

## **webMethods EntireX**

**EntireX RPC Server for IBM® AS/400®**

Version 10.9

April 2023

This document applies to webMethods EntireX Version 10.9 and all subsequent releases.

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

Copyright © 1997-2023 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

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

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

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

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

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

**Document ID: EXX-AS4RPC-109-20230403**

## Table of Contents

|  |    |
|--|----|
| 1 About this Documentation .....                         | 1  |
| Document Conventions .....                               | 2  |
| Online Information and Support .....                     | 2  |
| Data Protection .....                                    | 3  |
| 2 Introduction to the RPC Server for AS/400 .....        | 5  |
| Overview .....   | 6  |
| Post-installation Steps .....                            | 6  |
| Worker Models .....                                      | 7  |
| 3 Administering the RPC Server for AS/400 .....          | 9  |
| Customizing the RPC Server .....                         | 10 |
| Configuring the RPC Server Side .....                    | 12 |
| Configuring the IBM AS/400 Side .....                    | 15 |
| Using SSL/TLS with the RPC Server .....                  | 15 |
| Starting the RPC Server .....                            | 17 |
| Stopping the RPC Server .....                            | 17 |
| Pinging the RPC Server .....                             | 17 |
| Running an EntireX RPC Server as a Windows Service ..... | 18 |
| Application Identification .....                         | 18 |



# 1 About this Documentation

---

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

## Document Conventions

---

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

## Online Information and Support

---

### Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

In addition, you can also access the cloud product documentation via <https://www.software-ag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

### Product Training

You can find helpful product training material on our Learning Portal at <https://knowledge.softwareag.com>.

## Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://tech-community.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/publishers/softwareag> and discover additional Software AG resources.

## Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

## Data Protection

---

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

---



# 2 Introduction to the RPC Server for AS/400

---

- Overview ..... 6
- Post-installation Steps ..... 6
- Worker Models ..... 7

## Overview

---

The RPC Server for AS/400 acts on one side as an RPC server and on the other side as a client for calling AS/400 server programs.



For local extraction, all source files have to be stored locally on the same machine where the Designer is running. For existing AS/400 programs, use the *IDL Extractor for COBOL* to extract the *Software AG IDL File* in the IDL Editor documentation for the RPC clients.

## Post-installation Steps

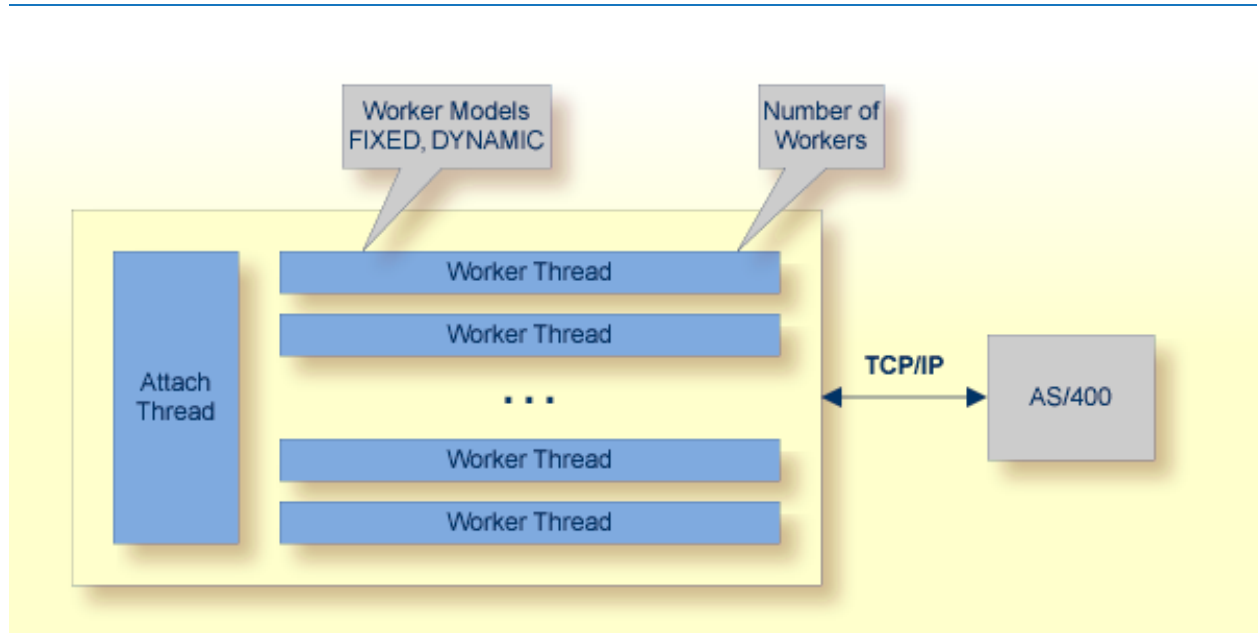
---

After installation of EntireX, the EntireX RPC Server for IBM® AS/400® cannot be started successfully. To access the AS/400 system you need the IBM ToolBox for Java (JTOpen).

### ➤ To download and install the IBM ToolBox for Java

- 1 From <https://sourceforge.net/projects/jt400/>, click **Downloads**. From <https://sourceforge.net/projects/jt400/>, select the latest version.
- 2 From the downloaded zip file, extract the file *lib/java8/jt400.jar*.
- 3 Copy the file *jt400.jar* to directory *EntireX/Classes*.

## Worker Models



RPC requests are worked off inside the RPC server in worker threads. If you are using RPC conversations, each RPC conversation requires its own thread during the lifetime of the conversation. The RPC Server for AS/400 can adjust the number of worker threads to the number of parallel requests. The 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.

- **DYNAMIC**

The *dynamic* model creates worker threads depending on the incoming load of RPC requests.

For configuration and technical details, see property `entirex.server.fixedservers` under *Administering the RPC Server for AS/400*.



# 3 Administering the RPC Server for AS/400

---

- Customizing the RPC Server ..... 10
- Configuring the RPC Server Side ..... 12
- Configuring the IBM AS/400 Side ..... 15
- Using SSL/TLS with the RPC Server ..... 15
- Starting the RPC Server ..... 17
- Stopping the RPC Server ..... 17
- Pinging the RPC Server ..... 17
- Running an EntireX RPC Server as a Windows Service ..... 18
- Application Identification ..... 18

The EntireX RPC Server for IBM® AS/400® enables you to remotely call server programs written in all ILE-based languages, for example C, COBOL, RPG.

This chapter describes how to administer the RPC Server for AS/400.

## Customizing the RPC Server

---

The following are used to set up the RPC Server for AS/400:

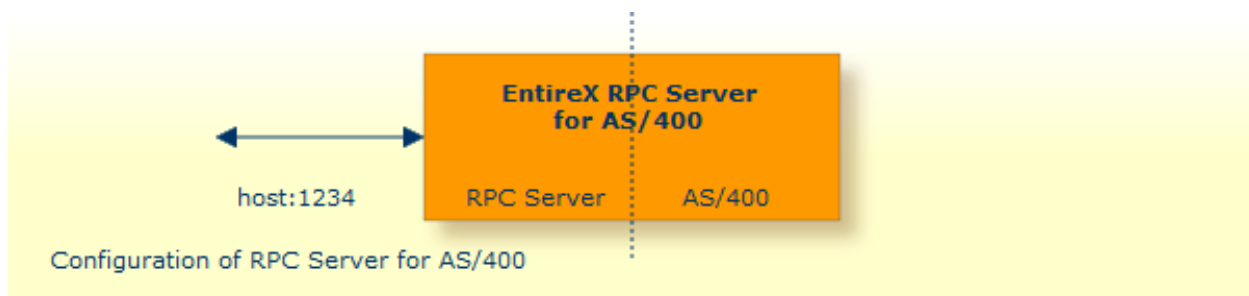
- [Configuration File](#)
- [Start Script](#)

### Configuration File

The default name of the configuration file is *entirex.as400.properties*. The RPC Server for AS/400 searches for this file in the current working directory.

You can set the name of the configuration file with `-Dentirex.server.properties=<your file name>` with `/"` as file separator.

The configuration file contains the configuration for both parts of the RPC Server for AS/400.



### Configuring more than one RPC Server

If you configure more than one RPC Server for AS/400 that connect to the same broker, the following items must be distinct:

- the trace output file (property `entirex.server.logfile`)
- the log for the Windows Service (property `entirex.server.serverlog`)

**Start Script**

The start script for the RPC Server for AS/400 is called *as400server.bsh* (Linux) or *as400server.bat* (Windows) and is provided in the *bin* folder of the installation directory. You may customize this file. The RPC Server for AS/400 itself is contained in the file *entirex.jar*.

## Configuring the RPC Server Side

The RPC Server for AS/400 uses the properties that start with “entirex.server” for configuring the RPC server side.

Alternatively to the properties, you can use the command-line options. These have a higher priority than the properties set as Java system properties, and these have higher priority than the properties in the configuration file.

| Property Name                | Command-line Option | Default          | Explanation  |
|------------------------------|---------------------|------------------|--|
| entirex.server.brokerid      | -broker             | localhost        | Broker ID.   |
| entirex.server.serveraddress | -server             | RPC/SRV1/CALLNAT | Server address.  |
| entirex.server.userid        | -user               | AS400RPCServer   | The user ID for access to the broker.  |
| entirex.server.fixedservers  |                     | no               | <p>NO The number of worker threads balances between what is specified in entirex.server.minservers and what is specified in entirex.server.maxservers. This is done by a so-called attach thread. At startup, the number of worker threads is the number specified in entirex.server.minservers. A new worker thread starts if the broker has more requests than there are worker threads waiting. If more than the number specified in entirex.server.minservers are waiting for requests, a worker thread stops if its receive call times out. The timeout period is configured with entirex.server.waitserver. See worker model <a href="#">DYNAMIC</a>.</p> <p>YES The number of worker threads specified in entirex.server.minservers is started and the server can process this number of parallel requests. See worker model <a href="#">FIXED</a>.</p> |
| entirex.server.minservers    |                     | 1                | Minimum number of server threads.  |



| Property Name                | Command-line Option | Default            | Explanation  |
|------------------------------|---------------------|--------------------|--|
| entirex.server.maxservers    |                     | 32                 | Maximum number of server threads.  |
| entirex.server.restartcycles | -restartcycles      | 15                 | Number of restart attempts if the Broker is not available. This can be used to keep the RPC Server for AS/400 running while the Broker is down for a short time.   |
| entirex.server.compresslevel | -compresslevel      | 0 (no compression) | Enter the text or the numeric value:<br><br>BEST_COMPRESSION      9<br>BEST_SPEED              1<br>DEFAULT_COMPRESSION -1<br>(mapped to              6)<br>DEFLATED                 8<br>NO_COMPRESSION        0<br>N                            0<br>Y                            8  |
| entirex.server.password      | -password           | yes                | The password for secured access to the broker. The password is encrypted and written to the property <code>entirex.server.password.e</code> .<br><br><ul style="list-style-type: none"> <li>■ To change the password, set the new password in the properties file.</li> <li>■ To disable password encryption, set <code>entirex.server.passwordencrypt=no</code>. Default=<code>yes</code>.</li> </ul> |
| entirex.server.security      | -security           | no                 | Valid values:<br>no   yes   auto   name of BrokerSecurity object.  |
| entirex.server.waitattach    |                     | 600S               | Wait timeout for the attach server thread.   |
| entirex.server.waitserver    |                     | 300S               | Wait timeout for the worker threads.   |
| entirex.timeout              |                     | 20                 | TCP/IP transport timeout.  |
|                              | -help               |                    | Display usage of the command-line parameters.  |
| entirex.server.logfile       | -logfile            | standard output    | Name of the log file.  |

| Property Name | Command-line Option | Default | Explanation  |
|---------------|---------------------|---------|--|
| entirex.trace | -trace              | 0       | <p>Trace level.</p> <p>0 No tracing, default.</p> <p>1 Trace all broker calls and other major actions.</p> <p>2 Dump the send and receive buffer.</p> <p>3 Dump the buffers sent to the broker and received from the broker.</p> |

## Configuring the IBM AS/400 Side

These properties are used to configure the connection to AS/400.

| Name                              | Default Value                             | Explanation   |
|-----------------------------------|---|---|
| as400.host                        |   | Host name of AS/400 system (IBM i). Mandatory.  |
| as400.programpath                 | /QSYS.LIB/%library%.<br>LIB/%program%.PGM | The fully qualified integrated file system path name to the program. %library% is replaced by the IDL file library name and %program% is replaced by the IDL file program name. The library and program name must each be 10 characters or less.  |
| entirex.bridge.<br>targetencoding | cp037                                     | Specify the appropriate EBCDIC encoding which corresponds to the CCSID (Coded Character Set Identifier) of your AS/400 system. This codepage is also used when communicating with the EntireX Broker.<br><br>Enable character conversion in the broker by setting the service-specific attribute <code>CONVERSION</code> to "SAGTRPC". See also <i>Configuring ICU Conversion</i> under <i>Configuring Broker for Internationalization</i> in the platform-specific Administration documentation. More information can be found under <i>Internationalization with EntireX</i> .<br><br>Default "cp037" is EBCDIC codepage with full Latin-1 character set. |
| as400.timeout                     | 50  | Maximum time to run the program in seconds.   |
| as400.userid                      |   | The user profile name to use to authenticate to the system.   |
| as400.password                    |   | The user profile password to use to authenticate to the system.   |

## Using SSL/TLS with the RPC Server

To use SSL with the RPC Server for AS/400, you need to configure the RPC Server Side.

RPC servers can use Secure Sockets Layer/Transport Layer Security (SSL/TLS) as the transport medium. The term "SSL" in this section refers to both SSL and TLS. RPC-based servers are always SSL clients. The SSL server can be either the EntireX Broker or Broker SSL Agent. For an introduction see *SSL/TLS, HTTP(S), and Certificates with EntireX* in the platform-independent Administration documentation.

**> To use SSL**

- 1 To operate with SSL, certificates need to be provided and maintained. Depending on the platform, Software AG provides sample certificates, but we strongly recommend that you create your own. See *SSL/TLS Sample Certificates Delivered with EntireX* in the EntireX Security documentation.
- 2 Set up the RPC Server for AS/400 for an SSL connection.

Use the *URL-style Broker ID* with protocol `ssl://` for the Broker ID. If no port number is specified, port 1958 is used as default. Example:

```
ssl://localhost:22101?trust_store=C:\SoftwareAG\EntireX\etc\ExxCACert.p12&trust_passwd=ExxCACert&verify_server=no
```

If the SSL client checks the validity of the SSL server only, this is known as *one-way SSL*. Two SSL parameters must be specified on the SSL client side: `trust_store` and `trust_passwd`. The mandatory `trust_store` parameter specifies the file name of a PKCS#12 certificate store that must contain the certificate chain of the trusted certificate authority (CA) that issued the SSL server's certificate.

To unlock this certificate store, the password has to be set with SSL parameter `trust_passwd`. By default a check is made that the certificate of the SSL server is issued for the hostname specified in the Broker ID. The common name of the subject entry in the server's certificate is checked against the hostname. If they do not match, the connection will be refused.

You can disable this check with SSL parameter `verify_server=no`.

If the SSL server additionally checks the identity of the SSL client, this is known as *two-way SSL*. In this case the SSL server requests a client certificate (the parameter `verify_client=yes` is defined in the configuration of the SSL server). Two additional SSL parameters must be specified on the SSL client side: `key_store` and `key_passwd`. This keystore must contain the private key of the SSL client. The password that protects the private key is specified with `key_passwd`.

The ampersand (&) character cannot appear in the password.

SSL parameters are separated by ampersand (&). See also *SSL/TLS Parameters for SSL Clients*.

- 3 Make sure the SSL server to which the RPC side connects is prepared for SSL connections as well. The SSL server can be EntireX Broker or Broker SSL Agent. See:
  - *Running Broker with SSL/TLS Transport* in the platform-specific Administration documentation
  - *Broker SSL Agent* in the platform-specific Administration documentation

## Starting the RPC Server

---

### ➤ To start the RPC Server for AS/400

- Use the *Start Script*.

Or:

Under Windows you can use the RPC Server for AS/400 as a Windows Service. See *Running an EntireX RPC Server as a Windows Service*.

## Stopping the RPC Server

---

### ➤ To stop the RPC Server for AS/400

- Use the command-line utility `etbcmd`. See `ETBCMD` under *Broker Command-line Utilities* in the platform-specific Administration documentation.

Or:

Use `CTRL-C` in the session where you started the RPC server instance.

Or:

Under Linux, enter command `kill -process-id`.

## Pinging the RPC Server

---

### ➤ To ping the RPC Server for AS/400

- Enter the following command:

```
java -classpath "$EXXDIR/classes/entirex.jar" ↵  
com.softwareag.entirex.rpcping.RPCServerPing -p <admin_port>
```

where *admin\_port* is the number of the administration port.

The `ping` command returns "0" if the server is reachable, and "1" if the server cannot be accessed.

## Running an EntireX RPC Server as a Windows Service

---

For general information see *Running an EntireX RPC Server as a Windows Service* in the Windows Administration documentation.

### ➤ To run the RPC Server for AS/400 as a Windows Service

- 1 Customize the *Start Script* according to your system installation.



**Note:** The script must pass external parameters to the RPC server and use the reduced signaling of the JVM (option `-Xrs`):

```
java -Xrs com.softwareag.entirex.as400bridge.AS400RPCServer %*
```

If `-Xrs` is not used, the JVM stops and an entry 10164002 is written to the event log when the user logs off from Windows.

See also *Starting the RPC Server*.

- 2 Test your RPC server to see whether it will start if you run your script file.
- 3 Use the *EntireX RPC Service Tool* and install the `RPCService` with some meaningful extension, for example `MyServer`. If your *Start Script* is `as400server.bat`, the command will be

```
RPCService -install -ext MyServer -script install_path\EntireX\bin\as400server.bat
```

The log file will be called `RPCservice_MyServer.log`.

- 4 In **Windows Services** menu (**Control Panel > Administrative Tools > Services**) select the service: `Software AG EntireX RPC Service [MyServer]` and change the property `Startup Type` from "Manual" to "Automatic".

## Application Identification

---

The application identification is sent from the RPC Server for AS/400 to the Broker. It is visible with Broker Command and Information Services.

The identification consists of four parts: name, node, type, and version. These four parts are sent with each Broker call and are visible in the trace information.

For the RPC Server for AS/400, these values are:

---

| Identification Part | Value                             |
|---------------------|-----------------------------------|
| Application name    | ANAME=RPC Server for IBM® AS/400® |
| Node name           | ANODE=< <i>host name</i> >        |
| Application type    | ATYPE=Java                        |
| Version             | AVERS=10.9.0.0                    |

