

webMethods EntireX

Using EntireX RPC for CL under IBM i

Version 10.1

October 2017

This document applies to webMethods EntireX Version 10.1 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 © 1997-2017 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-RPC-101-20191129CL

Table of Contents

1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
2 Using EntireX RPC for CL under IBM i	5
Creating a Sample Server in CL	6
Verifying the Server	8
Software AG IDL to CL Mapping	8

1 About this Documentation

▪ Document Conventions	2
▪ Online Information and Support	2
▪ Data Protection	3

Document Conventions

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

Online Information and Support

Software AG Documentation Website

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

Software AG Empower Product Support Website

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

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

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

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

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

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

Software AG TECHcommunity

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

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

Data Protection

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

2 Using EntireX RPC for CL under IBM i

■ Creating a Sample Server in CL	6
■ Verifying the Server	8
■ Software AG IDL to CL Mapping	8

Creating a Sample Server in CL

This section describes how to build a server application using the IBM i ILE language CL. The sample server will be named SENDMESS. Using the IBM i command SNGPGMMMSG (send program message), it sends a message to a given IBM i user and returns a confirmation to the RPC client.

This section tells you how to

- [Create the Client/Server Interface](#)
- [Create the Server](#)
- [Compile and Link the Server](#)

Create the Client/Server Interface

Using the EntireX Workbench on your PC, create a Software AG IDL file similar to the following:

```
Library 'EXAMPLE' Is
Program 'SENDMESS' Is
Define Data Parameter
  1 UserID          (A10)  In
  1 Message_Text    (A70)  In
  1 Confirmation    (A40)  Out
End-Define
```

Section [Software AG IDL to CL Mapping](#) describes how IDL data types are mapped to CL data items.

Create the Server

The server is implemented as an ILE CL program of type *PGM.

For our IDL example SENDMESS, the implemented server looks similar to the example below:

```
          PGM          PARM(&USER &MESSTEXT &CONFIRM)
/*-----*/
DCL      VAR(&USER) TYPE(*CHAR) LEN(10) /* the user ID */
DCL      VAR(&MESSTEXT) TYPE(*CHAR) LEN(70) /* the text */
DCL      VAR(&CONFIRM) TYPE(*CHAR) LEN(40) /* returned text */
/*-----*/
CHGVAR   VAR(&CONFIRM) VALUE(' ') /* clean it */
SNDPGMMSG MSG(&MESSTEXT) TOUSR(&USER) MSGTYPE(*COMP)

MONMSG   MSGID(CPF0000) EXEC(GOTO CMDLBL(BAD))
CHGVAR   VAR(&CONFIRM) +
          VALUE('Message sent to user' *BCAT &USER)
GOTO     CMDLBL(DONE) /* sending was ok */
/*-----*/
```

```
BAD:   CHGVAR      VAR(&CONFIRM) +
        VALUE('Message sending failed')
DONE:  ENDPGM
```

Because servers are running in a multithreaded environment, your application programs must be thread-safe. This implies that all commands and subprograms accessed in your servers must allow multithreads.

Compile and Link the Server

Compile the server source using the IBM i command CRTBNDC (create bound CL program).

The following example procedure demonstrates how to compile and bind an ILE CL program:

```
PGM      /* Compile and Bind a CL Server program      */
/*-----*/
DCL      VAR(&MODNAME) TYPE(*CHAR) LEN(10) VALUE(SENDMESS)
DCL      VAR(&LIBL) TYPE(*CHAR) LEN(10) VALUE(EXAMPLE)
DCL      VAR(&SRCF) TYPE(*CHAR) LEN(10) VALUE(QCLSRC)
DCL      VAR(&OPTL) TYPE(*CHAR) LEN(10) VALUE(*NONE)
DCL      VAR(&DBGV) TYPE(*CHAR) LEN(10) VALUE(*ALL)
/*-----*/
MONMSG   MSGID(CPF6801) EXEC(GOTO CMDLBL(DONE))
        /* If PF12 is pressed */

CRTBNDC  ??PGM(&LIBL/&MODNAME) ??SRCFILE(&LIBL/&SRCF) +
        ??SRCMBR(&MODNAME) DFTACTGRP(*NO) +
        ACTGRP(*CALLER) OUTPUT(*PRINT) +
        OPTIMIZE(&OPTL) DBGVIEW(&DBGV)

MONMSG   MSGID(LNC9001) EXEC(GOTO CMDLBL(ERRXT))
GOTO     CMDLBL(DONE)
/*-----*/
ERRXT:   SNDPGMMSG  MSG('MSG: Program Linkage Failed')
DONE:    RETURN
ENDPGM
```



Important: When linking/binding servers, the binding parameter ACTGRP(*CALLER) must be specified. This guarantees that the server application runs in the same activation group as the calling RPC Server.

Name the resulting server program like the program name in the IDL file and put it in a library whose name corresponds to the library name in the IDL file.

Example:

If a client performs an RPC which is based on the IDL program SENDMESS in the IDL library EXAMPLE, the remote RPC server will dynamically try to execute the ILE server program

SENDMESS in the IBM i library EXAMPLE. If no corresponding program can be found, the access will fail.

The principles of calling a server are described under *Administering the RPC Server*.

See *Step 3: Verify the RPC Server using COBOL* for how to start an RPC server that can execute the server program SENDMESS.

Verifying the Server

To verify the server program SENDMESS, you can use the EntireX IDL Tester.

➤ To verify the server

- 1 Confirm that an EntireX Broker and an EntireX RPC server are available in your network.
- 2 Start/submit an RPC server on your IBM i machine as described under *Starting the RPC Server*.
- 3 Create a *Software AG IDL File* in the IDL Editor documentation using the *IDL Editor* as described under *Create the Client/Server Interface*.
- 4 Start the IDL Tester.

Software AG IDL to CL Mapping

This section describes the specific mapping of Software AG IDL data types to the CL programming language. Please note also the remarks and hints on the Software AG IDL data types valid for all language bindings under section *Software AG IDL File* in the IDL Editor documentation.

The following topics are covered here:

- [Mapping IDL Data Types to CL Data Types](#)
- [Mapping Program and Library Names](#)
- [Mapping Arrays, Groups and Structures](#)

- [Mapping the Direction Attributes In, Out, InOut](#)

Mapping IDL Data Types to CL Data Types

In the table below, the following metasymbols and informal terms are used for the IDL.

- The metasymbols "[" and "]" enclose optional lexical entities.
- The informal term *number* (or in some cases *number1.number2*) is a sequence of numeric characters, for example 123.

Software AG IDL	Description	CL Data Type	See Notes
A_n	Alphanumeric	TYPE(*CHAR) LEN(n)	
$P(n - p)[.p]$	Packed decimal	TYPE(*DEC) LEN($n [p]$)	1

See also the hints and restrictions valid for all language bindings under *IDL Data Types*.



Notes:

1. n must be less than or equal to 15. The maximum value for p is 9.
For example, the IDL definition $P_{10.2}$ corresponds to `TYPE(*DEC) LEN(12 2)`

Other IDL data types have no appropriate equivalent in the CL language.

Mapping Program and Library Names

Do not use the special characters '#', '\$', '&', '+', '-', '.', '/' and '@' within names of programs and libraries in the IDL file. These characters are not allowed within names of server programs and libraries created on IBM i.

Mapping Arrays, Groups and Structures

Arrays, Groups and Structures are not supported for the CL language.

Mapping the Direction Attributes In, Out, InOut

The IDL syntax allows you to define parameters as IN parameters, OUT parameters, or IN OUT parameters (which is the default if nothing is specified). This direction specification is reflected in the stubless call of the RPC Server as follows:

- Direction attributes do not change the call interface because parameters are always treated as “called by reference”.
- Usage of direction attributes may be useful to reduce data traffic between RPC client and RPC server.
- Parameters with the IN attribute are sent from the RPC client to the RPC server.

- Parameters with the `OUT` attribute are sent from the RPC server to the RPC client.
- Parameters with the `IN` and `OUT` attribute are sent from the RPC client to the RPC server and then back to the RPC client.

Note that only the direction information of the top-level fields (Level 1) is relevant. Group fields always inherit the specification from their parent. A different specification is ignored.

See the `attribute-list` under *Software AG IDL Grammar* in the IDL Editor documentation for the syntax on how to describe attributes within the Software AG IDL file and refer to `direction-attribute`.