✔ software AG

webMethods EntireX

Post-installation Steps under UNIX

Version 10.7

October 2020

WEBMETHODS

Table of Contents

1   About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	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://documenta-tion.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.soft-wareag.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 Post-installation Steps under UNIX

Under UNIX and Windows, EntireX is now installed using the Software AG Installer, which you download from the Software AG Empower website at **Empower**. You can view the documentation for the Software AG Installer under *Cross-Product Guides > Software AG Installer and Update Manager Using the Software AG Installer>* in the Software AG Documentation. Alternatively, you can install and configure EntireX using Command Central; see *Installing EntireX using Command Central*.

This chapter assumes you have already installed EntireX using the Software AG Installer. It covers installation instructions for components not included in the Software AG Installer; instruction on how to verify the installation and set up EntireX Security.

See also *General Installation Information* for topics that apply to multiple operating systems.

Shell Environment Settings

EntireX requires some shell environment variables to be set for proper operation. The installation provides the shell script *exxenv* in the *EntireX/INSTALL* directory. Source this script from within the *.profile* of the EntireX users.

Sourcing this script, which includes additional environment scripts, defines the following variables:

Variable	Description
EXXDIR	Identifies the base installation directory for EntireX (typically */opt/softwareag/EntireX*)
EXXVERS	Identifies the product version. This variable is deprecated and is set to "." for reasons of backward compatibility.
ETBLNK	Identifies the absolute path to the broker stubs library if EntireX Broker has been installed: $EXXDIR/lib/broker.so.
JAVA_HOME	Identifies the Java directory in the installation.

In addition, the script modifies the environment variables PATH, LD_LIBRARY_PATH (LIBPATH under AIX).

- directory */<Install_Dir>/EntireX/bin* is added to the list of directories in the PATH environment variable

- the directory */<Install_Dir>/EntireX/lib* is added to LD_LIBRARY_PATH (LIBPATH under AIX).

Broker Instance Created Automatically during Installation

If check box **Turn on Autostart for default EntireX Broker** is checked, the installation starts the default broker ETB001. This broker instance listens on the TCP/IP port defined in the custom panel during installation. Default port numbers are 1971 (TCP/IP) and 1958 (SSL), but only the TCP/IP port is enabled by default.

≫ To change the default port of the default broker

1   Stop the broker, using one of the following methods:

  ▪ Use Command Central to stop the broker. See *Administering EntireX Components with Command Central* in the EntireX documentation or the separate Command Central documentation and online help for details.

  ▪ Enter command:

    <Installation_Dir>/EntireX/bin/defaultbroker stop

2   Edit the configuration file, using one of the following methods:

  ▪ Use Command Central to change. See *Configuring a Broker Instance > Ports* using the Command Central GUI | Command Line.

  ▪ Edit the broker attribute file *config/etb/ETB001/ETB001.atr* and change the TCP/IP and SSL port numbers to a different value.

3   Start the broker, using one of the following methods:

  ▪ Use Command Central to start the broker. See *Administering EntireX Components with Command Central* in the EntireX documentation or the separate Command Central documentation and online help for details.

  ▪ Enter command:

    <Installation_Dir>/EntireX/bin/defaultbroker start

Startup Daemon for Broker Administration

When installation is complete, the daemon is running and ready to be used by `etbsrv` script located in directory *<Installation_Dir>/EntireX/bin*. This script can be used, for example, to start or stop the broker.

≫ **To start the daemon**

■ Enter the following command:

```
- <Installation_Dir>/EntireX/bin/sagexx107 start
```

≫ **To stop the daemon**

■ Enter the following command:

```
- <Installation_Dir>/EntireX/bin/sagexx107 stop
```

It is also registered to startup at boot time, therefore the installation generates additional scripts in a location that depends on the operating system:

Operating System	Location	Note
Solaris, Linux	/etc/init.d	Recent Linux versions use systemd instead of init scripts.
AIX	/etc	

Considerations if you did not Install EntireX Broker using sudo

If you installed EntireX Broker on a UNIX system and did not use sudo, perform the following:

1. Restart the Administration Service with the following command:

```
/etc/init.d/sagnumberexxrelease_number restart
```

where *<number>* matches the number in the sagexx107 script in your *<Software AG_directory>/EntireX/bin* directory.

2. If you want to enable autostart, see for example *Configuring a Broker Instance > Autostart* using the Command Central GUI | Command Line.

Verifying the EntireX Broker Installation

This section describes how to verify the installation of the following EntireX components:

- EntireX Broker
- Broker Stubs
- Sample Programs for Client (bcoc) and Server (bcos)
- Environment Variables

EntireX Broker

The following methods are available to verify successful broker installation:

- `etbinfo` under *Broker Command-line Utilities* in the platform-specific Administration documentation
- `etbcmd` under *Broker Command-line Utilities* in the platform-specific Administration documentation
- *Sample Programs for Client (`bcoc`) and Server (`bcos`)*

⟫ To verify the broker installation

1   If you have not already created a default broker, create one using Command Central. See *Creating a Broker Instance* under *Administering EntireX Broker using the Command Central GUI*.

2   Start the broker using Command Central. See *Starting a Broker Instance* under *Administering EntireX Broker using the Command Central GUI*.

3   Use one of the following methods to check if the broker is running:

- Run `etbinfo` to query the broker for information. Start a Command Prompt session, change your directory to the EntireX *bin* directory and use the following command:

```
etbinfo -blocalhost -dBROKER -pbroker.pro
```

This assumes that you use the default port 1971. Otherwise use the command:

```
etbinfo -blocalhost:port -dBROKER -pbroker.pro
```

where *port* is the port number of your broker.

This should display a formatted report with information about the broker. If the broker is not active, you get response 02150148 "EntireX Broker not active".

- Run the `etbcmd` utility.

```
etbcmd -bETB048 -cPING -dBROKER
```

The broker ID is ETB0248. If successful, the message `PING broker ETB048 successfully performed` is returned. If the broker is not active, you get response 02150148 "EntireX Broker not active".

■ Use the verification of **Broker Stubs**. This also verifies EntireX Broker installation.

Broker Stubs

≫ To verify broker stubs installation

1 Ensure that you have installed the EntireX Broker on the target node and that it is up and running.

2 Ensure that the variable `ETBLNK` is set to /<Install_Dir>/EntireX/lib/broker.s[o|l] and that the `ETB_TRANSPORT` variable is set correctly, by entering the command:

```
env | grep ET
```

3 Enter the commands to run the test programs. The Broker ID is ETB048 and the `REPEATS` value is 100. The parameters are described under *Sample Programs for Client (`bcoc`) and Server (`bcos`)*.

```
cd /<Install_Dir>EntireX/installation_verification/conversational/C
bcos -i100 -bETB048 -2&
bcoc -i100 -bETB048 -2
```

If you receive error 02150148 "EntireX Broker not active", verify that the broker is running on the target machine.

Sample Programs for Client (bcoc) and Server (bcos)

The programs `bcoc` and `bcos` are client and server programs provided for test purposes. They are delivered as executables in the *<Install_Dir>/Entirex/bin/* directory.

If you have problems, check that environment variable `ETBLNK` refers to the broker shared library in directory */<Install_Dir>/EntireX/lib*. If the broker is not active, you get 02150148 "EntireX Broker not active".

bcoc Parameters

Parameter	Description	ACI Field
`-a`*`locale`*	*`locale_string,environment`*	`LOCALE-STRING,` `ENVIRONMENT`
`-b`*`broker_id`*	Broker ID (same as the database ID). This is an EntireX Broker configuration parameter. Example: `bcoc -bETB045`	`BROKER-ID`
`-c`*`class`*	Server class (part of the server definition). Default is `ACLASS`. Value must be configured in the *Broker Attributes*. Together with `server` and `service` it forms the fully qualified server name.	`SERVER-CLASS`
`-g`*`input_file_name`*	Use a file as send buffer.	
`-h`	Display usage information.	
`-i`*`num_messages[,num_uows]`*	If *`num_uows`* is not specified, the number of messages sent. If *`num_uows`* is specified, the number of UOWs and messages therein sent. Example: `bcoc -i10`	
`-j`*`reply_error`*	Test `REPLY-ERROR`.	
`-k`*`conversation_flag`*	Using conversational mode.	
`-l`	Silent mode. Suppress output.	
`-n`*`token`*	Specify the security token, if desired.	`TOKEN`
`-p`*`message_size`*	The size, in bytes, of the packets that will be transmitted. Used to specify send and receive length in the broker API. Example: `bcoc -p10000`	
`-r`*`compress`*	Compression level. Values: `N|Y|0-9`.	`COMPRESSLEVEL`
`-s`*`server`*	Name of server. Default is `ASERVER`. Value must be configured in *Broker Attributes*. Together with `class` and `service` it forms the fully qualified server name.	`SERVER-NAME`
`-t`*`wait_time`*	Set the wait time.	`WAIT`
`-u`*`user_id`*	User ID. With a secure broker, this will be used to perform authentication.	`USER-ID`
`-v`*`service`*	Name of service (part of the server definition). The default is `ASERVICE`. Value must be configured in *Broker Attributes*. Together with `class` and `server` it forms the fully qualified server name.	`SERVER-NAME`
`-w`*`password`*	Password. With a secure broker, this will be used to perform authentication.	`PASSWORD`
`-x`*`ssl_parms`*	SSL parameters. For example: `-x"TRUST_STORE=../ExxCACert.pem&VERIFY_SERVER=NO"` See also *SSL/TLS Parameters for SSL Clients*.	
`-y`*`sleep_time`*	Delay between messages in seconds.	
`-z`*`store[uwstatp[,uowtime]]`*	Test unit of work.	

Parameter	Description	ACI Field
`-1..12`	By default, if omitted, the highest possible API version is used. Note that if you specify the API version, certain features might depend on a minimum API version.	`API-VERSION`

bcos Parameters

Most `bcoc` parameters apply to `bcos`, too. The following table lists additional parameters or parameters that have a different meaning.

Parameter	Description	ACI Field
`-g output-file-name`	Name of output file for receive buffer (if not running in silent mode).	
`-i num_messages[,num_uows]`	If `num_uows` is not specified, the number of messages accepted. If `num_uows` is specified, the number of UOWs and messages therein that are accepted before deregistration. Example: `bcos -i10`	
`-i unit_of_work_flag`	Test unit of work.	

Environment Variables

This section describes the environment variables required for EntireX.

≫ **To check the definitions of the environment variables**

■ Enter the command `printenv`.

If the definitions are not correct, set them as described.

Environment Variable		Description
`EXXDIR`		This is the top level directory. It is set during the installation procedure.
`PATH`		This points to the directories /<Install_Dir>/EntireX/examples/bin and /<Install_Dir>/EntireX/bin.
`CLASSPATH`		This points to the current directory and to /<Install_Dir>/EntireX/classes/entirex.jar, which contains the class libraries and package for EntireX Java ACI, Java Wrapper, XML/SOAP Wrapper and EntireX Broker Agents.
`LD_LIBRARY_PATH`	(Solaris and Linux)	This points to the directory /<Install_Dir>/EntireX/lib and /<Install_Dir>/EntireX/examples/lib to load the shared library at runtime.
`LIBPATH`	AIX	

Setting up EntireX Security under UNIX

- Setting up EntireX Security for Broker Kernel
- Setting up EntireX Security for Broker Stubs

> **Note:** You must set up EntireX Security for broker kernel and - if your application(s) use API version 7 or below - also for broker stubs.

Setting up EntireX Security for Broker Kernel

≫ To set up EntireX Security for broker kernel under UNIX

1   Determine that all prerequisites for EntireX components have been met before setting up EntireX Security. See *UNIX Prerequisites*.

2   Insert the following statements into the DEFAULTS=BROKER section of the Broker attribute file:

- SECURITY=YES

3   Modify the security-specific section of the broker attribute file according to your requirements. See *Security-specific Attributes*. These parameters are used to determine whether you will use UNIX local security or LDAP-based authentication.

> **Note:** If you are using LDAP-based authentication, define your authorization rules in LDAP and reference these with Security-specific attribute `LDAP-AUTHORIZATION-RULE`.

> **Note:** UNIX authentication utilizes the /etc/passwd and /etc/shadow files. Functions used are: `getspnam()` and/or `getpwnam()`. These functions usually do not return the encrypted password to ordinary processes; therefore, USRSEC employs a daemon process to perform these functions.
> The daemon process (sagssxauthd2) must be owned by the "root" user (usually uid = 0) and have the setuid flag enabled. This process will automatically be started at the first authentication request and terminated when the kernel is shut down. In addition if sagssxauthd2 is terminated while the Broker kernel is running, a new sagssxauthd2 will be started automatically, if required.

Setting up EntireX Security for Broker Stubs

> **Note:** If you are running your application(s) at ACI version 7 or below, the following steps are required to install EntireX Security for the broker stubs in all environments where applications execute either as clients or servers. See *Platform Coverage* for where EntireX Security for broker stubs is supported.

These steps are not required if you are running your application(s) at ACI version 8 or above.

≫ To set up EntireX Security for broker stubs under UNIX

■ Copy the module secuexit.s[o∣l] (depending on platform) from the directory */<Install_Dir>/EntireX/examples/lib* to the directory */<Install_Dir>/EntireX/lib*.

> **Notes:**

1. *Secuexit.dll* is needed for backward compatibility if your applications issue any commands using ACI version 7 or below. Applications using ACI version 8 or above do not require this additional component in the stub.

2. For encrypted transport we strongly recommend using the Secure Sockets Layer/Transport Layer Security protocol. See *SSL/TLS, HTTP(S), and Certificates with EntireX* in the platform-independent Administration documentation.

> **Caution:** If stub tracing level is "> 1", unencrypted contents of the send/receive buffers are exposed in the trace.

Setting of EntireX Security for Broker stubs is now complete. Now you can set up the security components for the Broker stubs on the remaining operating systems where your application components are located.

EntireX Mini Runtime Considerations

The EntireX Mini Runtime is a set of components that can be used for the deployment of applications using the Broker ACI, the Broker RPC, and - under Windows only - DCOM Wrapper objects and the Broker ActiveX Control. Unlike the full EntireX Runtime, the Mini Runtime does not include Java capabilities. The EntireX Mini Runtime is provided as a separate package in the Software AG Installer.

If you need the EntireX Mini Runtime with your application, you may have to extend the path variable LD_LIBRARY_PATH (UNIX and Linux) or LIBPATH (AIX) to *<inst_root>/EntireX/lib* before starting your application.