software AG

webMethods EntireX

Post-installation Steps under UNIX

Version 9.6

April 2014

webMethods EntireX

This document applies to webMethods EntireX Version 9.6.

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

Table of Contents

1 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 **http://empower.softwareag.com/**. You can view the documentation for the Software AG Installer under *webMethods Product Documentation > webMethods Product Suite > Documentation by Product > System Requirements, Installation, and Upgrade* on the **Software AG Product Documentation** website.

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.

Shell Environment Settings

EntireX requires some shell environment variables to be set for proper operation. The installation provides two shell scripts *exxenv* and *exxenv.csh* in the *EntireX/INSTALL* directory. Source one of these scripts 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, and SHLIB_PATH under HP-UX).

- directory */opt/softwareag/EntireX/bin* is added to the list of directories in the PATH environment variable

- the directory */opt/softwareag/EntireX/lib* is added to LD_LIBRARY_PATH (LIBPATH under AIX, and SHLIB_PATH under HP-UX).

EntireX Mini Runtime

The EntireX Mini Runtime is a set of components that may 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 Security, the Mini Runtime does not include Java capabilities.

The file *exxminrt.cpz* is always copied during installation. Its target path is */opt/softwareag/EntireX/etc*.

Installing the Mini Runtime

▶ **To install the EntireX Mini Runtime**

■   For Linux, enter command

```
zcat < <path-to-minrt>/exxminrt.cpz | cpio -idvB -H newc
```

For other UNIX platforms, enter command

```
zcat < <path-to-minrt>/exxminrt.cpz | cpio -idmvcB
```

This command unpacks the Mini Runtime files in a directory *EntireX*, which in turn contains subdirectories *INSTALL*, *lib* and *userexits*. The *INSTALL* subdirectory contains setup files *exxminrtenv* and *exxminrtenv.csh* that set up the environment required by the Mini Runtime. After unpacking the Mini Runtime, these two setup files must be adapted manually: set the environment variable EXXDIR to the directory to which you unpacked the Mini Runtime.

Make sure that you set the file permissions correctly so that all users who need to access the Mini Runtime have the appropriate access rights.

📄   **Note:** Do not install the EntireX Mini Runtime over an already installed EntireX version.

Updating the Mini Runtime

The EntireX Mini Runtime is delivered as archive *exxminrt.cpz* located in directory */opt/software-ag/EntireX/etc* of the installation.

▶ **To update the EntireX Mini Runtime**

1   Go to the directory where you installed the EntireX Mini Runtime.

2   For Linux, enter command

```
zcat < <path-to-minrt>/exxminrt.cpz | cpio -idvB -H newc
```

For other UNIX platforms, enter command

```
zcat < <path-to-minrt>/exxminrt.cpz | cpio -idmvcB
```

This command replaces all updated files of the EntireX Mini Runtime. It also replaces the two setup files *exxminrtenv* and *exxminrtenv.csh*, so either make a backup of these files or readapt them manually after the update by setting environment variable `EXXDIR` to the directory to which you unpacked the Mini Runtime.

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 and SSL ports defined in the custom panel during installation. Default port numbers are 1971 (TCP/IP) and 1958 (SSL).

▶ **To change the default port of the default broker**

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

   ▪ Use System Management Hub. See *Stopping a Local Broker* under *Broker Administration using System Management Hub* in the UNIX administration documentation.

   ▪ Source the EntireX environment file *<Installation_Dir>/EntireX/INSTALL/exxenv[.csh]* and enter command:

   ```
   <Installation_Dir>/EntireX/bin/defaultbroker stop
   ```

2   Edit the configuration file *config/entirex.config* and change the TCP/IP and SSL port numbers to a different value. For other broker instances, see `PORT` under *Broker Attributes* in the platform-independent administration documentation.

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

   ▪ Use System Management Hub. See *Starting a Local Broker* under *Broker Administration using System Management Hub* in the UNIX administration documentation.

   ▪ Enter command:

```
<Installation_Dir>/EntireX/bin/defaultbroker start
```

Startup Daemon 'etbsrv'

This daemon runs in the background for the System Management Hub agents to administer broker instances. It is installed as `etbsrv` in the directory /opt/softwareag/EntireX/bin.

▶ **To start the daemon**

■   Enter the following command:

```
- /etc/init.d/sag<n>etbsrv start
```

where `<n>` is a sequential, installation-dependent number.

This ensures that `etbsrv` is always running and ready to receive start/stop commands from System Management Hub agents. Note that the startup script *sag<n>etbsrv* sources the SAG environment file *EntireX/INSTALL/exxenv*.

▶ **To stop the daemon**

■   Enter the following command:

```
- /etc/init.d/sag<n>etbsrv 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
Solaris, Linux	/etc/init.d
AIX	/etc
HP-UX	/sbin/init.d

See also *Broker Administration using System Management Hub* in the UNIX administration documentation.

Verifying the EntireX Installation

- EntireX Broker
- Developer's Kit
- Broker Stubs
- Environment Variables

EntireX Broker

To verify successful installation, you can use the etbcmd utility in the directory */opt/softwareag/EntireX/bin*. It checks whether the Broker is up and running and responds to various requests as described below.

1. Start the Broker by executing shell script *etbstart* in the */opt/softwareag/EntireX/bin* directory:

```
etbstart ETBnn
```

2. Execute:

```
etbcmd -b<broker-id> -cPING -dBROKER
```

If successful, the message `PING broker broker-id successfully performed` is returned.

In a Natural environment, you can also verify correct installation under UNIX by installing the EntireX Broker Tutorial. See *Installing the Natural-based EntireX Broker Tutorial under UNIX*.

Developer's Kit

▶ **To verify the installation**

1   Check that EntireX Broker is installed on the target node and is up and running.

2   Create and run a sample application (see below).

▶ **To create and run a sample application**

■   Create and run a sample application provided in the directory */opt/softwareag/EntireX/examples/RPC/basic/example/CServer*:

The following steps describe how to run the sample application locally:

1. Verify that the EntireX environment is adapted to your environment.

2. Verify that the EntireX Broker stub is installed in your EntireX environment.

3. Create an example RPC server with the following steps:

   a. Change to directory */opt/softwareag/EntireX/examples/BrokerRPC/RPC/basic/example/CServer*.

   ```
   cd /opt/softwareag/EntireX/examples/RPC/basic/example/CServer
   ```

   b. Create the server library with the command `make`.

   c. Add this directory to the `LD_LIBRARY_PATH` (`LIBPATH` under AIX and `SHLIB_PATH` under HP-UX).

   d. Modify the configuration *server.cfg* file to set up the Broker address for your environment.

   e. Start the RPC server with

   ```
   ../../../bin/rpcserver CFG=server.cfg
   ```

4. Create an example RPC client with the following steps:

   a. Change to directory */opt/softwareag/EntireX/examples/RPC/basic/example/CClient*.

   ```
   cd /opt/softwareag/EntireX/examples/RPC/basic/example/CClient
   ```

   b. Create the client with the command:

   ```
   make
   ```

   The client will be built in the current directory.

   c. Start the RPC client with

   ```
   client <your_broker>@RPC/SRV1/CALLNAT
   ```

   Set `<your_broker>` to the configured Broker of the server configuration.

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 /opt/softwareag/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. In the following example, both programs are run on the local node. The Broker ID is ETB048 and the REPEATS value is 100 (see *convClt Parameters* below). The value -2 indicates that the Broker is Version 2.1.1 or above.

```
cd /opt/softwareag/EntireX/examples/ACI/conversational/C
convSvr -i100 -bETB048 -2&
convClt -i100 -bETB048 -2
```

If you have problems, check that environment variable `ETBLNK` refers to the Broker shared library in directory */opt/softwareag/EntireX/lib*. If not, rerun `SAGINST`. If you receive error `02150148`, verify that the Broker is running on the target machine.

convClt Parameters

Parameter	Description	ACI Field
-a*locale*	*locale_string,environment*	`LOCALE-STRING,` `ENVIRONMENT`
-b*brokerid*	Broker ID (same as the database ID). This is an EntireX Broker configuration parameter. For example: `convSrv -bETB045`	`BROKER-ID`
-c*class*	Server class (part of the server definition).	`SERVER-CLASS`
-e*encryptionlevel*	Values: 0,1,2. Force encrypted communication for client - broker - server. See *Encryption* under *Writing Applications using EntireX Security* in the ACI Programming documentation.	`ENCRYPTION-LEVEL`
-g*input_file_name*	Use a file as send buffer.	
-i*repeats*	The number of requests that are accepted by the example program before it deregisters. For example: `convSrv -i10`	
-l	Silent mode. Suppress output for every incoming request.	
-n*token*	Specify the security token, if desired.	`TOKEN`
-p*packetsize*	`convClt` only. The size, in bytes, of the packets that will be transmitted (i.e. size of the send buffer). For example: `convClt -p10000`	
-r*compression_level*	Compression level. Values: N|Y|0-9.	`COMPRESSLEVEL`
-s*server*	Name of server.	`SERVER-NAME`
-u*userid*	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 fully qualified server name used by the example program to	`SERVICE`

Parameter	Description	ACI Field
	register. The default is ACLASS, ASERVER, ASERVICE. Values must be configured in the EntireX Broker.	
-w*password*	Password. With a secure broker, this will be used to perform authentication.	PASSWORD

-x*sslparms*	SSL parameters. For example:
	`-x"TRUST_STORE=C:\SoftwareAG\EntireX\`
	`etc\ExxCACert.pem&VERIFY_SERVER=NO"`	

-1..9	Specify the API version (note that certain features might depend on a specific minimum level).	API-VERSION

convSrv Parameters

Most `convClt` parameters apply to `convSrv`, 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).	

Environment Variables

This section describes the environment variables required for Developer's Kit.

▶ **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 */opt/softwareag/EntireX/examples/bin and /opt/softwareag/EntireX/bin*.
CLASSPATH		This points to the current directory and to */opt/softwareag/EntireX/classes/entirex.jar*, which contains the class libraries and package for EntireX Java ACI, Java Wrapper, XML/SOAP Wrapper and EntireX Broker Agents.
SHLIB_PATH	(HP-UX)	This points to the directory */opt/softwareag/EntireX/lib* and */opt/softwareag/EntireX/examples/lib* to load the shared library at runtime.
LD_LIBRARY_PATH	(Solaris and Linux)	
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* in the EntireX Release Notes.

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

- SECURITY=YES

3   Modify the DEFAULTS=SECURITY section of the Broker attribute file according to your requirements. These parameters are used to determine whether you will use UNIX local security or LDAP-based authentication. See *Security-specific Attributes* (DEFAULTS=SECURITY) under *Broker Attributes* in the platform-independent administration documentation. If you are using LDAP-based authentication, authorization rules are not available to you.

> **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 *List of Components per Platform* under *Platform Coverage* in the EntireX Release Notes 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**

1   Copy the module secuexit.s[o|l] (depending on platform) from the directory */opt/softwareag/EntireX/examples/lib* to the directory */opt/softwareag/EntireX/lib*.

> **Note:** *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   To use encrypted communication, set the environment variable `NA2_BKPRIV=1`. For no encryption, set `NA2_BKPRIV=0` (default).

> **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.

Installing the Natural-based EntireX Broker Tutorial under UNIX

▶ **To install the Tutorial**

1   Use the Natural utility SYSOBJH to load all objects contained in *SYSETB.TRA* in directory */opt/softwareag/EntireX/etc*. See *Object Handler* in the Natural Tools and Utilities documentation for more information.

2   Invoke Natural, logon to library *SYSETB* and issue the CATALL command to catalog all the programs in the library.