

webMethods EntireX

EntireX BS2000/OSD Batch RPC Server

Version 9.6

April 2014

This document applies to webMethods EntireX Version 9.6.

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

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://documentation.softwareag.com/legal/>.

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://documentation.softwareag.com/legal/> 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 and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

Document ID: EXX-BS2RPC-96-20140628

Table of Contents

1 Introduction to the BS2000/OSD Batch RPC Server	2
Introduction to the BS2000/OSD Batch RPC Server	2
Usage of SVM 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	18
Stopping the RPC Server	19
Activating Tracing for the RPC Server	21
3 Handling SVM Files under BS2000/OSD	23
SVM Files in the EntireX Workbench	24
SVM Files in the RPC Server	24
Source Control of SVM Files	25
Change Management of SVM Files	25
Compare SVM Files	25
List Deployed SVM Files	25
Check if an SVM File Revision has been Deployed	26
Access Control: Secure SVM File Deployment	26
When is an SVM File Required?	26
Is There a Way to Smoothly Introduce SVM Files?	28
4 Extractor Service under BS2000/OSD	29
Introduction	30
Scope	31
Enabling the Extractor Service	31
Disabling the Extractor Service	32
5 Deployment Service under BS2000/OSD	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 Introduction to the BS2000/OSD Batch RPC Server

- Introduction to the BS2000/OSD Batch RPC Server 2
- Usage of SVM Files 6

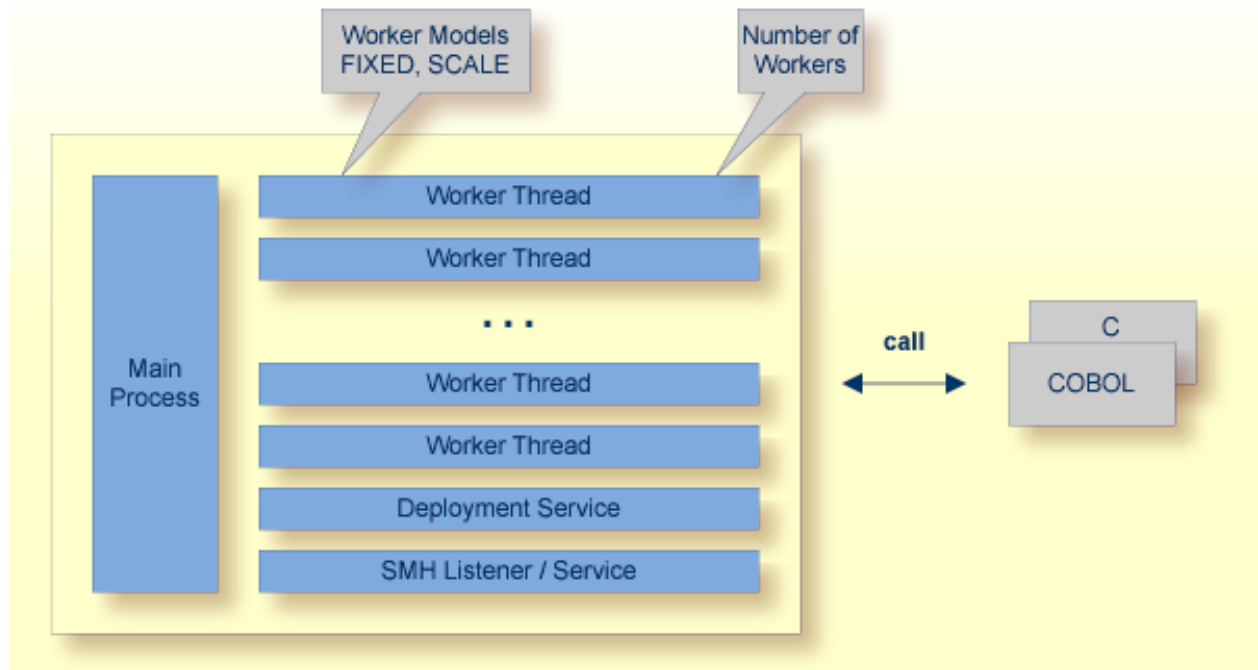
The EntireX BS2000/OSD Batch RPC Server allows standard RPC clients to communicate with RPC servers on the operating system BS2000/OSD. It supports the programming languages COBOL and C.

Introduction to the BS2000/OSD Batch 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.

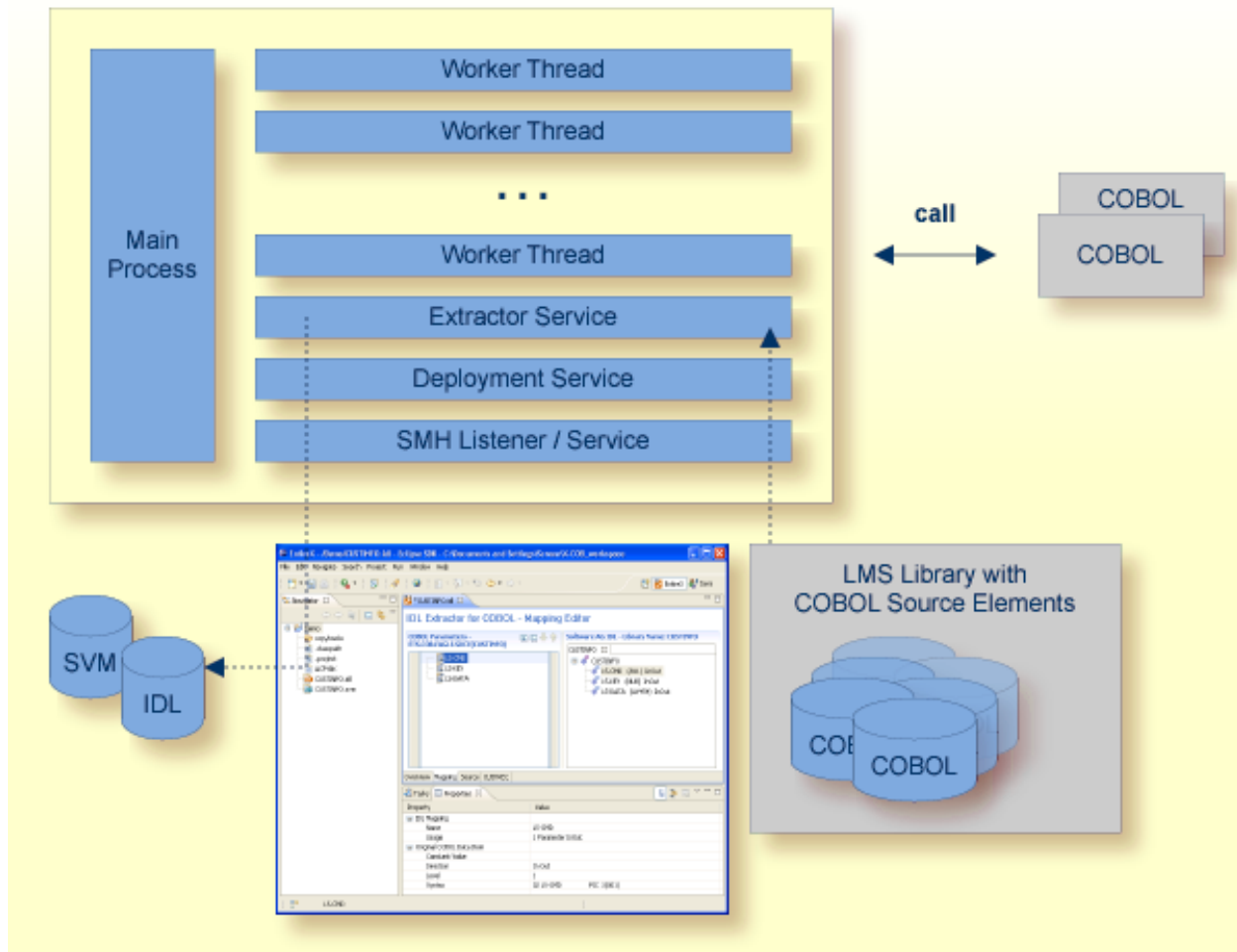
Inbuilt Services

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

- Extractor Service
- Deployment Service
- SMH Listener Service

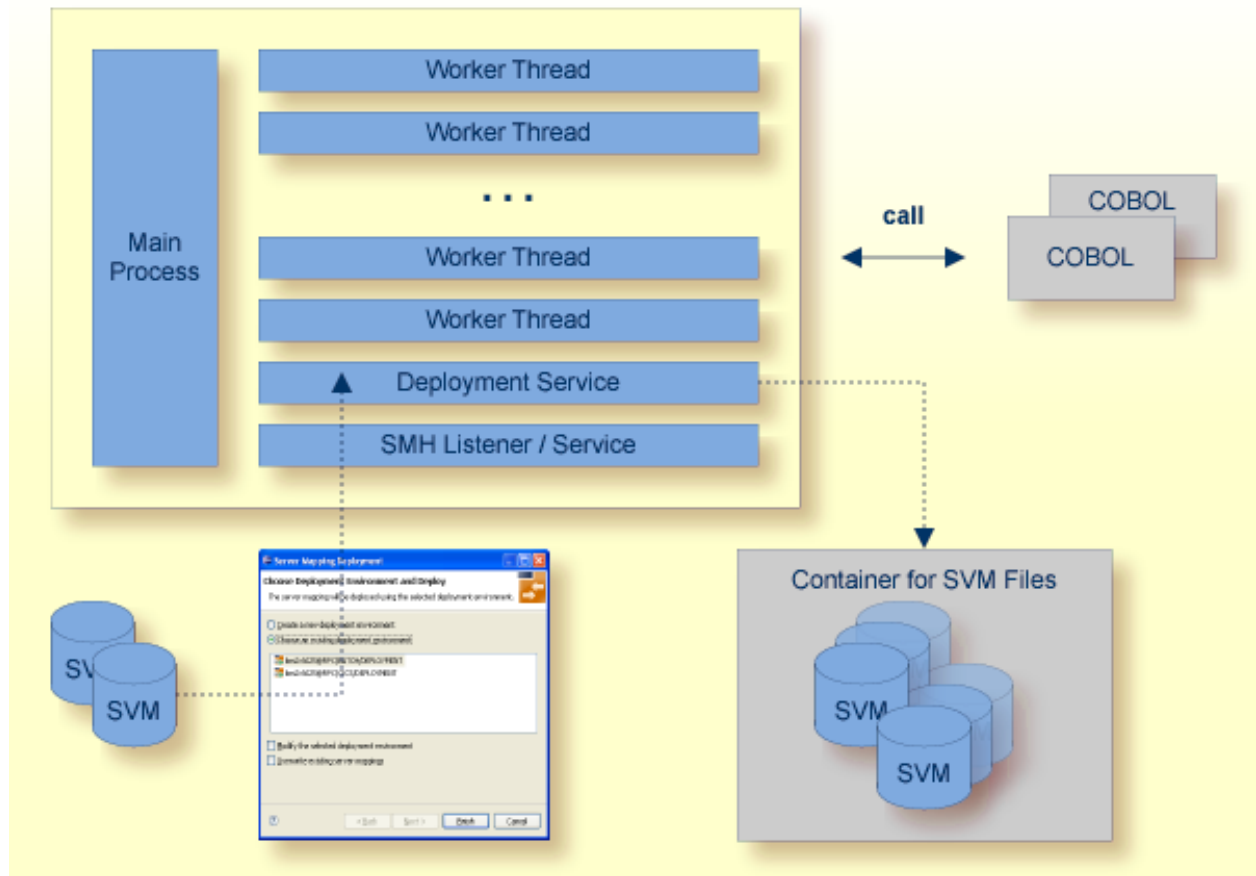
Extractor Service

The Extractor Service is a prerequisite for remote extractions with the *IDL Extractor for COBOL* and *IDL Extractor for PL/I*. See [Extractor Service under BS2000/OSD](#) for more information.



Deployment Service

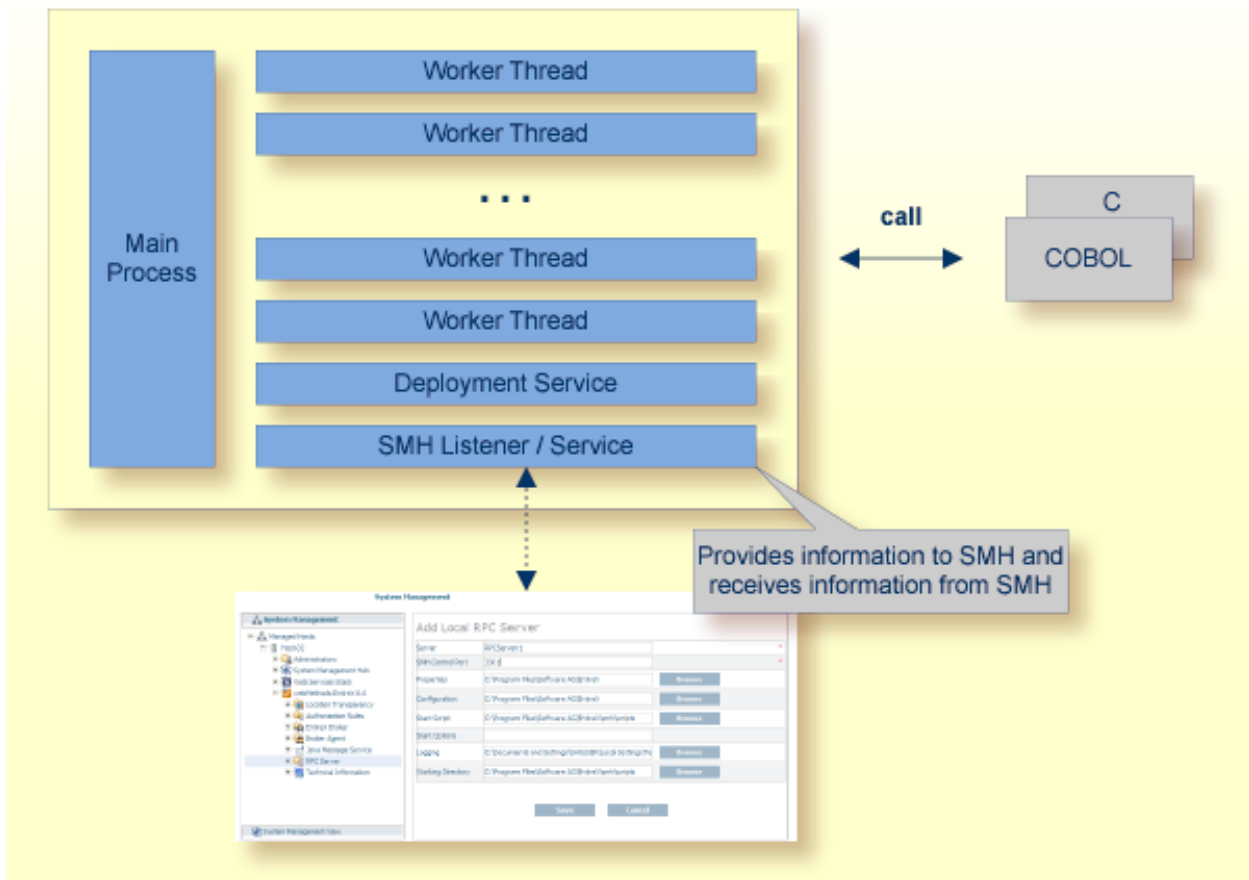
The Deployment Service allows you to deploy server mapping files (SVM files) interactively using the Deployment Wizard (see *Server Mapping Deployment*). On the RPC server side, the SVM files are stored in an ISAM file as the container. See *Deployment Service under BS2000/OSD* for configuration information.



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.

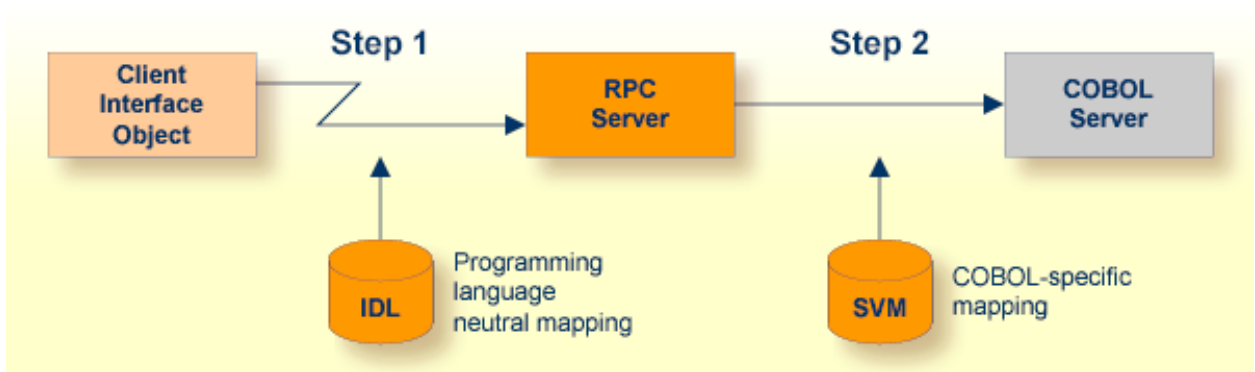
The SMH Service is switched on if the parameter `smhport` is set. See parameter `smhport` under *Configuring the RPC Server*.



Usage of SVM Files


The BS2000/OSD Batch RPC Server is optimized to call COBOL servers originally written for Fujitsu Siemens. For this purpose, the RPC server requires in many situations a server mapping file (SVM).

SVM files contain COBOL-specific mapping information that is not included in the IDL file and therefore *not* sent by an EntireX RPC client to the RPC server. See also [When is an SVM File Required?](#) under [SVM Files](#).



The RPC server marshalls 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 SVM file (Step 2). In this way the COBOL server can be called as expected.

The SVM files are retrieved as a result of the *IDL Extractor for COBOL* extraction process and the *COBOL Wrapper* if a COBOL server is generated.

 **Note:** SVM files are used for COBOL only.

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	18
▪ Stopping the RPC Server	19
▪ Activating Tracing for the RPC Server	21

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 mapping file
- 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
<code>brokerid</code>	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
<code>class</code>	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
<code>codepage</code>	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>	
<u>compresslevel</u>	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>compresslevel= 0 1 2 3 4 5 6 7 8 9 Y <u>N</u></p> <p>0-9 0=no compression 9=max. compression</p> <p><u>N</u> No compression. Y Compression level 6.</p> <p>Example: compresslevel=6</p>	O
<u>deployment</u>	NO	<p>Activates the deployment service, see <i>Deployment Service under BS2000/OSD</i>. Required to use the deployment wizard. See <i>Server Mapping Deployment Wizard</i> in the COBOL Wrapper 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 ENCRYPTION-LEVEL 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 <i>term_exit</i>.</p> <p>Example: init_exit=myExit</p>	O
<u>extractor</u>	NO	<p>The extractor service is a prerequisite for remote extractions. See <i>Extractor Service under BS2000/OSD</i>.</p> <p>extractor=YES NO</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>YES 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 (SVM) files are used to call the COBOL server correctly if one is available. See <i>Server Mapping Deployment</i>.</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 SVM files.</p> <p>SVM=<u>PREFERRED</u> NO</p> <p>PREFERRED This setting is to support COBOL servers built with the EntireX Workbench that do not have a server mapping file (SVM), plus COBOL servers built with an SVM file. If an SVM file is found, it is used to unmarshal the data and call the server. If no SVM file is found, the BS2000/OSD Batch RPC Server tries to unmarshal the data without the SVM file and call the server. This is the default.</p> <p>NO Server mapping files (SVMs) are not used to unmarshal the data. The BS2000/OSD Batch RPC Server tries to</p>	O

Parameter	Default	Values	Req/ Opt
		<p>unmarshal the data without the SVM file and call the server.</p> <p>Example for BS2000/OSD: SVM=NO</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 init_exit.</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 restartcycles.</p> <p>Example: <code>timeout=300</code></p>	O
<code>tracedestination</code>	ERXTrace.nnn.log	Trace output is written to SYSOUT.	O
<code>tracelevel</code>	None	<p>Trace level for the server. See also Activating Tracing for the RPC Server.</p> <pre>tracelevel = None Standard Advanced ↵ Support</pre> <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: <code>tracelevel=standard</code></p>	O
<code>userid</code>	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: <code>userid=MyUid</code></p>	R

Parameter	Default	Values	Req/ Opt								
workermodel	SCALE,1,3,slowshrink	<p>The BS2000/OSD Batch RPC Server can be configured to</p> <ul style="list-style-type: none"> adjust the number of worker threads to the current number of client requests: <pre data-bbox="841 485 1411 583">workermodel=(SCALE,from,thru [,slowshrink ← fastshrink])</pre> use a fixed number of worker threads: <pre data-bbox="841 695 1411 730">workermodel=(FIXED,number)</pre> <table border="1" data-bbox="808 789 1419 1797"> <tr> <td data-bbox="808 789 954 898">FIXED</td> <td data-bbox="954 789 1419 898">A fixed <i>number</i> of worker threads is used by the BS2000/OSD Batch RPC Server.</td> </tr> <tr> <td data-bbox="808 898 954 1797">SCALE</td> <td data-bbox="954 898 1419 1797"> <p>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.</p> <table border="1" data-bbox="954 1157 1419 1797"> <tr> <td data-bbox="954 1157 1143 1514">slowshrink</td> <td data-bbox="1143 1157 1419 1514">The RPC server stops all worker threads not used in the time specified by the timeout parameter, except for the number of workers specified as minimum value. This is the default if SCALE is used.</td> </tr> <tr> <td data-bbox="954 1514 1143 1797">fastshrink</td> <td data-bbox="1143 1514 1419 1797">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.</td> </tr> </table> </td> </tr> </table> <p data-bbox="813 1801 1162 1871">Example: workermodel=(SCALE,2,5)</p>	FIXED	A fixed <i>number</i> of worker threads is used by the BS2000/OSD Batch RPC Server.	SCALE	<p>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.</p> <table border="1" data-bbox="954 1157 1419 1797"> <tr> <td data-bbox="954 1157 1143 1514">slowshrink</td> <td data-bbox="1143 1157 1419 1514">The RPC server stops all worker threads not used in the time specified by the timeout parameter, except for the number of workers specified as minimum value. This is the default if SCALE is used.</td> </tr> <tr> <td data-bbox="954 1514 1143 1797">fastshrink</td> <td data-bbox="1143 1514 1419 1797">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.</td> </tr> </table>	slowshrink	The RPC server stops all worker threads not used in the time specified by the timeout parameter, except for the number of workers specified as minimum value. This is the default if SCALE is used.	fastshrink	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.	O
FIXED	A fixed <i>number</i> of worker threads is used by the BS2000/OSD Batch RPC Server.										
SCALE	<p>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.</p> <table border="1" data-bbox="954 1157 1419 1797"> <tr> <td data-bbox="954 1157 1143 1514">slowshrink</td> <td data-bbox="1143 1157 1419 1514">The RPC server stops all worker threads not used in the time specified by the timeout parameter, except for the number of workers specified as minimum value. This is the default if SCALE is used.</td> </tr> <tr> <td data-bbox="954 1514 1143 1797">fastshrink</td> <td data-bbox="1143 1514 1419 1797">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.</td> </tr> </table>	slowshrink	The RPC server stops all worker threads not used in the time specified by the timeout parameter, except for the number of workers specified as minimum value. This is the default if SCALE is used.	fastshrink	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.						
slowshrink	The RPC server stops all worker threads not used in the time specified by the timeout parameter, except for the number of workers specified as minimum value. This is the default if SCALE is used.										
fastshrink	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.										

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](#). Depending on the setting of the parameter `marshalling`, the BS2000/OSD Batch RPC Server supports COBOL object modules or C object modules.

COBOL

The approach used to derive the COBOL object module name for the RPC server depends on whether so-called server mapping files are used or not. See *Usage of SVM Files* in the BS2000/OSD administration documentation for an introduction.

- If SVM files are used, the IDL library and IDL program names are used to form a key to locate the SVM entry in the SVM container. If an SVM entry is found, the COBOL object module name of the RPC server is derived from the SVM entry. In this case the IDL program name can be different to the COBOL object module name if it is renamed during the wrapping process (see *Customize Automatically Generated Server Names*) or during the extraction process in the COBOL Mapping Editor (see *The Software AG IDL Tree Pane*).
- If no SVM files are 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* in the BS2000/OSD administration documentation or *Scenario II: Writing a New COBOL Server* in the BS2000/OSD administration documentation,

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=EXP960.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 EXP960.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>ip:port:TCP</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>ETBnnn:SVCmmm:NET</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	EXX960.JOBS
EXX-LIB	EntireX Broker module library	EXX960.LIB
WAL-MOD	WAL module library	WAL826.MOD

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

/CALL-PROCEDURE (EXP960.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 `EXP960.JOBS`.

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

3 Handling SVM Files under BS2000/OSD

- SVM Files in the EntireX Workbench 24
- SVM Files in the RPC Server 24
- Source Control of SVM Files 25
- Change Management of SVM Files 25
- Compare SVM Files 25
- List Deployed SVM Files 25
- Check if an SVM File Revision has been Deployed 26
- Access Control: Secure SVM File Deployment 26
- When is an SVM File Required? 26
- Is There a Way to Smoothly Introduce SVM Files? 28

A server mapping file (SVM) enables the RPC server to correctly support special COBOL syntax such as `REDEFINES`, `JUSTIFIED`, `SYNCHRONIZE` and `OCCURS DEPENDING ON` clauses, `LEVEL-88` fields, etc. If one of these elements is used, the EntireX Workbench automatically extracts an SVM file in addition to the IDL (interface definition language), or an SVM file is generated by the COBOL Wrapper for a server skeleton. The SVM file is used at runtime to marshal and unmarshal the RPC data stream.

SVM Files in the EntireX Workbench

In the *EntireX Workbench*, an SVM file has to relate to an appropriate IDL file. Therefore, you always have to keep the IDL file and the SVM file together in the same folder.

If there is an SVM file and a corresponding IDL file,

- at least one of the IDL programs in the corresponding IDL file requires server-mapping information to correctly call the target server. For those IDL programs, there is an SVM entry (line) in the Workbench SVM file.
- deployment of the SVM file to the RPC server is mandatory, see *Server Mapping Deployment*.

If there is an IDL file but no corresponding SVM file,

- there is no IDL program that requires server mapping information.

SVM Files in the RPC Server

Under BS2000/OSD, SVM entries of Workbench SVM files are stored as records of one ISAM file (containing all SVM entries from all Workbench SVM files). The unique key of the ISAM file consists of the first 255 bytes of the record: for the type (1 byte), the IDL library (127 bytes) and the IDL program (127 bytes).

If *one* server requires an SVM file, you need to provide this to the RPC server:

- Development environments: to allow the deployment of new SVM files, enable the deployment service. See [Enabling the Deployment Service](#).
- Production environments: provide SVM files to the RPC server. See configuration parameter `svm`.

If *no* server requires an SVM file, you can execute the RPC server without SVM files:

- Development environments: you can disable the deployment service. See [Disabling the Deployment Service](#).

- Production environments: there is no need to provide SVM files to the RPC server. See configuration parameter `svm`.

Source Control of SVM Files

Because SVM entries within an SVM file contain text data only, a Workbench SVM file is text-based (although it is not intended for human consumption). Therefore, you can include it in your source control management together with the IDL file and the COBOL source(s) as a triplet that should always be kept in sync.

Change Management of SVM Files

Under BS2000/OSD, change management for an ISAM file (SVM container) is similar to change management for an ordinary file. The complete ISAM file can be backed up at any time. All updates to the ISAM file done after a backup must be kept.

All Workbench SVM files added since the last backup should be available.

Compare SVM Files

For SVM files in the *EntireX Workbench* format, you can use a third party file/text compare tool to check if two files are identical.

The SVM entries (corresponding to lines in a Workbench SVM file) contain a creation timestamp at offset 276 (decimal) in the format `YYYYMMDDHHIISST`. The precision is 1/10 of a second.

List Deployed SVM Files

To list the SVM entries of Workbench SVM files, use the command:

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

where `<server-mapping-file>` is the ISAM file containing all SVM entries from all Workbench SVM files. See [SVM Files in the RPC Server](#).

Check if an SVM File Revision has been Deployed

SVM entries (corresponding to lines in Workbench SVM files) 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 ISAM file (SVM container).

Access Control: Secure SVM 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](#).

When is an SVM File Required?

For the IDL Extractor for COBOL

Interface Type	COBOL Syntax	COBOL Mapping Editor	SVM Required	More Information
CICS with DFHCOMMAREA Calling Convention and IN different to OUT	all		yes	<i>CICS with DFHCOMMAREA Calling Convention under Introduction to the IDL Extractor for COBOL CICS DFHCOMMAREA under COBOL Parameter Selection</i>
CICS Channel Container Calling Convention	all		yes	<i>CICS with Channel Container Calling Convention</i>
CICS with DFHCOMMAREA Large Buffer Interface	all		yes	<i>CICS with DFHCOMMAREA Large Buffer Interface</i>
IMS MPP Message Interface (IMS Connect)	all		yes	<i>IMS MPP Message Interface (IMS Connect)</i>
IMS BMP with Standard Linkage Calling Convention	all		yes	<i>IMS BMP with Standard Linkage Calling Convention</i>
Micro Focus with Standard Linkage Calling Convention	BINARY clause		yes	<i>Micro Focus with Standard Linkage Calling Convention</i>

Interface Type	COBOL Syntax	COBOL Mapping Editor	SVM Required	More Information
all	OCCURS DEPENDING ON clause		yes	<i>Tables with Variable Size - DEPENDING ON Clause under COBOL to IDL Mapping in the IDL Extractor for COBOL documentation</i>
all	REDEFINES clause		yes	<i>REDEFINE Clause</i>
all	TRAILING [SEPARATE] clause		yes	<i>SIGN LEADING and TRAILING SEPARATE Clause</i>
all	LEADING [SEPARATE] clause		yes	<i>SIGN LEADING and TRAILING SEPARATE Clause</i>
all	ALIGNED RIGHT attribute		yes	
all	all	Rename of program	yes	<i>The Software AG IDL Tree Pane under Mapping Editor User Interface in the IDL Extractor for COBOL documentation</i>
all	all	Map to operation	yes	<i>Context Menu under The COBOL Parameters Pane</i>
all	all	Map to constant	yes	<i>Context Menu</i>
all	all	Suppress	yes	<i>Context Menu</i>
other combinations			no	

For the COBOL Wrapper

This depends on the interface type chosen and the IDL type:

Interface Type	IDL Type	COBOL Wrapper	SVM Required	More Information
CICS with DFHCOMMAREA Large Buffer Interface	all		yes	<i>CICS with DFHCOMMAREA Large Buffer Interface under COBOL Server Interface Types</i>
CICS with Channel Container Calling Convention	all		yes	<i>CICS with Channel Container Calling Convention</i>
IMS BMP with Standard Linkage Calling Convention	all		yes	<i>IMS BMP with Standard Linkage Calling Convention</i>
Micro Focus	I2 or I4		yes	<i>Micro Focus with Standard Linkage Calling Convention IDL Data Types under</i>

Interface Type	IDL Type	COBOL Wrapper	SVM Required	More Information
				<i>Software AG IDL File</i> in the IDL Editor documentation
all	IDL unbounded array		yes	<i>array-definition</i> under <i>Software AG IDL Grammar</i> in the IDL Editor documentation
all	IDL unbounded group		yes	<i>group-parameter-definition</i> under <i>Software AG IDL Grammar</i> in the <i>IDL Editor</i> documentation
all	all	IDL program name is not a valid COBOL name and is therefore adapted, or the COBOL program name is customized	yes	<i>Customize Automatically Generated Server Names</i>
other combinations			no	

Is There a Way to Smoothly Introduce SVM Files?

All EntireX RPC servers can be executed without SVM files. There is no need to install the SVM container (see [SVM Files in the RPC Server](#)) as long as you do not use features that require SVM files (see [When is an SVM File Required?](#)). You can also call COBOL servers generated or extracted with previous versions of EntireX mixed with a COBOL server that requires SVM files. All EntireX RPC servers are backward compatible.

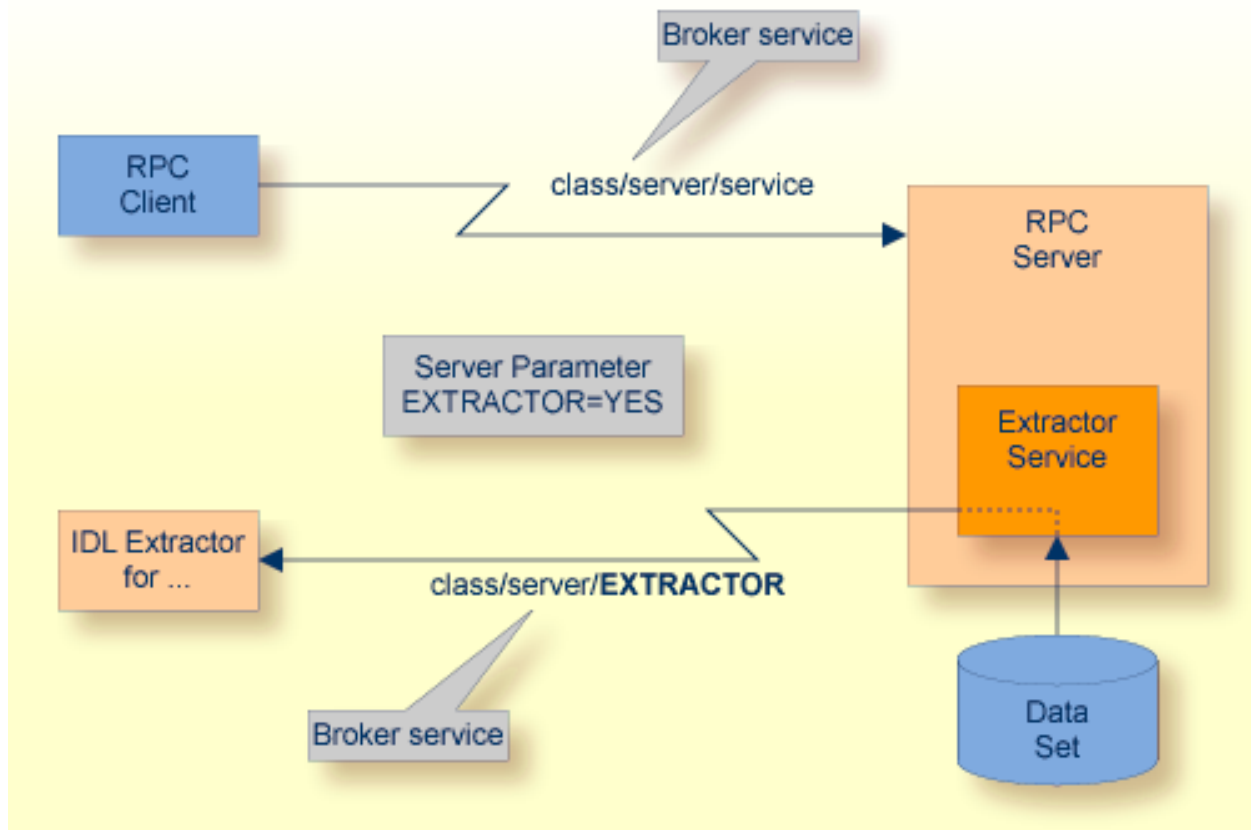
4 Extractor Service under BS2000/OSD

- 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* in the platform-independent administration documentation.

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.

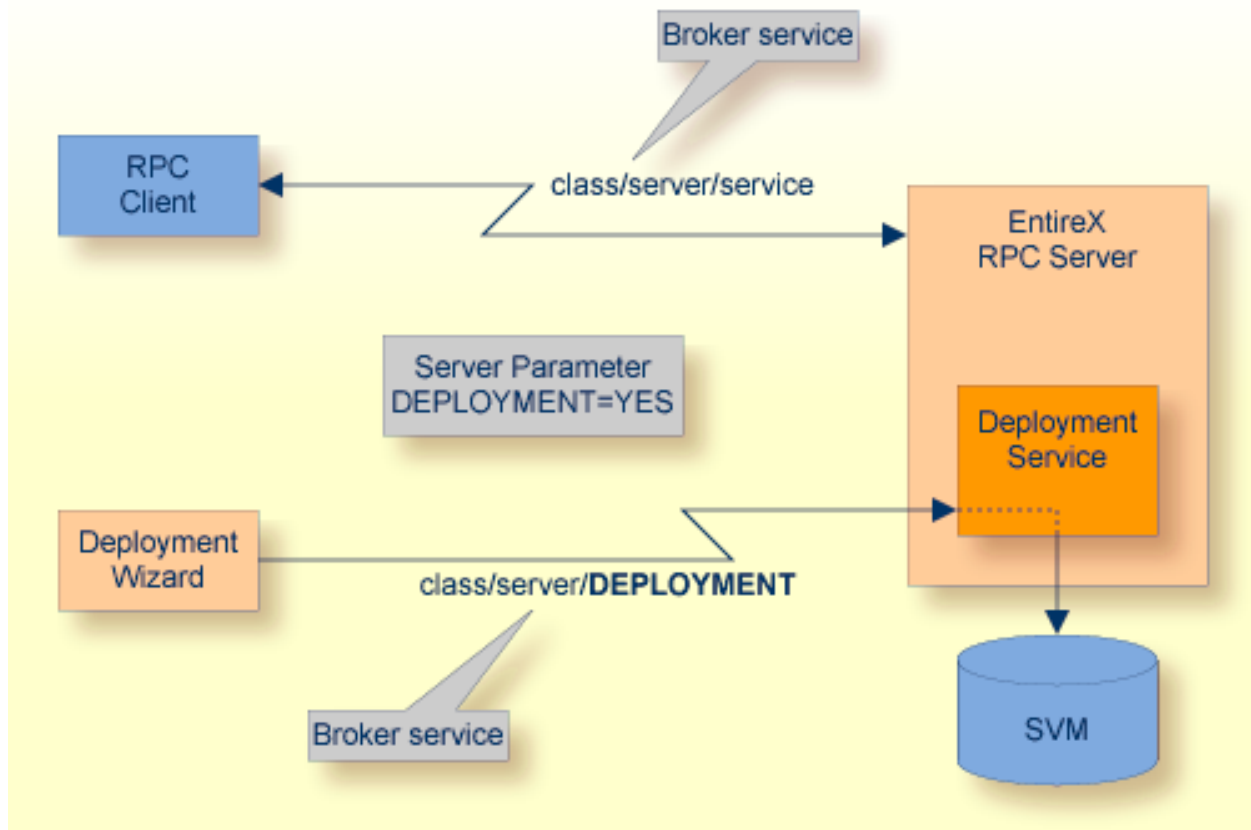
5 Deployment Service under BS2000/OSD

- 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*.
- 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.



Scope

The deployment service is used for the

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

See *Server Mapping Deployment Wizard*.

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 mapping file ISAM (container) must be installed and configured. See *Step 1: Define an RPC SVM File* 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 of the extraction, also an SVM file.
- 2 Build an EntireX RPC client using any EntireX wrapper. See *EntireX Wrappers*. 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*

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 of the extraction, also an SVM file. 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. See *EntireX Wrappers*. 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*

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. See *EntireX Wrappers*. 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*

