port * ?

SSL port * ?


Autostart ☒ ?





Security ☐ ?

Next **Cancel**

Parameter	Description
Broker name	Required. Name of broker.
TCP port	Required if SSL port is not set; optional if SSL port is set. TCP port number in range from 1025 to 65535.
SSL port	Required if TCP port is not set; optional if TCP port is set. SSL port number in range from 1025 to 65535.
Autostart	Enables/disables autostart. Default is no (disabled). If autostart is enabled, the broker starts automatically with the next system start.
Security	Enables/disables EntireX Security. Default is no (disabled).

6 Press **Finish**.

 **Instances**

Component	Status	Alerts
EntireX Broker ETB001		
EntireX Broker ETB12031		
EntireX Broker ETB12047		

Operation triggered

Job operation is started successfully.

View Job

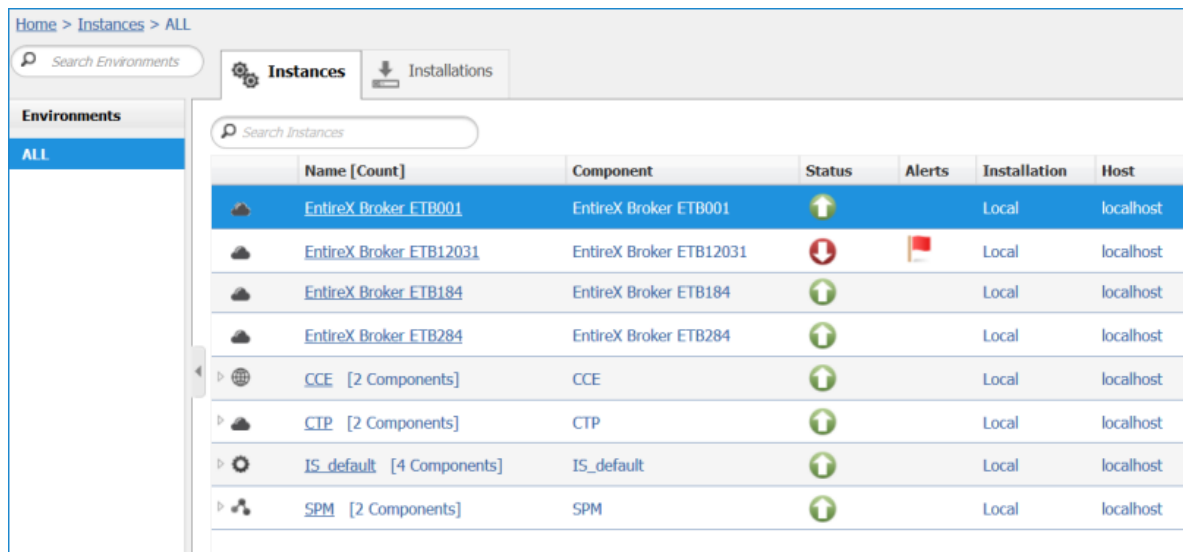
Finish

The new instance MyNewBroker appears in the list.

Configuring a Broker Instance

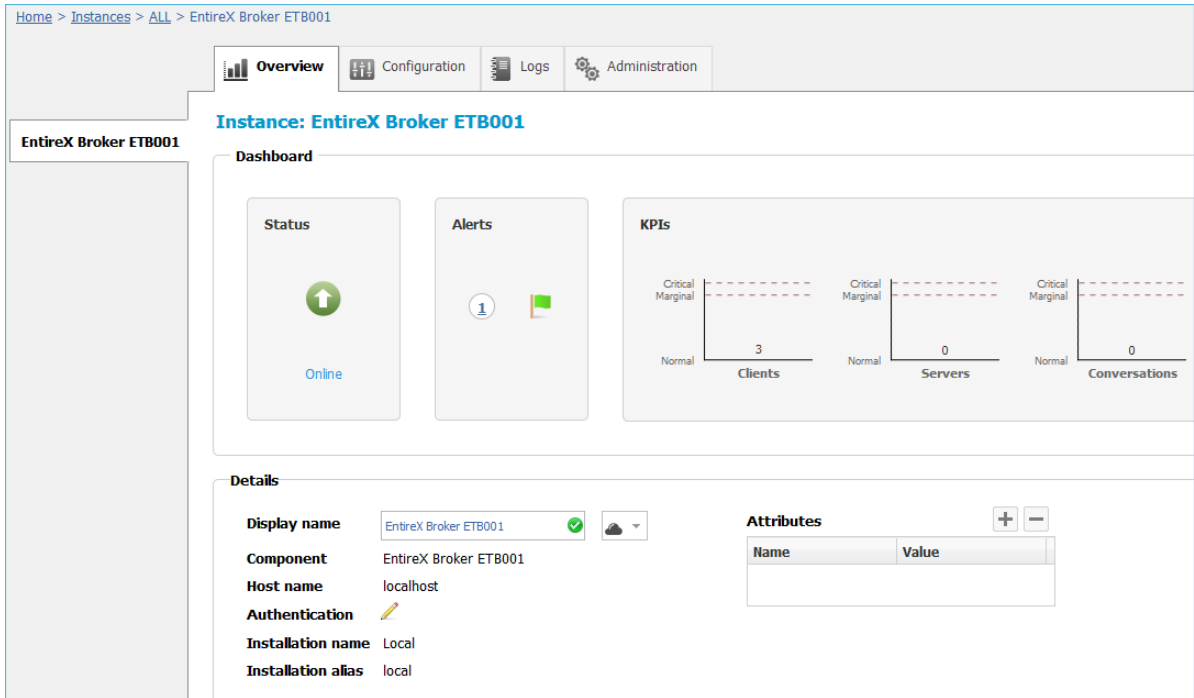
> To configure a Broker instance

- 1 In the Command Central **Home** page, click the **Instances** tab and select the broker instance you wish to configure.



Name [Count]	Component	Status	Alerts	Installation	Host
EntireX Broker ETB001	EntireX Broker ETB001			Local	localhost
EntireX Broker ETB12031	EntireX Broker ETB12031			Local	localhost
EntireX Broker ETB184	EntireX Broker ETB184			Local	localhost
EntireX Broker ETB284	EntireX Broker ETB284			Local	localhost
CCE [2 Components]	CCE			Local	localhost
CTP [2 Components]	CTP			Local	localhost
IS_default [4 Components]	IS_default			Local	localhost
SPM [2 Components]	SPM			Local	localhost

- 2 Click the link associated with the Broker instance you want to configure. This takes you to the following page:



- 3 If you want to monitor a secure broker instance, add under **Authentication** a user name and password combination that is allowed to access and administer the broker by the security system used by the broker. For example:
 - If SECURITY-SYSTEM in the broker attribute file is set to OS (or not specified), enter a user name and password valid for the local operating system where the broker is running.
 - If SECURITY-SYSTEM in the broker attribute file is set to LDAP, enter a valid LDAP user name and password.

See also SECURITY-SYSTEM under *Security-specific Broker Attributes*.



Note: You cannot change credentials accepted by this broker. This has to be done in the security system used by the broker.

The screenshot shows the 'Authentication Mode' dialog box. On the left, the 'Details' tab is active, showing fields for 'Display name' (EntireX Broker ETB001), 'Component' (EntireX Broker ETB001), 'Host name' (localhost), 'Authentication' (highlighted with a red box and a pencil icon), 'Installation name' (Local), and 'Installation alias' (local). The 'Authentication Mode' dialog is open on the right, showing radio buttons for 'System default', 'None', 'Trusted', 'Delegated', and 'Fixed user'. The 'Fixed user' option is selected and highlighted with a red box. Below it are input fields for 'User name' and 'Password'. 'OK' and 'Cancel' buttons are at the bottom right.

- 4 Click the **Configuration** tab. EntireX supports the following configuration types, which are presented in a drop-down box when you click the down arrow below the **Configuration** tab label:

The screenshot shows the 'Configuration' tab selected in the Command Central GUI. The 'Application Monitoring' dropdown menu is open, showing a list of configuration types: 'Application Monitoring' (highlighted), 'Attribute File', 'Autostart', 'Licenses', 'Monitoring KPIs', 'Persistent Store', 'Ports', 'Security', 'Trace Level', and 'Trace Location'. The background shows the 'Overview' and 'Configuration' tabs, and a section titled 'by this EntireX' with a 'No' radio button and some input fields.

- 5 Click **Edit** to modify the parameters. The parameters are described in the tables below.
 - *Application Monitoring*
 - *Attribute File*
 - *Autostart*

- [Licenses](#)
- [Monitoring KPIs](#)
- [Persistent Store](#)
- [Ports](#)
- [Security](#)
- [Trace Level](#)
- [Trace Location](#)

Application Monitoring

Here you can modify the Application Monitoring configuration.

Parameter	Description
Set Data Collector used by this EntireX Broker	
Enabled	Enable/disable application monitoring in EntireX Broker. Default is <code>false</code> (disabled).
Host name	Required. The host name on which the Application Monitoring Data Collector accepts the monitoring data.
Port	Required. The TCP/IP port on which the Application Monitoring Data Collector accepts the monitoring data.

Attribute File

Here you can view/edit the attribute file of the broker. See *Broker Attributes* for description of attributes.

Autostart

Here you can control the start behavior of an EntireX Broker.

Parameter	Description
Autostart	Enables/disables autostart. Default is <code>no</code> (disabled). If autostart is enabled, the broker starts automatically with the next system start.

Licenses

Here you can view/set the license file in the EntireX installation. For details see *Import Product License Keys for Instances or Components* under *Getting Started with Command Central* in the separate Command Central documentation under Software AG Suite & Cross-Product Guides.



Note: The license file is used for all EntireX instances in this installation.

Monitoring KPIs

Here you can modify margins of monitored key performance indicators (KPIs) available for the broker: Clients, Servers and Conversations.

Key performance indicators (KPIs) enable you to monitor the health of your broker. The following KPIs help you administer, troubleshoot, and resolve performance issues:

Parameter	Description
Clients	
Maximum	Maximum number of clients in the overview graph.
Marginal	Marginal limit for numbers of clients in the overview graph.
Critical	Critical limit for numbers of clients in the overview graph.
Servers	
Maximum	Maximum number of servers in the overview graph.
Marginal	Marginal limit for numbers of servers in the overview graph.
Critical	Critical limit for numbers of servers in graph.
Conversations	
Maximum	Maximum number of conversations in the overview graph.
Marginal	Marginal limit for numbers of conversations in the overview graph.
Critical	Critical limit for numbers of conversations in the overview graph.

Persistent Store

Here you can modify the EntireX Broker persistent store configuration.

Parameter	Description
Set Persistent Store	
Enabled	Enable/disable the persistent store in EntireX Broker. Default is <code>false</code> (disabled). See <i>Concepts of Persistent Messaging</i> .
Initialization mode	Defines the status of the persistent store at broker startup, including the condition of persistent units of work (UOWs): COLD - Default. Persistent UOWs are not restored during initialization, and the persistent store is considered empty. HOT - Persistent UOWs are restored to their prior state during initialization.
Type	The type of persistent store driver required: CTREE - Default. c-tree database. ADABAS - Adabas database.
Configure Persistent Store (c-tree)	
Maximum size	Required for type CTREE. The maximum size of the c-tree persistent store in gigabytes (G) or megabytes (M), for example 1G, 2M, 3. If no unit is specified, megabytes are used.

Parameter	Description
Page size	Required for type CTREE. The page size of the c-tree persistent store in kilobytes (K). Valid values: 8 or greater.
Path	Required for type CTREE. The directory where the c-tree persistent store resides. It will be created in the Broker's configuration directory.
Configure Persistent Store (Adabas)	
Database ID	Required for type ADABAS. The ID of the Adabas database where the persistent store resides.
File number	Required for type ADABAS. The file number of the broker persistent store file.

SSL Port

Here you can modify the EntireX Broker SSL port configuration. Select a port in the list for configuring the port.

Parameter	Description
Enabled	Enable/disable the SSL port. Default is <code>false</code> (disabled).
Port number	Required. The port number in range from 1025 to 65535.
Key file	Required. The name of file that contains the broker's private key.
Key password encrypted	Required. The encrypted password used to protect the private key.
Key store	Required. The location of the store containing SSL certificate; may contain the private key.
Trust store	Required. The location of the store containing certificates of trust Certificate Authorities (or CAs).
Client-side trust store	Required. The location of the trust store used by the platform manager.
Client-side trust password	The password of the trust store used by the platform manager.

TCP Port

Here you can modify the EntireX Broker TCP port configuration. Select a port in the list for configuring the port.

Parameter	Description
Enabled	Enable/disable the TCP port. Default is <code>true</code> (enabled).
Port number	Required. The port number in range from 1025 to 65535.

Security

Here you can modify the EntireX Broker security configuration.

Parameter	Description
Set EntireX Broker Security	
Enabled	Enable/disable EntireX Broker Security. Default is <code>false</code> (disabled).
Authentication type	Authentication Type. Operating System - Default. Authentication is performed against the local operating system. LDAP - Authentication and authorization are performed against the LDAP repository.
Configure LDAP for EntireX Broker Security	
Repository URL	Required for authentication type LDAP: URL of the LDAP repository.
Distinguished name	Required for authentication type LDAP: The distinguished name of user for authenticated access to the LDAP server. Default value: <code>cn=admin,dc=software-ag,dc=de</code>

Trace Level

Here you can modify the EntireX Broker trace level configuration.

Parameter	Description
Configure the EntireX Broker trace level	
Trace level	One of the following levels: 0 - Default. No tracing. 1 - Incoming request, outgoing replies, resource usage and conversion errors if SAGTRPC is used. 2 - All of trace level 1, plus all main routines executed. 3 - All of trace level 2, plus all routines executed. 4 - All of trace level 3, plus EntireX Broker ACI control block displays.

Trace Location

Here you can modify the EntireX Broker trace location configuration. This specifies the directory that contains the EntireX Broker trace file.

Parameter	Description
Configure the EntireX Broker trace location	
Trace location	The location of the EntireX Broker trace file. If left empty, the default location in your EntireX installation is used.

- Click **Test** to check the correctness of your input or **Save** to save your changes.

Viewing the Runtime Status

➤ To view the runtime status of a Broker instance

- In the Command Central **Home** page, click the **Instances** tab and select the Broker instance for which you want to see the runtime status (same as Step 1 under [Configuring a Broker Instance](#)).

The visual key performance indicators (KPIs) and alerts enable you to monitor EntireX Broker's health.



The following KPIs help you administer, troubleshoot, and resolve performance issues in EntireX Broker:

KPI	Description
Clients	Number of active clients.
Servers	Number of active servers.
Conversations	Number of active conversations.

Starting a Broker Instance

➤ To start a Broker instance from the Instances tab

- 1 In the Command Central home page, click the **Instances** tab.

Name [Count]	Component	Status	Alerts	Installation	Host
CCE [1 Components]	CCE			Local	MCPH002
SPM [2 Components]	SPM			Local	MCPH002
EntireX Broker ETB001	EntireX Broker ETB001			0.1	localhost
SPM [1 Components]	SPM			0.1	localhost

- 2 Select the status, and from the context menu choose **Start**.

➤ To start a Broker instance from its Overview tab

- 1 In the Command Central home page, click the **Instances** tab and select the Broker instance you want to start (same as Step 1 under [Configuring a Broker Instance](#)).

The screenshot shows the Command Central GUI for the 'EntireX Broker ETB001' instance. The breadcrumb navigation is 'Home > Instances > ALL > EntireX Broker ETB001'. The 'Overview' tab is selected, showing a dashboard with 'Status', 'Alerts', and 'KPIs' sections. The 'Status' section has a red circular arrow icon. A 'Lifecycle Actions' context menu is open over this icon, with 'Start' highlighted. The 'KPIs' section shows 'Clients' with a value of 2 and 'Servers' with a value of 0, both in a 'Normal' state.

- 2 Select the status, and from the context menu choose **Start**.

Stopping a Broker Instance

> To stop a Broker instance from the Instances tab

- 1 In the Command Central home page, click the **Instances** tab.

Name [Count]	Component	Status	Alerts	Installation	Host
CCE [1 Components]	CCE	↑		Local	MCPHO02
SPM [2 Components]	SPM	↑		Local	MCPHO02
EntireX Broker ETB001	EntireX Broker ETB001	↑		0.1	localhost
SPM [1 Components]	SPM	↑		0.1	localhost

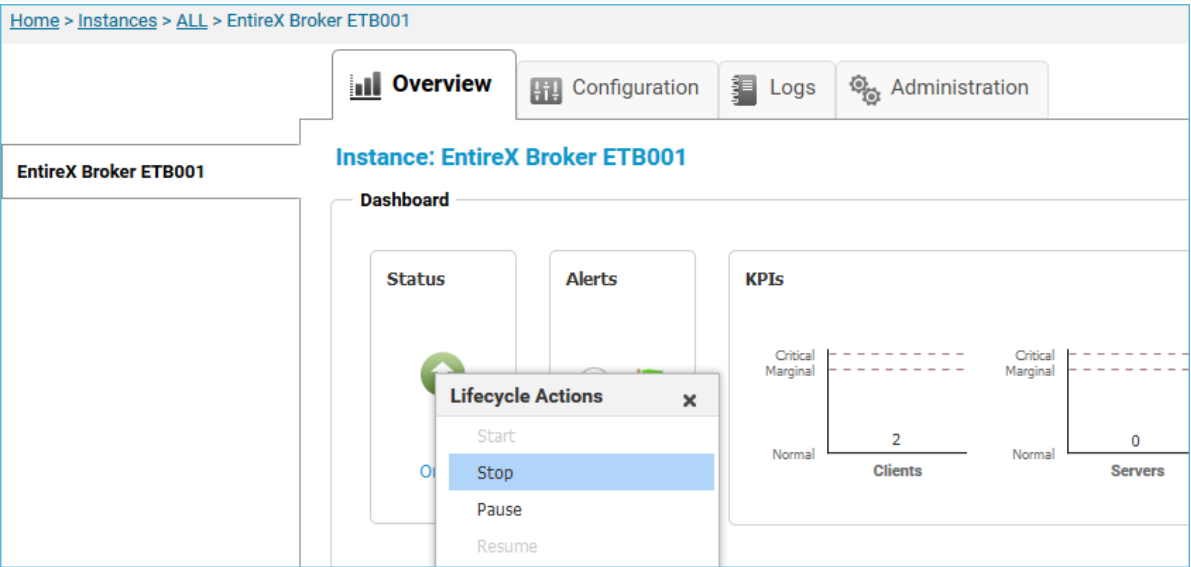
Lifecycle Actions x

- Start
- Stop
- Pause
- Resume

- 2 Select the status, and from the context menu choose **Stop**.

> To stop a Broker instance from its Overview tab

- 1 In the Command Central home page, click the **Instances** tab and select the Broker instance you want to stop (same as Step 1 under [Configuring a Broker Instance](#)).

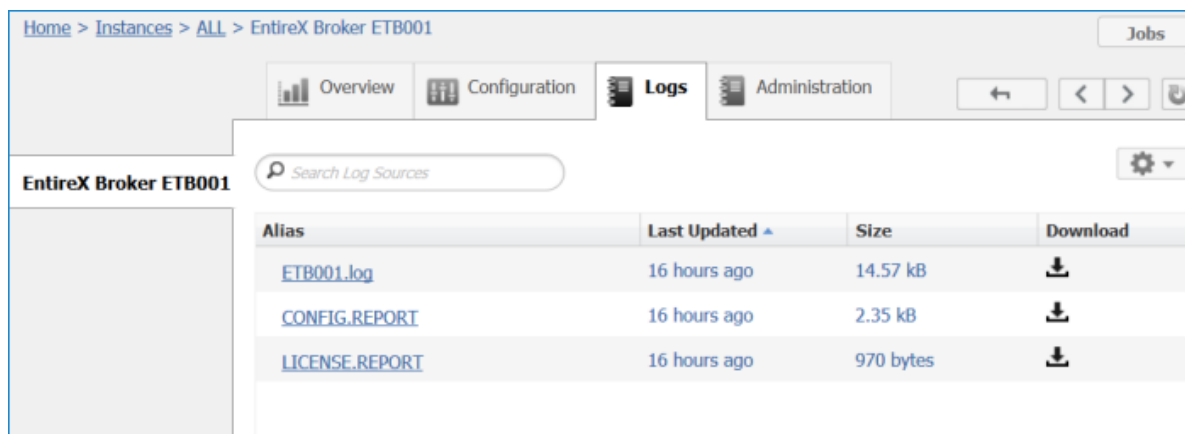


- 2 Select the status, and from the context menu choose **Stop**.

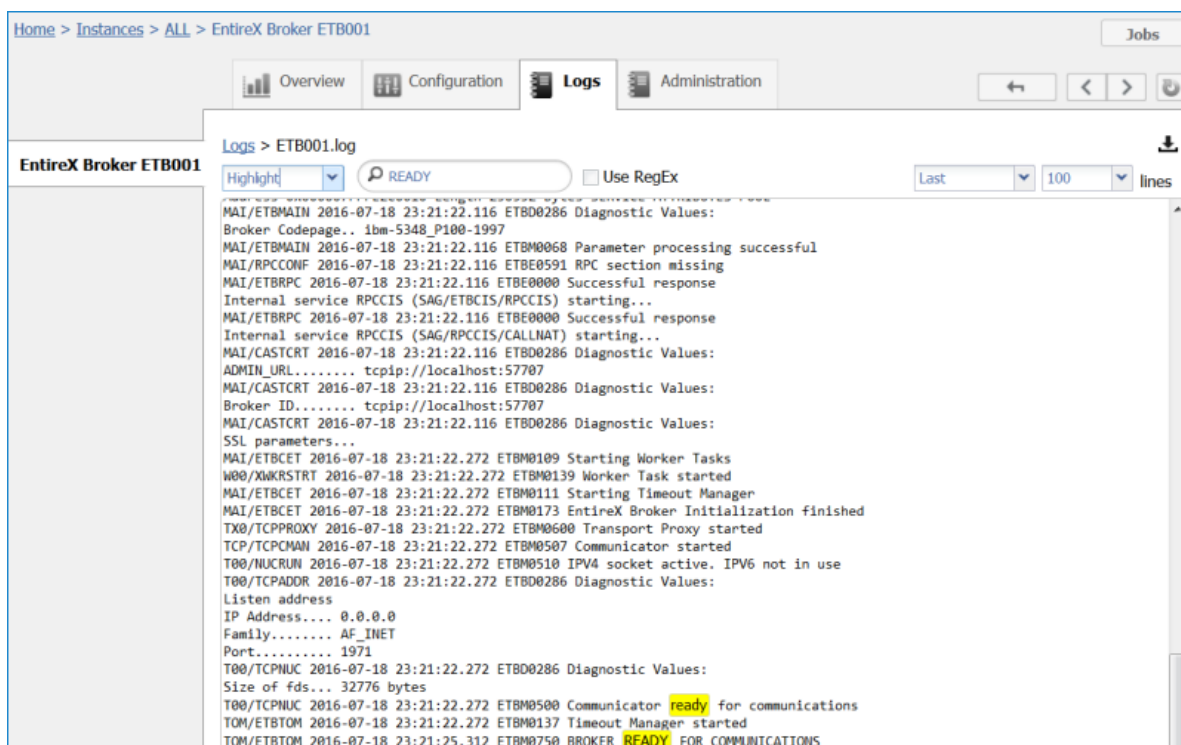
Inspecting the Log Files

➤ To inspect the log files of a Broker instance

- 1 In the Command Central **Home** page, click the **Instances** tab then click the link associated with the Broker instance for which you want to inspect the log files (same as Step 1 under [Configuring a Broker Instance](#)).
- 2 Click the **Logs** tab:



- 3 In the **Alias** column, click the link of the log file you want to inspect, for example ETB001.log:



Here you can search for certain text and highlight or filter the matching lines, specify how many first or last lines of the log you would like to see. You can also download the whole file.

Displaying the Statistics

➤ To display the Statistics of a running Broker instance

- 1 In the Command Central **Home** page, click the **Instances** tab, then click the link associated with the Broker instance for which you want to see its statistics (same as Step 1 under [Configuring a Broker Instance](#)).
- 2 Click on the **Administration** tab
- 3 Choose **Statistics** in the drop-down box.
- 4 Choose tab **General** to see important settings and statistics, or choose tab **Units of Work** to see UOW statistics (summarized per service).

Overview

Configuration

Logs

Administration

←

<

>

↺

Statistics ▼

General

Units of Work

Settings


Host name	
Platform	Windows 7 SP1
Broker version	10.5.0.00
Security	No
Persistent store	No
Transport	TCP (Port: 1971)
Application monitoring	No

Statistic

Running since	00h 01m 22s
CPU usage	0 %
Total storage usage	30 MB
Open TCP connections	1 (<1% of max. connections)
Open SSL connections	Not available for this Broker version
Authentication failures	0
Authorization failures	0

The screenshot shows the Command Central GUI with the **Administration** tab selected. Under the **Statistics** section, the **Units of Work** sub-tab is active. A search bar is present above a table with the following data:

Class/Server/Service	Bytes in all UOW Messages	Newest UOW	Accepted	Delivered	Postponed	Processed	Failed
RPC/UOW/MAILSERVICE-1	548	Mon Jun 17 08:11:34 2019	1	0	0	0	1
RPC/UOW/MAILSERVICE-2	812	Mon Jun 17 08:11:34 2019	1	0	0	0	2
RPC/UOW/MAILSERVICE-3	547	Mon Jun 17 08:11:34 2019	1	0	0	0	0

 **Note:** The returned value “UOWStatusFailed” represents the sum of UOW status “backed-out”, “cancelled”, “timeout” and “discarded”.

Displaying Services

> To display services registered with a Broker instance

- 1 In the Command Central **Home** page, click the **Instances** tab, then click the link associated with the Broker instance for which you want to see its registered services. (same as Step 1 under [Configuring a Broker Instance](#)).
- 2 Click on the **Administration** tab:

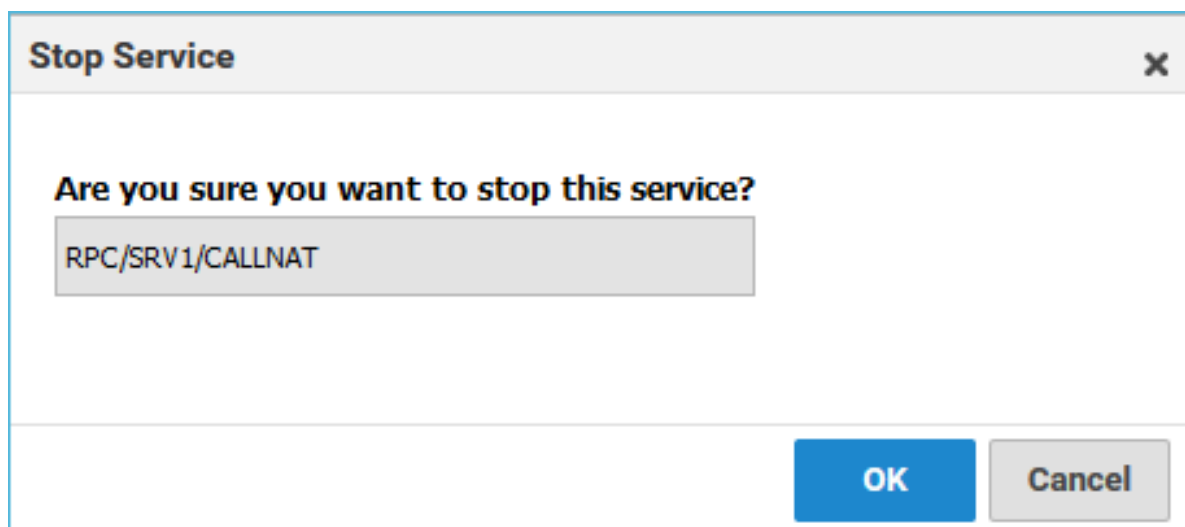
The screenshot shows the Command Central GUI with the **Administration** tab selected. Under the **Services** section, the **Currently Running Services** sub-tab is active. A search bar is present above a table with the following data:

Class/Server/Service	Server Instances	Requests	Wait for Servers	Conversations	Units of Work	
RPC/SRV1/CALLNAT	2	0	0 (0%)	0	0	Stop

Stopping a Service

➤ To stop a service

- 1 In the Command Central home page, click the **Instances** tab then click the link associated with the broker instance for which you want to stop a service (same as Step 1 under [Configuring a Broker Instance](#)).
- 2 Click on the **Administration** tab (same as [Displaying Services](#)).
- 3 To stop the service, click on **Stop** at the end of the service row.



- 4 Press **OK** to confirm. The selected service is removed from the list of currently running services.

Displaying Server Instances of a Service

➤ To display server instances of a service

- 1 In the Command Central home page, click the **Instances** tab then click the link associated with the broker instance for which you want to display server instances of a service (same as Step 1 under [Configuring a Broker Instance](#)).
- 2 Click on the **Administration** tab (same as [Displaying Services](#)).
- 3 In the **Class/Server/Service** column of the **Services** table, click the link of the service for which you want to see the associated server instances:

Service Details

Service

RPC/SRV1/CALLNAT

Server Instances

Search Server Instances

Name	Type	Host	Version	User ID	Conv	Start Time
Java RPC Server	Server		10.3.0.0.113	JavaRPCServer	0	2017.11.07 07:40:41
Java RPC Server	Attach Server		10.3.0.0.113	JavaRPCServer	0	2017.11.07 07:40:41

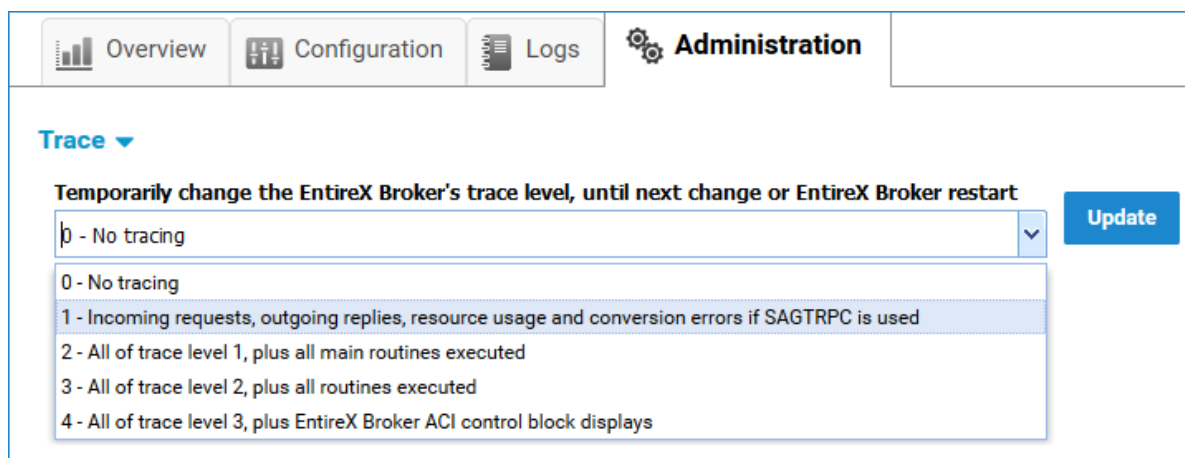
Close

The Start Time is displayed in the time zone of the Broker instance in sortable ISO-format.

Changing the Trace Level Temporarily

> To temporarily change the trace level of running broker

- 1 In the Command Central home page, click the **Instances** tab then click the link associated with the broker instance for which you want change the trace level temporarily (same as Step 1 under [Configuring a Broker Instance](#)).
- 2 In the **Administration** tab, select the trace level and press **Update**.



The screenshot shows the 'Administration' tab in the Command Central GUI. Under the 'Trace' section, there is a dropdown menu titled 'Temporarily change the EntireX Broker's trace level, until next change or EntireX Broker restart'. The dropdown is currently set to '0 - No tracing'. The dropdown menu is open, showing the following options:

- 0 - No tracing
- 1 - Incoming requests, outgoing replies, resource usage and conversion errors if SAGTRPC is used
- 2 - All of trace level 1, plus all main routines executed
- 3 - All of trace level 2, plus all routines executed
- 4 - All of trace level 3, plus EntireX Broker ACI control block displays

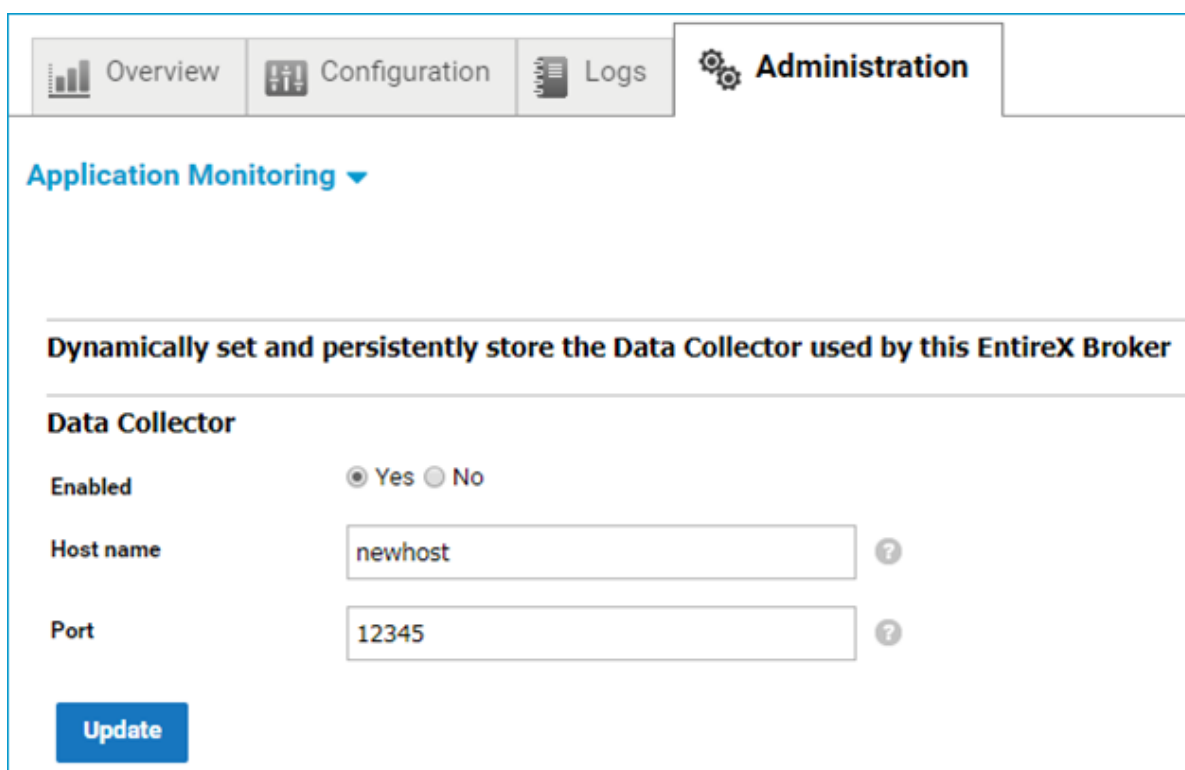
An 'Update' button is located to the right of the dropdown menu.

The change is effective immediately; there is no need to restart the EntireX Broker.

Changing the Application Monitoring Settings Dynamically

➤ To dynamically change the application monitoring settings of a running broker

- 1 In the Command Central home page, click the **Instances** tab then click the link associated with the broker instance for which you want change the application monitoring settings dynamically (same as Step 1 under [Configuring a Broker Instance](#)).
- 2 In the **Administration** tab, enter the application monitoring settings and press **Update**.




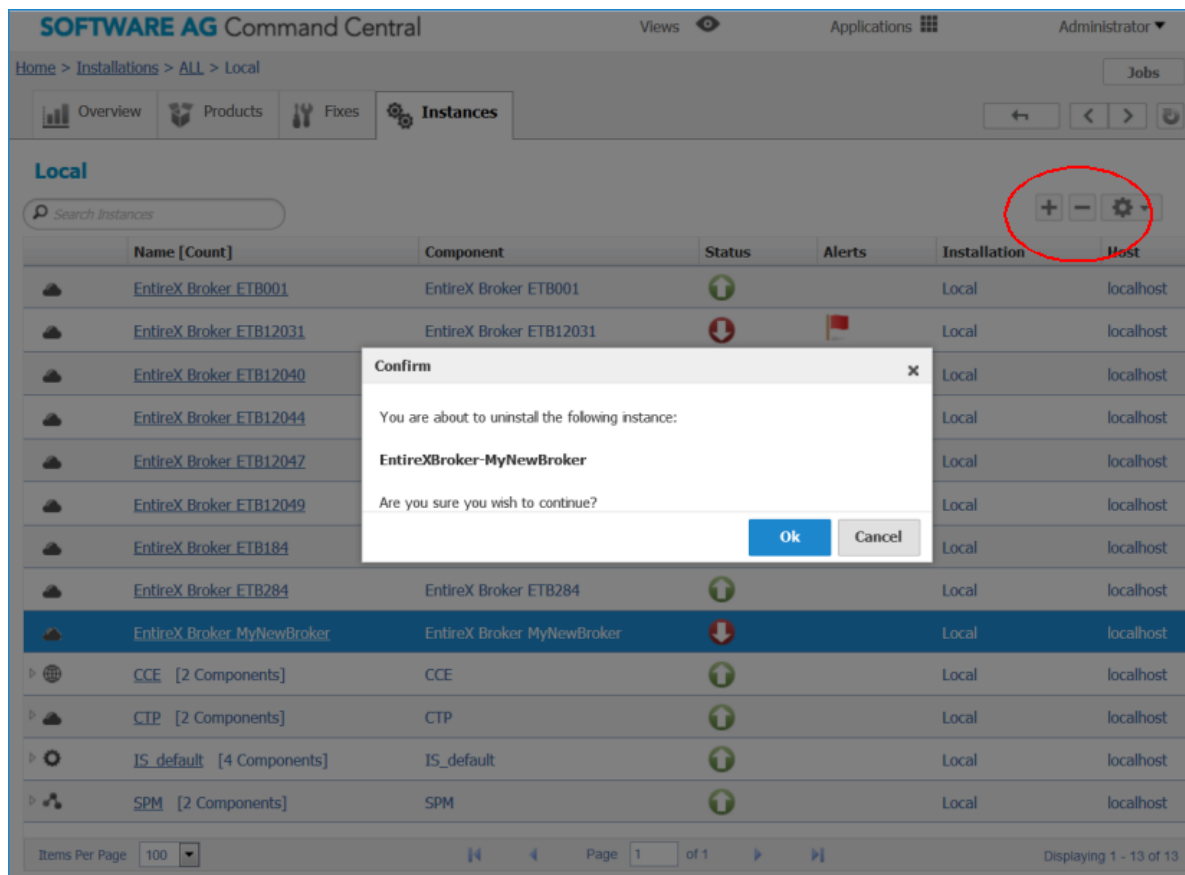
The screenshot shows the 'Administration' tab in the Command Central GUI. At the top, there are four tabs: 'Overview', 'Configuration', 'Logs', and 'Administration' (which is selected). Below the tabs, the 'Application Monitoring' section is expanded, showing a title bar with a dropdown arrow. Underneath, a heading reads 'Dynamically set and persistently store the Data Collector used by this EntireX Broker'. Below this heading is the 'Data Collector' section. It includes an 'Enabled' toggle with 'Yes' selected and 'No' unselected. There are two input fields: 'Host name' with the value 'newhost' and 'Port' with the value '12345'. Each input field has a help icon (a question mark in a circle) to its right. At the bottom left of the form is a blue 'Update' button.

The change is effective immediately; there is no need to restart the EntireX Broker.

Deleting a Broker Instance

> To delete a Broker instance

- 1 In the Command Central **Home** page, click the **Instances** tab and select the Broker instance you want to delete (same as Step 1 under [Configuring a Broker Instance](#)).
- 2 Click the  button in the upper right corner above the list:



- 3 Click **OK** to confirm the uninstall of this Broker instance.
- 4 In the next window, click **Finish**. The selected instance is removed from the list.

Troubleshooting

If a broker that you previously defined is not shown, one possible cause is that the Broker Administration service is not running and needs to be restarted.

Restarting the Broker Administration Service

- **Windows**

Select the broker instance with **Administration Tools > Services > Software AG EntireX Broker Admin 10.9 <installation_number>**, and from the context menu choose **Stop** then **Start** or **Restart**

- **Linux**

In directory *<inst_root>/EntireX/bin*, execute the following commands:

```
sagexx109 stop
```

and

```
sagexx109 start
```

or

```
sagexx109 restart
```


4 Administering EntireX Broker using the Command Central

Command Line

■ Creating a Broker Instance	38
■ Configuring a Broker Instance	39
■ Displaying the EntireX Inventory	61
■ Viewing the Runtime Status	62
■ Starting a Broker Instance	63
■ Stopping a Broker Instance	63
■ Restarting a Broker Instance	64
■ Inspecting the Log Files	65
■ Displaying the Statistics	66
■ Administering Services	68
■ Changing the Trace Level Temporarily	71
■ Changing the Application Monitoring Settings Dynamically	73
■ Deleting a Broker Instance	74
■ Troubleshooting	76

See also [Introduction](#) and [Administering EntireX Broker using the Command Central GUI](#).

Creating a Broker Instance

This table lists the parameters to include when creating an EntireX Broker instance, using the Command Central `create instances` command.

Command	Parameter	Value	Description
create instances	<i>node_alias</i>	<i>name</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>product</i>	EntireXBroker	Required. Must be set to "EntireXBroker".
	<i>instance.name</i>	<i>name</i>	Required. Name of broker.
	<i>tcp.port</i>	1025-65535	Required if SSL port is not set; optional if SSL port is set. TCP port number in range from 1025 to 65535.
	<i>ssl.port</i>	1025-65535	Required if TCP port is not set; optional if TCP port is set. SSL port number in range from 1025 to 65535.
	<i>autostart</i>	yes <u>no</u>	Enables/disables autostart. Default is no (disabled).
	<i>security</i>	yes <u>no</u>	Enables/disables security. Default is no (disabled).

Example

- To create a new instance of EntireXBroker, with name "MyBroker", listening on port 4711, in the installation with alias name "local":

```
sagcc create instances local EntireXBroker instance.name=MyBroker tcp.port=4711
```

Information about the creation job - including the job ID - is displayed.

Configuring a Broker Instance

Here you can administer the following parameters of the EntireX Broker. The newly configured parameters will be used the next time you start the EntireX Broker.

- [Application Monitoring](#)
- [Attribute File](#)
- [Autostart](#)
- [Monitoring KPIs](#)
- [Persistent Store](#)
- [SSL Ports](#)
- [TCP Ports](#)
- [Security](#)
- [Trace Level](#)
- [Trace Location](#)

Application Monitoring

Here you can administer the EntireX Broker Application Monitoring parameters. See the separate Application Monitoring documentation.

- [Available Application Monitoring Parameters](#)
- [Displaying Application Monitoring Parameters](#)
- [Updating Application Monitoring Parameters](#)

Available Application Monitoring Parameters

Parameter	Value	Description
Enabled	true false	Enable/disable application monitoring in EntireX Broker. Default is false (disabled).
CollectorHost	appmondc-host	Required. The host name on which the Application Monitoring Data Collector accepts the monitoring data.
CollectorPort	appmondc-port	Required. The TCP/IP port on which the Application Monitoring Data Collector accepts the monitoring data.

If Enabled=true is set, the Application Monitoring will be switched on with the next broker start.

Displaying Application Monitoring Parameters

This table lists the parameters to include when displaying the application monitoring parameters of a broker, using the Command Central `get configuration data` command.

Command	Parameter	Description
get configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "APPLICATION-MONITORING".
	<i>-o file</i>	Optional. Specifies the file where you want the output written.

Examples

- To display the application monitoring parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc get configuration data local EntireXBroker-MyBroker APPLICATION-MONITORING
```

- To store the application monitoring parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in the file `appmon.json` in the current working directory:

```
sagcc get configuration data local EntireXBroker-MyBroker APPLICATION-MONITORING ↵  
-o appmon.json
```

Updating Application Monitoring Parameters

This table lists the parameters to include when updating the application monitoring parameters of a broker, using the Command Central `update configuration data` command.

Command	Parameter	Description
update configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "APPLICATION-MONITORING".
	<i>-i file</i>	Optional. Specifies the file from where you want the input read.

Example

- To load the application monitoring parameters from the file `appmon.json` of the current working directory for the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc update configuration data local EntireXBroker-MyBroker APPLICATION-MONITORING ↵  
-i appmon.json
```

Attribute File

Here you can administer the attribute file of a broker. See the *Broker Attributes* documentation for details of individual attributes.

- [Displaying the Content of the Attribute File](#)
- [Updating the Content of the Attribute File](#)

Displaying the Content of the Attribute File

This table lists the parameters to include when displaying the attributes of a broker, using the Command Central `get configuration data` command.

Command	Parameter	Description
get configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "ATTRIBUTE-FILE".
	<i>-o file</i>	Optional. Specifies the file where you want the output written.

Examples

- To display the attributes of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc get configuration data local EntireXBroker-MyBroker ATTRIBUTE-FILE
```

- To store the content of the attribute file of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in the file myAttributeFile.txt in the current working directory:

```
sagcc get configuration data local EntireXBroker-MyBroker ATTRIBUTE-FILE -o ↵  
myAttributeFile.txt
```

Updating the Content of the Attribute File

This table lists the parameters to include when updating the attributes of a broker, using the Command Central `update configuration data` command.

Command	Parameter	Description
update configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "ATTRIBUTE-FILE".
	<i>-i file</i>	Optional. Specifies the file from where you want the input read.

Example

- To load the content of the EntireX Broker attribute file myAttributeFile.txt from the current working directory into the attribute file of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc update configuration data local EntireXBroker-MyBroker ATTRIBUTE-FILE -i ↵  
myAttributeFile.txt
```

Autostart

Here you can control the start behavior of an EntireX Broker. If `Enabled=true` is set, the broker starts automatically with the next system start.

- [Displaying the Autostart Parameter](#)
- [Updating the Autostart Parameter](#)

Displaying the Autostart Parameter

This table lists the parameters to include when displaying the autostart parameter of a broker, using the Command Central `get configuration data` command.

Command	Parameter	Description
get configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "BROKER-AUTOSTART".
	<i>-o file</i>	Optional. Specifies the file where you want the output written.

Examples

- To display the autostart parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc get configuration data local EntireXBroker-MyBroker BROKER-AUTOSTART
```

- To store the autostart parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in the file `autostart.json` in the current working directory:

```
sagcc get configuration data local EntireXBroker-MyBroker BROKER-AUTOSTART -o ↵  
autostart.json
```

Updating the Autostart Parameter

This table lists the parameters to include when updating the autostart parameter of a broker, using the Command Central `update configuration data` command.

Command	Parameter	Description
update configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "BROKER-AUTOSTART".
	<i>-i file</i>	Optional. Specifies the file from where you want the input read.

Example

- To load the autostart parameters from the file autostart.json of the current working directory into the attribute file of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc update configuration data local EntireXBroker-MyBroker BROKER-AUTOSTART -i ↵  
autostart.json
```

Monitoring KPIs

Key performance indicators (KPIs) enable you to monitor the health of your broker. Here you can administer margins of monitored KPIs available for the broker: Clients, Servers and Conversations.

- [Available KPIs](#)
- [Displaying the Monitoring KPIs](#)
- [Updating the Monitoring KPIs](#)

Available KPIs

The following KPIs help you administer, troubleshoot, and resolve performance issues:

Parameter	Value	Description
ClientsMaximum	1-2147483647	Required. Maximum number of clients in graph.
ClientsCriticalPercent	1-100	Required. Critical barrier of clients in graph in %.
ClientsMarginalPercent	1-100	Required. Marginal barrier of clients in graph in %.
ServersMaximum	1-2147483647	Required. Maximum number of servers in graph.
ServersCriticalPercent	1-100	Required. Critical barrier of servers in graph in %.
ServersMarginalPercent	1-100	Required. Marginal barrier of servers in graph in %.
ConversationsMaximum	1-2147483647	Required. Maximum number of conversations in graph.
ConversationsCriticalPercent	1-100	Required. Critical barrier of conversations in graph in %.
ConversationsMarginalPercent	1-100	Required. Marginal barrier of conversations in graph in %.

Do not change the other properties!

Displaying the Monitoring KPIs

This table lists the parameters to include when displaying the monitoring KPI parameters of a broker, using the Command Central `get configuration data` command.

Command	Parameter	Description
get configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "EXX-MONITORING-KPIS".
	<i>-f xml json</i>	Required. Specifies XML or JSON as output format.
	<i>-o file</i>	Optional. Specifies the file where you want the output written.

Examples

- To display the monitoring KPI properties of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc get configuration data local EntireXBroker-MyBroker EXX-MONITORING-KPIS -f ↵
json
```

- To store the monitoring KPI properties of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in the file *MyKpis.json* in the current working directory.

```
sagcc get configuration data local EntireXBroker-MyBroker EXX-MONITORING-KPIS -f ↵
json -o MyKpis.json
```

Updating the Monitoring KPIs

This table lists the parameters to include when updating the monitoring KPI parameters of a broker, using the Command Central `update configuration data` command.

Command	Parameter	Description
update configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "EXX-MONITORING-KPIS".
	<i>-i file</i>	Optional. Specifies the file from where you want the input read.

Example

- To load the content of the *MyKpis.json* file from the current working directory into the `spm.properties` of EntireX Broker with name = "MyBroker" in the installation with alias name "local":

```
sagcc update configuration data local EntireXBroker-MyBroker EXX-MONITORING-KPIS ↵
-i MyKpis.json
```

Persistent Store

Here you can administer the EntireX Broker persistent store parameters. See *Broker-specific Attributes* for description of the persistent store parameters. The new parameters will be used with the next broker start.

- [Available Persistent Store Parameters](#)
- [Displaying the Persistent Store Parameters](#)
- [Updating the Persistent Store Parameters](#)

Available Persistent Store Parameters

Parameter	Value	Description
Enabled	true <u>false</u>	Enable/disable the persistent store in EntireX Broker. Default is false (disabled). See <i>Concepts of Persistent Messaging</i> .
PStore	<u>COLD</u> HOT	Defines the status of the persistent store at broker startup, including the condition of persistent units of work (UOWs): COLD - Default. Persistent UOWs are not restored during initialization, and the persistent store is considered empty. HOT - Persistent UOWs are restored to their prior state during initialization.
PStoreType	<u>CTREE</u> ADABAS	The type of persistent store driver required: CTREE - Default. c-tree database. ADABAS - Adabas database.
PStoreCTreeMaxSize	<i>n</i> [G M]	Required for type CTREE. The maximum size of the c-tree persistent store in gigabytes (G) or megabytes (M), for example 1G, 2M, 3. If no unit is specified, megabytes are used.
PStoreCTreePageSize	<i>n</i> [K]	Required for type CTREE. The page size of the c-tree persistent store in kilobytes (K). Valid values: 8 or greater.
PStoreCTreePath	<i>ctree_directory</i>	Required for type CTREE. The directory where the c-tree persistent store resides. It will be created in the Broker's configuration directory.
PStoreAdabasDBID	<i>dbid</i>	Required for type ADABAS. The ID of the Adabas database where the persistent store resides.
PStoreAdabasFNR	<i>fnr</i>	Required for type ADABAS. The file number of the broker persistent store file.

Displaying the Persistent Store Parameters

This table lists the parameters to include when displaying the persistent store parameters of a broker, using the Command Central `get configuration data` command.

Command	Parameter	Description
get configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "BROKER-PSTORE".
	<i>-o file</i>	Optional. Specifies the file where you want the output written.

Examples

- To display the persistent store properties of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc get configuration data local EntireXBroker-MyBroker BROKER-PSTORE
```

- To store the persistent store parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in the file `pstore.json` in the current working directory:

```
sagcc get configuration data local EntireXBroker-MyBroker BROKER-PSTORE -o ↵  
pstore.json
```

Updating the Persistent Store Parameters

This table lists the parameters to include when updating the persistent store parameters of a broker, using the Command Central `update configuration data` command.

Command	Parameter	Description
update configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "BROKER-PSTORE".
	<i>-i file</i>	Optional. Specifies the file from where you want the input read.

Example

- To load the parameters described in the `pstore.json` file from the current working directory into the attribute file of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc update configuration data local EntireXBroker-MyBroker BROKER-PSTORE -i ↵  
pstore.json
```

SSL Ports

Here you can administer the EntireX Broker SSL port configuration. See *SSL/TLS-specific Attributes* for description of parameters. The new parameters will be used with the next broker start.

- [Available SSL Parameters](#)
- [Displaying the SSL Port Configuration](#)
- [Updating the SSL Port Configuration](#)

Available SSL Parameters

Parameter	Value	Description
Enabled	true <u>false</u>	Enable/disable the SSL port. Default is false (disabled).
Port	1025 - 65535	Required. The port number in range from 1025 to 65535.
KeyFile	<i>filename</i>	Required. The name of file that contains the broker's private key.
KeyPasswdEncrypted	<i>encrypted password</i>	Required. The encrypted password used to protect the private key.
KeyStore	<i>filename</i>	Required. The location of the store containing SSL certificate; may contain the private key.
TrustStore	<i>filename</i>	Required. The location of the store containing certificates of trust Certificate Authorities (or CAs).
ClientTrustStore	<i>filename</i>	Required. The location of the trust store used by the platform manager.
ClientTrustStorePasswd	<i>password</i>	The password of the trust store used by the platform manager.

Displaying the SSL Port Configuration

This table lists the parameters to include when displaying the SSL parameters of a broker, using the Command Central `get configuration data` command.

Command	Parameter	Description
get configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "SSL".
	<i>-o file</i>	Optional. Specifies the file where you want the output written.

Examples

- To display the SSL parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc get configuration data local EntireXBroker-MyBroker SSL
```

- To store the SSL parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in the file ssl.json in the current working directory:

```
sagcc get configuration data local EntireXBroker-MyBroker SSL -o ssl.json
```

Updating the SSL Port Configuration

Command	Parameter	Description
update configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "SSL".
	<i>-i file</i>	Optional. Specifies the file from where you want the input read.

Example

- To load the SSL parameters described in the ssl.json file from the current working directory into the attribute file of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc update configuration data local EntireXBroker-MyBroker SSL -i ssl.json
```


TCP Ports

Here you can administer the EntireX Broker TCP port configuration. See *TCP/IP-specific Attributes* for description of parameters. The new parameters will be used with the next broker start.

- [Available TCP Parameters](#)
- [Displaying the TCP Port Configuration](#)
- [Updating the TCP Port Configuration](#)

Available TCP Parameters

Parameter	Value	Description
Enabled	true false	Enable/disable the TCP port. Default is true (enabled).
Port	1025 - 65535	Required. The port number in range from 1025 to 65535.

Displaying the TCP Port Configuration

This table lists the parameters to include when displaying TCP parameters of a broker, using the Command Central `get configuration data` command.

Command	Parameter	Description
get configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "TCP".
	<i>-o file</i>	Optional. Specifies the file where you want the output written.

Examples

- To display the TCP parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc get configuration data local EntireXBroker-MyBroker TCP
```

- To store the TCP parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in the file tcp.json in the current working directory:

```
sagcc get configuration data local EntireXBroker-MyBroker TCP -o tcp.json
```

Updating the TCP Port Configuration

This table lists the parameters to include when updating TCP parameters of a broker, using the Command Central `update configuration data` command.

Command	Parameter	Description
update configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "TCP".
	<i>-i file</i>	Optional. Specifies the file from where you want the input read.

Example

- To load the parameters described in the `tcp.json` file from the current working directory into the attribute file of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc update configuration data local EntireXBroker-MyBroker TCP -i tcp.json
```

Security

Here you can administer the EntireX Security configuration. See *Security-specific Attributes* for description of parameters. The new parameters will be used with the next broker start.

- [Available Security Parameters](#)
- [Displaying the Security Parameters](#)
- [Updating the Security Parameters](#)

Available Security Parameters

Parameter	Value	Description
Enabled	true <u>false</u>	Enable/disable EntireX Broker Security. Default is false (disabled).
AuthType	<u>OS</u> LDAP	Authentication Type. Operating System - Default. Authentication is performed against the local operating system. LDAP - Authentication and authorization are performed against the LDAP repository.
LdapUrl	ldap:// <i>HostName</i> [: <i>PortNumber</i>]	Required for authentication type LDAP: URL of the LDAP repository.
LdapDn	<i>string</i>	Required for authentication type LDAP: The distinguished name of user for authenticated access to the LDAP server. Default value: cn=admin,dc=software-ag,dc=de

Displaying the Security Parameters

This table lists the parameters to include when displaying security parameters of a broker, using the Command Central `get configuration data` command.

Command	Parameter	Description
get configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "BROKER-SECURITY".
	-o <i>file</i>	Optional. Specifies the file where you want the output written.

Examples

- To display the broker security parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc get configuration data local EntireXBroker-MyBroker BROKER-SECURITY
```

- To store the security parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in the file security.json in the current working directory:

```
sagcc get configuration data local EntireXBroker-MyBroker BROKER-SECURITY -o ↵  
security.json
```

Updating the Security Parameters

This table lists the parameters to include when updating security parameters of a broker, using the Command Central `update configuration data` command.

Command	Parameter	Description
update configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "BROKER-SECURITY".
	<i>-i file</i>	Optional. Specifies the file from where you want the input read.

Example

- To load the parameters described in the security.json file from the current working directory into the attribute file of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc update configuration data local EntireXBroker-MyBroker BROKER-SECURITY -i ↵  
security.json
```

Trace Level

Here you can administer the EntireX Broker trace level configuration. The new trace level will be used with the next broker start. See also [Changing the Trace Level Temporarily](#).

- [Available Parameters](#)
- [Displaying the Trace Level Configuration](#)
- [Updating the Trace Level](#)

Available Parameters

Parameter	Value	Description
TraceLevel	0 1 2 3 4	One of the following levels: 0 - Default. No tracing. 1 - Incoming request, outgoing replies, resource usage and conversion errors if SAGTRPC is used. 2 - All of trace level 1, plus all main routines executed. 3 - All of trace level 2, plus all routines executed. 4 - All of trace level 3, plus EntireX Broker ACI control block displays.

Displaying the Trace Level Configuration

This table lists the parameters to include when displaying the trace level of a broker, using the Command Central `get configuration data` command.

Command	Parameter	Description
get configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "BROKER-TRACELEVEL".
	<i>-o file</i>	Optional. Specifies the file where you want the output written.

Examples

- To display the trace level configuration parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc get configuration data local EntireXBroker-MyBroker BROKER-TRACELEVEL
```

- To store the trace configuration parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in the file trace.json of the current working directory:

```
sagcc get configuration data local EntireXBroker-MyBroker BROKER-TRACELEVEL -o ↵  
trace.json
```

Updating the Trace Level

This table lists the parameters to include when updating the trace level of a broker, using the Command Central `update configuration data` command.

Command	Parameter	Description
update configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "BROKER-TRACELEVEL".
	<i>-i file</i>	Optional. Specifies the file from where you want the input read.

Example

- To load the parameters described in the `trace.json` file from the current working directory into the attribute file of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc update configuration data local EntireXBroker-MyBroker BROKER-TRACELEVEL ↵  
-i trace.json
```

Trace Location

Here you can administer the EntireX Broker trace location configuration. This specifies the directory that contains the EntireX Broker trace file. The new trace location will be used with the next broker start.

- [Available Parameters](#)
- [Displaying the Trace Location Configuration](#)

- [Updating the Trace Location](#)

Available Parameters

Parameter	Value	Description
TraceLocation	<i>string</i>	The location of the EntireX Broker trace file. If left empty, the default location in your EntireX installation is used.

Displaying the Trace Location Configuration

This table lists the parameters to include when displaying the trace location of a broker, using the Command Central `get configuration data` command.

Command	Parameter	Description
get configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "BROKER-TRACELOCATION".
	<i>-o file</i>	Optional. Specifies the file where you want the output written.

Examples

- To display the trace location configuration parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc get configuration data local EntireXBroker-MyBroker BROKER-TRACELOCATION
```

- To store the trace location configuration parameters of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in the file `trace.json` of the current working directory:

```
sagcc get configuration data local EntireXBroker-MyBroker BROKER-TRACELOCATION ↵  
-o trace.json
```

Updating the Trace Location

This table lists the parameters to include when updating the trace location of a broker, using the Command Central `update configuration data` command.

Command	Parameter	Description
update configuration data	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	<i>instanceid</i>	Required. Must be "BROKER-TRACELOCATION".
	<i>-i file</i>	Optional. Specifies the file from where you want the input read.

Example

- To load the parameters described in the `trace.json` file from the current working directory into the attribute file of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc update configuration data local EntireXBroker-MyBroker BROKER-TRACELOCATION ↵  
-i trace.json
```


Displaying the EntireX Inventory

Listing all Inventory Components

This table lists the parameters to include, when listing all EntireX instances, using the Command Central `list inventory components` command.

Command	Parameter	Description
list inventory components	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".

Example

- To list inventory components of instance EntireX Broker in the installation with alias name "local":

```
sagcc list inventory components local EntireXBroker*
```

A list of all EntireX Broker runtime components will be displayed.

Viewing the Runtime Status

This table lists the parameters to include when displaying the state of an EntireX component, using the Command Central `get monitoring state` command.

Command	Argument	Description
<code>sagcc get monitoring state</code>	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".

Example

- To display state information about the EntireX Broker "MyBroker" in the installation with alias name "local":

```
sagcc get monitoring state local EntireXBroker-MyBroker
```

Starting a Broker Instance

This table lists the parameters to include when starting an EntireX component, using the Command Central `exec lifecycle` command.

Command	Parameter	Description
exec lifecycle	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".

Example

- To start the EntireX Broker component "MyBroker" in the installation with alias name "local":

```
sagcc exec lifecycle start local EntireXBroker-MyBroker
```

Information about the job - including the job ID - will be displayed.

Stopping a Broker Instance

This table lists the parameters to include when stopping an EntireX component, using the Command Central `exec lifecycle` command.

Command	Parameter	Description
exec lifecycle	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".

Example

- To stop the EntireX Broker component MyBroker, in the installation with alias name "local":

```
sagcc exec lifecycle stop local EntireXBroker-MyBroker
```

Information about the job - including the job ID - will be displayed.

Restarting a Broker Instance

This table lists the parameters to include when restarting an EntireX component, using the Command Central `exec lifecycle` command.

Command	Parameter	Description
exec lifecycle	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".

Example

- To restart the EntireX Broker component "MyBroker" in the installation with alias name "local":

```
sagcc exec lifecycle restart local EntireXBroker-MyBroker
```

Information about the job - including the job ID - will be displayed.

Inspecting the Log Files

Here you can view, get the content of, or download the EntireX Broker log files. This table lists the parameters to include when displaying or modifying parameters of an EntireX component, using the Command Central `list diagnostics logs` command.

- [Listing all EntireX Broker Log Files](#)
- [Getting Content from or Downloading EntireX Broker Log Files](#)

Listing all EntireX Broker Log Files

Command	Parameter	Description
list diagnostics logs	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".

Example

- To list the log files of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc list diagnostics logs local EntireXBroker-MyBroker
```

Getting Content from or Downloading EntireX Broker Log Files

Command	Parameter	Description
get diagnostics logs	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	full tail head	Optional. Shows full log file content, or only tail or head.
	export -o <i>file</i>	Optional. Creates a zip file of the logs.

Examples

- To list the tail of the log file MyBroker.log of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc get diagnostics logs local EntireXBroker-MyBroker MYBROKER.log tail
```

- To create and download a zip file myfile.zip of the logs of the EntireX Broker with the name "MyBroker" in the installation with alias name "local".

```
sagcc get diagnostics logs local EntireXBroker-MyBroker export -o myfile.zip
```

Displaying the Statistics

- [Displaying the General Statistics of a Running EntireX Broker](#)
- [Displaying the UOW \(Unit of Work\) Statistics of a Running EntireX Broker](#)

Displaying the General Statistics of a Running EntireX Broker

Here you can display the current statistics of a running EntireX Broker.

Command	Parameter	Description
get administration	<i>component</i>	Specifies that a component will be administered.
	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	Statistics	Required. Specifies what is to be administered.
	loadStatisticGeneral	Required. Get the general settings.
	-f xml json	Required. Specifies XML or JSON as output format.
	-o file	Optional. Specifies the file where you want the output written.

Examples

- To display the current general setting of the running EntireX Broker with the name 'MyBroker' in the installation with alias name 'local'.broker in JSON format on stdout:

```
sagcc get administration component local EntireXBroker-MyBroker Statistics ↵  
loadStatisticGeneral -f json
```

- To display the current general setting of the running EntireX Broker with the name 'MyBroker' in the installation with alias name 'local' in XML format on stdout:

```
sagcc get administration component local EntireXBroker-MyBroker Statistics ↵
loadStatisticGeneral -f xml
```

Displaying the UOW (Unit of Work) Statistics of a Running EntireX Broker

Here you can display the current UOW statistics of a running EntireX Broker. The UOW statistics are summarized per service.



Note: The returned value “UOWStatusFailed” represents the sum of UOW status “backed-out”, “cancelled”, “timeout” and “discarded”.

Command	Parameter	Description
get administration	<i>component</i>	Specifies that a component will be administered.
	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	Statistics	Required. Specifies what is to be administered.
	loadStatisticUow	Required. Get the statistics of UOW usage.
	-f xml json	Required. Specifies XML or JSON as output format.
	-o file	Optional. Specifies the file where you want the output written.

Examples

- To display the current UOW statistics of the running EntireX Broker with the name 'MyBroker' in the installation with alias name 'local'.broker in JSON format on stdout:

```
sagcc get administration component local EntireXBroker-MyBroker Statistics ↵
loadStatisticUow -f json
```

- To display the current UOW statistics of the running EntireX Broker with the name 'MyBroker' in the installation with alias name 'local' in XML format on stdout:

```
sagcc get administration component local EntireXBroker-MyBroker Statistics ↵
loadStatisticUow -f xml
```

Administering Services

Here you can view and control EntireX Broker Services. This table lists the parameters to include when displaying or modifying parameters of a broker, using the Command Central `list administration` command. This section covers the following topics:

- [List Running Services](#)
- [Stop Running Services](#)
- [List Server Instances](#)

List Running Services

Command	Parameter	Description
list administration	<i>component</i>	Specifies that a component will be administered.
	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	Services	Required. Specifies what is to be administered.
	listServices	Required. List all services.
	-f xml json	Required. Specifies XML or JSON as output format.

Examples

- To display a list of services of the running EntireX Broker with the name "MyBroker" in the installation with alias name "local".broker in JSON format:

```
sagcc list administration component local EntireXBroker-MyBroker Services ↵
listServices -f json
```

- To store a list of services of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in JSON format in the file services.json of the current working directory:

```
sagcc list administration component local EntireXBroker-MyBroker Services ↵
listServices -o services.json
```


Stop Running Services

Command	Parameter	Description
exec administration	component	Specifies that a component will be administered.
	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	Services	Required. Specifies what is to be administered.
	stopService	Required. Stop a service.
	serviceName	Required. Name of a service to stop in format <i>class/server/service</i>
	-f xml json	Required. Specifies XML or JSON as output format.

Example

- To stop all servers of the service RPC/SRV1/CALLNAT of the EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc exec administration component local EntireXBroker-MyBroker Services ↵
stopService serviceName=RPC/SRV1/CALLNAT -f json
```

List Server Instances

Command	Parameter	Description
list administration	component	Specifies that a component will be administered.
	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	Services	Required. Specifies what is to be administered.
	listServers	Required. List all servers.
	serviceName	Required. Shows only servers of this service. Format: <i>class/server/service</i>
	-f xml json	Required. Specifies XML or JSON as output format.

Examples

- To display a list of servers of the current service of the EntireX Broker with the name "MyBroker" in the installation with alias name "local" in XML format:

```
sagcc list administration component local EntireXBroker-MyBroker Services ↵  
listServers serviceName=RPC/SRV1/CALLNAT -f xml
```

- To store a list of servers in JSON format in the file services.json of the current working directory:

```
sagcc list administration component local EntireXBroker-MyBroker Services ↵  
listServers serviceName=RPC/SRV1/CALLNAT -o server.json
```

Changing the Trace Level Temporarily

Here you can temporarily change the trace level of a running EntireX Broker. This table lists the parameters to include when displaying or modifying parameters of an EntireX component, using the Command Central `exec administration` command. The change is effective immediately; there is no need to restart the EntireX Broker.



Note: If you want to set the trace level permanently, see [Trace Level](#) under *Configuring a Broker Instance*.

- [Displaying the Trace Level of a Running EntireX Broker](#)
- [Updating the Trace Level of a Running EntireX Broker](#)

Displaying the Trace Level of a Running EntireX Broker

Command	Parameter	Description
exec administration	component	Required. Specifies that a component will be administered.
	node_alias	Required. Specifies the alias name of the installation in which the runtime component is installed.
	componentid	Required. The component identifier. The prefix is "EntireXBroker-".
	Trace	Required. Specifies what is to be administered.
	load tracelevel=?	Required. Get the trace level.
	-f xml json	Required. Specifies XML or JSON as output format.

Examples

- To display the current tracelevel of the running EntireX Broker with the name "MyBroker" in the installation with alias name "local" in JSON format on stdout:

```
sagcc get administration component local EntireXBroker-MyBroker Trace load -f json
```

- To display the current trace level of the running EntireX Broker with the name "MyBroker" in the installation with alias name "local" in XML format on stdout:

```
sagcc get administration component local EntireXBroker-MyBroker Trace load -f xml
```

Updating the Trace Level of a Running EntireX Broker

Command	Parameter	Description
exec administration	component	Required. Specifies that a component will be administered.
	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	Trace	Required. Specifies what is to be administered.
	update tracelevel	Required. Update temporarily the trace level of a running broker.
	-f xml json	Required. Specifies XML or JSON as output format.

Example

- To change the current trace level of the running EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc exec administration component local EntireXBroker-MyBroker Trace update ↵  
tracelevel=2 -f json
```

Changing the Application Monitoring Settings Dynamically

Here you can dynamically set and persistently store the EntireX Broker Application Monitoring parameters. This table lists the parameters to include when displaying or modifying parameters of an EntireX component, using the Command Central `get administration` and `exec administration` commands. The change is effective immediately; there is no need to restart the EntireX Broker.

- [Displaying the Application Monitoring Settings of a Running EntireX Broker](#)
- [Updating the Application Monitoring Settings of a Running EntireX Broker](#)

Displaying the Application Monitoring Settings of a Running EntireX Broker

Command	Parameter	Description
get administration	component	Required. Specifies that a component will be administered.
	node_alias	Required. Specifies the alias name of the installation in which the runtime component is installed.
	componentid	Required. The component identifier. The prefix is "EntireXBroker-".
	Appmon	Required. Specifies what is to be administered.
	load	Required. Get the application monitoring settings.
	-a xml json	Required. Specifies XML or JSON as output format.
	-o file	Optional. Specifies the file where you want the output written.

Examples

- To display the current application monitoring settings of the running EntireX Broker with the name "MyBroker" in the installation with alias name "local" in JSON format on stdout:

```
sagcc get administration component local EntireXBroker-MyBroker Appmon load -a json
```

- To display the current application monitoring settings of the running EntireX Broker with the name "MyBroker" in the installation with alias name "local" in XML format on stdout:

```
sagcc get administration component local EntireXBroker-MyBroker Appmon load -a xml
```

Updating the Application Monitoring Settings of a Running EntireX Broker

Command	Parameter	Description
exec administration	component	Required. Specifies that a component will be administered.
	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".
	Appmon	Required. Specifies what is to be administered.
	update enabled collectorHost collectorPort	Required. Update the application monitoring settings of a running broker.

Example

- To change the current application monitoring settings of the running EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc exec administration component local EntireXBroker-MyBroker Appmon update ↵
enabled=true collectorHost=newhost collectorport=12345
```

Deleting a Broker Instance

This table lists the parameters to include when deleting an EntireX Broker instance, using the Command Central `delete instances` command.

Command	Parameter	Description
delete instances	<i>node_alias</i>	Required. Specifies the alias name of the installation in which the runtime component is installed.
	<i>componentid</i>	Required. The component identifier. The prefix is "EntireXBroker-".

Example

- To delete an instance of an EntireX Broker with the name "MyBroker" in the installation with alias name "local":

```
sagcc delete instances local EntireXBroker-MyBroker
```

Information about the deletion job - including the job ID - is displayed.

Troubleshooting

If you have already defined a broker and an empty list is returned or you receive an unexpected message such as "connection refused", one possible cause is that the Broker Administration Service is not running and needs to be restarted.

Restarting the Broker Administration Service

■ Windows

Select the broker instance with **Administration Tools > Services > Software AG EntireX Broker Admin 10.9 <installation_number>**, and from the context menu choose **Stop** then **Start** or **Restart**

■ Linux

In directory *<inst_root>/EntireX/bin*, execute the following commands:

```
sagexx109 stop
```

and

```
sagexx109 start
```

or

```
sagexx109 restart
```