software AG

webMethods EntireX

General Installation Information

Version 10.5

October 2019

WEBMETHODS

Table of Contents

Preface

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 *http://documentation.softwareag.com* > *Cross-Product Guides* > *Software AG Installer and Update Manager*. Alternatively, you can install and configure EntireX using Command Central; see *Installing EntireX using Command Central*.

The EntireX installation documentation for UNIX and Windows assumes you have successfully installed the product using the Installer or Command Central. This section covers the following topics:

License Concept	License concept for webMethods EntireX.
Installation Packages	EntireX Installation Packages.
Software AG Mainframe Product Licensing	Software AG's mainframe product licensing procedures, license check software and license file.
Port Numbers	Ensuring unique port numbers; port number configuration.
Migration	Migrating EntireX data from an earlier version (UNIX and Windows).
Installing Fixes	Installing fixes for EntireX.
Installing using Command Central	Installing EntireX using Command Central.
Template-based Provisioning	Installing and configuring EntireX components using Command Central.
Building an EntireX Docker Image	Using Command Central and scripts provided by EntireX to build a Docker image and run Docker container for EntireX Broker, RPC Server for XML/SOAP and RPC Server for Java.
Design-time	Installing the EntireX design-time separately (outside Designer).

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://documentation.softwareag.com**.

In addition, you can also access the cloud product documentation via **https://www.softwareag.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.

I EntireX Installation Packages

2 EntireX Installation Packages

EntireX consists of multiple installable components. Some components can be installed alone; some have dependencies on other components. These dependencies are handled automatically by the installation. For example, if you select package EntireX Broker, the EntireX Mini Runtime is installed automatically. This chapter describes the individual components and any dependencies on other EntireX packages.

> **Note:** There may be additional dependencies on other products or components, for example infrastructure components, that are not explicitly listed here.

Adapters > EntireX Adapter

The Adapters > EntireX Adapter package (under the **Adapters** node) contains the EntireX Adapter that runs in webMethods Integration Server. You can run services between mainframe platforms and the Integration Server. Run most scenarios from the EntireX Adapter directly. Use the EntireX Broker for additional scenarios. Connect COBOL, Natural or PL/I with z/OS, z/VSE or BS2000. See *EntireX Adapter*.

For installing this package with Command Central command line, the product ID is `EntireXAdapter`.

Designer > EntireX Designer

The Designer > EntireX Designer package (under the **Designer** node) contains EntireX design-time. It lets you create services to/from mainframe platforms and the webMethods Integration Server easily. Also Java or Web services. Let the EntireX wizards guide you through extracting, mapping and testing the generated RPC clients/servers. See *Software AG Designer*.

For installing this package with Command Central command line, the product ID is `DesignerEXXEntireX`.

EntireX Core Files

The EntireX Core Files package contains all files to be able to run any number of RPC servers (for all EntireX supported languages, C, Java, .NET, etc.) together with pre-configured property files. You can run EntireX clients and servers, using the runtime provided by this package. See *EntireX RPC Servers and Listeners*

For installing this package with Command Central command line, the product ID is `EntireXCore`.

If you select this package for installation, the package EntireX MiniRuntime is also installed automatically.

EntireX Administration and Monitoring

The EntireX Administration and Monitoring node expands to the following three subpackages:

- Application Monitoring Data Collector
- Command-line Scripts
- Mainframe Broker Monitoring

Application Monitoring Data Collector

This package includes the Application Monitoring Data Collector. See *Starting and Stopping the Application Monitoring Data Collector* in the Application Monitoring documentation.

For installing this package with the Command Central command line, the product ID is `EntireXAppMonDC`.

Command-line Scripts

This package allows you to administer and monitor EntireX using scripts and utilities. See *Monitoring EntireX with Command-line Scripts*.

For installing this package with the Command Central command line, the product ID is `EntireXCommandLineScripts`.

Mainframe Broker Monitoring

This package allows you to monitor EntireX mainframe brokers using Command Central. See *Introduction to Monitoring EntireX Mainframe Broker using Command Central*.

For installing this package with the Command Central command line, the product ID is `EntireXMainframeProxy`.

EntireX Broker

The EntireX Broker package contains all necessary files to be able to run any number of brokers. It contains the EntireX Broker administration services, which are necessary for administrating brokers using Command Central, and a pre-configured EntireX Default Broker ETB001. See *EntireX Broker*.

For installing this package with Command Central command line, the product ID is `EntireXBroker`.

If you select this package for installation, the package EntireX MiniRuntime is also installed automatically.

EntireX Development Tools

The EntireX Development Tools package provides Windows .NET developers with files and the Microsoft Visual Studio Wizard for EntireX .NET Wrapper. It contains examples for COBOL and PL/I programmers (z/OS, z/VSE, BS2000, Micro Focus), and header files to develop C applications. See *Microsoft Visual Studio Wizard for EntireX .NET Wrapper | Delivered Examples for the COBOL Wrapper | Delivered Examples for the PL/I Wrapper | C Wrapper*.

For installing this package with Command Central command line, the product ID is `EntireXDevelopmentTools`.

If you select this package for installation, the package EntireX MiniRuntime is also installed automatically.

EntireX Mini Runtime

The EntireX Mini Runtime package contains files that enable you to run EntireX clients written in .NET, C and DCOM. See *EntireX Mini Runtime Considerations*.

For installing this package with Command Central command line, the product ID is `EntireXMiniRuntime`.

EntireX WebService

The EntireX WebService package installs the Software AG Runtime and deploys the Listener for XML/SOAP and the RPC Server for XML/SOAP there. This package handles dependencies on other packages outside of EntireX. You will be able to provide and access Web services from EntireX RPC endpoints using the Software AG Runtime. See *Running the RPC Server for XML/SOAP in the Software AG Runtime* and *Listener for XML/SOAP*.

For installing this package with Command Central command line, the product ID is `EntireXWebService`.

If you select this package for installation, the packages EntireX Core and EntireX MiniRuntime are also installed automatically.

Overview of Packages

This table provides an overview of the EntireX packages, the corresponding Command Central product ID, and the additional packages that are installed automatically.

Package			Command Central Product ID	Additionally Installed Packages [*]
Adapters				
	EntireX Adapter		`EntireXAdapter`	none
Designer				
	EntireX Designer		`DesignerEXXEntireX`	none
EntireX				
	Core Files		`EntireXCore`	EntireX Mini Runtime
	Administration and Monitoring			
		Application Monitoring Data Collector	`EntireXAppMonDC`	none
		Command Line Scripts	`EntireXCommandLineScripts`	none
		Mainframe Broker Monitoring	`EntireXMainframeProxy`	none
	Broker		`EntireXBroker`	EntireX Mini Runtime, Command Line Scripts
	Development Tools		`EntireXDevelopmentTools`	EntireX Mini Runtime
	Mini Runtime		`EntireXMiniRuntime`	none
	Web Services		`EntireXWebService`	EntireX Core Files, EntireX Mini Runtime

[*] There may be additional dependencies on other products or components, for example infrastructure components, that are not explicitly listed here.

II License Concept for webMethods EntireX

3 License Concept for webMethods EntireX

License Key Checks

EntireX on all applicable platforms must have a valid license key. If the license key does not match for a component, the component cannot be used. The license may be checked during installation and/or when the component is started.

For EntireX Broker (all platforms) and EntireX RPC servers under UNIX and Windows, the license file is checked once daily. If the license file is due to expire in the next 30 days, a warning is issued. This is written to the log file.

> **Notes:**

1. The documentation delivered with this product covers all components of EntireX on all platforms, including those for which you may not have a valid license. The inclusion of a component or feature in this documentation does not imply that this component/feature is available for your installed version of the product.

2. The fact that you are able to download a component or start and run a component, does not mean you are entitled to do so. In any case you have to have a valid license for that component.

License Key for z/OS received by E-Mail

Normally the EntireX license key is delivered directly on the installation medium as file EXX105.LICS. This file contains the license information necessary to start EntireX broker. It is no longer necessary to transfer the file to the host file system.

However, if you received the EntireX license key file EXX105.LICS by e-mail or diskette, you need to transfer this file to the host file system, using FTP. You will need:

- an FTP client (comes with Windows) and an FTP server on the mainframe, or
- an FTP server on Windows and an FTP client on the mainframe.

> **To transfer license file EXX105.LICS to your mainframe host**

1   Transfer the file EXX105.XML to the host, using FTP.

   As the target of the host, use the data set EXX105.LICS provided with the EntireX kit or allocate a new data set on z/OS with the following attributes:

```
DSORG=PS
RECFM=FB
BLKSIZE=800
LRECL=80
```

You must use *binary* transfer mode to transfer the file EXX105.XML to the host. This ensures that the license key remains unaltered.

> **Note:** You cannot "cut and paste" the certificate using a 3270 emulator due to format incompatibilities: the mainframe is an EBCDIC system, whereas the certificate is an ASCII-formatted binary file.

2   Ensure that the LICENSE DD card in the broker startup JCL references data set EXX105.LICS.

Example FTP Commands

```
open <host>
<userid>
<password>
binary
put EXX105.xml '<hlq>.EXX105.LICS'
close
bye
```

License Key on UNIX and Windows

On UNIX and Windows platforms, the license file is requested during installation and must be provided by the user performing the installation. The Software AG Installer renames this file to "*license.xml*" and copies it to the *EntireX/config* directory.

> **Note:** You cannot change the name or location of the license file. You may update the license file at a later time, but make sure the name and location are the same.

License Key under z/VSE

The EntireX license file *EXX105.LICS* is delivered on the installation medium. It is referenced in the EntireX Broker startup job control. For this purpose, you can copy the license file to a sublibrary or to a sequential disk file. See *Step 2: Install the EntireX License File* for detailed information.

III   Software AG Mainframe Product Licensing

4 Software AG Mainframe Product Licensing

This document describes Software AG's mainframe product licensing procedures, license check software and license file.

Notations Used in this Document:

- *vrs*, *vr***or** *v.r.s***:**
  When used in this document, the notation *vrs*, *vr* or *v.r.s* represents the three-digit or two-digit version number of the Software AG product.

- *ppp***:**
  When used in this document, the notation *ppp* represents the three-character product code of the Software AG product: ADA for Adabas, NAT for Natural, EXX for EntireX, COM for Complete, WCP for Entire Net-Work and NPR for Entire System Server.

Distributed License Check Software

This section describes the data sets (files) containing this software. All licensing data sets have names starting with the characters MLC. The *vrs* in the names represents the version of the license check software, which is not necessarily the same as the version of your Software AG mainframe product.

For instructions on copying the data sets from the product installation medium to disk and loading the libraries contained on the data set, refer to the appropriate installation procedure for your mainframe product.

- z/OS Licensing Data Sets
- z/VSE Licensing Data Sets
- BS2000 Licensing Data Sets

z/OS Licensing Data Sets

The following licensing data sets are provided in z/OS environments:

Data Set Name	Description
MLC*vrs*.LOAD	Mainframe license check load modules
MLC*vrs*.JOBS	Mainframe license check sample jobs

z/VSE Licensing Data Sets

The following licensing data sets are provided in z/VSE environments:

Data Set Name	Description
MLC*vrs*.LIBR	Mainframe license check load modules
MLC*vrs*.LIBJ	Mainframe license check sample jobs

BS2000 Licensing Data Sets

The following licensing data sets (files) are provided in BS2000 environments:

Data Set Name	Description
MLC*vrs*.MOD	Mainframe license check load modules
MLC*vrs*.JOBS	Mainframe license check sample jobs

License File

You must install a valid license file on all mainframe platforms in which your Software AG mainframe product is installed.

The license file is provided as an XML document (encoding is US-ASCII). If you receive the license file as an email attachment, save it directly to disk. To avoid unwarranted modification, do not save it from a browsing tool or text editor. For example, Internet Explorer might change the XML encoding tag value to its local encoding.

The XML document can be viewed using a browsing tool or text editor on a PC. It can also be viewed on the mainframe using the **DISPLAY** function of the license utility, LICUTIL, described later in this document. The license file contains text, which represents the licensing information and a digital signature, the license key. Among other things, it displays Software AG legal notices and environmental information.

⚠️ **Caution:** The license file must remain in ASCII format -- even on the mainframe. It must not be modified. Any modification of the license file will invalidate the digital signature and the license check will fail. If the check fails, please contact your Software AG technical support representative.

Delivery and Installation

Your individual product license is shipped on the product installation medium as *pppvrs*.LICS data set (file) or as an e-mail attachment in the file format *pppvr*.xml.

Once the license file is received, either copy it from the product installation medium to disk or use native FTP commands to transfer it to your mainframe host before using it during the mainframe product installation.

During the mainframe product installation process, the license file is then loaded and processed as required by your Software AG mainframe product. Refer to the appropriate installation procedure for your Software AG mainframe product for further information.

Copying a License File to Disk

For instructions on copying a *pppvrs*.LICS data set (file) from the product installation medium to disk, refer to the appropriate installation procedure for your Software AG mainframe product.

Transferring a License File from PC to a z/OS Host Using FTP

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands provided in this section.

⛔ **Caution:** Using utilities instead of native FTP commands for the license file transfer may corrupt the license key.

⚠️ **Important:** Make sure to switch to binary transfer and verify that the resulting data set has `RECFM=FB` or `RECFM=F` and `LRECL=80`.

≫ **To transfer a license file from the PC to a z/OS host, perform the following steps:**

1   Save the product license file e-mail attachment as *pppvr*.xml on your PC's hard disk.

2   Start an FTP session for communication with the z/OS host using the following FTP command:

```
ftp host-name
```

where *host-name* is the name of the z/OS host.

Enter your z/OS host login ID and password.

3   Switch to binary data mode (the license file must retain its format as ASCII during the transfer):

```
binary
```

4   Specify that the data set for the license file must be written with `RECFM=FB` and `LRECL=80`:

```
quote site RECFM=FB LRECL=80 BLKSIZE=4000
```

`RECFM=F` is also supported for product license files.

5   Write the license file as a data set on the z/OS system:

```
put pppvr.xml 'hilev.pppvrs.LICS'
```

where *hilev* is the high-level qualifier (for example, `SAG`) to be used for the data set.

This command will create a data set called *hilev*.`pppvrs`.`LICS` with `RECFM=FB`, `LRECL=80`, and the license information stored in the data set will be in ASCII format.

Transferring a License File from PC to a z/VSE Host Using FTP

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands described in this section.

The FTP sample session described in these steps loads the license file onto a member of a z/VSE library. It is possible to load the license file onto a sequential z/VSE data set instead. For more information, refer to your FTP documentation.

🛑   **Caution:** Using utilities instead of native FTP commands for the license file transfer may corrupt the license key.

⚠️   **Important:** Make sure to switch to binary transfer and verify that the resulting data set has `RECFM=FB` or `RECFM=F` and `LRECL=80`.

≫ **To transfer a license file from the PC to a z/VSE host, perform the following steps:**

1   Save the product license file e-mail attachment as *pppvr*.xml on your PC's hard disk.

2   Start an FTP session for communication with the z/VSE host using the following FTP command:

```
ftp host-name
```

where *host-name* is the name of the z/VSE host.

Enter your z/VSE host login ID and password.

3   Once the FTP session has been established, you can optionally delete the default working prefix. To determine what the default working prefix is, enter the following command:

```
pwd
```

To delete the default working prefix, enter the following command:

```
cd ..
```

4   Point to the z/VSE sublibrary for the license file, for example, `USERLIB.MYSUBLIB`:

```
cd USERLIB.MYSUBLIB
```

5   Switch to binary data mode (the license file must retain its ASCII format during the transfer):

```
binary
```

6   Specify that the data set for the license file must be written with `RECFM=FB` and `LRECL=80`:

```
quote site recfm fb 80
```

`RECFM=F` is also supported for product license files.

7   Write the license as a member in the libraries on the z/VSE system you identified in Step 4:

```
put pppvr.xml pppvrs.LICS
```

This command will write the license file *pppvr*.xml to a member called *pppvrs*.LICS. If you requested a library and sublibrary of `USERLIB` and `MYSUBLIB` in Step 4, the *pppvrs*.LICS file would reside in the `USERLIB.MYSUBLIB` library. The data set will have `RECFM=FB`, `LRECL=80`, and the license information stored in the member will be in ASCII format.

Transferring a License File from PC to a BS2000 Host Using FTP

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands described in this section.

⬤ **Caution:** Using utilities instead of native FTP commands for the license file transfer may corrupt the license key.

⚠ **Important:** Make sure to switch to binary transfer and verify that the resulting data set has file type SAM and `RECFORM=V`.

≫ **To transfer a license file from the PC to a BS2000 host, perform the following steps:**

1 Save the product license file e-mail attachment as *pppvr*.xml on your PC's hard disk.

2 Start an FTP session for communication with the BS2000 host using the following FTP command:

```
ftp host-name
```

where *host-name* is the name of the BS2000 host.

Enter your BS2000 host user ID, password and account number (if relevant).

3 Switch to binary data mode (the license file must retain its ASCII format during the transfer):

```
binary
```

4 Specify that the file for the license file must be written with `FCBTYPE=SAM` and `RECFORM=V`:

```
quote file pppvrs.LICS,FCBTYPE=SAM,RECFORM=V
```

5 Write the license file to the BS2000 host:

```
put pppvr.xml pppvrs.LICS
```

Product License Check FAQs

- Why has Software AG introduced a license check for products on the mainframe?
- Which information is contained in a product license?
- What exactly is meant by the CPU ID?
- Which Software AG mainframe products require a product license?
- How is the product license file installed?
- How can I read the product license file?
- How can I display machine-specific data?
- When is the product license checked?
- Which items of a product license are checked?
- Does the Software AG mainframe license check software support a disaster recovery scenario?
- How do I find out when the product license expires?
- What happens if the product license is incorrect, insufficient or not installed?
- How can I get a new product license file if the delivered license file is insufficient for my environment?
- Do I need separate product licenses for different machines?

Why has Software AG introduced a license check for products on the mainframe?

Software AG wants to ensure that customers run our software products only on mainframe machines for which they have valid and sufficient product licenses. This ensures that we have better control of the use of our software products. On other platforms (UNIX, Windows), the product license check has been established for many years.

Which information is contained in a product license?

A product license is a sequential file with US-ASCII text in XML format containing the following items:

- Software AG header

- customer information (Name, ID)

- encrypted license key

- license expiration date (or unlimited)

- product information (product code, version, name)

- environment information, including operating system type, CPU ID, LPAR ID, system name, and capacity measured in million service units (MSUs)

What exactly is meant by the CPU ID?

IBM defines the CPU ID as the "central processing complex node descriptor sequence number". It is the unique hexadecimal machine serial number without the machine model number.

The CPU ID and other machine data can be viewed by using the **DISPLAY** function of the license utility, LICUTIL, described later in this document.

Which Software AG mainframe products require a product license?

- Adabas and the following add-ons:
  - Adabas for zIIP (AZPAD), Event Replicator for zIIP (AZPRP), Adabas Cluster Services (ALS), Adabas Parallel Services (ASM), Adabas Caching Facility (ACF), Adabas Online Services (AOS), Adabas Delta Save (ADE)
- Natural
- EntireX
- Com-plete
- Entire Net-Work
- Entire System Server

How is the product license file installed?

Transfer the license file from the product installation medium or PC to disk as described in the installation documentation for your Software AG mainframe product or the various *Transferring a License File from PC to a xxx Host section* sections respectively. Then proceed as described in the appropriate installation steps for your mainframe product.

How can I read the product license file?

- The product license file can be read on a PC by means of an XML editor (file type .xml) or any PC text editor.
- The license utility, LICUTIL, provided with the Software AG mainframe license check software includes a function that allows you to display the license (**DISPLAY** function). For more information about the license utility, read *Using the License Utility: LICUTIL*.

How can I display machine-specific data?

The license utility, LICUTIL, provided with the Software AG mainframe license check software includes a function that allows you to display machine-specific data (**DISPLAY** function). For more information about the license utility, read *Using the License Utility: LICUTIL*.

When is the product license checked?

The product license is checked every time the product is initialized. In addition, the product license is checked once a day.

Thirty days before the license expires, license check failure messages will warn you that it is time to obtain a new license.

If a license check fails, contact your Software AG technical support representative.

Which items of a product license are checked?

Product license items are divided into three logical license groups (License Key, Product-Specific Information and Machine-Specific Information) and checked in that order. If an inconsistency in any one of these groups is detected by the license checker, the succeeding items of that logical license group are also checked. Then the check terminates with corresponding error messages. For example, if the license key expires, the license key check terminates with an error before the license checker checks any product-specific or machine-specific information.

The following table identifies the product license items comprising each logical group:

Checking Order	Logical License Group Name	Product License Item Included
1	License Key	The encrypted license key
2	Product-Specific Information	■ The expiration date (if any) ■ The operating system (z/OS, z/VSE or BS2000) ■ The product code (for example, ADA, NAT or EXX) ■ The product version, in either $v.r.$ or $v.r.s$ format.
3	Machine-Specific Information	■ The machine CPU ID ■ The machine capacity measured in MSU (million service units); only in z/OS and z/VSE environments ■ The logical partition (LPAR) ID (if applicable)

Does the Software AG mainframe license check software support a disaster recovery scenario?

Yes. Please contact your Software AG sales representative to obtain a license file which is able to support disaster recovery scenarios. The parameter DisasterRecoveryCPUIDs in this license file allows you to specify one or more CPU IDs of the machines to be used when your production machine fails. If you do not have these CPU IDs available, it is possible to specify the value `Unknown`. In this case, the license checker issues a warning message when a disaster recovery scenario is assumed.

In a disaster recover scenario, no machine data besides the CPU ID is checked.

How do I find out when the product license expires?

Thirty days before the license expires, license check failure messages will warn you that it is time to obtain a new license.

What happens if the product license is incorrect, insufficient or not installed?

An error message is issued and the product for which the license is checked either continues to run or terminates. The message is either output as a primary message or on the system console.

How can I get a new product license file if the delivered license file is insufficient for my environment?

Contact your Software AG sales representative to get a correct product license file according to your contract. It can be shipped by e-mail either in ASCII format or in the converted assembler input format.

Do I need separate product licenses for different machines?

No, it is possible to have multiple CPU IDs defined in one license file.

Using the License Utility: LICUTIL

The license utility, LICUTIL, is a batch utility that is used to check the license file, convert the license file into an assembler source module and display the license file and machine data.

LICUTIL is provided with Software AG's mainframe license check software contained on the data sets MLC*vrs*.LOAD (z/OS), MLC*vrs*.MOD (BS2000) and MLC*vrs*.LIBR (z/VSE).

Error messages for the license utility are documented with the rest of your Software AG product messages.

This section provides detailed information of each function provided by the LICUTIL utility. The functions can be executed by using the sample JCL described later in this section. The syntax of the functions that can be supplied with the LICUTIL command is shown in the following diagram:

```
CHECK {ZIIP}
DISPLAY {ALL | LICENSE | MACHINE}
END | EXIT
HELP
LOADLIC module-name
MAKE

SET          ⎧  DATE={yyyy-mm-dd | yyyymmdd}     ⎫
             ⎨  FORMAT={FREE|LEFT|STRUCTURED}    ⎬
             ⎪  PRODVER=pppv.r.s                 ⎪
             ⎩  RMODE={ANY|24}                   ⎭
```
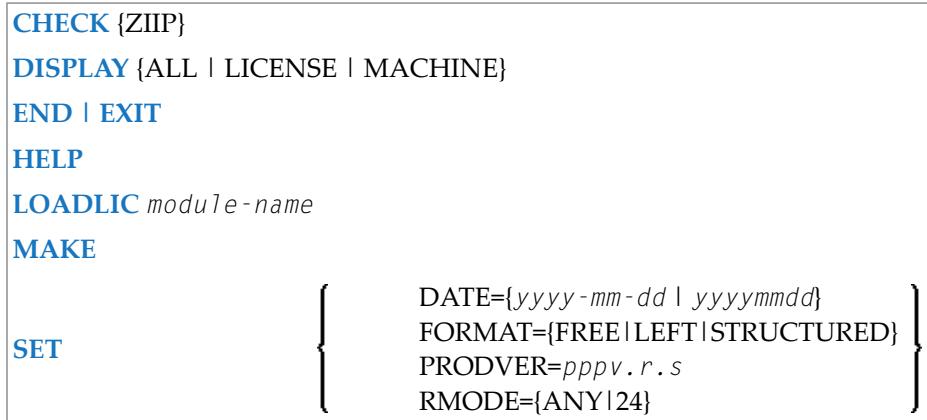
The parameters and variable values that can be supplied with each function are described in the following section. For explanations of the syntax symbols used in the diagram, please read the documentation for your specific Software AG mainframe product.

- CHECK Function: Check the Product License
- DISPLAY Function: Display License Items and Machine Data
- END/EXIT Functions: Terminate the License Utility
- HELP Function: View LICUTIL Help
- MAKE Function: Convert the License File to Assembler Source
- LOADLIC Function
- SET Function: Set LICUTIL Options
- JCL Examples and Requirements
- LICUTIL Licensing Messages

CHECK Function: Check the Product License

The CHECK function can be used to check the license file against the product's execution environment. The original license file must be provided as input along with the product code and version. If the license file is not valid, appropriate error messages are returned. LICUTIL terminates with one of the following return codes:

Return Code	Meaning
0	The license check has been successful and the product continues to run.
4	Warning message(s) are written to the job log but the product continues to run.
8	Warning message(s) are written to the user terminal (or job log) but the product continues to run.
12	Error message(s) are written to the user terminal (or job log) and the product terminates.

⚠ **Important:** In order to run the CHECK function successfully, you must have previously specified the **SET PRODVER** function to provide the product code and version number. The parameter ZIIP must be specified if the license is for a zIIP enabling product, for example, Natural Batch for zIIP.

The following items are checked, in the following order:

1. License key.

2. Product-specific data, including the product code, version, operating system and expiration date.

3. Machine-specific data, including the CPU IDs, the machine capacity (measured in MSU, million service units) on only z/OS and z/VSE systems, and the logical partition (LPAR) ID.

DISPLAY Function: Display License Items and Machine Data

Use the DISPLAY function to display the contents of the product license items (license file, product-specific information and machine-specific information) and to display machine data (CPU ID, machine type, model ID, LPAR ID, MSU) about the machine on which the utility is being run.

The parameters of this function are described in the following table. At least one parameter is required.

Parameter	Description
ALL	Display the product license items and machine data. ALL is the default.
LICENSE	Display only the product license items.
MACHINE	Display only the machine data.

END/EXIT Functions: Terminate the License Utility

Use the END or EXIT function to terminate the license utility. These functions should be used if LICUTIL is running interactively in environments other than batch (for example, TSO under z/OS).

HELP Function: View LICUTIL Help

Use the HELP function to review a list of the available LICUTIL functions and their syntax.

MAKE Function: Convert the License File to Assembler Source

Use the MAKE function to convert the license file into an assembler source. After assembling this source (and linking it), the resulting license module can be used by defined Software AG products (for example, Adabas or Natural) instead of the license files. For more information about using license modules, refer to the installation documentation provided with your Software AG product.

Sample Output

Here is some sample output from this function when run on a z/OS system:

```
MLC9006 License converted to assembler output. 042 records written
        to  SYSUT2.
```

LOADLIC Function

This function applies only on z/OS.

Use the LOADLIC function to load the license module after you converted (**MAKE** function), assembled and linked the assembler source. `module-name` denotes the name of the license module in the load library.

LOADLIC can be specified only once per LICUTIL session.

After the license module has been successfully loaded by LOADLIC, you can further process the module using other LICUTIL functions such as **CHECK** and **DISPLAY**.

SET Function: Set LICUTIL Options

Use the SET function to set options for the license utility.

The parameters of this function are described in the following table. At least one parameter is required.

Parameter	Description
DATE	Sets the date for the license check. The default is today. The format of the date can be either `yyyy-mm-dd` or `yyyymondd`, where `yyyy` is the four-character year, `mm` is the two-digit month, `mon` is the three-character month abbreviation and `dd` is the two-digit day.
FORMAT	Sets the display format for the **DISPLAY** function. There are three different formats: ■ Specify FREE (the default) to produce output without XML tags. ■ Specify LEFT to produce output with each XML tag in a new line. ■ Specify STRUCTURED to produce output with each XML tag indented.

Parameter	Description
PRODVER	Sets the product code and version for the **CHECK** function. The format in which the product code and version should be specified is *pppv.r.s*. For a zIIP enabling product such as Natural Batch for zIIP, the product code can consist of up to five characters: *pppppv.r.s*.
RMODE	Sets the RMODE for the assembler source produced by the **MAKE** function: ■ Specify 24 (the default) for 24-bit addressing mode. ■ Specify ANY for either 24- or 31-bit addressing mode.

JCL Examples and Requirements

Sample JCL of the license utility functions can be found in the following members of the licensing data sets (files): MLC*vrs*.JOBS for z/OS and BS2000, and MLC*vrs*.LIBJ for z/VSE.

Member	Functions Included	Demonstrates how to...
LICCHECK	SET PRODVER=*pppv.r.s* DISPLAY CHECK	Check a license file with LICUTIL. For a zIIP enabling product such as Natural Batch for zIIP, the product code can consist of up to five characters: *pppppv.r.s*.
LICMAKE	MAKE	Convert the license file to assembler source.
LICMDATA	DISPLAY MACHINE	Display machine data. This function can be used to send Software AG the required data for a license file.

This section covers the following topics:

- z/OS JCL Requirements
- z/VSE JCL Requirements
- BS2000 JCL Requirements

z/OS JCL Requirements

The following file assignments are used by the license utility in z/OS environments:

DD Name	Description
SYSIN	Input of LICUTIL control parameters.
SYSPRINT	Output of LICUTIL run.
SYSUT1	Input of original license file (*pppvrs*.LICS data set).
SYSUT2	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	DD Name			
	SYSIN	SYSPRINT	SYSUT1	SYSUT2
CHECK	X	X	X	
DISPLAY ALL	X	X	X	
DISPLAY LICENSE	X	X	X	
DISPLAY MACHINE	X	X		
MAKE	X	X	X	X
HELP	X	X		

z/VSE JCL Requirements

The following file assignments are used by the license utility in z/VSE environments:

File Assignment	Description
SYSIPT	Input of LICUTIL control parameters and the input of original license file (*pppvrs*.LICS data set). **Note:** The license file has to be stored in a sublibrary.
SYSLIST	Output of LICUTIL run.
SYSPCH	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	File Assignment		
	SYSIPT	SYSLIST	SYSPCH
CHECK	X	X	
DISPLAY ALL	X	X	
DISPLAY LICENSE	X	X	
DISPLAY MACHINE	X	X	
MAKE	X	X	X
HELP	X	X	

BS2000 JCL Requirements

The following file assignments are used by the license utility in BS2000 environments:

System File	Description
SYSDTA	Input of LICUTIL control parameters.
SYSOUT	Output of LICUTIL run.
Link Name	
SYSUT1	Input of original license file (*pppvrs*.LICS data set).
SYSUT2	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	File Assignment			
	SYSDTA	SYSOUT	SYSUT1	SYSUT2
CHECK	X	X	X	
DISPLAY ALL	X	X	X	
DISPLAY LICENSE	X	X	X	
DISPLAY MACHINE	X	X		
MAKE	X	X	X	X
HELP	X	X		

LICUTIL Licensing Messages

The messages in this document may be produced when using the LICUTIL utility.

MLC1001	**CPU-ID {cpuid} is not defined in your product license**
Explanation	The CPU ID listed in the message (*cpuid*) is not defined in your product license. Either your license file is invalid or you have attempted to run the Software AG mainframe product on a machine for which it is not licensed.
Action	Install and run the Software AG product on the machine for which it is licensed. If the problem persists, contact your Software AG sales representative for assistance.

MLC1002	**LPAR name {lparname} is not defined in your product license**
Explanation	The LPAR name listed in the message (*lparname*) is not defined in your product license. Either your license file is invalid or you have attempted to run the Software AG mainframe product on a machine for which it is not licensed.
Action	Install and run the Software AG product on the machine for which it is licensed. If the problem persists, contact your Software AG sales representative for assistance.

MLC1003	**Machine capacity {value} higher than MSUs in your product license**
Explanation	The machine or the partition capacity (*value*) of the CEC or LPAR respectively on which you have attempted to install and run the Software AG mainframe product exceeds the capacity allowed in your product license. Either your license file is invalid or you have attempted to run the Software AG mainframe product on a machine for which it is not licensed.
Action	Contact your Software AG sales representative for assistance.

MLC1004	**The product license will expire on {date}**
Explanation	This warning message provides the date (YYYY/MM/DD) when the license will expire for this Software AG mainframe product. The product cannot be started with the current product license after this date. For example, an expiration date of 2018/09/05 indicates that you can still work on 2018/09/05 until 23:59 (local time) but need a new license from the next day (2018/09/06).
Action	No action is required for this warning message. However, you should contact your Software AG sales representative to obtain an updated license soon.

MLC1005	**We assume CPU-ID {cpuid} as your DR or DR testing environment**
Explanation	The CPU ID listed in the message (*cpuid*) is not defined in your product license. Therefore, we assume that you are running in a disaster recovery environment.
Action	If you are running in a disaster recovery environment, no action is required for this warning message. If you are not running in a disaster recovery environment, contact your Software AG sales representative to obtain an updated license.

MLC2001	**The product license is invalid**
Explanation	Your product license is invalid.
Action	Contact your Software AG sales representative for assistance.

MLC2002	**The product license key is invalid**
Explanation	The license file for the Software AG mainframe product is invalid.
Action	Verify that the license file was transferred correctly to the mainframe host and that it is still in ASCII format. If the problem persists, contact your Software AG sales representative for assistance.

MLC2003	**The product license is not for {value}**
Explanation	The license file for the Software AG mainframe product is not for the operating system (OS), product code, or product version listed in the message (*value*).
Action	Use the correct license file. If the problem persists, contact your Software AG sales representative for assistance.

MLC2004	**The product license has expired on {date}**
Explanation	The product license has expired. The date (YYYY/MM/DD) it expired is given in the message (*date*).
Action	Contact your Software AG sales representative for assistance.

MLC2005	**Tag missing in product license: {tagname}**
Explanation	The tag with the name listed in the message (*tagname*) is missing in your product license.
Action	Contact your Software AG sales representative for assistance.

MLC2006	**Invalid tag value in product license: {tagname}**
Explanation	The value of the tag with the name listed in the message (*tagname*) is invalid in your product license.
Action	Contact your Software AG sales representative for assistance.

MLC2007	**Current CPU-ID not found for product license check**
Explanation	The CPU ID of the CPU on which the product license check was run is not found in the license file.
Action	Contact your Software AG sales representative for assistance.

MLC2008	**The number of MSU values in the product license is incorrect**
Explanation	The machine capacity covered by the product license is incorrect.
Action	Contact your Software AG sales representative for assistance.

MLC2009	**ExtendedRights setting {value} not allowed under this system**
Explanation	The ExtendedRights setting listed in the message (*value*) is not supported by your system.
Action	Contact your Software AG sales representative for assistance.

MLC2010	**Soft capping is not allowed with ExtendedRights {value}**
Explanation	Your product license does not allow soft capping with the ExtendedRights setting listed in the message (*value*).
Action	Contact your Software AG sales representative for assistance.

MLC2011	**Defined hardware class in the license exceeded {value}**
Explanation	The hardware class (PriceQuantity) in your product license is too small for the current machine (*value*).
Action	Contact your Software AG sales representative for assistance.

MLC2012	**The number of zIIPs ({value}) exceeds the license definition**
Explanation	The number of zIIP processors in the current machine exceeds the number of zIIPs (PriceQuantity) in your product license.
Action	Contact your Software AG sales representative for assistance.

MLC2013	**The number of zIIP values in the product license is incorrect**
Explanation	The number of zIIP values (zIIPs) in your product license does not match with the number of LPAR values (LPARs).
Action	Contact your Software AG sales representative for assistance.

MLC3001	**Invalid LCPCB length**
Explanation	The license check routine has been incorrectly called.
Action	Contact your Software AG support representative for assistance.

MLC3002	**Work area too small**
Explanation	The size of the work area for the license check routine is too small.
Action	Contact your Software AG support representative for assistance.

MLC3003	**Invalid function code**
Explanation	The license check routine has been incorrectly called.
Action	Contact your Software AG support representative for assistance.

MLC3004	**Invalid operating system**
Explanation	The ID of the operating system passed through the license check routine is invalid.
Action	Contact your Software AG support representative for assistance.

MLC3005	**Invalid product code**
Explanation	The product code passed through the license check routine is invalid.
Action	Contact your Software AG support representative for assistance.

MLC3006	**Invalid product version**
Explanation	The product version passed through the license check routine is not numeric.
Action	Contact your Software AG support representative for assistance.

MLC3007	**Invalid current date**
Explanation	The current date passed through the license check routine is invalid. The date must be in the format *yyyy-mm-dd*. The dash (-) delimiters are not checked but the numbers must be a valid year, month and day, respectively.
Action	Contact your Software AG support representative for assistance.

MLC3008	**Tag not found**
Explanation	The license check routine cannot find the requested XML tag.
Action	Contact your Software AG support representative for assistance.

MLC3009	**Component Id tag not found**
Explanation	The license check routine cannot find the XML tag `Component Id`.
Action	Contact your Software AG support representative for assistance.

MLC3010	**Invalid search tag**
Explanation	The XML search tag requested by the license check routine is invalid.
Action	Contact your Software AG support representative for assistance.

MLC3011	**Insufficient I/O area size**
Explanation	The size of the I/O area for the license check routine is too small.
Action	Contact your Software AG support representative for assistance.

MLC3012	**Output record length too small**
Explanation	The length of the output record for the license check routine is too small.
Action	Contact your Software AG support representative for assistance.

MLC3013	**Error message text not found**
Explanation	The license check routine cannot find the text that belongs to the requested error message.
Action	Contact your Software AG support representative for assistance.

MLC3014	**Too many errors occurred**
Explanation	The license check routine detected too many errors. Some messages are lost.
Action	Contact your Software AG support representative for assistance.

MLC9001	**License exceeds maximum size of 8000**
Explanation	Your license file is too large.
Action	Contact your Software AG sales representative for assistance.

MLC9002   **Error opening license file input**

Explanation   An error occurred during an attempt to open the license file input.

Action   Investigate the cause of the error, paying special attention to the corresponding error messages produced by the operating system. Contact your Software AG sales or support representative for assistance.

MLC9003   **Error reading license file input**

Explanation   An error occurred during an attempt to read the license file input.

Action   Investigate the cause of the error, paying special attention to the corresponding error messages produced by the operating system. Contact your Software AG sales or support representative for assistance.

MLC9004   **Error opening license file assembler output**

Explanation   An error occurred during an attempt to open the license file assembler output.

Action   Investigate the cause of the error, paying special attention to the corresponding error messages produced by the operating system. Contact your Software AG sales or support representative for assistance.

MLC9005   **Error writing license file assembler output**

Explanation   An error occurred during an attempt to write the license file assembler output.

Action   Investigate the cause of the error, paying special attention to the corresponding error messages produced by the operating system. Contact your Software AG sales or support representative for assistance.

MLC9006   **License converted to assembler output records written to output file**

Explanation   The license file was successfully converted to an assembler source module. Output records were written to the output file.

Action   No action is required for this informational message.

MLC9007   **Error loading license module {modulename} {errorcode}**

Explanation   The LOADLIC function (z/OS only) tried to load a license module from the load library and received an error (*errorcode*) from the operating system.

Action   Correct the LICUTIL input, removing duplicate LICUTIL function calls. Check that the license module (*modulename*) is in the load library, that the correct library is referenced in the JCL, and that the module name is specified correctly in the LOADLIC statement (case sensitive, a blank between LOADLIC and *modulename*).

MLC9009	**DMS Error**
Explanation	A DMS error occurred (only in BS2000 environments).
Action	Investigate the cause of the BS2000 DMS error. If you need assistance, contact your Software AG support representative.

MLC9010	**LOADLIC can only be issued once, before any DISPLAY, CHECK, MAKE or READ**
Explanation	The LOADLIC function (z/OS only) can only be performed once per LICUTIL session. LOADLIC must not be preceded by the DISPLAY, CHECK or READ function. These functions can only locate the license module *after* the LOADLIC.
Action	Correct the LICUTIL input, removing duplicate LICUTIL function calls. Change the call order so that the LOADLIC precedes any DISPLAY, CHECK or READ.

IV Port Numbers in EntireX

5 Port Numbers in EntireX

Ensuring Unique Port Numbers

The following four port numbers must be unique for each installation of EntireX.

- **Default EntireX Broker**

    - TCP/IP port

    - SSL port

- **Administration**

    - TCP/IP port

- **Application Monitoring Data Collector**

    - TCP/IP port

Uniqueness checks are performed during installation with the Software AG Installer. If one port in block **Default EntireX Broker** or **Administration** is in use, both ports in this block are recalculated. You can change these values as required.

Any new port number must be a positive integer greater than 1024. If you enter a port number that is already in use, an error message is displayed. All four ports must be assigned correctly before the installation can continue.

Port Number Configuration

In versions before 8.2, the default broker ID was `"localhost"`, which implied `"localhost:1971"` (default port number). From version 8.2, however, the default port number can be changed during installation, and *must* be changed in the case of multiple installations (see **Ensuring Unique Port Numbers**).

The following RPC server configuration/property files in directory *EntireX*/*config* contain a default broker ID:

- `dotNetServer.cfg` (Windows only)

- `cserver.cfg`

- `microfocusserver.cfg`

- `exxatm.cfg`

- `entirex.cicseci.properties`

- `entirex.imsconnect.properties`

- `entirex.javarpcserver.properties`

- `entirex.rpcacibridge.properties`

- `entirex.wmqbridge.properties`

- `entirex.wmqbridgelistener.properties`

- `entirex.xmlrpcserver.properties`

The configuration/property files are updated automatically during installation. `"BrokerID=localhost"` is now `"BrokerID=localhost:1971"`, or, for example, if port 1988 was specified during installation, `"BrokerID=localhost"` becomes `"BrokerID=localhost:1988"`.

If the port numbers are changed *after* installation, you need to update the config/property files in the list above accordingly.

Port Numbers when Installing with Command Central

Since version 10.1 you can also install EntireX using Command Central. When installing EntireX using Command Central, no parameters can be specified. This means:

- The number of the TCP/IP port for the Application Monitoring Data Collector is set to 57900 (same default as standard installation).

- Free TCP and SSL ports are determined automatically by the installation. You can modify these ports after installation as described under *Post-installation Configuration* in section *Installing EntireX using Command Central*.

V Migrating EntireX Data from Earlier Versions

6 Migrating EntireX Data from Earlier Versions

This chapter describes how to migrate an earlier version of EntireX to version 10.5 under UNIX or Windows. EntireX version 10.5 can only be installed in parallel to an existing EntireX version. In this scenario you can migrate user data from one installation to the other easily, using the steps described here. The information applies to both UNIX and Windows systems; any differences are described explicitly.

> **Note:** In this chapter, *source* refers to an earlier EntireX installation, and *target* refers to the current EntireX version 10.5.

Migrating EntireX Broker Data

The broker attribute files are located at *<inst_dir>/EntireX/config/etb*. Follow the steps below to migrate broker data from one installation to the other:

≫ To migrate broker data

1 Stop all running EntireX brokers in both the source and target installations. EntireX brokers from other installations and platforms are not affected.

2 Stop all running RPC Servers that are registered to EntireX brokers in source or target installations. This also includes remote RPC servers or RPC servers from other EntireX installations if these RPC servers are registered with the brokers to be shut down.

3 Stop the Broker Administration Service/Daemon.

■ **Under UNIX:**
Stop the Administration Service, using the script

```
EntireX/bin/sag<n>exx105 stop
```

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

`105` is the EntireX version (major and minor version number)

For example:

```
sag5exx105 stop
```

■ **Under Windows:**
From the Control Panel in both installations, choose **Administration > Services >Stop Software AG EntireX Broker Administration Service**

> **Note:** The first installation does not have a number to identify it. The second and subsequent installations are appended with (2), (3) etc.

4   Copy all relevant directories of defined brokers from source to target installation. These directories - one for each broker - are located under *<inst_dir>/EntireX/config/etb*.

> **Note:** Copy the complete directories, not just the attribute files.

5   Check the broker attribute files for path names and adapt them to the target installation.

6   Restart the Broker Administration Service/Daemon.

- **Under UNIX:**
  Start the Administration Service for both installations, for example with command

  ```
  sag5exx105 start
  ```

- **Under Windows:**
  From the Control Panel, restart the Software AG EntireX Broker Administration Service for both installations.

Following the above steps, autostart settings and user credentials are not migrated. Apply these manually in the new installation. When you restart the Broker Administration service, all brokers will be shown. Readjust the autostart settings and user credentials, using Command Central.

You do not need to move Broker definitions (attribute file etc.) that are stored outside the EntireX installation directory. Simply makes sure when you start a broker stored externally that the broker executable from version 10.5 is used. This may apply to your scripts or when you start a broker manually.

Migrating Authorization Rules

In EntireX version 9.10 and below, the LDAP configuration was defined in the *xds.ini* file. From version 9.12, the configuration is defined with the following new attributes in the Broker attribute file. See *Security-specific Attributes*. The following table gives an overview of the old and the new configuration:

Configuration File *xds.ini*	Broker Attribute File
host=<host>	LDAP-AUTHORIZATION-URL="ldap://host:port".
port=<port>	LDAP-AUTHORIZATION-URL="ldap://host:port".
protocol=<protocol>	LDAP-AUTHORIZATION-URL="ldap://host:port".
authDN=<user>	LDAP-AUTH-DN="cn=admin,dc=software-ag,dc=de".
authPass=<ldap_password>	LDAP-AUTH-PASSWD=encrypted_ldap_password.
baseDN=<DN>	LDAP-BASE-DN="dc=software-ag,dc=de".

Migrating RPC Servers

> **To migrate RPC servers**

1   Copy the file *<inst_dir>/EntireX/config/rpc.server.properties* from source to target installation.

2   In this file, adapt path names that reference your source installation to your target installation.

     Paths of RPC servers other than the source installation can remain unchanged.

Migrating Web Services

Please note the following when migrating Web services to a more recent version of EntireX:

■ Artifacts such as XMM or AAR files can be reused without regeneration.

■ Make sure Web Services Stack and EntireX are on the same level.

■ Modify your scripts (for example the start script of the RPC Server for XML/SOAP) so that they use the new versions of the relevant JAR files of Web Services Stack and EntireX.

Migrating XML/SOAP Components

This section applies to EntireX version 8 and above and describes migration steps for the following components:

> ■ RPC Server for XML/SOAP
> ■ XML/SOAP Wrapper
> ■ Listener for XML/SOAP

RPC Server for XML/SOAP

Update the start script to set the paths to current installation. If the properties file contains a path to the configuration file, you may need to adjust the path. The properties and configuration files can be used as before. See *Configuration File* for more information.

XML/SOAP Wrapper

Update the start script to set the paths to current installation.

Listener for XML/SOAP

The migration steps depend on the Web server you are using:

■ **Using the Software AG Web Server based on Apache Tomcat / Software AG Runtime from Software AG Installation**
Save your Web services (AAR files) and modified configuration files (i.e. *axis2.xml*). You do not need to re-generate your Web services (AAR files). If the configuration was modified, perform the same modifications for the new installation and restart Software AG Web Server. Redeploy the web services (AAR files).

> **Note:** The new installation will use another port by default, which means that the Web clients need to be adjusted.

■ **Using another Web Server**
Save your Web services (AAR files) and modified configuration files (i.e. *axis2.xml*). You do not need to re-generate your Web services (AAR files). For deployment of file *wsstack.war* and copying file *entirex.jar* to Web Services Stack, see the separate Web Services Stack documentation in the *Software AG Infrastructure Administrator's Guide*, also available under *http://documentation.softwareag.com* > *Guides for Tools Shared by Software AG Products*.

If configuration was modified, perform the same modifications for new installation and restart the Web server. Redeploy the Web services (AAR files).

Designer Considerations

We recommend you import existing projects into a new workspace of your target installation. We do not recommend reusing an existing Eclipse workspace. This may damage data or result in strange behavior.

This section describes how to import the projects of Designer from an earlier version into the workspace of the new Designer 10.5, then redeploy the Web services to the new Web Services Stack 10.5. It covers the following topics:

- Overview
- Migrating your Preferences (Optional)
- Migrating Projects to Designer 10.5.0

- Redeploying Web Services

Overview
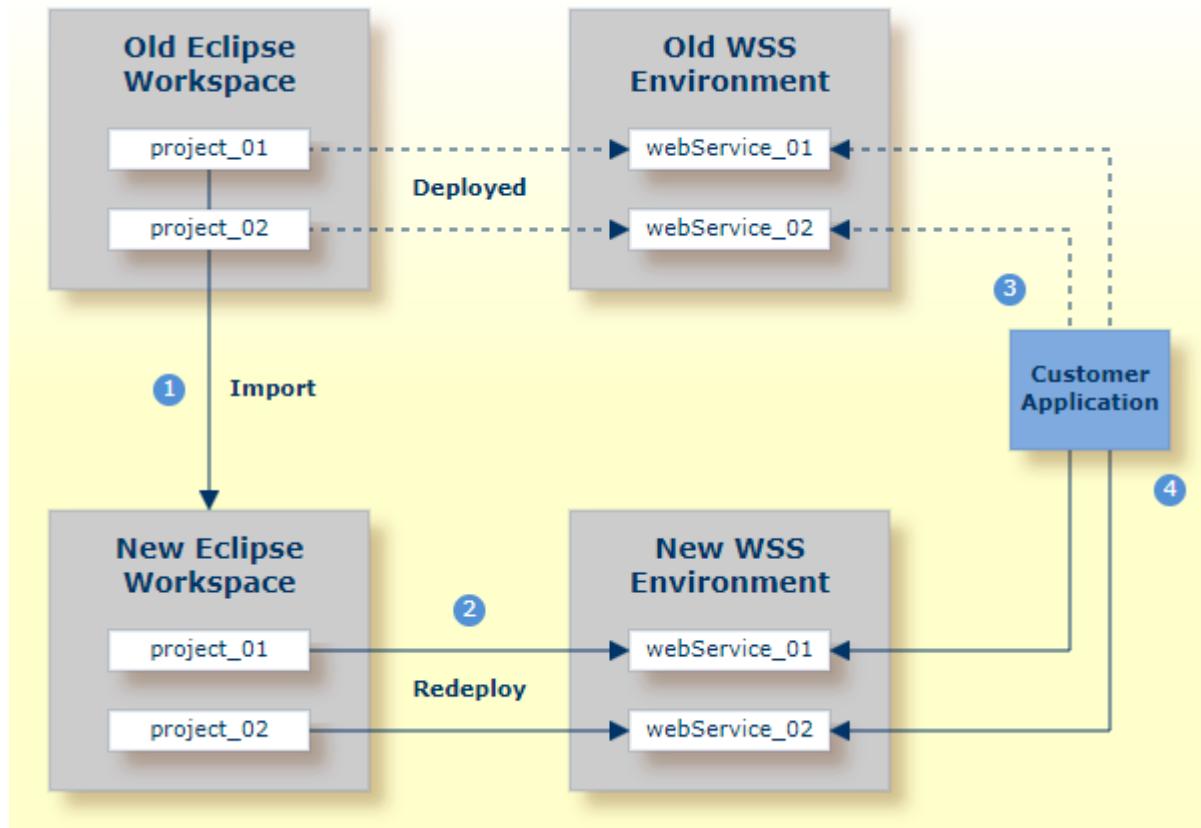
📄 **Notes:**

1. We recommend importing your old Eclipse workspace when you install Designer version 10.5. See also *http://documentation.softwareag.com* > *Upgrading webMethods Products* > *Migrate Designer Projects*.

2. We recommend you redeploy your Web services to the new Web Services Stack environment.

3. Existing applications can continue to use web Services in an old Web Services Stack environment.

4. However, if you wish to access the redeployed web Services in the new 10.5 Web Services Stack environment, your existing applications must be modified accordingly.

Migrating your Preferences (Optional)

Before you migrate your old projects, you can optionally migrate your existing preferences to your new 10.5 installation.

≫ **To migrate the preferences to the new workspace (optional)**

1   Open Designer 10.5 and point to a new 10.5 workspace. Accept the default "workspace105" or specify another one.

2   Export your preferences before upgrading:

1. Start the old version of Designer with the old workspace.

2. Choose **File > Export...**. In the **Select** panel, choose **General > Preferences** and click **Next**.

3. In the **To preference file** field, specify the EPF (Eclipse Process Framework) file to which you want to export your preferences, then click **Finish**.

3   Import these preferences to the new workspace:

1. Start Designer 10.5 with the new workspace.

2. Choose **File > Import...**. In the **Select** panel, choose **General > Preferences** and click **Next**.

3. In the **From preference file** field, specify the EPF file to which you exported your preferences, then click **Finish**.

Migrating Projects to Designer 10.5.0

≫ **To import projects into Designer 10.5**

1   Choose **File > Import...**. In the Select panel, choose **General > Existing Projects into Workspace** and click **Next**.

2   In the **Import Projects** panel, choose **Select Root Directory** and go to the old workspace. The projects appear in the **Projects** box. Select the projects to import into Designer 10.5.

3   Choose **Copy projects into workspace** and click **Finish**.

Redeploying Web Services

≫ **To redeploy the Web services**

1   Select an AAR file and from the context menu choose **Web Services Stack > Deploy Services Package**.

2   Select the correct URL that points to the Web Services Stack 10.5 installation.

VI   Installing Fixes

7 Installing Fixes

Fixes for EntireX and other Software AG products are available from the following sources, depending on your platform:

- **Mainframe**
  You can download fixes for for mainframe platforms from Software AG's Empower Portal. See the Resolution Handling Advice on the fix for details.

- **UNIX/Windows**
  For UNIX and Windows platforms you can download fixes using the Software AG Update Manager (SUM) or Software AG Command Central. See your Software AG Update Manager or Command Central documentation for details.

  > **Note:** Software AG's Empower Portal also provides readme files for UNIX and Windows fixes; the fixes themselves can only be installed using Software AG Update Manager or Command Central.

VII Installing EntireX using Command Central

8 Installing EntireX using Command Central

EntireX can also be installed using Command Central instead of the Software AG Installer.

The installation itself is described in the separate Command Central product documentation, also available under **Cross-Product Guides** on the Software AG documentation website. The key sections are *Creating and Updating Product and Fix Repositories* (if you have not already done so) and *Working with Standalone Product Installations > Install Products*.

This chapter provides EntireX-specific information and describes the post-installation configuration steps that are required.

See also *Template-based Provisioning*.

Introduction

When installing EntireX using Command Central, no parameters can be specified. This means:

- The installation does not demand a license file. Copy a valid EntireX license file after installation as described below.

- The Administration Service/Daemon is started automatically when the installation is finished. Free TCP and SSL ports are determined automatically by the installation. You can modify these ports after installation as described below.

- The number of the TCP/IP port for the Application Monitoring Data Collector is set to 57900 (same default as standard installation) and the autostart option is not set.

- The default broker ETB001 is created, but the autostart option is not set. This means it is not started automatically when the installation is finished. Free TCP and SSL ports are determined automatically by the installation. You can modify these ports after installation using Command Central or you can edit the *ETB001.atr* file directly.

Post-installation Configuration

≫ **To configure EntireX after installation with Command Central**

Depending on the EntireX components installed, perform the following steps as required:

1   Provide a valid license file, using one of the methods below. This step is mandatory.

- Copy a valid EntireX license file to directory *<inst_root>/EntireX/config* and rename to *license.xml* if necessary.

    or

■ Register a license file with the License File Manager of Command Central, using the following steps:

1. Go to **Views > Licensing** and add a license file.

2. Go to **Views > Landscape > Installations** and select the installation you have just installed.

3. Go to **Instances** and select any EntireX instance, e.g. EntireX Broker ETB001.

4. Go to **Configuration > Licenses > Edit** and specify the license you registered previously.

The selected license file will be copied automatically to *<inst_root>/EntireX/config/license.xml*.

2   You can change the port of the Administration Service in the configuration file *etc/brokeradminwrapper.conf*. In this case you need to restart the Administration service.

■ `wrapper.java.additional.101=-Dcom.sun.management.jmxremote.port=57909`

See **Port Numbers in EntireX**.

3   If you modified any port numbers, perform the following steps for these new settings to take effect (skip this step if you did not modify any ports).

■ Restart the Broker Administration Service/Daemon manually:

     ■ **Windows**
     Select the broker instance with **Administration Tools > Services > Software AG EntireX Broker Admin 10.5 <installation_number>**, and from the context menu choose **Stop** then **Start** or **Restart**

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

```
sagexx105 stop
```

and

```
sagexx105 start
```

or

```
sagexx105 restart
```

■ Restart the Software AG Platform Manager manually:

     ■ **Windows**
     Select the Windows service **Administration Tools > Services > Software AG Platform Manager 10.5 <installation_number>**, and from the context menu choose **Stop** and then **Start** or **Restart**.

- **UNIX**

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

   ```
   sagexx105 stop
   ```

   and

   ```
   sagexx105 start
   ```

   or

   ```
   sagexx105 restart
   ```

4   The Default Broker now appears in Command Central and can be started there.

5   If applicable, change the port number of the TCP/IP port for the Application Monitoring Data Collector as described under *Configuration for Application Monitoring* and start the Data Collector as described under *Starting and Stopping the Application Monitoring Data Collector* in the Application Monitoring documentation.

See *Administering EntireX Components with Command Central*.

VIII   Template-based Provisioning

9 **Template-based Provisioning**

EntireX supports template-based provisioning using Command Central since version 10.1. This means you can control and automate the installation, creation and configuration of EntireX component instances (in this case EntireX Broker or RPC servers) using templates. The example in this chapter shows how to create a distributed environment with a broker and an RPC server.

This chapter provides EntireX-specific information. For general information on template-based provisioning, see *Automation and Template-based Provisioning* in the *Software AG Command Central Help*, available under **Cross-Product Guides** on the Software AG Documentation Website.

Location of Templates

Templates are available at GitHub under https://github.com/SoftwareAG/sagdevops-templates/tree/master/templates. Each template is in a separate directory prefixed with "sag-exx".

Each directory contains a readme file with a short description of how to use the template files.

Example

- Introduction
- Template File
- Executing the Templates

Introduction

This example assumes you have Command Central installed on a central host and shows two installations on two remote hosts where the Software AG Platform Manager is already installed. All commands are started from the host where Command Central is installed, and are executed on the target host. This example uses sample files provided in the *example* folder.

> **Note:** You can of course also install, create and configure locally on the same host where Command Central is installed.
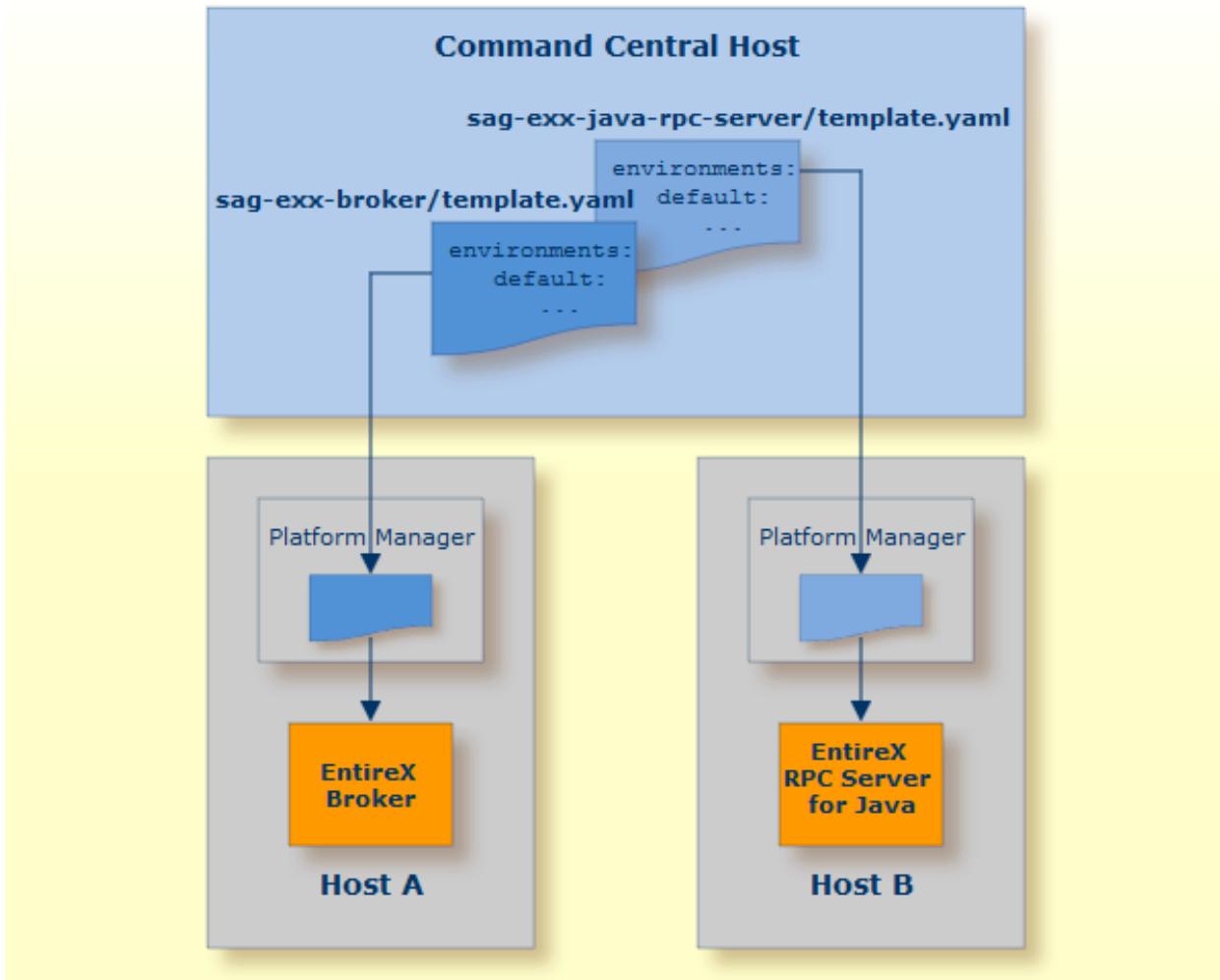
- On Host A, package EntireX Broker is installed and a broker instance created. This is done with file `template.yaml` in directory `sag-exx-broker`.

- On Host B, package EntireX Core is installed and an instance of the RPC Server for Java is created. This is done with the template `template.yaml` in directory `sag-exx-java-rpc-server`.

Template File

Installation and creation of instances is controlled by a template file, which consists of the following sections:

- environments Section
- provision Section
- layers Section
- templates Section

environments Section

In the `environments` section you can define default values for the parameters of an instance (EntireX Broker or RPC server). These parameters are used later when creating an instance.

provision Section

In the `provision` section you define a layer containing the individual actions for the host you want to install (installation, creation of an instance, configuration of an instance). Our example uses the layer `runtime`.

layers Section

The section `layers` contains the various actions to be performed with corresponding reference to the template that is to be used. In our example, it first defines the product repository, the fix repository and the template to be executed, in this case `java.server`. You need to make sure that you have defined a valid product and fix repository in Command Central.

templates Section

The section `templates` contains the actions to be executed. It defines the license to be used from the Command Central license repository, the product component to be installed, the fixes to be applied and the instance to be created. In the following configuration section, all possible configurations for an existing instance can be made.

When the template is executed, it executes different steps:

- First the defined component is installed if not already installed.

- Then the defined fixes are applied if not yet applied.

- An instance is created if it does not already exit.

- Configuration based on the parameters in the configuration section is done.

Executing the Templates

After you have adapted the files with the necessary parameters, execute the files to run the specified actions. For this, the YAML files have to be imported into the Command Central repository. Each YAML file defines an alias name under which the YAML file is registered in Command Central. In our example these names are `sag-exx-broker` and `sag-exx-java-rpc-server`.

> **Note:** All actions are started on the host where Command Central is installed, and are executed on the target host.

≫ **To register the YAML files**

■ Run the following commands:

- ```
  sagcc exec templates composite import -i sag-exx-broker/template.yaml ↵
  [overwrite=true]
  ```

- ```
  sagcc exec templates composite import -i sag-exx-java-rpc-server/template.yaml ↵
  [overwrite=true]
  ```

> **Note:** The overwrite option is only necessary when the file is already registered in Command Central and you want to replace it with a newer version.

Now the two templates can be referenced by their alias names and can be executed. If no additional parameters are specified, the default action in the `provision` section is executed. This can consist of multiple steps: in our example the different steps are installation, creating an instance and configuring an instance as defined in the `layers` section. Some of the parameters in the YAML files in the example should be overwritten with other values defined in the corresponding property file.

≫ **To execute the actions on the two different hosts**

■ Run the following commands:

- **On Host A**

  ```
  sagcc exec templates composite apply sag-exx-broker nodes=A ↵
  repo.product=<your-product-repository> repo.fix=<your-fix-repository>
  ```

- **On Host B**

```
sagcc exec templates composite apply sag-exx-java-rpc-server nodes=B ↵
repo.product=<your-product-repository> repo.fix=<your-fix-repository>
```

This installs the EntireX Broker package on host A with an EntireX Broker instance "ETB009" that has a TCP port 7922, an SSL port 7923 and a trace level 1. On host B, the EntireX Core package is installed together with an RPC Server for Java "JavaRpcSrv" and trace level 1.

Supported EntireX Instance Types

The example above shows how to install different packages of EntireX and to install and configure an EntireX Broker and an RPC Server for Java. But EntireX supports some additional RPC servers.

EntireX RPC servers have common parameters valid for all RPC servers and type-specific parameters; these type-specific differences are reflected in the respective template files. This section lists all components currently supported by EntireX and the corresponding template files:

- EntireX Broker (UNIX/Windows)
- EntireX Mainframe Broker
- EntireX RPC Server for Java
- EntireX RPC Server for C
- EntireX RPC Server for IBM® MQ
- EntireX RPC Server for IMS Connect
- EntireX RPC Server for .NET
- EntireX RPC Server for XML/SOAP
- EntireX RPC Server for CICS Socket Listener

Notes:

1. All actions described below are started on the host where Command Central is installed and executed on the target host. You need to make sure that you have imported a valid license file and defined a product and fix repository.

2. As in the example above, the product component is installed, available fixes are applied, an instance is created and configured if not already done.

3. When you use the Broker template for first time on your UNIX host to install the Broker component, run the script <inst_root>/bin/afterInstallAsRoot.sh with root permission if you want to register the EntireX Broker Administration Service as daemon (started when the system is booted).

EntireX Broker (UNIX/Windows)

Use template file `template.yaml` in directory `sag-exx-broker` to create and/or configure an instance of an EntireX Broker. All available parameters are specified in the template file; you can change them here or overwrite them with a corresponding attribute file. See *Advanced Configuration*.

≫ **To install and configure an EntireX Broker instance**

1   Import the template with the following command:

```
sagcc exec templates composite import -i sag-exx-broker/template.yaml ↵
[overwrite=true]
```

2   Execute the template with its alias name "sag-exx-broker" to create the Broker:

```
sagcc exec templates composite apply sag-exx-broker nodes=local ↵
repo.product=<your-product-repository> repo.fix=<your-product-repository>
```

3   Reconfigure the Broker. After changing the values in the template, you need to import the template again as in step 1, then execute it again as in step 2. If the instance already exists, only the modifications of the configurations are executed.

See *Creating a Broker Instance* and *Configuring a Broker Instance* under *Administering EntireX Broker using the Command Central Command Line* for description of parameters.

EntireX Mainframe Broker

Use template file `template.yaml` in directory `sag-exx-mf-broker` to create and/or configure a proxy instance of an EntireX Mainframe Broker. All available parameters are specified in the template file; you can change them here or overwrite them with a corresponding attribute file. See *Advanced Configuration*.

≫ **To install and configure an EntireX Mainframe Broker proxy instance**

1   Import the template with the following command:

```
sagcc exec templates composite import -i sag-exx-mf-broker/template.yaml ↵
[overwrite=true]
```

2   Execute the template with its alias name "sag-exx-mf-broker" to create a proxy instance of an EntireX Mainframe Broker:

```
sagcc exec templates composite apply sag-exx-mf-broker nodes=local ↵
repo.product=<your-product-repository> repo.fix=<your-product-repository>
```

3   Reconfigure the Broker. After changing the values in the template, you need to import the template again as in step 1, then execute it again as in step 2. If the instance already exists, only the modifications of the configurations are executed.

See *Creating an EntireX Mainframe Broker Connection* and *Configuring the EntireX Mainframe Broker* in the Platform-independent Administration documentation for description of parameters.

EntireX RPC Server for Java

Use template file `template.yaml` in directory `sag-exx-java-rpc-server` to create and/or configure an instance of an RPC Server for Java. All available parameters are specified in the template file; you can change them here or overwrite them with a corresponding attribute file. See *Advanced Configuration*.

≫ **To create and configure an RPC Server for Java**

1   Import the template with the following command:

```
sagcc exec templates composite import –i sag-exx-java-rpc-server/template.yaml ↵
[overwrite=true]
```

2   Execute the template with its alias name "sag-exx-java-rpc-server" to create the RPC server:

```
sagcc exec templates composite apply sag-exx-java-rpc-server nodes=local ↵
repo.product=<your-product-repository> repo.fix=<your-product-repository>
```

3   Reconfigure the RPC Server for Java. After changing the values in the template, you need to import the template again as in step 1, then execute it again as in step 2. If the instance already exists, only the modifications of the configurations are executed.

See *Creating an RPC Server Instance* and *Configuring an RPC Server Instance* in the RPC Server for Java documentation for description of parameters.

EntireX RPC Server for C

Use template file `template.yaml` in directory `sag-exx-c-rpc-server` to create and/or configure an instance of an RPC Server for C. All available parameters are specified in the template file; you can change them here or overwrite them with a corresponding attribute file. See *Advanced Configuration*.

≫ To create and configure an RPC Server for C

1   Import the template with the following command:

```
sagcc exec templates composite import -i sag-exx-c-rpc-server/template.yaml ↵
[overwrite=true]
```

2   Execute the template with its alias name "sag-exx-c-rpc-server" to create the RPC server:

```
sagcc exec templates composite apply sag-exx-c-rpc-server nodes=local ↵
repo.product=<your-product-repository> repo.fix=<your-product-repository>
```

3   Reconfigure the RPC Server for C. After changing the values in the template, you need to import the template again as in step 1, then execute it again as in step 2. If the instance already exists, only the modifications of the configurations are executed.

See *Creating an RPC Server Instance* and *Configuring an RPC Server Instance* in the RPC Server for C documentation for description of parameters.

EntireX RPC Server for IBM® MQ

Use template file `template.yaml` in directory `sag-exx-mq-rpc-server` to create and/or configure an instance of an RPC Server for IBM MQ. All available parameters are specified in the template file; you can change them here or overwrite them with a corresponding attribute file. See *Advanced Configuration*.

≫ To create and configure an RPC Server for IBM MQ

1   Import the template with the following command:

```
sagcc exec templates composite import -i sag-exx-mq-rpc-server/template.yaml ↵
[overwrite=true]
```

2   Execute the template with its alias name "sag-exx-mq-rpc-server" to create the RPC server:

```
sagcc exec templates composite apply sag-exx-mq-rpc-server nodes=local ↵
repo.product=<your-product-repository> repo.fix=<your-product-repository>
```

3 Reconfigure the RPC Server for IBM MQ. After changing the values in the template, you need to import the template again as in step 1, then execute it again as in step 2. If the instance already exists, only the modifications of the configurations are executed.

See *Creating an RPC Server Instance* and *Configuring an RPC Server Instance* in the RPC Server for IBM MQ documentation for description of parameters.

EntireX RPC Server for IMS Connect

Use template file `template.yaml` in directory `sag-exx-ims-rpc-server` to create and/or configure an instance of an RPC Server for IMS Connect. All available parameters are specified in the template file; you can change them here or overwrite them with a corresponding attribute file. See *Advanced Configuration*.

≫ **To create and configure an RPC Server for IMS Connect**

1 Import the template with the following command:

```
sagcc exec templates composite import -i sag-exx-ims-rpc-server/template.yaml ↵
[overwrite=true]
```

2 Execute the template with its alias name "sag-exx-ims-rpc-server" to create the RPC server:

```
sagcc exec templates composite apply sag-exx-ims-rpc-server nodes=local ↵
repo.product=<your-product-repository> repo.fix=<your-product-repository>
```

3 Reconfigure the RPC Server for IMS Connect. After changing the values in the template, you need to import the template again as in step 1, then execute it again as in step 2. If the instance already exists, only the modifications of the configurations are executed.

See *Creating an RPC Server Instance* and *Configuring an RPC Server Instance* in the RPC Server for IMS Connect documentation for description of parameters.

EntireX RPC Server for .NET

Use template file `template.yaml` in directory `sag-exx-net-rpc-server` to create and/or configure an instance of an RPC Server for .NET. All available parameters are specified in the template file; you can change them here or overwrite them with a corresponding attribute file. See *Advanced Configuration*.

≫ **To create and configure an RPC Server for .NET**

1   Import the template with the following command:

```
sagcc exec templates composite import -i sag-exx-net-rpc-server/template.yaml ↵
[overwrite=true]
```

2   Execute the template with its alias name "sag-exx-net-rpc-server" to create the RPC server:

```
sagcc exec templates composite apply sag-exx-net-rpc-server nodes=local ↵
repo.product=<your-product-repository> repo.fix=<your-product-repository>
```

3   Reconfigure the RPC Server for .NET. After changing the values in the template, you need to import the template again as in step 1, then execute it again as in step 2. If the instance already exists, only the modifications of the configurations are executed.

See *Creating an RPC Server Instance* and *Configuring an RPC Server Instance* in the RPC Server for .NET documentation for description of parameters.

EntireX RPC Server for XML/SOAP

Use template file `template.yaml` in directory `sag-exx-xml-rpc-server` to create and/or configure an instance of an RPC Server for XML/SOAP. All available parameters are specified in the template file; you can change them here or overwrite them with a corresponding attribute file. See *Advanced Configuration*.

≫ **To create and configure an RPC Server for XML/SOAP**

1   Import the template with the following command:

```
sagcc exec templates composite import -i sag-exx-xml-rpc-server/template.yaml ↵
[overwrite=true]
```

2   Execute the template with its alias name "XMLRpcServer" to create the RPC server:

```
sagcc exec templates composite apply sag-exx-xml-rpc-server nodes=local ↵
repo.product=<your-product-repository> repo.fix=<your-product-repository>
```

3   Reconfigure the RPC Server for XML/SOAP After changing the values in the template, you
    need to import the template again as in step 1, then execute it again as in step 2. If the instance
    already exists, only the modifications of the configurations are executed.

See *Creating an RPC Server Instance* and *Configuring an RPC Server Instance* in the RPC Server for
XML/SOAP documentation for description of parameters.

EntireX RPC Server for CICS Socket Listener

Use template file `template.yaml` in directory `sag-exx-cics-socket-listener-rpc-server` to
create and/or configure an instance of an RPC Server for CICS Socket Listener. All available
parameters are specified in the template file; you can change them here or overwrite them with a
corresponding attribute file. See *Advanced Configuration*.

≫ **To create and configure an RPC Server for CICS Socket Listener**

1   Import the template with the following command:

```
sagcc exec templates composite import -i ↵
sag-exx-cics-socket-listener-rpc-server/template.yaml [overwrite=true]
```

2   Execute the template with its alias name "sag-exx-cics-socket-listener-rpc-server" to create the
    RPC server:

```
sagcc exec templates composite apply sag-exx-cics-socket-listener-rpc-server ↵
nodes=local repo.product=<your-product-repository> ↵
repo.fix=<your-product-repository>
```

3   Reconfigure the RPC Server for CICS Socket Listener. After changing the values in the template,
    you need to import the template again as in step 1, then execute it again as in step 2. If the
    instance already exists, only the modifications of the configurations are executed.

See *Creating an RPC Server Instance* and *Configuring an RPC Server Instance* in the RPC Server for
CICS Socket Listener documentation for description of parameters.

Advanced Configuration

You can also specify a complete EntireX Broker attribute file or RPC server configuration file when you configurer an instance. First create the instance with a template, then instead of reconfiguring individual properties, specify an attribute file or configuration file.

> **Note:** The attribute and configuration files must be on the host where Command Central is installed.

There are two additional templates files in the directories `sag-exx-broker-attr` and `sag-exx-java-rpc-server-config`.

- The template `sag-exx-broker-attr/template.yaml` shows how to create a broker with a given attribute file in the template.
- The template `sag-exx-java-rpc-server-config/template.yaml` shows how to create an RPC Server for Java with a complete configuration file in the template.

There is also a readme file with a short description of how to use the templates.

IX Building an EntireX Docker Image

10 Building an EntireX Docker Image

Using Scripts provided by EntireX

With scripts provided by EntireX you can build a Docker image for the following components:

- EntireX Broker

- RPC Server for Java

- RPC Server for XML/SOAP

This is described in detail in the relevant section of the EntireX documentation.

> **Note:** Software AG has tested the Docker images with CentOS 7 as base image and is supporting this. You are free to choose any other Linux base image, but Software AG will fix only issues that are reproducible with the officially supported CentOS 7 base image.

Using Command Central

Command Central enables you to build the following product images using the Command Central Docker Builder:

- EntireX Broker

- RPC Server for Java

- RPC Server for XML/SOAP

Instructions are provided centrally for all supported Software AG products under https://github.com/SoftwareAG/. To view the documentation you will first need to create a github account. The specific instruction and necessary files to build the EntireX images can be found in the corresponding subfolders of https://github.com/SoftwareAG/sagdevops-templates/tree/master/containers.

X   Installing EntireX Design-time outside the Designer

11 Installing EntireX Design-time outside the Designer

> **Note:** This section is applicable if you want to install the EntireX design-time outside of the Software AG Designer, that is, not as part of the full webMethods EntireX installation.

Prerequisites

Prerequisites for all EntireX components are described centrally. See *Prerequisites* in the EntireX Release Notes. EntireX supports the Eclipse version provided with the respective webMethods suite.

Scope

The webMethods installation contains a folder *<installation_root>\Designer\updates* with archived update sites of all the features installed. To install the EntireX design-time into Eclipse, use the archived update sites:

- *eclipse.exx.UpdatePackage.zip*

- *com.softwareag.common.zip*

- *eclipse.wss.105.UpdatePackage.nnnn.zip*

The Eclipse features *com.softwareag.common.zip* and *com.softwareag.wsstack.ui_n.n.n.nnn* are pre-requisites for installing the EntireX design-time outside the Designer. All features listed above can be installed in one installation step.

See *Software AG Designer* for more information.

Installation Steps

Installation in GUI Mode

Follow the instruction in the Eclipse online help under **Help > Help Contents > Workbench User Guide > Tasks > Updating and installing software**. Install the three features listed above. You can use drag-and-drop to add the three archived update sites from the folder *<suite installation folder>\Designer\updates* to the list of update sites to visit.

Installing Updates to your own Eclipse

To install updates to your own Eclipse you need a Software AG Designer installation with EntireX design-time of the same version as for your Eclipse.

≫ **To install the updates**

1   Use the Software AG Update Manager to install the latest fixes of EntireX design-time into the Software AG Designer. After installation you will find a folder *<installation_root>\install\fix\backup\Designer\repo*.

2   ▪ If your own Eclipse is on the same machine, you can define this folder as local Update Site for your Eclipse.

   ▪ If your Eclipse is on a different machine, transfer this folder and all its subfolders to the other machine and define it there as local Update Site.

   Under some circumstances, an Eclipse error message indicates that file *repo* could not be found. You can ignore this message.

3   Choose **Help > Check for Update** and select the EntireX update.

4   Install the Update.

Follow these steps again for every new update.

Uninstalling the EntireX Design-time

Uninstalling the Designer plug-ins is done with the Uninstall wizard described under **Help > Help Contents > Workbench User Guide > Tasks > Updating and installing software**.