

webMethods EntireX

EntireX BS2000/OSD Batch RPC Server

Version 9.7

October 2014

This document applies to webMethods EntireX Version 9.7.

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

Copyright © 1997-2014 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-BS2RPC-97-20160805

Table of Contents

1 Inside the RPC Server	2
Inside the RPC Server	2
Usage of Server Mapping Files	6
2 Administering the BS2000/OSD Batch RPC Server	9
Customizing the RPC Server	10
Configuring the RPC Server	11
Locating and Calling the Target Server	18
Starting the RPC Server	19
Stopping the RPC Server	20
Activating Tracing for the RPC Server	22
3 Server-side Mapping Files	23
Server-side Mapping Files in the RPC Server	24
Deploying Server-side Mapping Files to the RPC Server	25
Undeploying Server-side Mapping Files to the RPC Server	25
Change Management of Server-side Mapping Files	25
List Deployed Server-side Mapping Files	26
Check if a Server-side Mapping File Revision has been Deployed	26
Access Control: Secure Server Mapping File Deployment	26
Is There a Way to Smoothly Introduce Server-side Mapping Files?	26
4 Extractor Service	29
Introduction	30
Scope	31
Enabling the Extractor Service	31
Disabling the Extractor Service	32
5 Deployment Service	33
Introduction	34
Scope	35
Enabling the Deployment Service	35
Disabling the Deployment Service	36
6 Scenarios	37
COBOL Scenarios	38
C Scenarios	39

1 Inside the RPC Server

■ Inside the RPC Server	2
■ Usage of Server Mapping Files	6

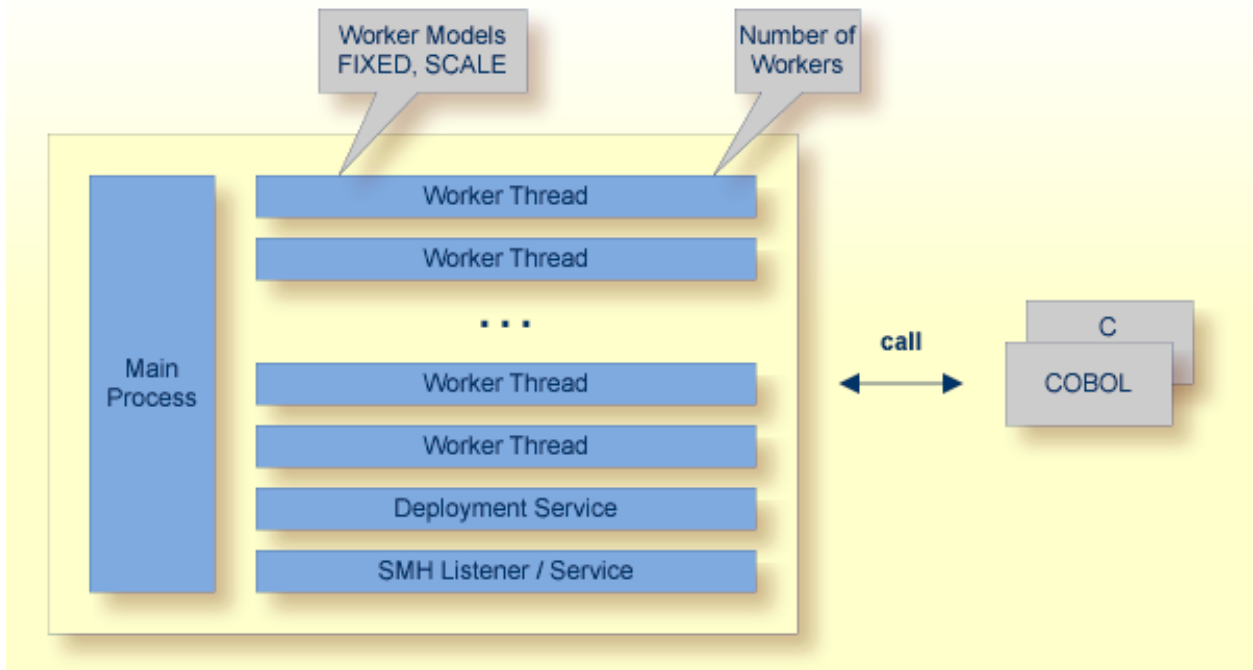
The EntireX BS2000/OSD Batch RPC Server allows standard RPC clients to communicate with RPC servers on the operating system BS2000/OSD. It supports the programming languages COBOL and C.

Inside the RPC Server

- [Worker Models](#)

- Inbuilt Services

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 BS2000/OSD 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.

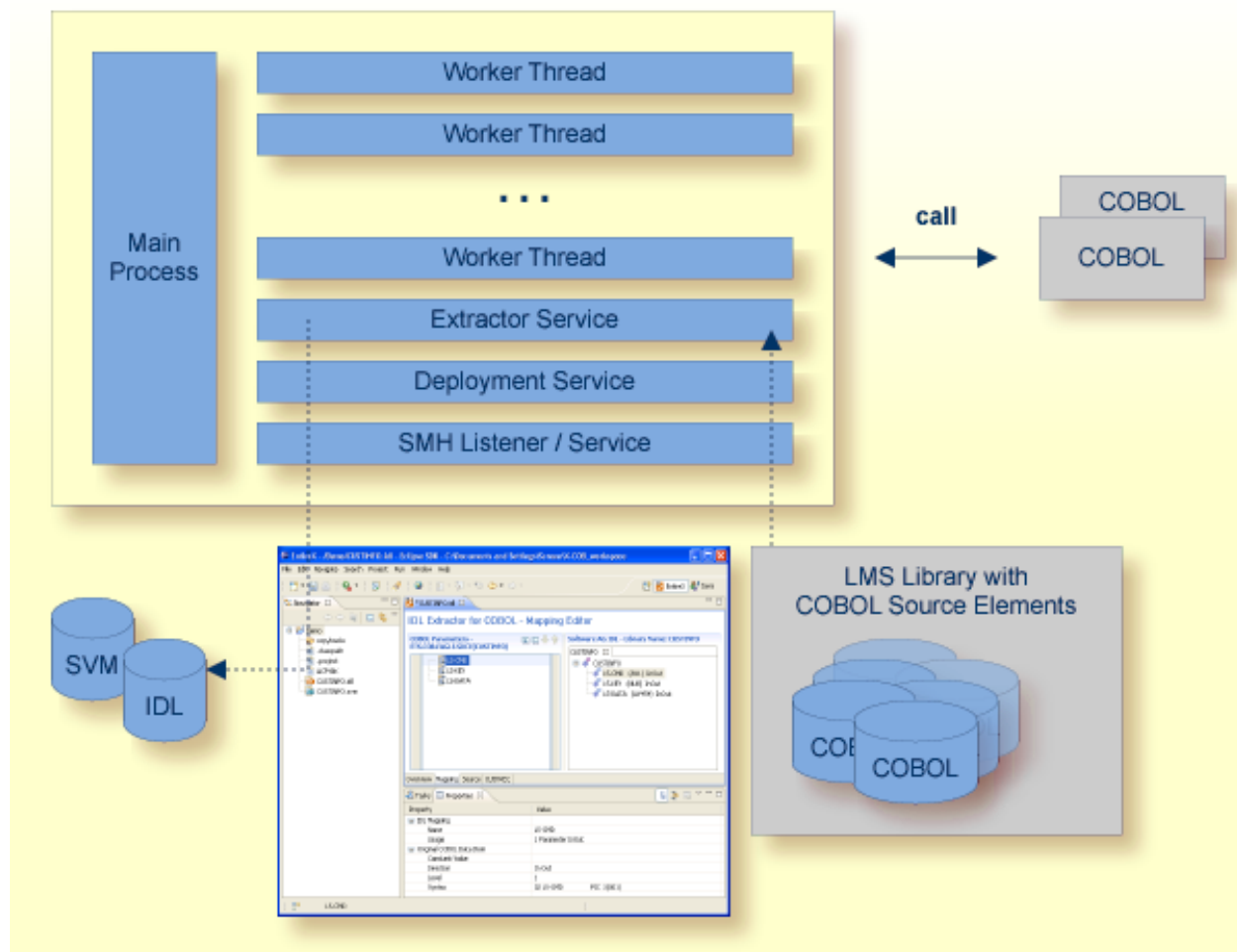
Inbuilt Services

BS2000/OSD Batch RPC Server provides the following services for ease-of-use:

- Extractor Service
- Deployment Service
- SMH Listener Service

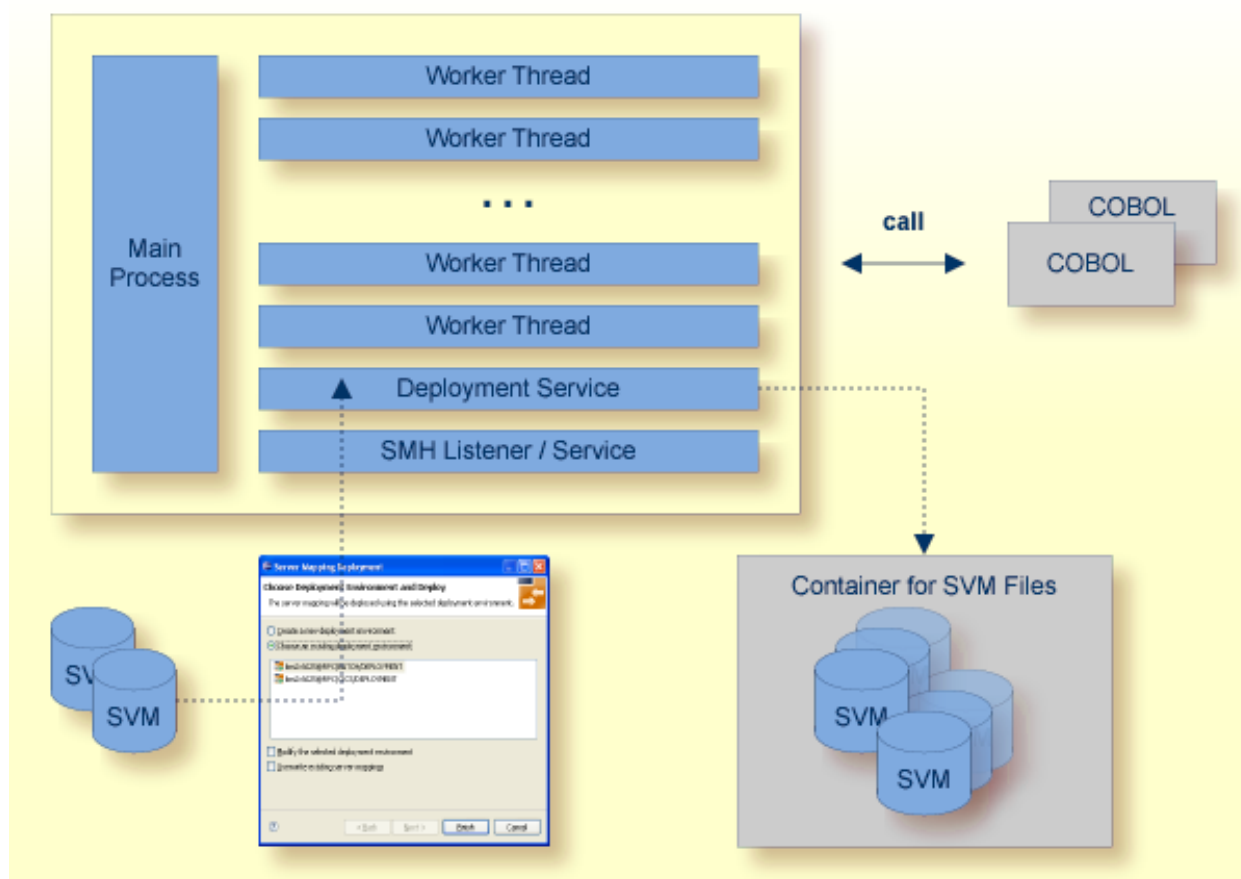
Extractor Service

The Extractor Service is a prerequisite for remote extractions with the *IDL Extractor for COBOL* and *IDL Extractor for PL/I*. See [Extractor Service](#) for more information.



Deployment Service

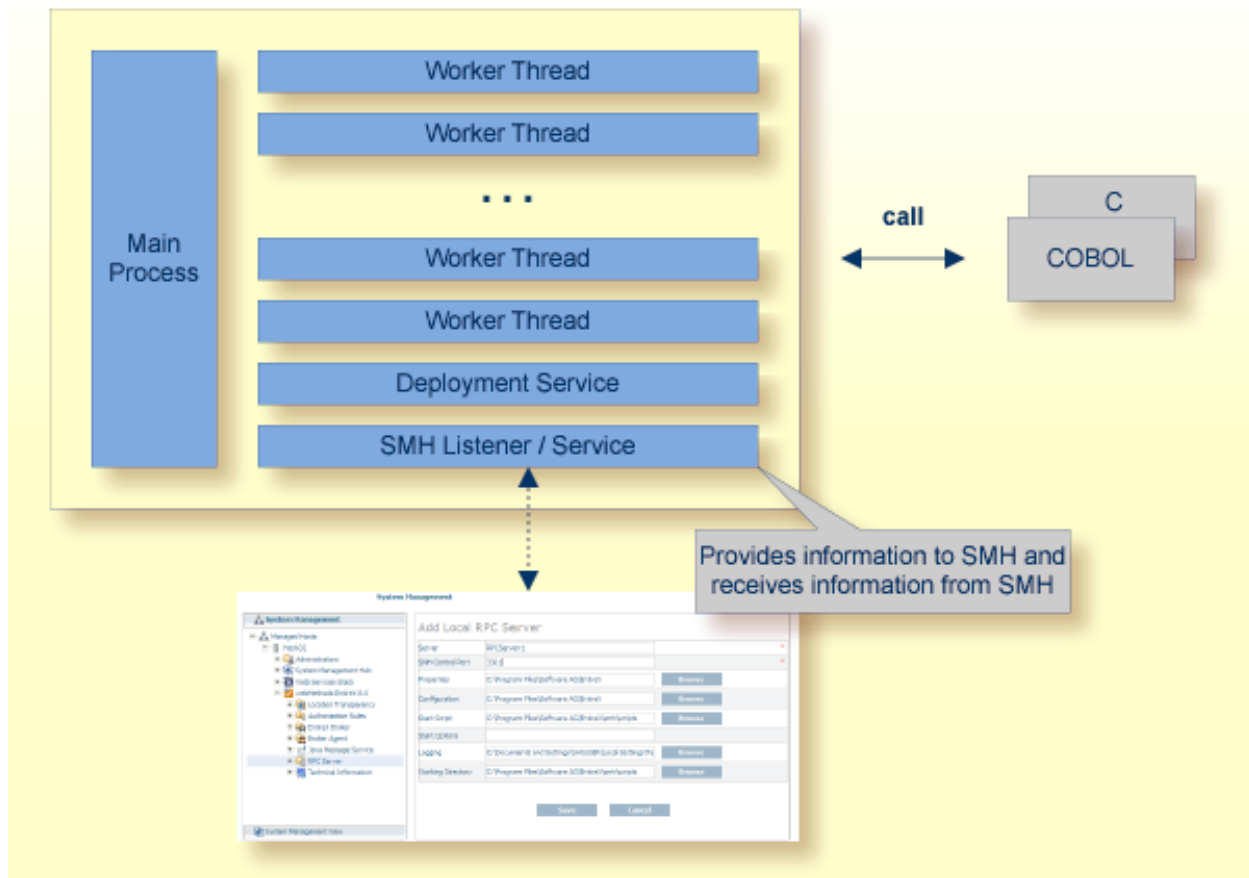
The Deployment Service allows you to deploy server-side mapping files (EntireX Workbench files with extension .svm) interactively using the *Server Mapping Deployment Wizard*. On the RPC server side, the server-side mapping files are stored in a server-side mapping container (ISAM file). See [Server-side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

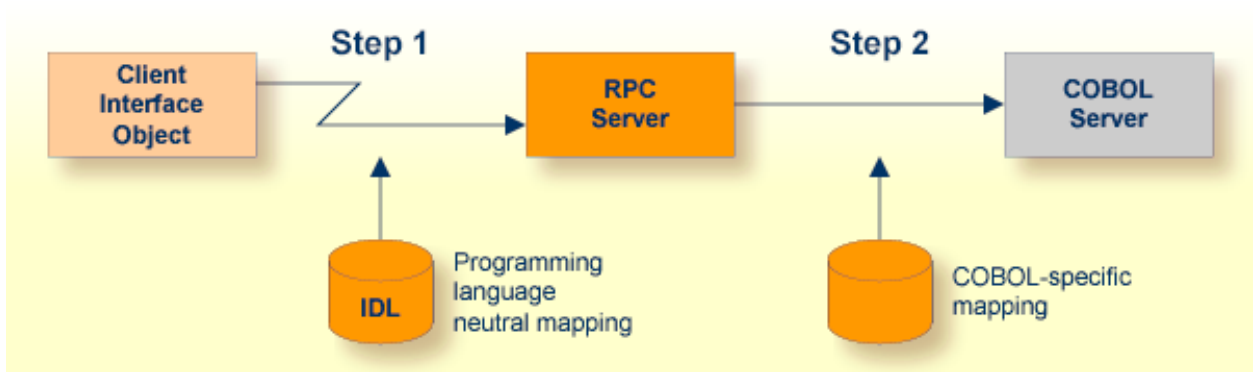
The SMH Service is switched on if the parameter `smhport` is set. See parameter `smhport` under *Configuring the RPC Server*.



Usage of Server Mapping Files

There are many situations where the BS2000/OSD 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](#).

2 Administering the BS2000/OSD Batch RPC Server

■ Customizing the RPC Server	10
■ Configuring the RPC Server	11
■ Locating and Calling the Target Server	18
■ Starting the RPC Server	19
■ Stopping the RPC Server	20
■ Activating Tracing for the RPC Server	22

The EntireX BS2000/OSD Batch RPC Server allows standard RPC clients to communicate with RPC servers on the operating system BS2000/OSD. It supports the programming languages COBOL and C.

Customizing the RPC Server

The following elements are used for setting up the BS2000/OSD Batch RPC Server:

- [Common Runtime Environment \(CRTE\)](#)
- [Configuration File](#)
- [Start Procedure](#)

Common Runtime Environment (CRTE)

When the BS2000/OSD Batch RPC Server calls COBOL or C server programs, the BS2000/OSD Common Runtime Environment (CRTE) is loaded dynamically into the corresponding address space of the worker task.

There is no need to bind the CRTE statically to the called server object modules. If this is needed for any reason, the CRTE must be linked as a subsystem. All entries must be hidden to prevent duplicates. Linking the CRTE statically will occupy resources and slow down the load time of the server object modules.

The CRTE is not delivered with this package. For a detailed description, see the *CRTE (BS2000/OSD) User's Guide*.

Configuration File

The name of the delivered example configuration file is "RPC-CONFIG". The configuration file contains the configuration for the BS2000/OSD 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](#).

Start Procedure

The name of the start S-procedure for the BS2000/OSD Batch RPC Server is "START-RPC-SERVER". The start procedure contains the following:

- the location of the Common Runtime Environment (CRTE)
- the target server library name of the called COBOL or C server
- the configuration file used; see [Configuration File](#)
- etc.

Configuring the RPC Server

The following rules apply:

- 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
<u>brokerid</u>	localhost	Broker ID used by the server. See <i>Using the Broker ID in Applications</i> in the RPC Programming documentation. Example: <code>brokerid=myhost.com:1971</code>	R
<u>class</u>	RPC	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</i> (DEFAULTS=SERVICE) under <i>Broker Attributes</i> in the platform-independent administration documentation). Case-sensitive, up to 32 characters. Corresponds to CLASS. Example: <code>class=MyRPC</code>	R
<u>codepage</u>	no codepage transferred	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	R

Parameter	Default	Values	Req/ Opt
		<p>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> under <i>Introduction to Internationalization</i>, the correct codepage (locale string) must be provided. This means it must:</p> <ul style="list-style-type: none"> ■ follow the rules described under <i>Locale String Mapping</i> in the internationalization documentation ■ be a codepage supported by the broker ■ be the codepage used in your environment for file and terminal IO, otherwise unpredictable results may occur. <p>Example: codepage=EDF041</p>	
<code>compresslevel</code>	N	<p>Enforce compression when data is transferred between broker and server. See <i>Data Compression in EntireX Broker</i> in the general administration documentation.</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: <code>compresslevel=6</code></p>	O
<code>deployment</code>	NO	<p>Activates the deployment service, see Deployment Service. 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>	O

Parameter	Default	Values	Req/ Opt
		<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>	
<u>encryptionlevel</u>	0	<p>Enforce encryption when data is transferred between client and server. Requires EntireX Security. See <code>ENCRYPTION-LEVEL</code> under <i>Broker ACI Fields</i>.</p> <p>0 Encryption is enforced.</p> <p>1 Encryption is enforced between server and broker kernel.</p> <p>2 Encryption is enforced between server and broker kernel, and also between client and broker.</p> <p>Example: encryptionlevel=2</p>	O
<u>init_exit</u>		<p>Initialization exit. The BS2000/OSD Batch RPC Server provides user exits that allow you to plug in code during initialization and to terminate RPC worker tasks. This parameter specifies the name of an executable module that is loaded and executed during initialization of each worker task. See also term_exit.</p> <p>Example: init_exit=myExit</p>	O
<u>extractor</u>	NO	<p>The extractor service is a prerequisite for remote extractions. See Extractor Service.</p> <p>extractor=YES <u>NO</u></p> <p>Example: extractor=yes</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><u>YES</u> Logon/logoff functions are executed.</p> <p>Example:</p>	O

Parameter	Default	Values	Req/ Opt
		logon=no	
<u>marshalling</u>	COBOL	<p>The BS2000/OSD Batch RPC Server can be configured to support either COBOL or C. See also Locating and Calling the Target Server.</p> <p>marshalling=(LANGUAGE=<u>COBOL</u> C)</p> <p>COBOL The BS2000/OSD Batch RPC Server supports COBOL. The COBOL servers are called directly without a server interface object. The COBOL server modules may be compiled as OM or LLM modules. So-called server mapping files are used to call the COBOL server correctly if one is available. See Usage of Server Mapping Files.</p> <p>C The BS2000/OSD Batch RPC Server supports C. The modules are called using a server interface object built with the <i>C Wrapper</i>.</p>	O
<u>password</u>	no default	<p>Password for broker logon. Case-sensitive, up to 32 characters. For more information see broker ACI control block field PASSWORD.</p> <p>Example: password=MyPwd</p>	O
<u>restartcycles</u>	15	<p>Number of restart attempts if the broker is not available. This can be used to keep the BS2000/OSD Batch RPC Server running while the broker is down for a short time. A restart cycle will be repeated at an interval which is calculated as follows:</p> <p>timeout + ETB_TIMEOUT + 60 seconds</p> <p>where timeout is the RPC server parameter (see this table), and ETB_TIMEOUT is the environment variable (see <i>Environment Variables in EntireX</i> in the general administration documentation)</p> <p>When the number of cycles is reached and a connection to the broker is not possible, the RPC server stops.</p> <p>Example: restartcycles=30</p>	O

Parameter	Default	Values	Req/ Opt
<u>servername</u>	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> (DEFAULTS=SERVICE) under <i>Broker Attributes</i> in the platform-independent administration documentation. Case-sensitive, up to 32 characters. Corresponds to SERVER of the broker attribute file.</p> <p>Example: servername=mySrv</p>	R
<u>service</u>	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> (DEFAULTS=SERVICE) under <i>Broker Attributes</i> in the platform-independent administration documentation. Case-sensitive, up to 32 characters. Corresponds to SERVICE attribute of the broker attribute file.</p> <p>Example: service=MYSERVICE</p>	R
<u>smhport</u>	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: smhport=3001</p>	O
<u>svm</u>	PREFERRED	<p>Usage of server mapping files. See Server-side Mapping Files in the RPC Server.</p> <p>SVM=<u>PREFERRED</u> NO</p> <p>PREFERRED This setting is to support COBOL server programs that do not have server-side mapping, plus COBOL server programs built with a server-side mapping file. If you use server-side mapping files, the server-side mapping container must be installed and configured. See <i>Step 1: Define a Server-side Mapping Container</i> in the BS2000/OSD Installation documentation. There are also client-side mapping files that do not require configuration here; see <i>Server Mapping Files for COBOL</i> in the EntireX Workbench documentation.</p>	O

Parameter	Default	Values	Req/ Opt
		<p>NO Server-side mapping files are not used.</p> <p>Example for BS2000/OSD: SVM=NO</p> <p>See also <i>Usage of Server Mapping Files</i>.</p>	
<code>term_exit</code>		<p>Termination exit. The BS2000/OSD Batch RPC Server provides user exits that allow you to plug in code during initialization and terminate RPC worker tasks. This parameter specifies the name of an executable module that is loaded and executed during termination of each worker task. See also <code>init_exit</code>.</p> <p>Example: <code>term_exit=myExit</code></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 WAIT for more information. Also influences <i>restartcycles</i>.</p> <p>Example: <code>timeout=300</code></p>	O
<code>tracedestination</code>	<code>ERXTrace.nnn.log</code>	Trace output is written to SYSOUT.	O
<code>tracelevel</code>	None	<p>Trace level for the server. See also <i>Activating Tracing for the RPC Server</i>.</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <code>tracelevel = None Standard Advanced Support</code> </div> <p>None No trace output.</p> <p>Standard For minimal trace output.</p> <p>Advanced For detailed trace output.</p> <p>Support This trace level is for support diagnostics and should only be switched on when requested by Software AG support.</p> <p>Example: <code>tracelevel=standard</code></p>	O

Parameter	Default	Values	Req/ Opt
<u>userid</u>	ERX-SRV	Used to identify the server to the broker. See broker ACI control block field USER-ID. Case-sensitive, up to 32 characters. Example: userid=MyUid	R
<u>workermodel</u>	SCALE,1,3,slowshrink	<div>The BS2000/OSD Batch RPC Server can be configured to</div> <div><div>■ adjust the number of worker threads to the current number of client requests:</div><div>workermodel=(SCALE,from,thru </div></div>	

Parameter	Default	Values		Req/ Opt
				conversation, except for the number of workers specified as minimum value.
		Example: workermodel=(SCALE,2,5)		

Locating and Calling the Target Server

Target server programs are loaded dynamically, using the BS2000 BLSLIB chain. The target server library name needs to be set up as `PROGRAM-LIB` in the parameter declaration section of the `START-RPC-SERVER` S-procedure, see [Start Procedure](#). Different mechanisms are used depending on the language:

- COBOL
- C

COBOL

The approach used to derive the COBOL object 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 COBOL object module name of the RPC server is derived from this mapping. In this case the IDL program name can be different to the COBOL object 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 COBOL object module name of the RPC server (the IDL library name is ignored).

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

➤ **To use the BS2000/OSD Batch RPC Server with COBOL**

- 1 Make sure that all target server programs called as RPC servers
 - are COBOL object modules
 - use COBOL calling conventions
- 2 Configure the parameter `marshalling` for COBOL, for example:

```
marshalling=COBOL
```

C

➤ **To use the BS2000/OSD Batch RPC Server with C**

- 1 Make sure that all target server programs called as RPC servers
 - are C object modules
 - use C calling conventions
- 2 Configure the parameter `marshalling` for C, for example:

```
marshalling=C
```

See *Scenario III: Writing a New C Server* in the BS2000/OSD administration documentation.

Starting the RPC Server

➤ **To start the BS2000/OSD Batch RPC Server**

- Use the following SDF command:

```
/ENTER-PROCEDURE *LIB(LIB=EXP811.JOBS,ELE=START-RPC-SERVER), -  
/JOB-NAME=RPCMAIN,LOG=*NO
```

Stopping the RPC Server

➤ To stop the BS2000/OSD Batch RPC Server using System Management Hub

- Use the RPC server agent in the SMH to stop the BS2000/OSD Batch RPC Server.

➤ To stop the BS2000/OSD Batch RPC Server from a privileged user ID

- Enter the command:

```
/INFORM-PROGRAM MSG='STOP',JOB-IDENTIFICATION=*TSN(TSN=tsn)
```

where *tsn* is the task number associated with the BS2000/OSD Batch RPC Server main task
(in the example above the TSN of RPCMAIN)

All other tasks that were created as a result of starting the batch RPC server will be stopped automatically.

➤ To stop the BS2000/OSD Batch RPC Server from an operator console

- Enter the command:

```
/INTR tsn,STOP
```

where *tsn* is the task number associated with the BS2000/OSD Batch RPC Server main task
(in the example above the TSN of RPCMAIN)

All other tasks that were created as a result of starting the batch RPC server will be stopped automatically.

➤ To stop the BS2000/OSD Batch RPC Server from a non-privileged user ID

- Use S-procedure STOP-RPC-SERVER in EXP811.JOBS.

Startup Parameter	Description	Default
BROKER-ID	<p>Depending on the communication method, the broker ID can be specified in two different formats:</p> <p>■ TCP Transport Method</p> <pre><i>ip:port:TCP</i></pre> <p>where <i>ip</i> is the address or DNS host name, <i>port</i> is the port number that EntireX Broker is listening on, and TCP is the protocol name</p> <p>■ NET Transport Method</p> <pre><i>ETBnnn:SVCmmm:NET</i></pre> <p>where <i>nnn</i> is the ID under which EntireX Broker is connected to the Adabas ID table, <i>mmm</i> is the SVC number under which the Adabas ID table can be accessed, and NET is the protocol name</p>	none
CLASS	The class name under which the RPC server is registered at the EntireX Broker.	RPC
SERVER	The server name under which the RPC server is registered at the EntireX Broker.	SRV1
SERVICE	The service name under which the RPC server is registered at the EntireX Broker.	CALLNAT
USERID	If EntireX Broker is running with EntireX Security, a user ID needs to be supplied	none
PASSWORD	If EntireX Broker is running with EntireX Security, a password needs to be supplied	none
EXX-JOBS	EntireX Broker jobs library	EXX811.JOBS
EXX-LIB	EntireX Broker module library	EXX811.LIB
WAL-MOD	WAL module library	WAL826.MOD

Set the broker ID in the `PARAMETER-DECLARATION` section and enter following command:

```
/CALL-PROCEDURE (EXP811.JOBS, STOP-RPC-SERVER)
```

Activating Tracing for the RPC Server

➤ To switch on tracing for the RPC server

- Set the parameter `TRACELEVEL` in S-element `RPC-CONFIG` in `EXP811.JOBS`.

To evaluate the return codes, see *Error Messages and Codes*.

3

Server-side Mapping Files

▪ Server-side Mapping Files in the RPC Server	24
▪ Deploying Server-side Mapping Files to the RPC Server	25
▪ Undeploying Server-side Mapping Files to the RPC Server	25
▪ Change Management of Server-side Mapping Files	25
▪ List Deployed Server-side Mapping Files	26
▪ Check if a Server-side Mapping File Revision has been Deployed	26
▪ Access Control: Secure Server Mapping File Deployment	26
▪ Is There a Way to Smoothly Introduce Server-side Mapping Files?	26

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 BS2000/OSD, 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 ISAM 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 ISAM 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 (ISAM 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 (ISAM file) to the RPC server. See configuration parameter `svm`.

Deploying Server-side Mapping Files to the RPC Server

Deploy a server-side mapping file (Workbench file with extension .svm) with the Server Mapping Deployment Wizard. See *Server Mapping Files for COBOL* in the EntireX Workbench documentation.

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

- 1 Make sure the 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** COBOL > Deploy/Synchronize Server Mapping and call the Deployment Wizard. See *Server Mapping Deployment Wizard* in the EntireX Workbench documentation.

Undeploying Server-side Mapping Files to 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 BS2000/OSD, change management for an ISAM file (server-side mapping container, see [Server-side Mapping Files in the RPC Server](#)) is similar to change management for an ordinary file. All updates to the ISAM 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 the command `SHOW-FILE` to list the contents of the server-side mapping container. See [Server-side Mapping Files in the RPC Server](#).

```
SHOW-FILE <server-mapping-file>
```

where `<server-mapping-file>` is the server-side mapping container (ISAM file) containing all server-side mapping information.

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.T`. 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 (ISAM 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](#).

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.

4

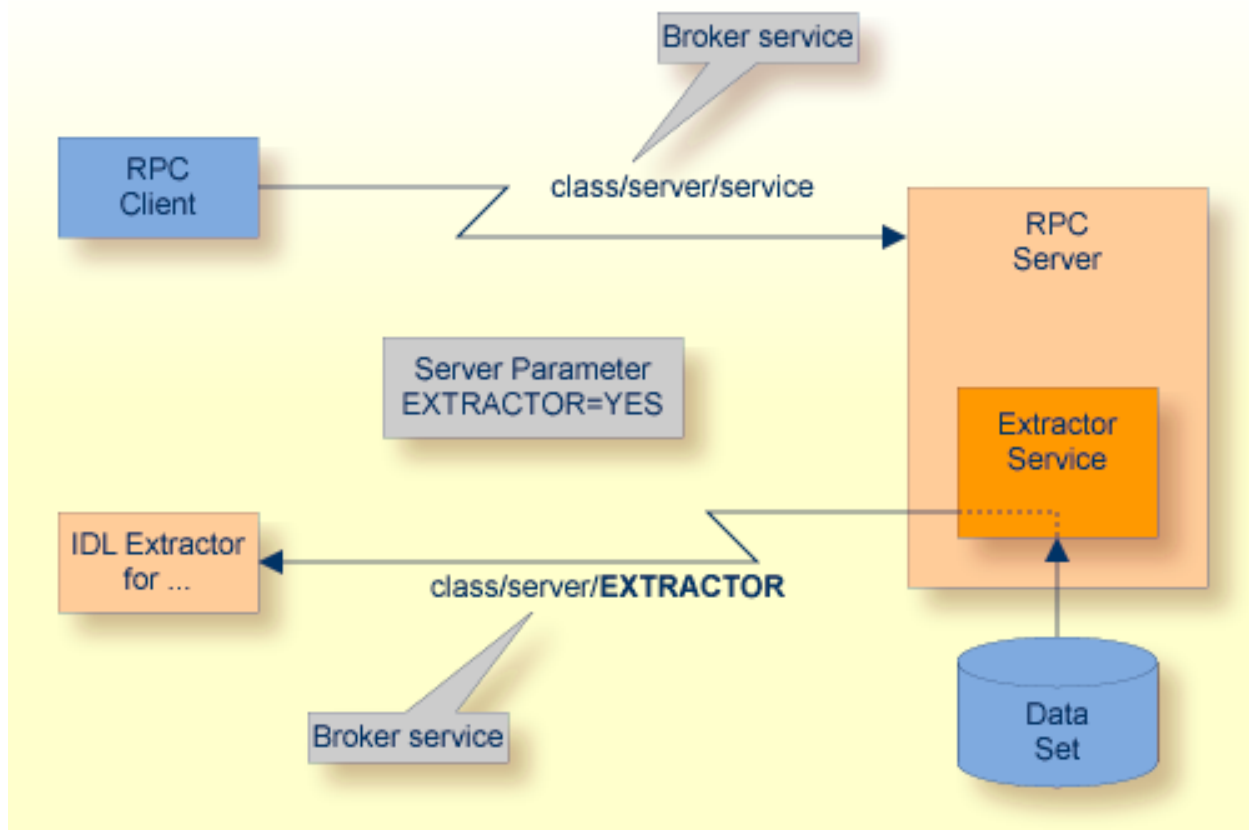
Extractor Service

■ Introduction	30
■ Scope	31
■ Enabling the Extractor Service	31
■ Disabling the Extractor Service	32

Introduction

The extractor service

- provides access to LMS libraries
- is a built-in service of the RPC server, which can be enabled/disabled by RPC server configuration settings
- depending on the platform where the broker is running, usage can be restricted to certain users or group of users, using EntireX Security; see *Authorization of Client and Server* under *Overview of EntireX Security* in the EntireX Security documentation.



Scope

The extractor service is a prerequisite for the

■ IDL Extractor for COBOL

used together with a remote extractor environment, see *Step 2: Select a COBOL Extractor Environment or Create a New One* in the IDL Extractor for COBOL documentation.

The extractor service uses the same class and server names as defined for the RPC server, and "EXTRACTOR" as the service name, resulting in `class/server/EXTRACTOR` as the broker service. Please note "EXTRACTOR" is a service name reserved by Software AG. See `SERVICE` under *Broker Attributes*.

Enabling the Extractor Service

➤ To enable the extractor service

- 1 Set the Batch RPC Server parameter `extractor=yes`. See `extractor` under *Configuring the RPC Server*.
- 2 Define in the broker attribute file, under the RPC service, an additional broker service with "EXTRACTOR" 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 extractor service requires the following additional service definition in the Broker attribute file:

```
CLASS = RPC    SERVER = SRV1    SERVICE = EXTRACTOR
```

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

Disabling the Extractor Service

➤ To disable the extractor service

- Set the BS2000/OSD Batch RPC Server parameter `extractor=no`. See `extractor` under *Configuring the RPC Server*. The BS2000/OSD Batch RPC Server will not register the extractor service in the broker.

5

Deployment Service

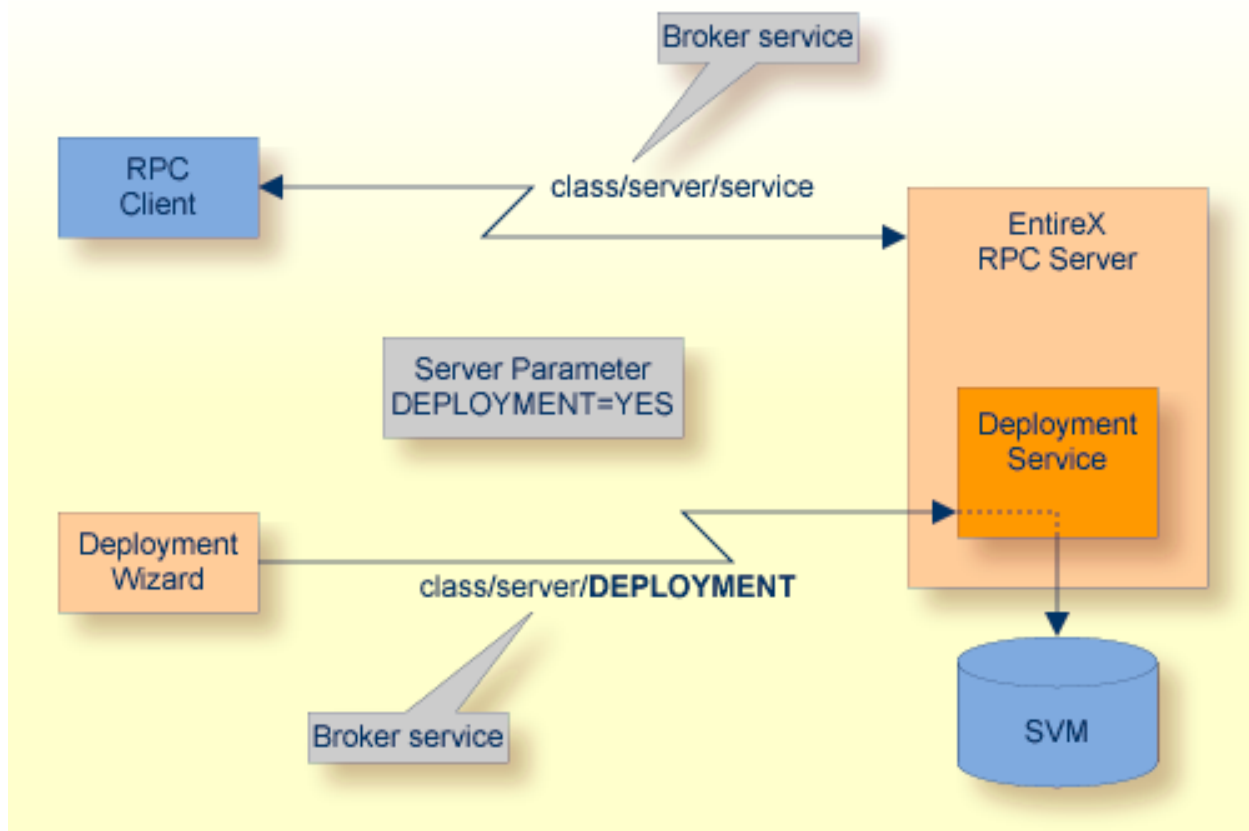
■ Introduction	34
■ Scope	35
■ Enabling the Deployment Service	35
■ Disabling the Deployment Service	36

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* under *Overview of EntireX Security* in the EntireX Security documentation.

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 BS2000/OSD Batch RPC Server, the server-side mapping container (ISAM file) must be installed and configured. See *Step 1: Define a Server-side Mapping Container* in the BS2000/OSD Installation documentation.
- 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* in the EntireX Security documentation.

- 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 BS2000/OSD Batch RPC Server parameter `deployment=no`. See [deployment](#) under *Configuring the RPC Server*.

The BS2000/OSD Batch RPC Server will not register the deployment service in the broker.

6 Scenarios

■ COBOL Scenarios	38
■ C Scenarios	39

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 BS2000/OSD* 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 BS2000/OSD* in the COBOL Wrapper documentation for COBOL RPC Server examples.

C Scenarios

Scenario III: Writing a New C Server

➤ To write a new C server

- 1 Use the *C Wrapper* to generate a C server skeleton and a C server interface object. Write your C server and proceed as described under *Using the C Wrapper for the Server Side (z/OS, UNIX, Windows, BS2000/OSD, IBM i)*.
- 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

