

## **webMethods EntireX**

### **EntireX z/VSE Batch RPC Server**

Version 9.10

April 2016

This document applies to webMethods EntireX Version 9.10 and all subsequent releases.

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

Copyright © 1997-2016 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

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

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

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

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

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

**Document ID: EXX-BATCHRPC-VSE-910-20181116**

## Table of Contents

1 About this Documentation .....	1
Document Conventions .....	2
Online Information and Support .....	2
Data Protection .....	3
2 Introduction to the Batch RPC Server .....	5
Worker Models .....	6
Inbuilt Services .....	7
Usage of Server Mapping Files .....	8
3 Administering the Batch RPC Server .....	11
Customizing the RPC Server with a Configuration File .....	12
Configuring the RPC Server .....	12
Locating and Calling the Target Server .....	18
Using SSL/TLS with the RPC Server .....	20
Starting the RPC Server .....	21
Stopping the RPC Server .....	22
Activating Tracing for the RPC Server .....	22
4 Deployment Service .....	23
Introduction .....	24
Scope .....	25
Enabling the Deployment Service .....	25
Disabling the Deployment Service .....	26
5 Server-side Mapping Files .....	27
Server-side Mapping Files in the RPC Server .....	28
Deploying Server-side Mapping Files to the RPC Server .....	29
Undeploying Server-side Mapping Files from the RPC Server .....	30
Change Management of Server-side Mapping Files .....	30
List Deployed Server-side Mapping Files .....	30
Check if a Server-side Mapping File Revision has been Deployed .....	31
Access Control: Secure Server Mapping File Deployment .....	31
Ensure that Deployed Server-side Mapping Files are not Overwritten .....	31
Is There a Way to Smoothly Introduce Server-side Mapping Files? .....	32
6 Scenarios .....	33
COBOL Scenarios .....	34



# 1 About this Documentation

---

- Document Conventions ..... 2
- Online Information and Support ..... 2
- Data Protection ..... 3

## Document Conventions

---

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

## Online Information and Support

---

### Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>. The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

### Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to [empower@softwareag.com](mailto:empower@softwareag.com) with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at <https://empower.softwareag.com/>.

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at [https://empower.softwareag.com/public\\_directory.asp](https://empower.softwareag.com/public_directory.asp) and give us a call.

### **Software AG TECHcommunity**

You can find documentation and other technical information on the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

## **Data Protection**

---

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

---

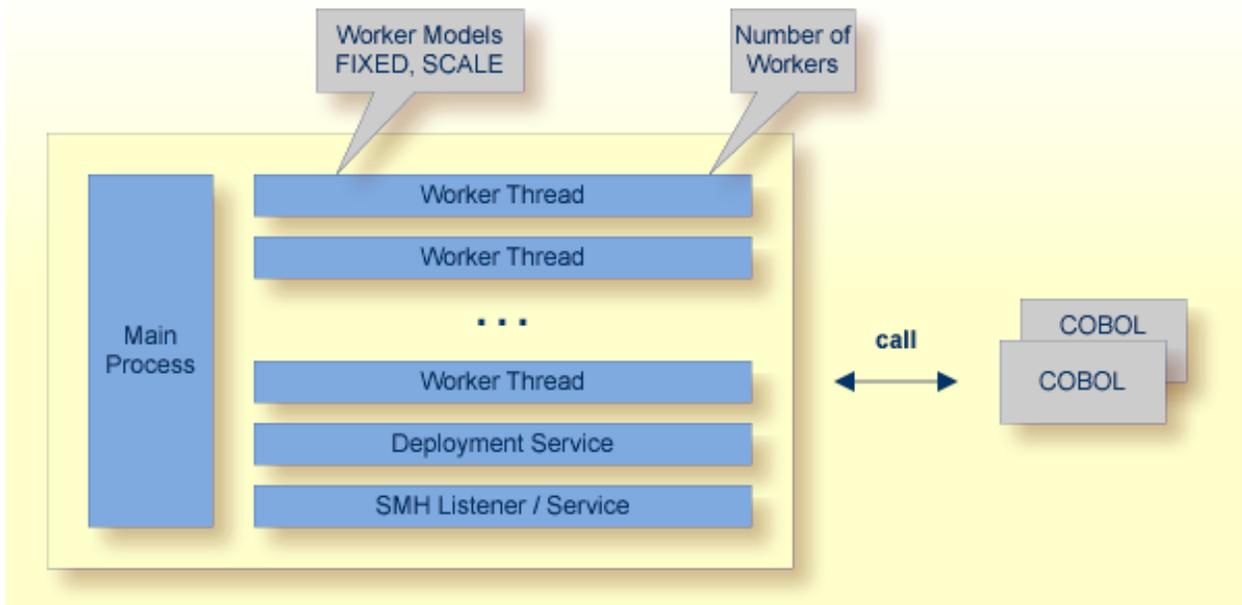
# 2 Introduction to the Batch RPC Server

---

- Worker Models ..... 6
- Inbuilt Services ..... 7
- Usage of Server Mapping Files ..... 8

The EntireX z/VSE Batch RPC Server allows standard RPC clients to communicate with RPC servers on the operating system z/VSE under Batch. It supports the programming language COBOL and works together with the COBOL Wrapper and IDL Extractor for COBOL.

## Worker Models



RPC requests are worked off inside the RPC server in worker threads, which are controlled by a main thread. Every RPC request occupies during its processing a worker thread. If you are using RPC conversations, each RPC conversation requires its own thread during the lifetime of the conversation. The Batch RPC Server provides two worker models:

- **FIXED**  
The *fixed* model creates a fixed number of worker threads. The number of worker threads does not increase or decrease during the lifetime of an RPC server instance.
- **SCALE**  
The *scale* model creates worker threads depending on the incoming load of RPC requests.

A maximum number (thru value of the `workermodel` parameter) of worker threads created can be set to restrict the system load. The minimum number (from value of the `workermodel` parameter), allows you to define a certain number of threads - not used by the currently executing RPC request - to wait for new RPC client requests to process. In this way the RPC server is ready to handle many RPC client requests arriving at the same time.

See parameter `workermodel` under [Configuring the RPC Server](#).

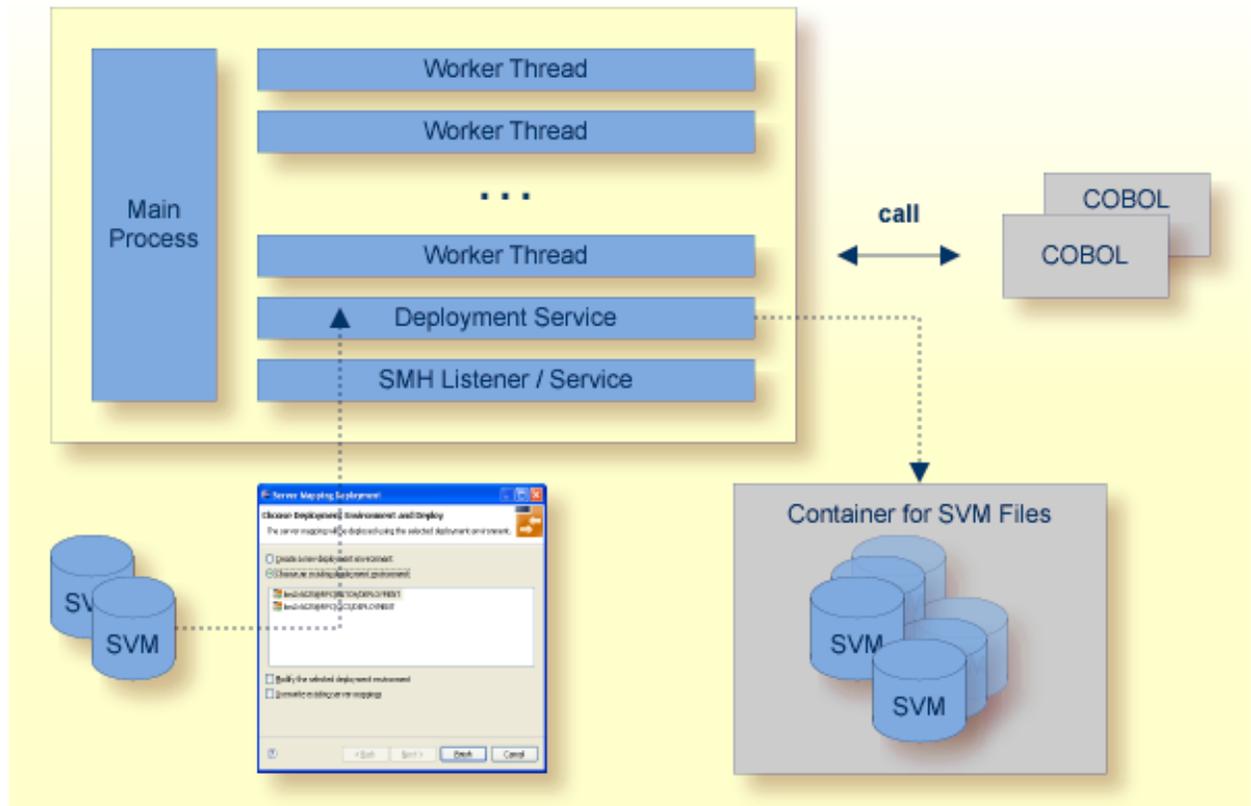
## Inbuilt Services

The Batch RPC Server provides the following services for ease-of-use:

- Deployment Service
- SMH Listener Service

### Deployment Service

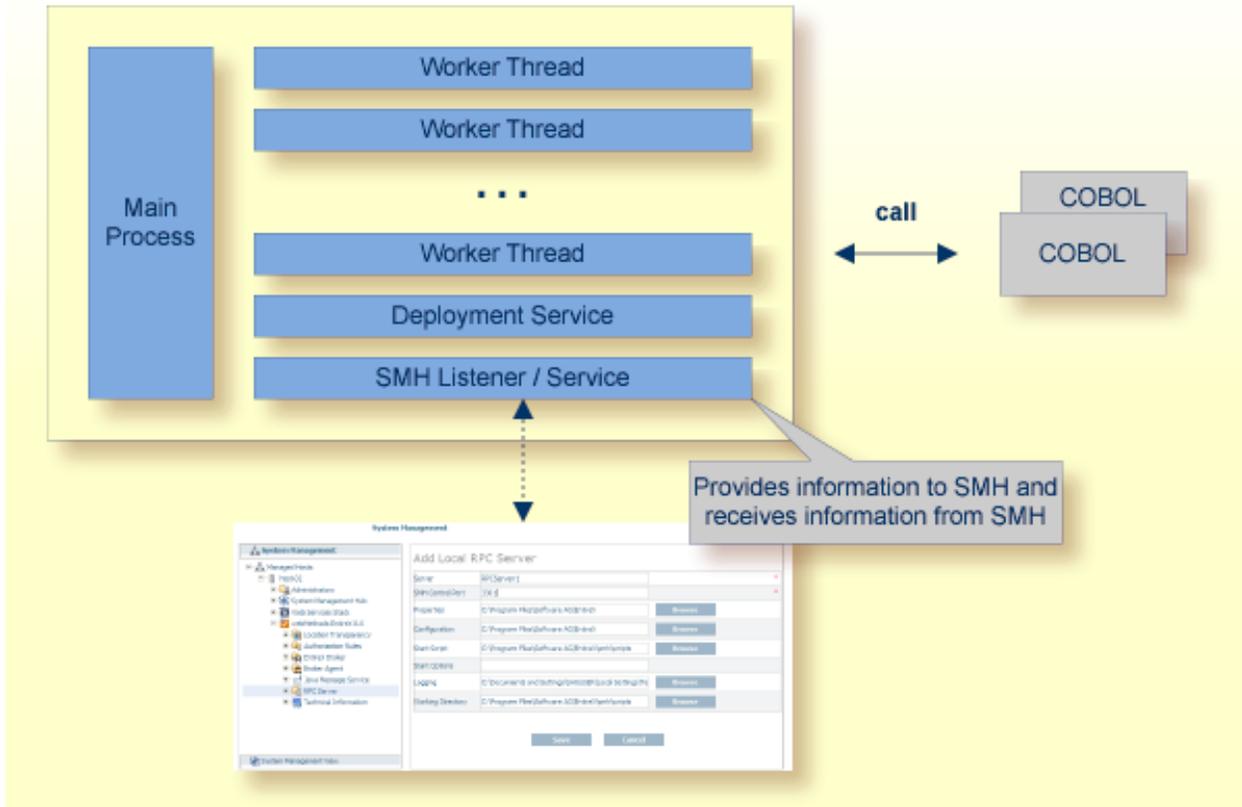
The Deployment Service allows you to deploy server-side mapping files (EntireX Workbench files with extension .svm) interactively using the *Server Mapping Deployment Wizard*. On the RPC server side, the server-side mapping files are stored in a server-side mapping container (VSAM file). See *Server-side Mapping Files in the RPC Server* and *Deployment Service* for configuration information.



## SMH Listener Service

With the SMH Listener Service you use the System Management Hub to monitor the RPC server. See *Administering the EntireX RPC Servers using System Management Hub* in the UNIX and Windows Administration documentation.

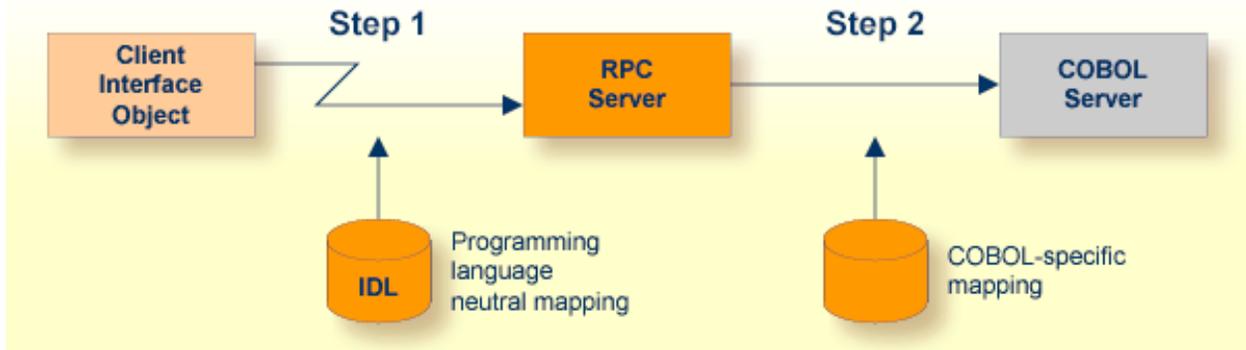
The SMH Service is switched on if the parameter `smhport` is set. See parameter `smhport` under *Configuring the RPC Server*.



## Usage of Server Mapping Files

There are many situations where the Batch RPC Server requires a server mapping file to correctly support special COBOL syntax such as REDEFINES, SIGN LEADING and OCCURS DEPENDING ON clauses, LEVEL-88 fields, etc.

Server mapping files contain COBOL-specific mapping information that is not included in the IDL file, but is needed to successfully call the COBOL server program.



The RPC server marshals the data in a two-step process: the RPC request coming from the RPC client (Step 1) is completed with COBOL-specific mapping information taken from the server mapping file (Step 2). In this way the COBOL server can be called as expected.

The server mapping files are retrieved as a result of the *IDL Extractor for COBOL* extraction process and the *COBOL Wrapper* if a COBOL server is generated. See *When is a Server Mapping File Required?*

There are *server-side* mapping files (*EntireX Workbench* files with extension `.svm`) and *client-side* mapping files (*Workbench* files with extension `.cvm`). See *Server Mapping Files for COBOL* and *How to Set the Type of Server Mapping Files*.

If you are using server-side mapping files, you need to customize the server-side mapping container with parameter `svm`. See *Configuring the RPC Server*.

 **Note:** Server mapping files are used for COBOL only.

---

# 3 Administering the Batch RPC Server

---

- Customizing the RPC Server with a Configuration File ..... 12
- Configuring the RPC Server ..... 12
- Locating and Calling the Target Server ..... 18
- Using SSL/TLS with the RPC Server ..... 20
- Starting the RPC Server ..... 21
- Stopping the RPC Server ..... 22
- Activating Tracing for the RPC Server ..... 22

The EntireX z/VSE Batch RPC Server allows standard RPC clients to communicate with RPC servers on the operating system z/VSE under Batch. It supports the programming language COBOL and works together with the COBOL Wrapper and IDL Extractor for COBOL.

## Customizing the RPC Server with a Configuration File

---

The name of the delivered example configuration file is `RPCPARM.CFG` (see sublibrary EXP960). The configuration file contains the configuration for the Batch RPC Server. The following settings are important:

- connection information such as broker ID, server address (class, name, service)
- location and usage of server-side mapping container, see [Usage of Server Mapping Files](#)
- scalability parameters
- trace settings
- etc.

For more information see [Configuring the RPC Server](#).

## Configuring the RPC Server

---

The following rules apply:

- In the configuration file:
  - Comments must be on a separate line.
  - Comment lines can begin with '\*', '/' and ';'.
  - Empty lines are ignored.
  - Headings in square brackets [`<topic>`] are ignored.
  - Keywords are not case-sensitive.
- Underscored letters in a parameter indicate the minimum number of letters that can be used for an abbreviated command.

For example, in `brokerid=localhost`, `brok` is the minimum number of letters that can be used as an abbreviation, i.e. the commands/parameters `broker=localhost` and `brok=localhost` are equivalents.

Parameter	Default	Values	Req/ Opt
<code>brokerid</code>	localhost	<p>Broker ID used by the server. See <i>Using the Broker ID in Applications</i>.</p> <p>Example:  <code>brokerid=myhost.com:1971</code></p>	R
<code>class</code>	RPC	<p>Server class part of the server address used by the server. The server address must be defined as a service in the broker attribute file (see <i>Service-specific Attributes under Broker Attributes</i>). Case-sensitive, up to 32 characters. Corresponds to CLASS.</p> <p>Example:  <code>class=MyRPC</code></p>	R
<code>codepage</code>		<p>Depending on the internationalization approach, the codepage (locale string) where incoming data is provided to the COBOL server. Conversely, the COBOL server must provide outgoing data in the given codepage, otherwise unpredictable results occur. See <i>What is the Best Internationalization Approach to use?</i> under <i>Internationalization with EntireX</i> for information on which internationalization approach requires a codepage (locale string).</p> <p>By default, no codepage is transferred to the broker. For the most popular internationalization approach, <i>ICU Conversion</i>, the correct codepage (locale string) must be provided. This means it must:</p> <ul style="list-style-type: none"> <li>■ follow the rules described under <i>Locale String Mapping</i></li> <li>■ be a codepage supported by the broker</li> <li>■ be the codepage used in your environment for file and terminal IO, otherwise unpredictable results may occur.</li> </ul> <p>Example:  <code>codepage=ibm-273</code></p>	
<code>compresslevel</code>	N	<p>Enforce compression when data is transferred between broker and server. See <i>Data Compression in EntireX Broker</i>.</p> <p><code>compresslevel= 0   1   2   3   4   5   6   7   8   9   Y   N</code></p> <p>0-9 0=no compression  9=max. compression</p> <p>N No compression.  Y Compression level 6.</p> <p>Example:</p>	O

Parameter	Default	Values	Req/ Opt				
		compresslevel=6					
<u>deployment</u>	NO	<p>Activates the deployment service, see <a href="#">Deployment Service</a>. Required to use the Server Mapping Deployment Wizard. See <i>Server Mapping Deployment Wizard</i> in the EntireX Workbench documentation.</p> <p>YES Activates the deployment service. The RPC server registers the deployment service in the broker.</p> <p>NO The deployment service is deactivated. The RPC server does not register the deployment service in the broker.</p> <p>Example: deployment=yes</p>	O				
<u>etblnk</u>	BKIMB	<p>Define the broker stub to be used. See <i>Administering Broker Stubs under z/VSE</i> for available stubs.</p> <p>Example: ETBL=BKIMB</p>	O				
<u>logon</u>	YES	<p>Execute broker functions LOGON/LOGOFF in worker threads. Must match the setting of the broker attribute AUTOLOGON. Reliable RPC requires logon set to YES. See <i>Reliable RPC</i>.</p> <p>NO No logon/logoff functions are executed.</p> <p>YES Logon/logoff functions are executed.</p> <p>Example: logon=no</p>	O				
<u>marshalling</u>	COBOL	<p>The Batch RPC Server can be configured to support either COBOL or C. See also <a href="#">Locating and Calling the Target Server</a>.</p> <p>marshalling=(LANGUAGE=<u>COBOL</u>   C)</p> <table border="1"> <tr> <td>COBOL</td> <td>Server supports COBOL. The COBOL servers are called directly without a server interface object. So-called server mapping files are used to call the COBOL server correctly if one is available. See <a href="#">Usage of Server Mapping Files</a>.</td> </tr> <tr> <td>C</td> <td>Server supports C. The modules are called using a server interface object built with the C Wrapper.</td> </tr> </table>	COBOL	Server supports COBOL. The COBOL servers are called directly without a server interface object. So-called server mapping files are used to call the COBOL server correctly if one is available. See <a href="#">Usage of Server Mapping Files</a> .	C	Server supports C. The modules are called using a server interface object built with the C Wrapper.	O
COBOL	Server supports COBOL. The COBOL servers are called directly without a server interface object. So-called server mapping files are used to call the COBOL server correctly if one is available. See <a href="#">Usage of Server Mapping Files</a> .						
C	Server supports C. The modules are called using a server interface object built with the C Wrapper.						

Parameter	Default	Values	Req/ Opt
<code>password</code>	no default	<p>Password for broker logon. Case-sensitive, up to 32 characters. For more information see broker ACI control block field <code>PASSWORD</code>.</p> <p>Example:  <code>password=MyPwd</code></p>	O
<code>restartcycles</code>	15	<p>Number of restart attempts if the broker is not available. This can be used to keep the Batch RPC Server running while the broker is down for a short time. A restart cycle will be repeated every 60 seconds.</p> <p>When the number of specified cycles is reached and a connection to the broker is not possible, the Batch RPC Server stops.</p> <p>Example:  <code>restartcycles=30</code></p> <p>The server waits up to 30 minutes before it terminates due to a missing broker connection.</p>	O
<code>runoption</code>	no default	<p>This parameter is for special purposes. It provides the Batch RPC Server with additional information. The runoptions are normally set to meet the platform's requirements. Set this parameter only if a support representative provides you with an option and asks you to do so. The parameter can be defined multiple times.</p> <p>Example:  <code>runoption=&lt;option&gt;</code>  <code>runoption=&lt;option&gt;</code></p>	O
<code>servername</code>	SRV1	<p>Server name part of the server address used by the server. The server address must be defined as a service in the broker attribute file. See <i>Service-specific Attributes</i> under <i>Broker Attributes</i>. Case-sensitive, up to 32 characters. Corresponds to <code>SERVER</code> of the broker attribute file.</p> <p>Example:  <code>servername=mySrv</code></p>	R
<code>service</code>	CALLNAT	<p>Service part of the server address used by the server. The server address must be defined as a service in the broker attribute file. See <i>Service-specific Attributes</i> under <i>Broker Attributes</i>. Case-sensitive, up to 32 characters. Corresponds to <code>SERVICE</code> attribute of the broker attribute file.</p> <p>Example:  <code>service=MYSERVICE</code></p>	R

Parameter	Default	Values	Req/ Opt
<code>smhport</code>	0	<p>The port where the server listens for commands from the System Management Hub (SMH). If this port is 0 (default), no port is used and management by the SMH is disabled.</p> <p>Example:  <code>smhport=3001</code></p>	O
<code>svm</code>	ERXSVM	<p>Usage and location of server-side mapping files; see <a href="#">Server-side Mapping Files in the RPC Server</a>. If no <code>svm</code> parameter is given, the RPC server tries to open the server-side mapping container using DLBL name <code>ERXSVM</code>. If this DLBL name is not available, no server-side mapping files are used. If you use server-side mapping files, the server-side mapping container must be installed and configured; see <i>Step 3: Customize the Batch RPC Server Startup JCL - RUNRPC.J</i> under <i>Installing the z/VSE EntireX RPC Servers</i>. There are also client-side mapping files that do not require configuration here; see <i>Server Mapping Files in the EntireX Workbench</i> in the EntireX Workbench documentation.</p> <p><code>svm = no   dlblname</code></p> <p><code>no</code> No server-side mapping files are used.</p> <p><code>dlblname</code> DLBL name of the server-side mapping container in the startup JCL of the Batch RPC Server.</p> <p>Example:  <code>svm=MYSVM</code></p> <p>For the example above, define the DLBL name <code>MYSVM</code> in the startup JCL of the Batch RPC Server as</p> <pre>// DLBL ↵ MYSVM, 'ENTIREX.SVMDEV.KSDS', 0, VSAM, CAT=VSESPUC</pre> <p>See also <a href="#">Usage of Server Mapping Files</a>.</p>	O
<code>timeout</code>	60	<p>Timeout in seconds, used by the server to wait for broker requests. See broker ACI control block field <code>WAIT</code> for more information. Also influences <code>restartcycles</code>.</p> <p>Example:  <code>timeout=300</code></p>	O

Parameter	Default	Values	Req/ Opt						
<u>tracelevel</u>	None	<p>Trace level for the server. See also <a href="#">Activating Tracing for the RPC Server</a>.</p> <pre>tracelevel = None   Standard   Advanced   ↩</pre> <p>Support</p> <p>None No trace output.            Standard For minimal trace output.            Advanced For detailed trace output.            Support This trace level is for support diagnostics and should only be switched on when requested by Software AG support.</p> <p>Example:            tracelevel=standard</p>	O						
<u>userid</u>	ERX-SRV	<p>Used to identify the server to the broker. See broker ACI control block field USER-ID. Case-sensitive, up to 32 characters.</p> <p>Example:            userid=MyUid</p>	R						
<u>workermodel</u>	SCALE,1,3,slowshrink	<p>The Batch RPC Server can be configured to</p> <ul style="list-style-type: none"> <li>adjust the number of worker threads to the current number of client requests:               <pre>workermodel=(SCALE,from,thru [,slowshrink   fastshrink])</pre> </li> <li>use a fixed number of worker threads:               <pre>workermodel=(FIXED,number)</pre> </li> </ul> <table border="1"> <tr> <td>FIXED</td> <td>A fixed <i>number</i> of worker threads is used by the Batch RPC Server.</td> </tr> <tr> <td>SCALE</td> <td>The number of worker threads is adjusted to the current number of client requests. With the <i>from</i> value, the minimum number of active worker threads can be set. The <i>thru</i> value restricts the maximum number of worker threads.</td> </tr> <tr> <td>slowshrink</td> <td>The RPC server stops all worker threads not used in</td> </tr> </table>	FIXED	A fixed <i>number</i> of worker threads is used by the Batch RPC Server.	SCALE	The number of worker threads is adjusted to the current number of client requests. With the <i>from</i> value, the minimum number of active worker threads can be set. The <i>thru</i> value restricts the maximum number of worker threads.	slowshrink	The RPC server stops all worker threads not used in	O
FIXED	A fixed <i>number</i> of worker threads is used by the Batch RPC Server.								
SCALE	The number of worker threads is adjusted to the current number of client requests. With the <i>from</i> value, the minimum number of active worker threads can be set. The <i>thru</i> value restricts the maximum number of worker threads.								
slowshrink	The RPC server stops all worker threads not used in								

Parameter	Default	Values		Req/ Opt
				the time specified by the <code>timeout</code> parameter, except for the number of workers specified as minimum value. This is the default if SCALE is used.
			<code>fastshrink</code>	The RPC server stops worker threads immediately as soon as it has finished its conversation, except for the number of workers specified as minimum value.
		Example: <code>workermodel=(SCALE,2,5)</code>		

## Locating and Calling the Target Server

The IDL library and IDL program names that come from RPC client are used to locate the RPC server. See `library-definition` and `program-definition` under *Software AG IDL Grammar* in the IDL Editor documentation. This two-level concept (library and program) has to be mapped to the Batch RPC Server environment. Different mechanisms are used depending on the language:

- COBOL
- C

### COBOL

The approach used to derive the z/VSE module name for the RPC server depends on whether server mapping is used or not. See *Usage of Server Mapping Files* for an introduction.

1. If the RPC client sends a client-side type of server mapping with the RPC request, this server mapping is used first.
2. If no server mapping is available from step 1 above, and if server-side type of server mapping is used, the IDL library and IDL program names are used to form a key to locate the server mapping in the server-side mapping container. If a server mapping is found, this is then used.
3. If a server mapping is available from step 1 or 2 above, the z/VSE module name of the RPC server is derived from this mapping. In this case the IDL program name can be different to the z/VSE module name if it is renamed during wrapping process (see *Customize Automatically Generated Server Names*) or during the extraction process in the *COBOL Mapping Editor*.

4. If no server mapping is used at all, the IDL program name is used as the z/VSE module name of the RPC server (the IDL library name is ignored).

➤ **To use the Batch RPC Server with COBOL**

1. Make sure that all z/VSE modules called as RPC servers
  - are compiled with IBM's Language Environment (see [LE/VSE V1R4 Programming Guide](#) for more information)
  - use COBOL calling conventions
  - can be called dynamically ("fetched") from any Language Environment program
  - are accessible through the Batch RPC Server JCL LIBDEF chain.
2. Configure the parameter `marshalling` for COBOL, for example:

```
marshalling=COBOL
```

3. Configure the parameter `svm` depending on whether server-side mapping files are used or not. See [Usage of Server Mapping Files](#).

See also [Scenario I: Calling an Existing COBOL Server](#) or [Scenario II: Writing a New COBOL Server](#).

## C

The approaches needed to derive the names for the Batch RPC Server are more complex for C, for the following reasons:

- the limitation of characters per (physical) member name
- the maximum length of 128 characters per IDL library name. See *Rules for Coding Library, Library Alias, Program, Program Alias and Structure Names* under *Software AG IDL File* in the IDL Editor documentation.

You need to restrict yourself to short IDL library names.

➤ **To use the Batch RPC Server with C**

- Configure the parameter `marshalling` for C, for example

```
marshalling=C
```

See *Using the C Wrapper for the Server Side (z/OS, UNIX, Windows, BS2000/OSD, IBM i)*.

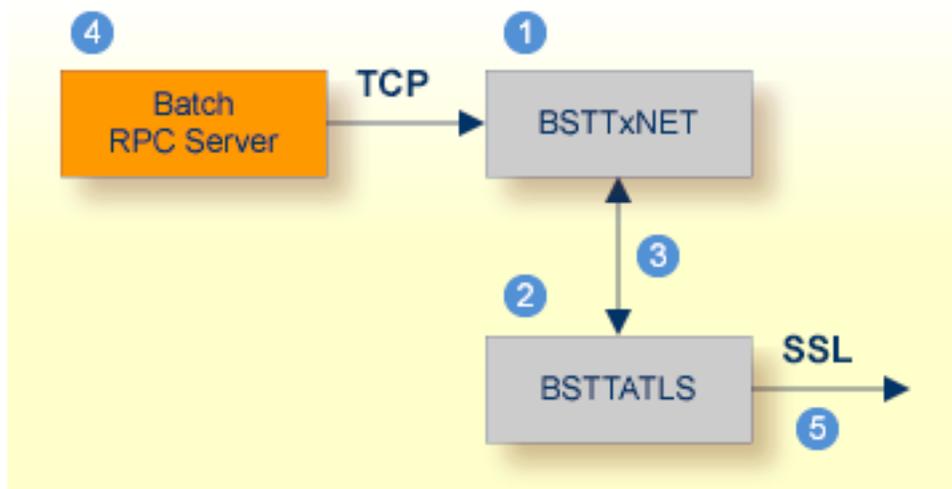
## Using SSL/TLS with the RPC Server

RPC servers can use Secure Sockets Layer/Transport Layer Security (SSL/TLS) as the transport medium. The term “SSL” in this section refers to both SSL and TLS. RPC-based servers are always SSL clients. The SSL server can be either the EntireX Broker, Broker SSL Agent, or Direct RPC in webMethods Integration Server (IS inbound). For an introduction see *SSL/TLS and Certificates with EntireX* in the EntireX Security documentation.

Establishing an SSL connection on z/VSE requires BSI's Automatic Transport Layer Security (ATLS). This facility is similar to z/OS Application Transparent - Transport Layer Security (AT-TLS). ATLS is supported by the BSI stack only.

### Using BSI's Automatic Transport Layer Security (ATLS)

Together with SSL parameters (to provide certificates), define ATLS rules for socket interception in the ATLS daemon startup job BSTTATLS <sup>2</sup>. If the rules match, the socket connection is turned into an SSL connection <sup>5</sup>. Refer to your IBM documentation for further information. For an overview, refer to the IBM Redbook *Enhanced Networking on IBM z/VSE*; for a more detailed description, refer to *BSI SSL Installation, Programming and User's Guide*.



- <sup>1</sup> BSI TCP/IP Stack, either BSTTINET (IPv4) or BSTT6NET (IPv6).
- <sup>2</sup> ATLS rules are defined manually. See Sample ATLS Daemon Configuration below.

- 3 BSTATLS is associated with a TCP/IP stack.
- 4 Application using TCP connection.
- 5 BSTATLS intercepts outbound TCP connection and converts it to SSL connection. For inbound, SSL connections can also be intercepted and converted to TCP connections.

#### ➤ To set up SSL with ATLS

- 1 To operate with SSL, certificates need to be provided and maintained. Depending on the platform, Software AG provides default certificates, but we strongly recommend that you create your own. See *Default Certificates Delivered with EntireX* under *SSL/TLS and Certificates with EntireX* in the EntireX Security documentation.
- 2 Set up the Batch RPC Server for a TCP/IP connection. On mainframe platforms, use *Transport-method-style Broker ID*. Example:

```
ETB024:1699:TCP
```

- 3 Configure ATLS to turn the TCP/IP connection to an SSL connection, see above.
- 4 Make sure the SSL server to which the Batch RPC Server connects is prepared for SSL connections as well. The SSL server can be EntireX Broker, Broker SSL Agent, or Direct RPC in webMethods Integration Server (IS inbound). See:
  - *Running Broker with SSL/TLS Transport* in the platform-specific Administration documentation
  - *Setting up and Administering the EntireX Broker SSL Agent* in the UNIX and Windows Administration documentation
  - *Support for SSL/TLS* in the EntireX Adapter documentation (for Direct RPC)

## Starting the RPC Server

---

#### ➤ To start the Batch RPC Server

- Run the job `RPCRPC.J`.

## Stopping the RPC Server

---

### > To stop the Batch RPC Server

- Use the console command `STOP`. For example:

```
task_id STOP
```

Or:

Use the System Management Hub. This method ensures that the deregistration from the Broker is correct.

## Activating Tracing for the RPC Server

---

### > To activate tracing for the Batch RPC Server

- 1 Set the parameter `tracelevel`.
- 2 Dynamically change the trace level with the operator command

```
port_number TRACELEVEL=tracelevel
```

See the table below for supported trace levels.

The `TRACELEVEL` command without `tracelevel` option will report the currently active trace.

# 4 Deployment Service

---

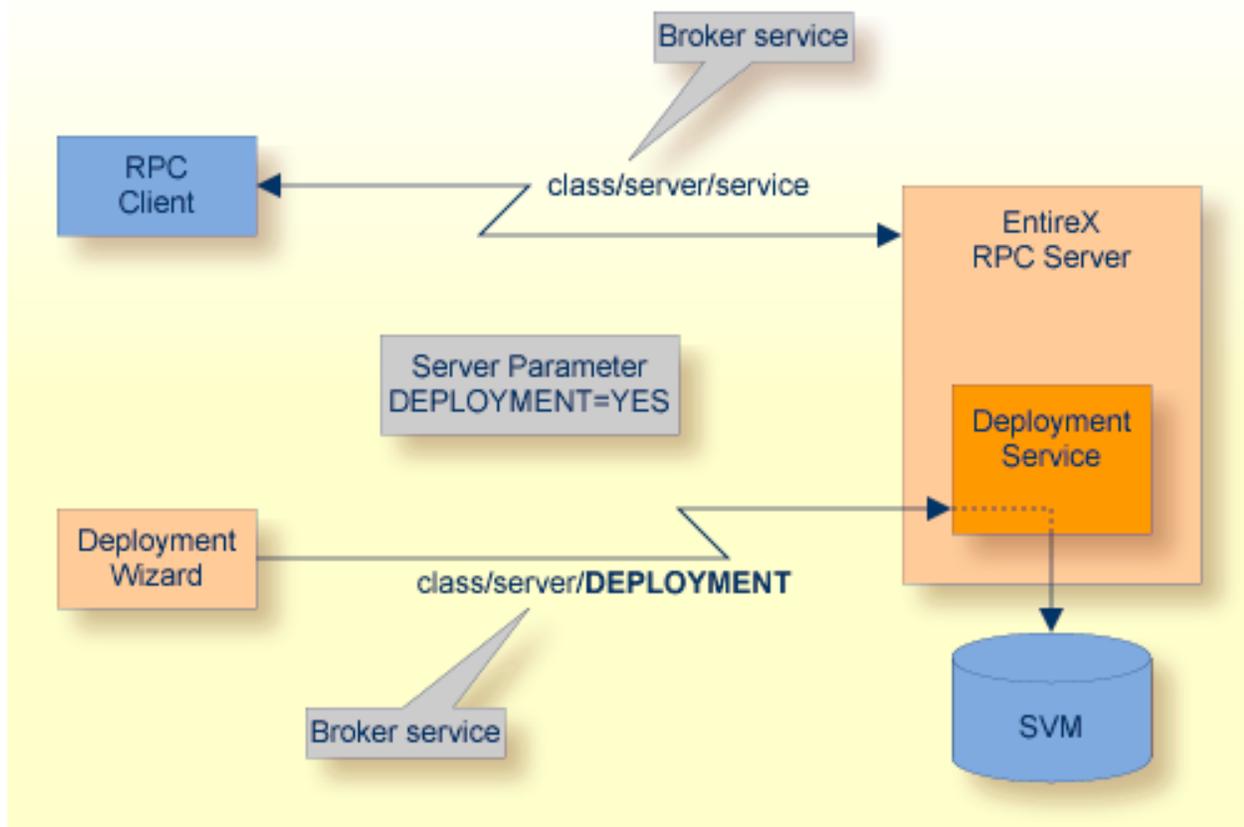
- Introduction ..... 24
- Scope ..... 25
- Enabling the Deployment Service ..... 25
- Disabling the Deployment Service ..... 26

## Introduction

The deployment service is the (server-side) counterpart to the deployment wizard; see *Server Mapping Deployment Wizard*. It is a built-in service of the EntireX RPC server, which can be enabled/disabled by EntireX RPC server configuration settings.

Usage can be restricted to certain users or group of users, using EntireX Security; see *Authorization of Client and Server*.

You need to configure the deployment service only when server-side mapping files are used. There are also client-side server mapping files that do not need configuration here; see *Server Mapping Files for COBOL* in the EntireX Workbench documentation.



## Scope

---

The deployment service is used in conjunction with the

- IDL Extractor for COBOL to deploy server-side mapping files with the deployment wizard;
- COBOL Wrapper for RPC server generation to deploy server-side mapping files with the deployment wizard.

See also [Deploying Server-side Mapping Files to the RPC Server](#).

The deployment service uses the same class and server names as defined for the EntireX RPC server, and `DEPLOYMENT` as the service name, resulting in `class/server/DEPLOYMENT` as the broker service. Please note `DEPLOYMENT` is a service name reserved by Software AG. See broker attribute `SERVICE`.

## Enabling the Deployment Service

---

### » To enable the deployment service

- 1 For a Batch RPC Server, the server-side mapping container (VSAM file) must be installed and configured. See *Step 1: Define a Server-side Mapping Container - VSAMDEF.J (Optional)* under *Installing the z/VSE EntireX RPC Servers*.
- 2 Set the RPC server parameter `deployment=yes`. See `deployment` under [Configuring the RPC Server](#).
- 3 Define in the broker attribute file, under the RPC service, an additional broker service with `DEPLOYMENT` as the service name and values for class and server identical to those used for the RPC service. For example, if your RPC service is named

```
CLASS = RPC      SERVER = SRV1      SERVICE = CALLNAT
```

the deployment service requires the following additional service definition in the broker attribute file:

```
CLASS = RPC      SERVER = SRV1      SERVICE = DEPLOYMENT
```

- 4 Optional. If you need to restrict the use of the deployment service to a selected group of users, use EntireX Security and define security rules for the `class/server/DEPLOYMENT` broker service. The service name `DEPLOYMENT` is a constant.
  - For a z/OS broker, see *Resource Profiles in EntireX Security*.

- For a UNIX or Windows broker, see *Administering Authorization Rules using System Management Hub* in the UNIX and Windows Administration documentation.
- Not applicable to a BS2000/OSD or z/VSE broker.

## Disabling the Deployment Service

---

### ➤ To disable the deployment service

- Set the Batch RPC Server parameter `deployment=no`. See [deployment](#) under *Configuring the RPC Server*.

The Batch RPC Server will not register the deployment service in the broker.

# 5 Server-side Mapping Files

---

- Server-side Mapping Files in the RPC Server ..... 28
- Deploying Server-side Mapping Files to the RPC Server ..... 29
- Undeploying Server-side Mapping Files from the RPC Server ..... 30
- Change Management of Server-side Mapping Files ..... 30
- List Deployed Server-side Mapping Files ..... 30
- Check if a Server-side Mapping File Revision has been Deployed ..... 31
- Access Control: Secure Server Mapping File Deployment ..... 31
- Ensure that Deployed Server-side Mapping Files are not Overwritten ..... 31
- Is There a Way to Smoothly Introduce Server-side Mapping Files? ..... 32

Server mapping enables the RPC server to correctly support special COBOL syntax such as `REDEFINES`, `SIGN LEADING` and `OCCURS DEPENDING ON` clauses, `LEVEL-88` fields, etc. If one of these elements is used, the IDL Extractor for COBOL automatically extracts a server mapping file in addition to the IDL file (interface definition language). Also, the COBOL Wrapper may generate a server mapping file for RPC server generation. The server mapping is used at runtime to marshal and unmarshal the RPC data stream. There are client-side mapping files (EntireX Workbench files with extension `.cvm`) and server-side mapping files (Workbench files with extension `.svm`). If you have not used server-side mapping, we recommend you use client-side mapping. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation.

See also *Source Control of Server Mapping Files* | *Comparing Server Mapping Files* | *When is a Server Mapping File Required?* | *Migrating Server Mapping Files* in the EntireX Workbench documentation.

## Server-side Mapping Files in the RPC Server

---

Under `z/VSE`, server-side mapping corresponds to lines of EntireX Workbench files with extension `.svm`. See *Server Mapping Files for COBOL*. The mapping information is stored as records within one VSAM file, the server-side mapping container. This container contains all server-side mapping entries from all EntireX Workbench files with extension `.svm`. The unique key of the VSAM file consists of the first 255 bytes of the record: for the type (1 byte), for the IDL library (127 bytes) and for the IDL program (127 bytes).

If *one* server requires a server-side mapping file, you need to provide this to the RPC server:

- Development environments: to deploy new server-side mapping files, see [Deploying Server-side Mapping Files to the RPC Server](#).
- Production environments: provide a server-side mapping container (VSAM file) containing all required server-side mapping files to the RPC server. See configuration parameter `svm`.

If *no* server requires server-side mapping, you can execute the RPC server without server mapping files:

- Development environments: you can disable the deployment service. See [Disabling the Deployment Service](#).
- Production environments: there is no need to provide a server-side mapping container (VSAM file) to the RPC server. See configuration parameter `svm`.

## Deploying Server-side Mapping Files to the RPC Server

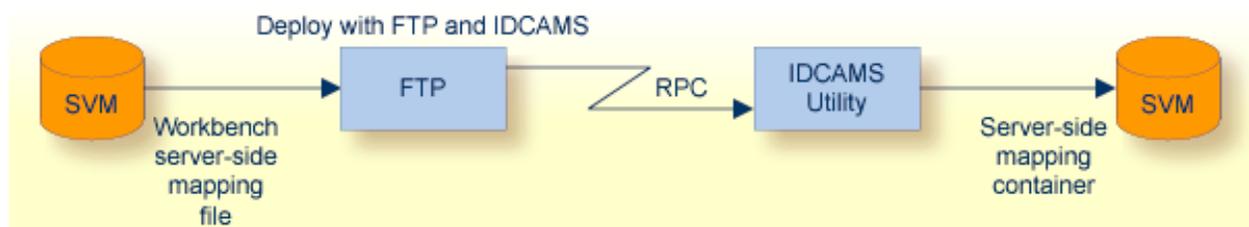
The following approaches are available to deploy a server-side mapping file (EntireX Workbench file with extension `.svm`; see *Server Mapping Files for COBOL*):

- Server Mapping Deployment Wizard
- FTP and IDCAMS

### ➤ To deploy a server-side mapping file with the Server Mapping Deployment Wizard

- 1 Make sure your RPC server is active and that the Deployment Service of the RPC server is properly configured. See [Deployment Service](#).
- 2 From the context menu of your IDL file, choose **COBOL > Deploy/Synchronize Server Mapping** and call the Server Mapping Deployment Wizard. See *Server Mapping Deployment Wizard* in the EntireX Workbench documentation.

### ➤ To deploy a server-side mapping file using FTP and IDCAMS



- 1 Make sure the server-side mapping container (VSAM file) is installed. See *Step 1: Define a Server-side Mapping Container - VSAMDEF.J (Optional)* under *Installing the z/VSE EntireX RPC Servers*.
- 2 Allocate a target sequential file on your mainframe.
- 3 Allow write access to the VSAM file mentioned above and usage of IDCAMS tools.
- 4 Transfer the server-side mapping file to the target host, using FTP. You have to switch to text mode and the codepage of the FTP service must be the same as the codepage (locale string) of the RPC server used.
- 5 Install the server mapping contained in the server-side mapping file into the server-side mapping container (VSAM file) with an appropriate IDCAMS job.

 **Note:** If you omit the keyword `REPLACE` or define `NOREPLACE` in the `SYSIN` data stream of IDCAMS instead, existing server mapping information is not overwritten. This protects server-side mapping records from being overwritten by duplicates.

## Undeploying Server-side Mapping Files from the RPC Server

---

Use the Server Mapping Deployment Wizard to undeploy a server-side mapping file (Workbench file with extension .svm). See *Server Mapping Files for COBOL*.

### ➤ To undeploy a server-side mapping file with the Server Mapping Deployment Wizard

- 1 Make sure your RPC server is active and that the Deployment Service of the RPC server is properly configured. See *Deployment Service*.
- 2 Make sure your IDL file is within an EntireX Workbench directory (folder) without the related server-side mapping file (.svm).
- 3 From the context menu of your IDL file, choose **COBOL > Deploy/Synchronize Server Mapping** and call the Server Mapping Deployment Wizard. See *Server Mapping Deployment Wizard* in the EntireX Workbench documentation. Because there is no related server-side mapping file in the Workbench, all server mapping information related to the IDL file in the RPC server will be removed.

## Change Management of Server-side Mapping Files

---

Under z/VSE, change management for a VSAM file (server-side mapping container, see *Server-side Mapping Files in the RPC Server*) is similar to change management for a database. The complete VSAM file can be backed up at any time, for example by using IDCAMS. All updates to the VSAM file done after a backup must be kept.

All EntireX Workbench server-side mapping files (.svm) added since the last backup should be available. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation.

## List Deployed Server-side Mapping Files

---

Use IDCAMS to list the contents of the server-side mapping container. See *Server-side Mapping Files in the RPC Server*.

```

* $$ JOB JNM=VSAMPRNT,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=K
/* ----- */
/* PRINT CONTENT OF AN SVM VSAM CLUSTER */
/* ----- */
// JOB VSAMPRNT
// DLBL ERXSVM,'ENTIREX.SVMDEV.KSDS',0,VSAM,CAT=VSESPUC
// EXEC IDCAMS,SIZE=AUTO
// PRINT INFILE(ERXSVM) CHAR
/*
/ &
* $$ EOJ

```

## Check if a Server-side Mapping File Revision has been Deployed

---

Server-side mapping records in the server-side mapping container correspond to lines of EntireX Workbench files with extension .svm. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation. The records contain a creation timestamp at offset 276 (decimal) in the format *YYYYMMDDHHIISS*. Precision is 1/10 of a second. The creation timestamp can be checked.

The timestamp can be found on the same offset in the records in the server-side mapping container (VSAM file). See *Server-side Mapping Files in the RPC Server*.

## Access Control: Secure Server Mapping File Deployment

---

For deployment with the *Server Mapping Deployment Wizard*, use EntireX Security if the broker is running on platforms z/OS, UNIX, Windows or z/VSE. See *Enabling the Deployment Service*.

For IBM deployment tool IDCAMS, use RACF to secure deployment.

## Ensure that Deployed Server-side Mapping Files are not Overwritten

---

For IDCAMS, use the `NOREPLACE` option to disallow overwriting of duplicate server-side mapping records in the server-side mapping container (VSAM file); see *Server-side Mapping Files in the RPC Server*. See also *Deploying Server-side Mapping Files to the RPC Server*.

## Is There a Way to Smoothly Introduce Server-side Mapping Files?

---

All EntireX RPC servers can be executed without server-side mapping files. See [Server-side Mapping Files in the RPC Server](#). There is no need to install the server-side mapping container if the following conditions are met:

- You do not use features that require server mapping; see *When is a Server Mapping File Required?*
- Server-side type of COBOL mapping is switched on in the EntireX Workbench. If you have not used server-side mapping, we recommend you use client-side mapping. See *Server Mapping Files for COBOL*.

You can also call COBOL servers generated or extracted with previous versions of EntireX mixed with a COBOL server that requires server-side mapping. All EntireX RPC servers are backward compatible.

# 6 Scenarios

---

- COBOL Scenarios ..... 34

## COBOL Scenarios

---

### Scenario I: Calling an Existing COBOL Server

➤ To call an existing COBOL server

- 1 Use the *IDL Extractor for COBOL* to extract the Software AG IDL and, depending on the complexity, also a server mapping file. See *When is a Server Mapping File Required?* in the EntireX Workbench documentation.
- 2 Build an EntireX RPC client using any EntireX wrapper. For a quick test you can:
  - use the IDL Tester; see *EntireX IDL Tester* in the EntireX Workbench documentation
  - generate an XML mapping file (XMM) and use the XML Tester for verification; see *EntireX XML Tester* in the XML/SOAP Wrapper documentation

See *Client and Server Examples for z/VSE Batch* in the COBOL Wrapper documentation for COBOL RPC Server examples.

### Scenario II: Writing a New COBOL Server

➤ To write a new COBOL server

- 1 Use the *COBOL Wrapper* to generate a COBOL server skeleton and, depending on the complexity, also a server mapping file. See *When is a Server Mapping File Required?* in the EntireX Workbench documentation. Write your COBOL server and proceed as described under *Using the COBOL Wrapper for the Server Side*.
- 2 Build an EntireX RPC client using any EntireX wrapper. For a quick test you can:
  - use the IDL Tester; see *EntireX IDL Tester* in the EntireX Workbench documentation
  - generate an XML mapping file (XMM) and use the XML Tester for verification; see *EntireX XML Tester* in the XML/SOAP Wrapper documentation

See *Client and Server Examples for z/VSE Batch* in the COBOL Wrapper documentation for COBOL RPC Server examples.