

COBOL Wrapper Reference

This chapter covers the following topics:

- The RPC Communication Area (Reference)
- Generic RPC Services Modules

The RPC Communication Area (Reference)

The RPC communication area is used to specify parameters that are needed to communicate with the broker and are not specific to client interface objects. These are, for example, the Broker ID, client parameters such as user ID, password and the server address such as class/servername/service etc. See the table below for a complete listing.

Notes:

1. See below the table for an explanation of column headings.
2. The RPC communication area is provided with the generated copybook ERXCOMM in the folder *include* for RPC client generation. See *Generating COBOL Source Files from Software AG IDL Files*.
3. See section *Using the RPC Communication Area* for the usage of the RPC communication area.

RPC Communication Area Field	Explanation	Req/ Opt/ Auto	In/ Out	Notes
ERXCOMM-HEADER	Label.	-	-	-
COMM-REQUEST	Label.	-	-	-
COMM-VERSION	Version of RPC communication area. Possible values: 2000.	R	I	-
COMM-FUNCTION	LO - log on to the Broker	O	I	1
	LF - log off from the Broker			1
	OC - open conversation			3
	CE - close conversation with commit			3
	CB - close conversation with backout			3
	CT - create Natural Security token			4
	RC - do reliable RPC commit			5
	RR - do reliable RPC rollback			5
	RS - get reliable status			5
	EC - end of conversation			5
COMM-RETURN-CODE	Message class and message code returned by COBOL Wrapper.	-	O	-

RPC Communication Area Field	Explanation	Req/ Opt/ Auto	In/ Out	Notes
COMM-MESSAGE-TEXT-EX	Message text provided by COBOL Wrapper (long versions).	-	O	-
COMM-MESSAGE-TEXT	Message text provided by COBOL Wrapper (short versions).	-	O	-
ERXCOMM-AREA1	Label.	-	-	-
COMM-USERID	Label.	-	-	-
COMM-USERID1	User ID (8 characters) used for Natural Security tokens.	O	I	4
COMM-USERID2	User ID extension.	O	I	-
COMM-PASSWORD	Password used for Natural Security tokens.	O	I	4
COMM-LIBRARY	Library information used by Natural Security token.	O	I	4
COMM-SECURITY-TOKEN-LENGTH	Length of Natural Security token.	-	O	4
COMM-SECURITY-TOKEN	Natural Security token.	-	O	-
COMM-IN-CONVERSATION	Control variable internally used by generic RPC services and client interface objects. If set to Y, RPC requests will use COMM-ETB-CONV-ID for conversationality.	A	I/O	3,6
COMM-IN-ACTIVE-UOW	Control variable internally used by generic RPC services and client interface objects for reliable RPC. If set to Y, RPC requests will use COMM-ETB-UOW-ID for reliability.	A	I/O	5,6
COMM-RELIABLE-STATE	Control variable used by the application to determine whether standard RPC requests or reliable RPC messages are used. Valid values: <div> <div>’ ’</div> <div>normal RPC requests</div> <div>(blank)</div> </div> <div> <div>A</div> <div>reliable RPC in</div> <div>AUTO-COMMIT mode</div> </div> <div> <div>C</div> <div>reliable RPC in</div> <div>CLIENT-COMMIT mode</div> </div>	R	I/O	5

RPC Communication Area Field	Explanation	Req/ Opt/ Auto	In/ Out	Notes
COMM-RELIABLE-STATUS	Result of a "get reliable status" call to generic RPC services, see field COMM-FUNCTION above. Values correspond to broker ACI field UOWSTATUS.		O	5
COMM-ETB-BROKER-ID	Corresponds to Broker ACI field BROKER-ID.	R	I	-
COMM-ETB-SERVER-CLASS	Corresponds to Broker ACI field SERVER-CLASS.	R	I	-
COMM-ETB-SERVER-NAME	Corresponds to Broker ACI field SERVER-NAME.	R	I	-
COMM-ETB-SERVICE-NAME	Corresponds to Broker ACI field SERVICE.	R	I	-
COMM-ETB-USER-ID	Corresponds to Broker ACI field USER-ID.	O	I	1,2
COMM-ETB-PASSWORD	Corresponds to Broker ACI field PASSWORD.	O	I	1,2
COMM-ETB-TOKEN	Corresponds to Broker ACI field TOKEN.	O	I/O	-
COMM-ETB-SECURITY-TOKEN	Internal field. Corresponds to Broker ACI field SECURITY-TOKEN.	A	I/O	6
COMM-ETB-CONV-ID	Internal field. Corresponds to Broker ACI field CONV-ID.	A	I/O	3,6
COMM-ETB-WAIT	Corresponds to Broker ACI field WAIT. Default: 60 seconds.	O	I	-
COMM-ETB-APIVERS	Corresponds to Broker ACI field API-VERSION. Default=4.	O	I	-
COMM-ETB-UOW-ID	Corresponds to Broker ACI field UOWID.	O	I/O	5
COMM-ETB-STORE	Corresponds to Broker ACI field STORE.	O	I/O	5
COMM-ETB-PROGRAM-OFFSET	Fields are used internally for accounting purposes. See <i>Accounting in EntireX Broker</i> under z/OS UNIX Windows z/VSE.	A	I/O	6
COMM-ETB-LIBRARY-OFFSET		A	I/O	6

RPC Communication Area Field	Explanation	Req/ Opt/ Auto	In/ Out	Notes
APPMON-SUPPORT	Fields are used internally to support <i>Application Monitoring</i>	A	I/O	6
APPMON-VERIFY		A	I/O	6
APPMON-TIMEVALUE		A	I/O	6
APPMON-TRANSPORT-BUFFER		A	I/O	6
APPMON-LEN-TRANSPORT-BUFFER		A	I/O	6
APPMON-RECEIVE-BUFFER		A	I/O	6
APPMON-LEN-RECEIVE-BUFFER		A	I/O	6
APPMON-LEN-DATA		A	I/O	6
APPMON-RETURN-CODE		A	I/O	6

RPC Communication Area field

Name of the field in the RPC communication area.

Explanation

Explanation of the purpose of the field.

Req/Opt/Auto

Indicates for input fields whether they have to be given by the RPC application (required) or may be given by the user (optional). Fields marked with "Auto" are managed internally by the *Generic RPC Services Modules* themselves.

In/Out

Indicates whether the field is an input field (to be given by the RPC application) or an output field (returned to your RPC application).

Notes:

1. See *Using Broker Logon and Logoff*.
2. Optional if broker does not require security, required if broker is secured.
3. RPC conversations are supported when communicating with an RPC server. For more information, see *Using Conversational RPC*.
4. Natural Security is only relevant if communicating with a Natural RPC server. See *Using the COBOL Wrapper with Natural Security and Impersonation*.
5. See *Reliable RPC for COBOL Wrapper*.
6. Field is managed internally by the *Generic RPC Services Modules* themselves. For these to work properly you need to initialize the RPC Communication Area before using it in your RPC client application. See *Step 1: Declare and Initialize the RPC Communication Area*. Do not change this field in your RPC client application.

Generic RPC Services Modules

This section covers the following topics:

- Introduction
- Generic RPC Services Modules Usage
- Delivered Modules for z/OS
- Delivered Modules for z/VSE
- Delivered Modules for BS2000/OSD
- Delivered Modules for IBM i
- Adapting the Used Broker Stub

Introduction

The generic RPC services module COBSRVI is required for RPC clients.

- It can be optionally generated during RPC client generation in the folder *client* in the container *folder*. Section *Generate Generic RPC Service for Module COBSRVI* under *Generating COBOL Source Files from Software AG IDL Files* explains how to generate the RPC service module COBSRVI.
- It contains functions needed for RPC communication where a client interface object(s) is not needed. Refer to the functions documented with the RPC communication area field COMM-FUNCTION under *The RPC Communication Area (Reference)* for a list of provided functions.
- It manages internal states held inside the RPC communication area for conversational RPC, reliable RPC etc. See *The RPC Communication Area (Reference)*.
- From a COBOL programmer's point of view, it is always called with the COBOL program name COBSRVI, even for the delivered mainframe sources COBSRVIB, COBSRVIC and COBSRVID.
- It contains the call to the broker stub.

Generic RPC Services Modules Usage

The delivered modules on mainframe platforms are mainly for a quick demonstration of the delivered examples. The best approach is to use the modules generated by the EntireX Workbench, for the following reasons:

- The modules delivered on mainframe platforms may be out-of-date.
- You can set generation options, for example String Literal (see *Characters Used for String Literals*), individually as required.

Environment	Scenarios	Source to be Used	Description	Installation Linkage Usage
z/OS and z/VSE	Batch. See <i>Using the COBOL Wrapper for Batch</i> (z/OS, BS2000/OSD, z/VSE and IBM i).	COBSRVIB	This module has a call interface to your COBOL RPC client application.	Linked to your client application or can be called dynamically.
	CICS with DFHCOMMAREA calling convention. See <i>Using the COBOL Wrapper for CICS with DFHCOMMAREA Calling Convention</i> (z/OS and z/VSE).	COBSRVIC	This module has an EXEC CICS LINK interface to your COBOL RPC client application.	Installed only once within CICS as a CICS program and shared by all COBOL RPC client applications.
	CICS with call interfaces. See <i>Using the COBOL Wrapper for CICS with Call Interfaces</i> (z/OS and z/VSE).	COBSRVID	This module has a call interface to your COBOL RPC client application.	Linked to your client application or can be called dynamically.
z/OS IMS	IMS. See <i>Using the COBOL Wrapper for IMS</i> (z/OS).	COBSRVIB	This module has a call interface to your COBOL RPC client application.	Linked to your client application or can be called dynamically.
z/OS IDMS/DC	IDMS/DC with call interfaces. See <i>Using the COBOL Wrapper for IDMS/DC with Call Interfaces</i> (z/OS).	Not delivered.	This module has a call interface to your COBOL RPC client application. Generate it with the EntireX Workbench.	Linked to your client application or can be called dynamically.
BS2000/OSD	Batch. See <i>Using the COBOL Wrapper for Batch</i> (z/OS, BS2000/OSD, z/VSE and IBM i).	COBSRVI	This module has a call interface to your COBOL RPC client application.	Linked to your client application or can be called dynamically.
IBM i	Batch. See <i>Using the COBOL Wrapper for Batch</i> (z/OS, BS2000/OSD, z/VSE and IBM i).	RPCSRVI	This module has a call interface to your COBOL RPC client application. Do not use this module; it is out of date. Generate it as COBSRVI with the EntireX Workbench.	Linked to your client application or can be called dynamically.

Delivered Modules for z/OS

Module	Data Set	Description	Notes
COBSRVIB	EXP970.SRCE	Batch generic RPC services with call interface.	2
COBSRVIC	EXP970.SRCE	CICS generic RPC services with EXEC CICS LINK interface.	2
COBSRVID	EXP970.SRCE	CICS generic RPC services with call interface.	2
COBIGYIC	EXP970.SRCE	JCL to compile the CICS generic RPC services module COBSRVIC with EXEC CICS LINK interface.	2
ERXCOMM	EXP970.INCL	RPC communication area.	1
ERXRCSR	EXP970.SRCE	C main module for application errors.	3
ERXRCSR	EXP970.LD00	Ready-to-use ERXRCSR module for application errors.	3
EXPCOBCL	EXP970.JOBS	JCL to compile the CICS generic RPC service module COBSRVIC with EXEC CICS LINK interface.	2

Module

Name of the delivered module.

vrs

Version, release and service pack.

EXP970.INCL

Generic RPC include data set. The generic RPC include data set may be delivered as a patch with a different name EXP970.IN nn , where nn is the patch level number. Make sure you install the highest patch level available. The data set is required to SYSLIB input for the COBOL compiler.

EXP970.SRCE

Generic RPC source data set. The generic RPC source data set may be delivered as a patch with a different name EXP970.S0 nn , where nn is the patch level number. Make sure you install the highest patch level available. The data set is required to SYSLIB input for the COBOL compiler.

Notes:

1. See *The RPC Communication Area (Reference)*.
2. See *Generic RPC Services Modules Usage*.
3. See *Aborting RPC Server Customer Code and Returning Error to RPC Client (z/OS | z/VSE)* in the CICS RPC Server documentation.

Delivered Modules for z/VSE

File	Sublibrary	Description	Notes
ERXCOMM	EXP970	RPC Communication area.	1
COBSRVIB.C	EXP970	Batch generic RPC services with call interface (source).	2, 3
COBSRVIB.OBJ	EXP970	Batch generic RPC services with call interface (object).	2, 3
COBSRVIC.C	EXP970	CICS generic RPC services with EXEC CICS LINK interface (source).	2, 3
COBSRVIC.OBJ	EXP970	CICS generic RPC services with EXEC CICS LINK interface (object).	2, 3
COBSRVID.C	EXP970	CICS generic RPC services with call interface (source).	2, 3
COBSRVID.OBJ	EXP970	CICS generic RPC services with call interface (object).	2, 3

File

Name of the delivered file.

Sublibrary

Name of the delivered sublibrary.

Description

Purpose of the file.

Notes:

1. See *The RPC Communication Area (Reference)*.
2. See *Generic RPC Services Modules Usage*.
3. We recommend you use module COBSRVI generated by the EntireX Workbench instead of the modules COBSRVIB, COBSRVIC and COBSRVID delivered with your z/VSE installation. The reason for this is that the EntireX Workbench is updated much more frequently. Section *Generate Generic RPC Service for Module COBSRVI* under *Generating COBOL Source Files from Software AG IDL Files* explains how to generate the RPC service module.

Delivered Modules for BS2000/OSD

Module	Data Set	Description	Notes
ERXCOMM	EXP811.COBC	RPC communication area.	1
COBSRVI.COB	EXP811.COBC	Batch generic RPC services with call interface.	2, 3

Notes:

1. See *The RPC Communication Area (Reference)*.
2. See *Generic RPC Services Modules Usage*.
3. We recommend you use module COBSRVI generated by the EntireX Workbench instead of the delivered module. The reason for this is that the EntireX Workbench is updated much more frequently. Section *Generate Generic RPC Service for Module COBSRVI* under *Generating COBOL Source Files from Software AG IDL Files* explains how to generate the RPC service module.

Delivered Modules for IBM i

Module	Source file	Description	Notes
ERXCOMM	QCBLLSRC	RPC communication area.	1
RPCSRVI	QCBLLSRC	Batch generic RPC services with call interface.	2, 3

Notes:

1. See *The RPC Communication Area (Reference)*.
2. See *Generic RPC Services Modules Usage*
3. Do not use module RPCSRVI delivered with your IBM i installation. It does not support all the features described here, for example reliable RPC. Use module COBSRVI generated by the EntireX Workbench instead. Section *Generate Generic RPC Service for Module COBSRVI* under *Generating COBOL Source Files from Software AG IDL Files* explains how to generate the RPC service module.

Adapting the Used Broker Stub

Because multiple broker stubs may be offered per operating system and environments, it may be necessary to adapt the COBSRVI module to the correct broker stub that supports the required transport (TCP, SSL, NET). To do this, modify the COBOL subprogram DOBROKER inside the COBSRVI source file with a broker stub that meets your requirements.

For availability and information on broker stubs, see *Administration of Broker Stubs* under z/OS | UNIX | Windows | BS2000/OSD | IBM i.

Caution:

Do not make any modifications other than changing the broker stub name, and do not modify the COBOL subprogram COBSRVI inside the same COBSRVI program source. Unexpected behavior will occur.