

## **webMethods EntireX**

### **EntireX Micro Focus COBOL RPC Server**

Version 9.5 SP1

November 2013

This document applies to webMethods EntireX Version 9.5 SP1.

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

Copyright © 1997-2013 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-MICROFOCUSRPC-95SP1-20140628**

## Table of Contents

1 Introduction to Micro Focus RPC Server .....	1
Inside the RPC Server .....	2
Usage of SVM Files .....	5
Scenario I: Calling an Existing COBOL Server .....	6
Scenario II: Writing a New COBOL Server .....	7
2 Administering the Micro Focus RPC Server .....	9
Customizing the RPC Server .....	10
Configuring the RPC Server .....	11
Locating and Calling the Target Server .....	18
Using SSL or TLS with the RPC Server .....	19
Starting the RPC Server .....	20
Stopping the RPC Server .....	21
Activating Tracing for the RPC Server .....	22
3 Deployment Service with Micro Focus RPC Server .....	23
Introduction .....	24
Scope .....	25
Enabling the Deployment Service .....	25
Disabling the Deployment Service .....	26
4 Handling SVM Files with Micro Focus RPC Server .....	27
SVM Files in the EntireX Workbench .....	28
SVM Files in the RPC Server .....	28
Source Control of SVM Files .....	29
Change Management of SVM Files .....	29
Compare SVM Files .....	29
List Deployed SVM Files .....	30
Check if an SVM File Revision has been Deployed .....	30
Access Control: Secure SVM File Deployment .....	30
When is an SVM File Required? .....	30
5 Running an EntireX RPC Server as a Windows Service .....	33
Introduction .....	34
Sample Installation .....	34
EntireX RPC Service Tool .....	36
Customizing the Service .....	38
Removing the Service .....	39
Tracing for the Windows Service .....	39
Starting an RPC Server Using the Service .....	39
Stopping an RPC Server Using the Service .....	40
Running with EntireX Security .....	40
Windows-specific Folders .....	41



# 1 Introduction to Micro Focus RPC Server

---

▪ Inside the RPC Server .....	2
▪ Usage of SVM Files .....	5
▪ Scenario I: Calling an Existing COBOL Server .....	6
▪ Scenario II: Writing a New COBOL Server .....	7

The EntireX Micro Focus COBOL RPC Server allows standard RPC clients to communicate with COBOL servers written with Micro Focus COBOL. It works together with the *COBOL Wrapper* and the *IDL Extractor for COBOL*.

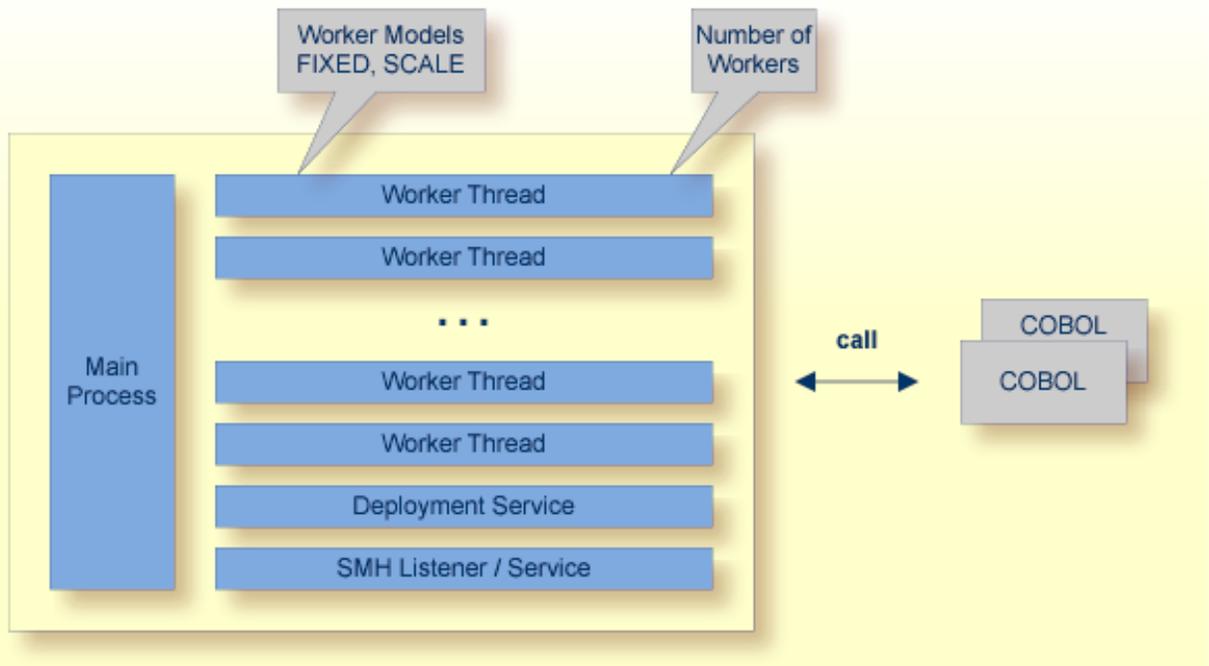
## Inside the RPC Server

---

- [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 Micro Focus 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.

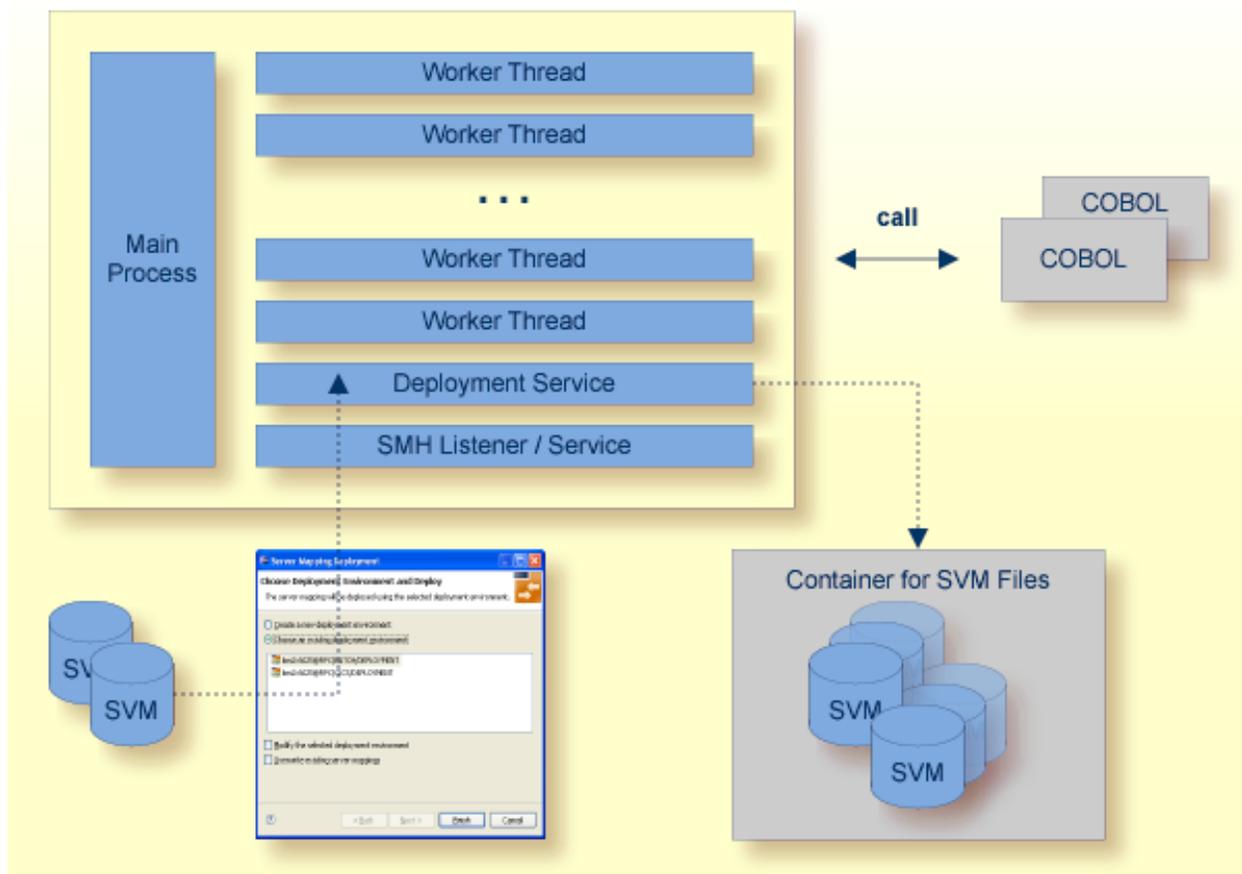
## Inbuilt Services

Micro Focus RPC Server provides the following services for ease-of-use:

- [Deployment Service](#)
- [SMH Listener Service](#)

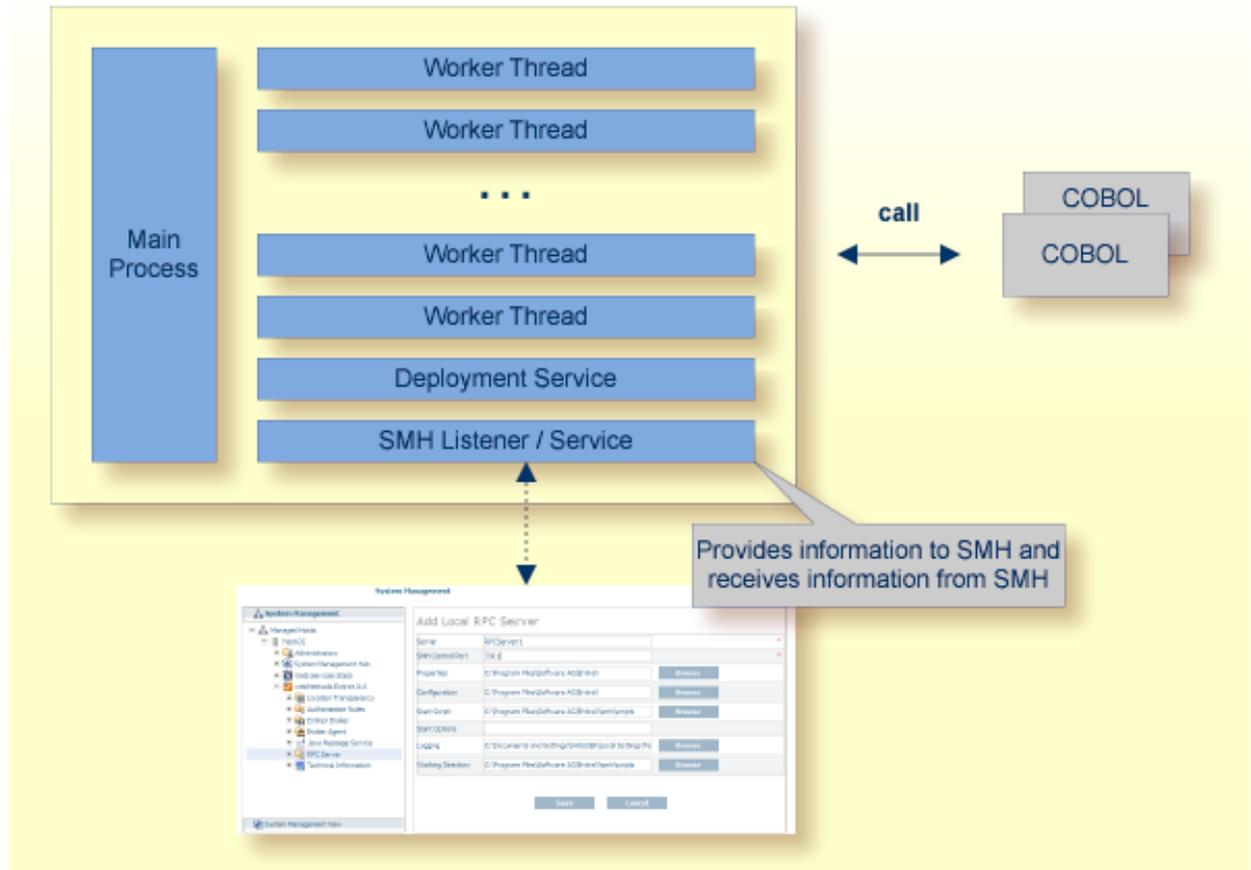
### Deployment Service

The Deployment Service allows you to deploy server mapping files (SVM files) interactively using the Deployment Wizard (see *Server Mapping Deployment*). On the RPC server side, the SVM files are stored in a directory as the container. See [Deployment Service with Micro Focus RPC Server](#) for configuration information.



## SMH Listener Service

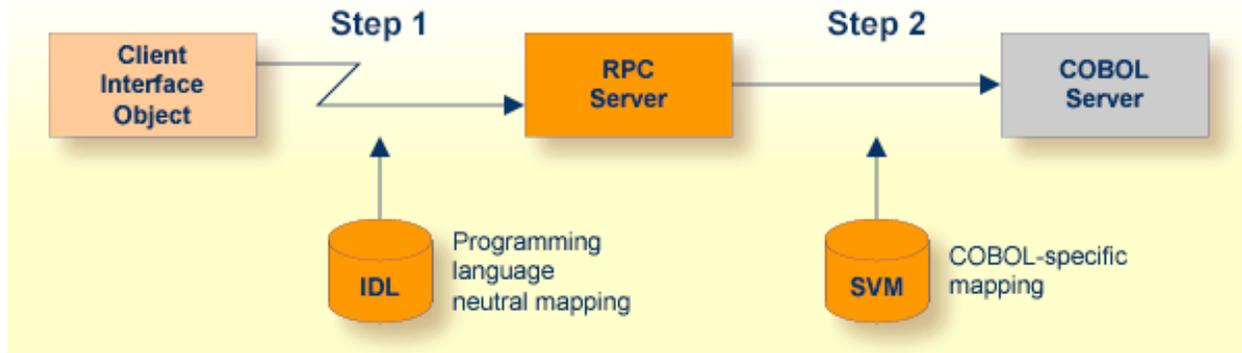
With the SMH Listener Service you use the System Management Hub to monitor the RPC server. See *Administering the EntireX RPC Servers using System Management Hub* in the UNIX and Windows administration documentation.



## Usage of SVM Files

The Micro Focus RPC Server is optimized to call COBOL servers originally written with Micro Focus COBOL, and servers ported from other environments (e.g. z/OS), to Micro Focus. 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 [Handling 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.

You can customize the usage of the SVM file using parameter `svm`. See [Configuring the RPC Server](#).

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

## 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 Micro Focus (UNIX and Windows)* 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 Micro Focus (UNIX and Windows)* for COBOL RPC Server examples.



## 2 Administering the Micro Focus RPC Server

---

▪ Customizing the RPC Server .....	10
▪ Configuring the RPC Server .....	11
▪ Locating and Calling the Target Server .....	18
▪ Using SSL or TLS with the RPC Server .....	19
▪ Starting the RPC Server .....	20
▪ Stopping the RPC Server .....	21
▪ Activating Tracing for the RPC Server .....	22

The EntireX Micro Focus COBOL RPC Server allows standard RPC clients to communicate with COBOL servers written with Micro Focus COBOL. It works together with the *COBOL Wrapper* and the *IDL Extractor for COBOL*.

## Customizing the RPC Server

---

The following elements are used for setting up the Micro Focus RPC Server:

- [Micro Focus COBOL Runtime](#)
- [Configuration File](#)
- [Start Script](#)

### Micro Focus COBOL Runtime

The COBOL runtime, for example *Micro Focus Server*, has to be installed according to the Micro Focus documentation. It is not delivered with this package. Provide the location of the COBOL runtime in the *Start Script*.

If a COBOL runtime is not provided, the Micro Focus RPC Server cannot be started and an error message is given.

### Configuration File

The name of the delivered example configuration file is *microfocussserver.cfg*. The configuration file contains the configuration for the Micro Focus RPC Server. The following settings are important:

- connection information such as broker ID, server address (class, name, service)
- location and usage of server mapping files (SVMs)
- scalability parameters
- trace settings
- etc.

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

## Start Script

The name of the start script is platform-dependent:

- UNIX: *microfocusserver.bsh*
- Windows: *microfocusserver.bat*

The start script for the Micro Focus RPC Server contains the following:

- the location of the Micro Focus COBOL runtime
- paths to the called COBOL server; see [Configuration Approaches](#)
- the configuration file used; see [Configuration File](#)
- etc.

## Configuring the RPC Server

The followings 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 equivalent.

Parameter	Default	Values	Req/Opt
<code>brokerid</code>	<code>localhost</code>	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

Parameter	Default	Values	Req/ Opt
<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</i> (DEFAULTS=SERVICE) under <i>Broker Attributes</i> in the administration documentation). 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> 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"> <li>■ follow the rules described under <i>Locale String Mapping</i> in the internationalization documentation</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=iso-8859-1</code></p>	R/O
<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><u>N</u> No compression.</p>	O

Parameter	Default	Values	Req/ Opt
		<p>Y Compression level 6.</p> <p>Example: compresslevel=6</p>	
<u>deployment</u>	NO	<p>Activates the deployment service, see <a href="#">Deployment Service with Micro Focus RPC Server</a>. 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> <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>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>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>Marshalling=(LANGUAGE=COBOL, flavor=MF) must be provided. Do not change these settings. The</p>	O

Parameter	Default	Values	Req/ Opt
		COBOL servers are called directly without a server interface object. So-called server mapping (SVM) files are used to call the COBOL server correctly if one is available. See <i>Server Mapping Deployment</i> and <i>Locating and Calling the Target Server</i> under <i>Administering the Micro Focus RPC Server</i> .	
<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 Micro Focus 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 <code>timeout</code> is the RPC server parameter (see this table), and  <code>ETB_TIMEOUT</code> 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
<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 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</p>	R

Parameter	Default	Values	Req/ Opt
		<p>the broker attribute file. See <i>Service-specific Attributes</i> (DEFAULTS=SERVICE) under <i>Broker Attributes</i> in the administration documentation. Case-sensitive, up to 32 characters. Corresponds to SERVICE attribute of the broker attribute file.</p> <p>Example: service=MYSERVICE</p>	
<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: smhport=3001</p>	O
<code>ssl_file</code>	no default	<p>Set the SSL parameters. See <a href="#">Using SSL or TLS with the RPC Server</a> for examples and more information.</p>	O
<code>svm</code>	xxx	<p>Usage and location (folder or directory) of SVM files.</p> <p>SVM=(PATH=<i>path</i>)</p> <p><i>path</i> The path to the location of SVM files can be absolute or relative to the current directory.</p> <p>Example for UNIX: SVM=(PATH=../config/svm)</p> <p>Example for Windows: SVM=(PATH=..\config\svm)</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 <a href="#">restartcycles</a>.</p> <p>Example: timeout=300</p>	O
<code>tracedestination</code>	ERXTrace. <i>nnn</i> .log	<p>The name of the destination file for trace output. By default the main trace file name is ERXTrace.<i>nnn</i>.log, where <i>nnn</i> can be in the range from 001 to 005.</p> <p>UNIX The trace file is located in the current working directory.</p> <p>Windows The trace file is located in a subfolder of the windows folder My Documents.</p> <p>Example:</p>	O

Parameter	Default	Values	Req/ Opt		
		tracedestination=ERXTrace.log			
<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   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:  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 Micro Focus 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] [,noisolation   isolation])</pre> </li> <li>use a fixed number of worker threads: <pre>workermodel=(FIXED,number [,noisolation   isolation])</pre> </li> </ul> <table border="1"> <tr> <td><b>FIXED</b></td> <td>A fixed <i>number</i> of worker threads is used by the Micro Focus RPC Server.</td> </tr> </table>	<b>FIXED</b>	A fixed <i>number</i> of worker threads is used by the Micro Focus RPC Server.	O
<b>FIXED</b>	A fixed <i>number</i> of worker threads is used by the Micro Focus RPC Server.				

Parameter	Default	Values	Req/ Opt
		<p>SCALE</p> <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>	
		<p>slowshrink</p> <p>The RPC server stops all worker threads not used in 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.</p>	
		<p>fastshrink</p> <p>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.</p>	
		<p>noisolation</p> <p>Calls to the COBOL server are executed within the Micro Focus RPC Server. If the COBOL server causes a COBOL runtime error, the Micro Focus RPC Server stops.</p>	
		<p>isolation</p> <p>Default. Calls to the COBOL server are executed in separate processes. If the COBOL server causes a COBOL runtime error, the Micro Focus RPC Server does not stop and continues.</p>	
		<p>Example: workermodel=(SCALE,2,5)</p>	

## Locating and Calling the Target Server

### Introduction

The Micro Focus RPC Server is able to call standard libraries (Windows DLLs or UNIX `so|sl`); Micro Focus proprietary formats such as intermediate code (`*.int`); generated code (`*.gnt`); and intermediate or generated code packaged in libraries (`*.lbr`). See the following table:

Executable Format	File Extension	File Name	Entry Point	Notes	Configuration
Operating system standard library with multiple server	<code>.so sl</code> (UNIX) or <code>.dll</code> (Windows)	IDL library	IDL program	1,2	1
Operating system standard library with single server	<code>.so sl</code> (UNIX) or <code>.dll</code> (Windows)	IDL program	IDL program	1,3,4	2
Micro Focus proprietary intermediate code	<code>.int</code>	IDL program		4	2
Micro Focus proprietary generated code	<code>.gnt</code>	IDL program		4	2
Micro Focus proprietary library with multiple server	<code>.lbr</code>	IDL library	IDL program	2,5	2
Micro Focus proprietary library with single server	<code>.lbr</code>	IDL program	IDL program	3,4,5	2

### Notes

1. This type of library is a standard library (UNIX shared library or Windows DLL).
2. This type of library may contain multiple COBOL servers. The IDL library name is used to form the operating system file name. The COBOL server names (entry points) are taken as follows:
  - if the COBOL Wrapper is used, by default from the IDL program names. The IDL program name can be different if it is renamed during the wrapping process, see *Customize Automatically Generated Server Names*
  - if the IDL Extractor for COBOL is used, from the COBOL program IDs. The IDL program name can be different if it is renamed during the extraction process in the COBOL Mapping Editor, see *The Software AG IDL Tree Pane*

If the IDL program name is different, an SVM file is required, see parameter `svm` under *Configuring the RPC Server*.

3. This type of library must contain one COBOL server only.
4. The IDL library name is not used. The COBOL server name (operating system file name and its entry point) are taken as follows:

- if the COBOL Wrapper is used, by default from the IDL program name. The IDL program name can be different if it is renamed during the wrapping process, see *Customize Automatically Generated Server Names*
- if the IDL Extractor for COBOL is used, from the COBOL program ID. The IDL program name can be different if it is renamed during the extraction process in the COBOL Mapping Editor, see *The Software AG IDL Tree Pane*

If the IDL program name is different, an SVM file is required, see parameter `svm` under *Configuring the RPC Server*.

5. Intermediate (\*.int) or generated (\*.gnt) code must be packaged in the library.

### Configuration Approaches

There are two approaches to access the COBOL server during runtime, which depend on the executable format (see table above):

1. The operating system's standard call mechanism is used to call libraries. Make sure your server(s) are accessible, for example:
  - under UNIX with the `LD_LIBRARY_PATH` environment variable
  - under Windows with the `PATH` environment variable
2. The Micro Focus environment variable `COBPATH` must be set before starting the RPC server. It lists all paths where a search for COBOL servers is to be performed. See the Micro Focus documentation for more information.

For both approaches, the start script of the Micro Focus RPC Server is an appropriate place to set the environment variables. See *Start Script*.

See also *Scenario I: Calling an Existing COBOL Server* or *Scenario II: Writing a New COBOL Server*.

## Using SSL or TLS with the RPC Server

There are two ways of specifying SSL or TLS, depending on the complexity of the parameters:

- as part of the Broker ID for short parameters, the simplest way
- using the SSL file, a text file containing more complex parameters.

For more information, see *SSL or TLS and Certificates with EntireX*.

## Specifying the SSL or TLS Parameters as Part of the Broker ID

The simplest way to specify SSL or TLS parameters is to add them to the Broker ID.

Example:

```
ssl://ETB001?TRUSTSTORE=whatever
```

## Specifying the SSL or TLS Parameters in a Separate File

For complex SSL or TLS parameters there is the SSL file, a text file containing the parameters.

The `SSL_FILE` keyword points to this text file.

### ▶ To specify the SSL or TLS parameters in the SSL file

- 1 Set the parameters as described under *Running Broker with SSL or TLS Transport* in the platform-specific administration documentation.
- 2 Prefix/suffix the Broker ID with the SSL key.

Example:

```
brokerid=SSL://ETB001  
.  
.  
ssl_file=C:\mySSLdirectory\mySSLParms.txt
```

## Starting the RPC Server

---

### ▶ To start the Micro Focus RPC Server

- Use the script *microfocusrpcserver* in the folder *bin* to start the Micro Focus RPC Server. You may customize this file.

Or:

Use the RPC server agent in the System Management Hub to configure and start the Micro Focus RPC Server.

See *Administering the EntireX RPC Servers using System Management Hub* in the UNIX and Windows administration documentation for details.

Or:

Use the Micro Focus RPC Server as a Windows service, see *Running an EntireX RPC Server as a Windows Service* in the Micro Focus RPC Server documentation.

## Stopping the RPC Server

---

### ▶ To stop the Micro Focus RPC Server

- Use the RPC server agent in the SMH to stop the Micro Focus RPC Server.

Or:

Use the agent for the broker. Use `Deregister` on the service, specified with the parameters `class/servername/service`.

## Activating Tracing for the RPC Server

---

▶ **To switch on tracing for the Micro Focus RPC Server**

- Set the parameters `tracelevel` and `tracedestination`. See [Configuring the RPC Server](#).

To evaluate the return codes, see *Error Messages and Codes*. See also *Tracing the RPC Server* in the UNIX and Windows administration documentation.

# 3 Deployment Service with Micro Focus RPC Server

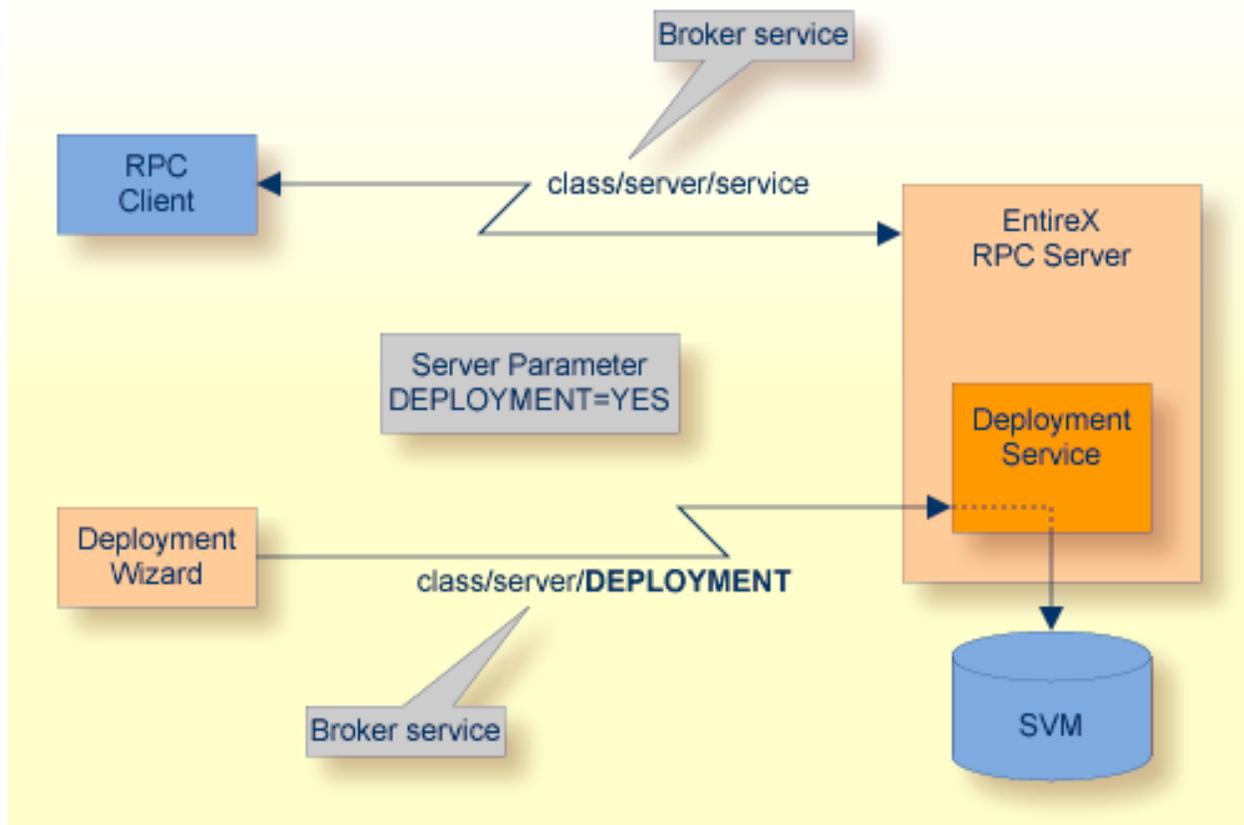
---

- 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*.
- 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 Micro Focus RPC Server, configure the server mapping file subparameter `path` of parameter `svm` to point to a directory with write access. See *Configuring the RPC Server* under *Administering the Micro Focus RPC Server*.
- 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 broker.

## Disabling the Deployment Service

---

### ▶ To disable the deployment service

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

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

# 4 Handling SVM Files with Micro Focus RPC Server

---

- SVM Files in the EntireX Workbench ..... 28
- SVM Files in the RPC Server ..... 28
- Source Control of SVM Files ..... 29
- Change Management of SVM Files ..... 29
- Compare SVM Files ..... 29
- List Deployed SVM Files ..... 30
- Check if an SVM File Revision has been Deployed ..... 30
- Access Control: Secure SVM File Deployment ..... 30
- When is an SVM File Required? ..... 30

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

---

For Micro Focus RPC Servers under UNIX or Windows, the SVM files of the Workbench are stored as directories (folders) and operating system files. For each IDL library, a directory is created by the deployment service during deployment and each SVM entry is stored as an operating system file within this directory. The anchor of the SVM directories is configured by the server mapping file subparameter "path" of parameter "svm", see *Configuring the RPC Server* under *Administering the Micro Focus RPC Server*. For example, deploying the file *example.svm* from the EntireX directory *examples/RPC/basic/example* results in folder EXAMPLE and operating system files for the SVM entries CALC and SQUARE:

```
./EXAMPLE  
/CALC.svm  
/SQUARE.svm
```

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

---

For the Micro Focus RPC Server under UNIX or Windows, change management within SVM directories (see [SVM Files in the RPC Server](#)) is similar to change management within ordinary operating system directories (folders). All updates to the SVM hierarchy 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 *YYYYMMDDHHIISS*. The precision is 1/10 of a second.

## List Deployed SVM Files

---

To list the contents of an SVM directory (see [SVM Files in the RPC Server](#)), use the Windows Explorer (for Windows) or the `ls` command (for UNIX).

## 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 `YYYYMMDDHHIISST`. Precision is 1/10 of a second. The creation timestamp can be checked.

The timestamp can be found on the same offset in the SVM files stored as operating system files in SVM directories.

## Access Control: Secure SVM File Deployment

---

For deployment with the *Server Mapping Deployment Wizard*, use EntireX Security if the broker is running on a platform other than BS2000/OSD. 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>

Interface Type	COBOL Syntax	COBOL Mapping Editor	SVM Required	More Information
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>
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 Software AG IDL File in the IDL Editor documentation</i>
all	IDL unbounded array		yes	<i>array-definition under Software AG IDL Grammar in the IDL Editor documentation</i>
all	IDL unbounded group		yes	<i>group-parameter-definition under Software AG IDL Grammar in the IDL Editor documentation</i>
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	

# 5

## Running an EntireX RPC Server as a Windows Service

---

▪ Introduction .....	34
▪ Sample Installation .....	34
▪ EntireX RPC Service Tool .....	36
▪ Customizing the Service .....	38
▪ Removing the Service .....	39
▪ Tracing for the Windows Service .....	39
▪ Starting an RPC Server Using the Service .....	39
▪ Stopping an RPC Server Using the Service .....	40
▪ Running with EntireX Security .....	40
▪ Windows-specific Folders .....	41

## Introduction

---

Any EntireX RPC server can be run as a Windows service. This has the following advantages:

- The server is started automatically when the PC is booted and shut down automatically when the PC is shut down.
- The service can be run under a system account that has different rights on a PC than the user account.
- The service can be run under any valid user ID.
- The Control Panel provides an overview of started servers and their status.

To run an RPC server as a Windows service, a batch script (.bat) that can start the RPC server is required. Sample batch scripts are provided in the EntireX *bin* folder:

- microfocusservice.bat
- cserver.bat
- cicseciservice.bat
- imconnectservice.bat
- dotNetServer.bat
- jrpservice.bat
- xmlrpcservice.bat
- ...

To install RPC server as windows services, use the EntireX RPC Service Tool. Multiple services can be installed if more than one RPC server is required.

## Sample Installation

---



**Note:** The steps below use the C RPC Server as an example, but the information applies to all RPC servers.

### ▶ To install an RPC Server as a Windows Service

- 1 Enter the EntireX installation directory and copy file *EntireX\bin\RPCService.bat*. Save the file under a different name, for example *EntireX\bin\RPCService\_C.bat*.

The EntireX installation directory is specified during installation. Later on when registering the service, the argument value of parameter `-ext` of `rpcservice.exe` must correspond with this extension, for example `-ext C`.

The extension `_C` will later correspond with the argument in parameter `-ext` when registering the service using `rpcservice.exe`. Also the name of the service will be extended by this value. This is needed to distinguish multiple server instances.

- 2 Modify batch file `RPCService_C.bat` and uncomment the line where the C RPC server batch file is referenced.

```
cserver.bat %*
```

The file contains entries for all the different types of RPC servers. The C RPC server is already uncommented, it is the default.

- 3 Adapt the corresponding configuration file, for example `\EntireX\config\cserver.cfg`, to your needs.
- 4 Optionally test whether the server is configured correctly before registering it as a Windows service by entering the following from a command prompt:

```
RPCService_C.bat
```

- 5 Register the Server as a Windows service. Enter:

```
rpcservice.exe -install -ext C -serverlog c:\serverlog.txt -trace  
c:\servertrace.txt -script c:\SoftwareAG\EntireX\bin\RPCservice_C.bat
```

where `c:\SoftwareAG` is the installation directory,

<code>-ext</code>	must correspond with the extension specified in step 1
<code>-serverlog</code>	is optional
<code>-trace</code>	is optional
<code>-script</code>	must point to file <code>RPCservice_C.bat</code> created in step 1

## EntireX RPC Service Tool

---

The EntireX RPC Service Tool is provided to install, remove, start and stop RPC services. An overview screen “Software AG EntireX RPC Service” lists the available arguments and options.

### Displaying the available Arguments and Options

▶ **To display the overview screen**

- In the command line, go to the EntireX *bin* directory and enter:

```
rpcservice.exe -help
```

The following screen is displayed:

```
Software AG EntireX RPC Service (V.n.n.n)
Service Tool to start/stop RPC servers
(c) Copyright Software AG 1997-20nn. All rights reserved.
-----

Usage:
rpcservice -install      [options]    to install the service
rpcservice -remove      [options]    to remove the service
rpcservice -help        [options]    prints this panel

Options:
  [-ext[ension]         <extension>]    use a service extension
  [-script              <path\script>]    set the server startup file
                                          for this service
  [-serverlog          <path\logfile>]    set the communication file
                                          between server and service
  [-userid             <userid>]         set userid for broker logon
  [-password           <password>]      set password for broker logon
  [-trace              [<tracefile>]]    used for diagnostics
```

## Explanation of Options

### ■ No option

This will install a service displayed as “Software AG EntireX RPC Service”. The service is registered as EXXRPCService (Service name under Properties). Required suboptions will be set to their defaults.

### ■ **-extension** <ext>

use this option to add an extension to the service to differentiate it from other instances, if you want to install multiple RPC service instances. The service will be displayed as “Software AG EntireX RPC Service [<ext>]”. The default log file *RPCservice\_<ext>.log*. will be added.

 **Important:** If you use this option, the program will by default search for a script with the same extension, *RPCService\_<ext>.bat*. Make sure it exists.

Example:

- For a Java RPC Server (see also *Running the Java RPC Server as a Windows Service* under *Administering the EntireX Java RPC Server* in the Windows administration documentation in the Windows administration documentation), you can use

```
-install -ext java -script <EntireX bin directory>\jrpcserver.bat
```

The service will be displayed as Software AG EntireX RPC Service [java], the log file will be called *RPCservice\_java.log*.

- For a Micro Focus RPC Server you can use:

```
-install -ext MicroFocus -script <EntireX bin directory>\microfocusserver.bat
```

The service will be displayed as Software AG EntireX RPC Service [MicroFocus], the log file will be called *RPCservice\_MicroFocus.log*.

- For a C RPC Server you can use:

```
-install -ext C -script cserver.bat
```

The service will be displayed as Software AG EntireX RPC Service [C], the log file will be called *RPCservice\_C.log*.

Default: none.

### ■ **-script**

use this option to specify a batch script other than the default scripts *RPCService.bat* or *RPCService\_<ext>.bat*.

 **Important:** The script file must pass external parameters to the RPC server. Please make sure that %\* remains at the end of the line that contains the RPC server call, so that parameters coming from the Windows Service are passed to the RPC server:

```
rpcserver.exe "cfg=..\config\cserver.cfg"%*
```

Default: <EntireX-installation>\bin\RPCService.bat  
or, if -ext <ext> is applied:  
<EntireX-installation>\bin\RPCService\_<ext>.bat

### ■ **-serverlog**

We recommend that you do not change the name of this file.

Default: CSIDL\_LOCAL\_APPDATA\Software AG\EntireX\RPCservice.log  
or, if -ext <ext> is applied:  
CSIDL\_LOCAL\_APPDATA\Software AG\EntireX\RPCservice\_<ext>.log

CSIDL\_LOCAL\_APPDATA is resolved at runtime according Microsoft's policy for user and application repositories. See [Windows-specific Folders](#).

### ■ **-userid and -password**

use these options if your target Broker requires EntireX Security.

These options are obsolete from EntireX versions 8.2 SP1. RPC servers communicate their user ID/password credentials via the server log file if required by EntireX Security. See [Running with EntireX Security](#). The options are still supported for compatibility with earlier versions or if the user ID and password of the RPC service have to be different from those of the RPC server(s). If the user ID and password are applied in this RPC service (and are different from the default), these values take precedence over the credentials that may be supplied by the RPC server(s).

Defaults:

User ID: ERX-USER

Password: PASSWORD



**Note:** This service requires the broker.dll. Make sure that when the service is executing under the control of the Windows Service Manager it can access the broker.dll in the standard path definitions or in the EntireX installation directory.

## Customizing the Service

---

You can set the service to start the RPC server automatically or manually.

### ▶ **To customize the service**

- 1 Go to **Control Panel > Administrative Tools > Services** and select the service and then **Properties**.
- 2 Set the Startup type as desired or
- 3 Start/stop the service as desired.

## Removing the Service

---

### ▶ To remove the service

- In the command line, go to the EntireX *bin* folder and enter:

```
rpcservice.exe -remove [-ext[ension] <extension>]
```

## Tracing for the Windows Service

---

Use the `-trace` option with the individual commands.

- with `-install`

The trace option will be stored in the System Registry and tracing will be enabled if the service is started/stopped by the Service Manager. It can only be switched off by removing the service.

- With the other commands trace runs as long as the command takes to finish its task.
- You can specify a target log file. By default the file is written to the Windows folder for user documents. See [Folder for User Documents](#).

## Starting an RPC Server Using the Service

---

If the startup type for the service is set to automatic, the server will be started when the PC is booted.

### ▶ To start the server manually

- Go to **Control Panel > Administrative Tools > Services** and select the service, then **Properties** and choose **Start**.

## Stopping an RPC Server Using the Service

---

If you selected startup type automatic in the service properties, the server will be stopped by the operating system when the PC is shut down.

### ▶ To stop the server manually

- Use the service, go to **Control Panel > Administrative Tools > Services** and select the service, then **Properties** and choose the **Stop** button.

If you choose to stop the server using the EntireX utilities, the Windows service will change the status from “running” to “stopped” and writes a warning message in the event log that the EntireX server has stopped for an unknown reason.

## Running with EntireX Security

---

### Starting an RPC Server running under EntireX Security

To start an RPC server, supply a valid user ID and password to log on to the broker. The ID and password are applied to the configuration of the individual RPC server(s).

### Stopping an RPC Server running under EntireX Security;

When you stop an RPC server, the RPC service uses the user ID/password from the credentials provided by the new RPC server(s), if applied. Otherwise user ID and password may be taken from the RPC service installation options (see above).



**Note:** Broker command service `etbcmd` is used to stop the broker. See *Broker Command and Information Services*. This means that users who can start RPC servers to certain broker services also need access rights to `Class=SAG, Server=ETBCIS, Service=*`.

## Windows-specific Folders

---

### Folder for Application Data

The file system directory that serves as a common repository for application-specific data.

A typical path is *C:\Documents and Settings\username\Application Data*.

Under Windows 7, this path for the SYSTEM user might resolve to:  
"C:\Windows\System32\config\systemprofile\AppData\Local".

### Folder for Local (Non-roaming) Application Data

The file system directory that serves as a data repository for local (non-roaming) applications.

A typical path is *C:\Documents and Settings\username\Local Settings\Application Data*.

### Folder for User Documents

The file system directory used to physically store a user's common repository of documents.

A typical path is *C:\Documents and Settings\username\My Documents*.

See [MSDN Library](#).

