

Introduction to the COBOL Wrapper

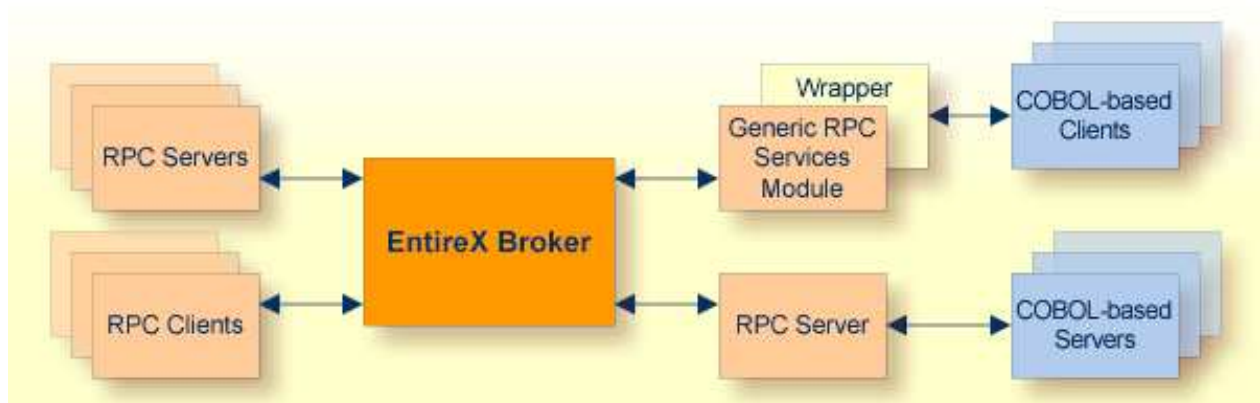
EntireX COBOL Wrapper provides access to RPC-based components from COBOL applications. It enables you to develop both client and server applications.

This chapter covers the following topics:

- Description
 - Generic RPC Services Module
 - COBOL Client Applications
 - COBOL Server Application
 - COBOL Server Interface Types
-

Description

The COBOL Wrapper provides access to RPC servers for COBOL client applications and access to COBOL servers for any RPC client. The COBOL Wrapper generation tools of the Workbench take as input a Software AG IDL file, which describes the interface of the RPC, and generate COBOL sources that implement the functions and data types of the interface.



The generated functions can be compiled with the COBOL compiler of your target platform.

The COBOL Wrapper works as follows:

- COBOL code is generated from the Software AG IDL file.
- Additionally for the client side, and depending on your target operating system and environment (e.g. Micro Focus, batch, CICS or IMS), a generic RPC services module is generated (see below).
- If required for the server side, a so-called server mapping file is created. A server mapping file is an EntireX Workbench file with extension `.svm` or `.cvm`. See *Server Mapping Files for COBOL*.

- The Software AG IDL Compiler and an appropriate template are used for the COBOL code generation.

Generic RPC Services Module

In order to minimize the amount of code generated for a specific IDL file, all service-type functionality that is not specific to a given IDL file required by the client interface object is generated in a generic RPC services module.

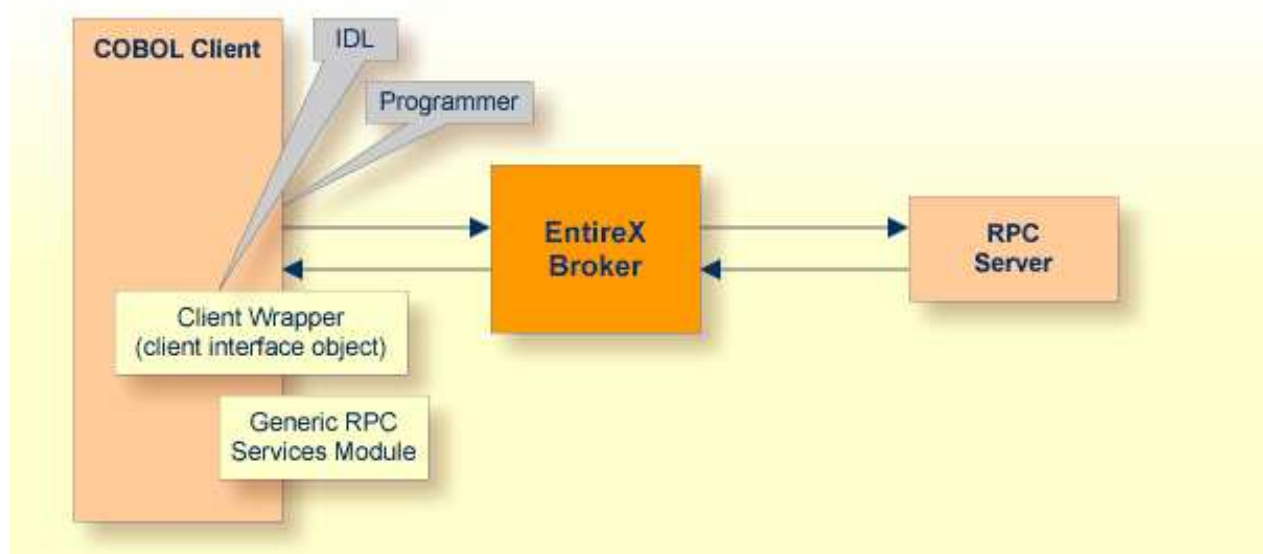
The generic RPC services module is used by RPC clients and contains the call to the broker stub, as well as other functions needed for RPC communication where an interface object is not needed, such as

- broker logon and logoff
- conversational support
- connecting RPC clients to RPC servers via the broker
- etc.

For more information, see *Generic RPC Services Modules*.

COBOL Client Applications

For a given IDL file, the Software AG IDL Compiler and a COBOL code generation template for clients are used to generate client interface objects and copybooks. The source code generated by the COBOL Wrapper can be compiled with your target COBOL compiler. Application developers use the generated generic RPC service module, the client interface object(s) and the copybooks to write COBOL applications that access RPC servers.

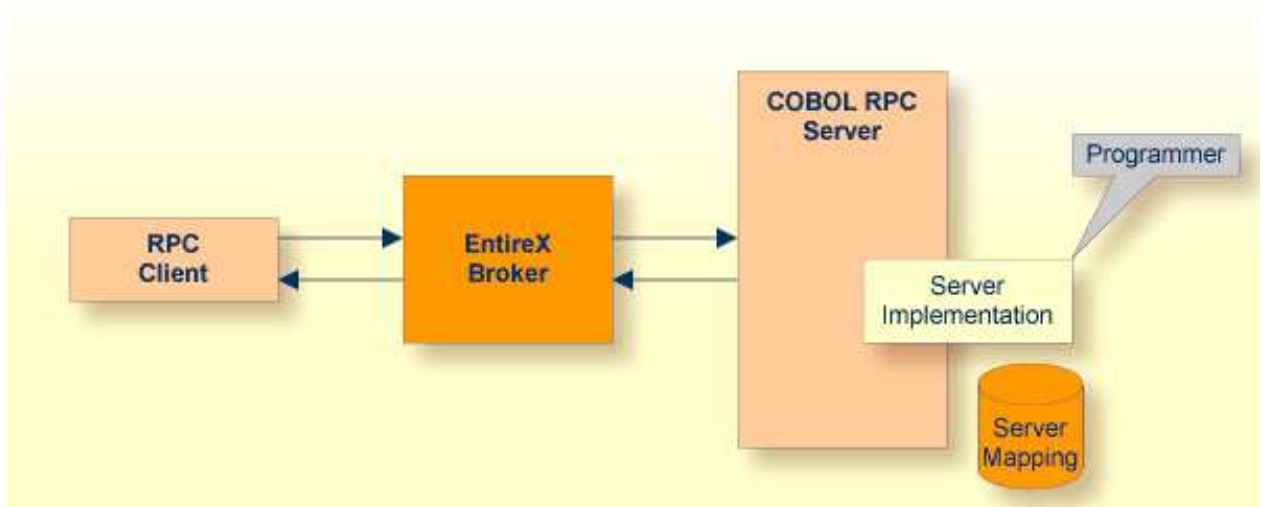


For more information, see *Using the COBOL Wrapper*.

COBOL Server Application

The Software AG IDL Compiler and a COBOL code generation template for servers are used to generate a server (skeleton) for a specific IDL. Additionally, depending on the IDL data types and whether IDL program names are customized, a so-called server mapping file is created. A server mapping file is an EntireX Workbench file with extension .svm or .cvm. See *When is a Server Mapping File Required?*.

Application developers use the generated server (skeleton) to write their own server code for each program in the IDL. The source code is compiled and linked with your target COBOL compiler. Client-side and server-side mapping files are handled differently. See *Server Mapping Files for COBOL* and *Using the COBOL Wrapper for the Server Side*.



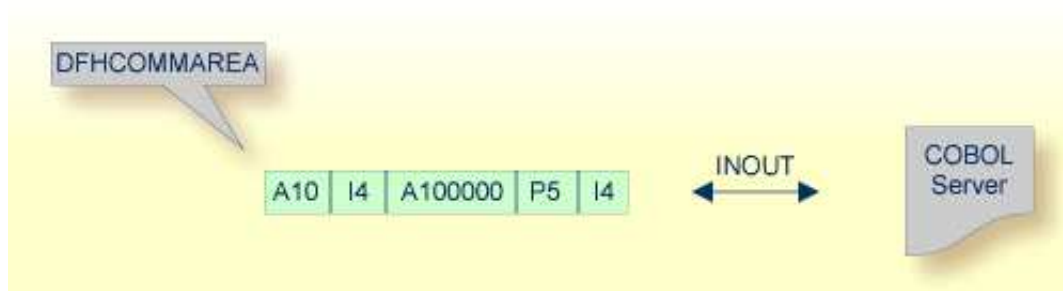
COBOL Server Interface Types

Depending on your requirements and generation settings, the COBOL Wrapper generates a server skeleton with one of the following interface types:

- CICS with DFHCOMMAREA Calling Convention
- CICS with Channel Container Calling Convention
- CICS with DFHCOMMAREA Large Buffer Interface
- Micro Focus with Standard Linkage Calling Convention
- Batch with Standard Linkage Calling Convention
- IMS BMP with Standard Linkage Calling Convention
- Compatibility between COBOL Interface Types and RPC Server

CICS with DFHCOMMAREA Calling Convention

CICS programs using the standard DFHCOMMAREA for parameter passing.



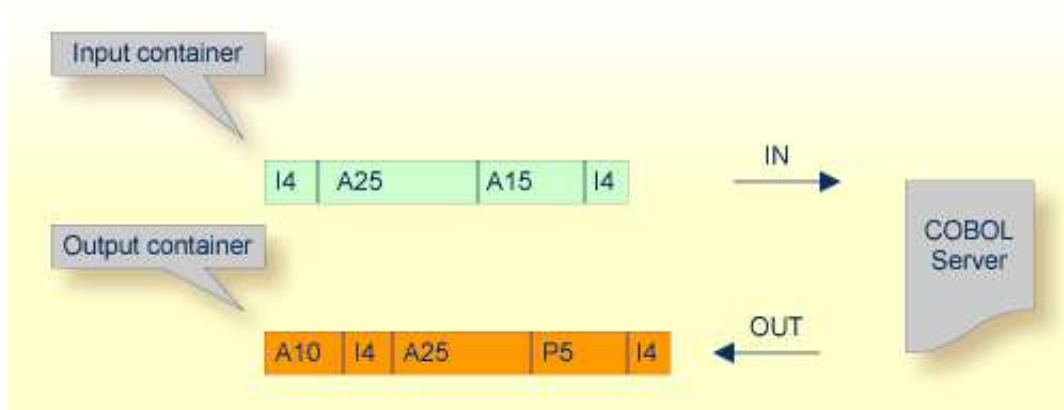
Technically, the generated COBOL server skeleton contains

- in the DFHCOMMAREA, the parameter structure

See *Server Interface Types* for more information on how to create COBOL servers with this interface type.

CICS with Channel Container Calling Convention

Channels and containers are IBM's approach to access more than 31 KB of data in CICS. There is no need for coding any channel container statements because all this is generated. Thus the programmer focus can be on the application logic.



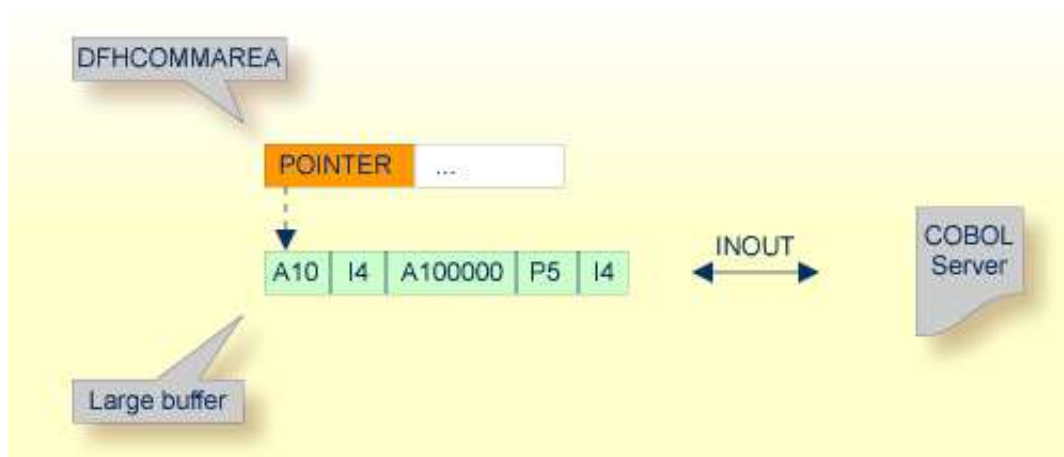
Technically, the generated COBOL server skeleton contains

- container layouts in the linkage section
- EXEC CICS CONTAINER statements for accessing the container on input and output

See *Server Interface Types* for more information on how to create COBOL servers with this interface type.

CICS with DFHCOMMAREA Large Buffer Interface

This type of program has a defined DFHCOMMAREA interface to access more than 31 KB of data in CICS. The interface is the same as the webMethods WMTLSRVR interface. This enables customers to use an easy and simple interface type to access more than 31 KB of data in CICS.



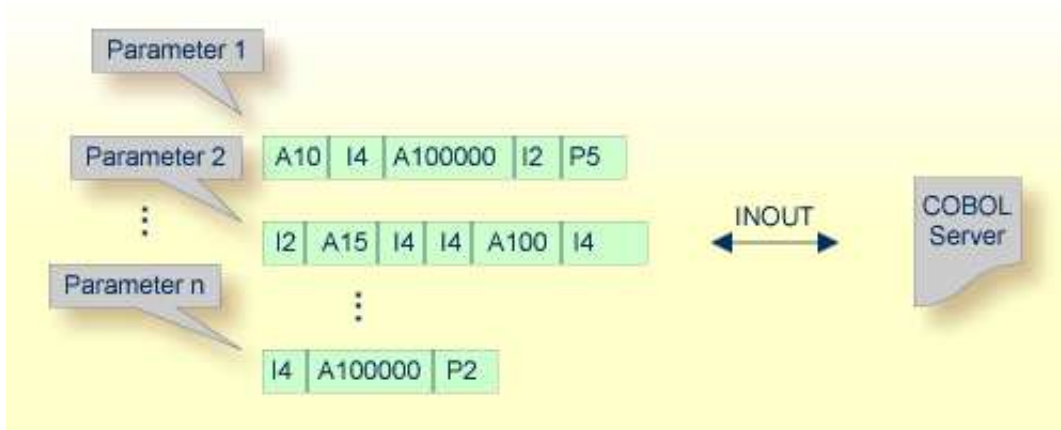
Technically,

- the generated server skeleton contains in the DFHCOMMAREA layout a *pointer* to a large buffer
- the parameter structure in the linkage section is accessed using COBOL's SET ADDRESS statement using the large buffer pointer

See *Server Interface Types* for more information on how to create COBOL servers with this interface type.

Micro Focus with Standard Linkage Calling Convention

Standard call interfaces with a given number of parameters are supported. Every parameter addresses a fixed COBOL structure.



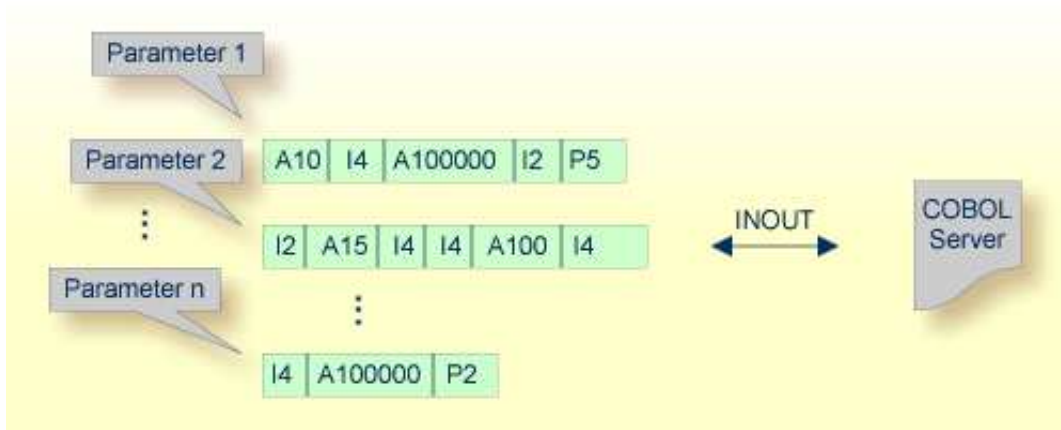
Technically, the generated COBOL server skeleton contains

- a parameter list `PROCEDURE DIVISION USING PARM1 PARM2 . . . PARMn`
- the parameters in the linkage section as COBOL data items on level 1

See *Server Interface Types* for more information on how to create COBOL servers with this interface type.

Batch with Standard Linkage Calling Convention

Standard call interfaces with a given number of parameters are supported. Every parameter addresses a fixed COBOL structure.



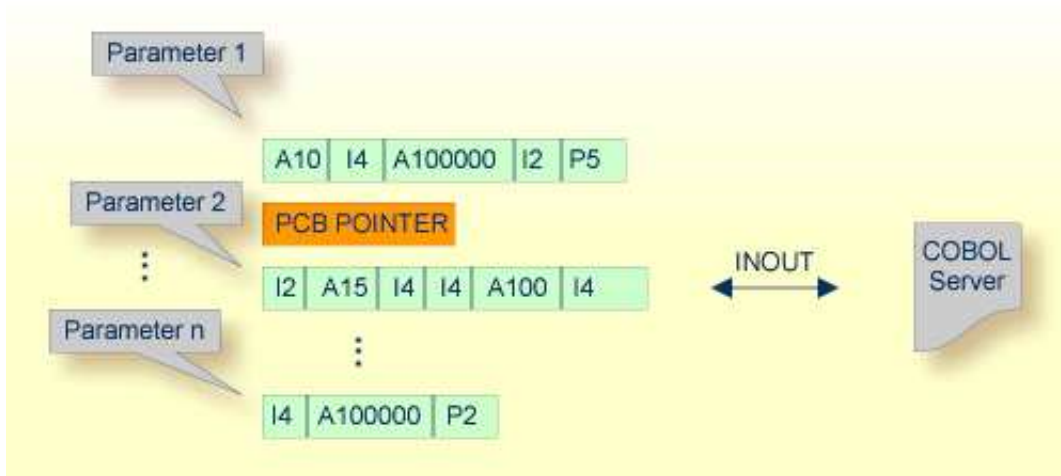
Technically, the generated COBOL server skeleton contains

- a parameter list `PROCEDURE DIVISION USING PARM1 PARM2 . . . PARMn`
- the parameters in the linkage section as COBOL data items on level 1

See *Server Interface Types* for more information on how to create COBOL servers with this interface type.

IMS BMP with Standard Linkage Calling Convention

IMS batch message processing programs (BMP) with PCB parameters are directly supported.



Technically, the generated COBOL server skeleton contains

- IMS-specific PCB *pointers* within a parameter list.

See *Server Interface Types* for more information on how to create COBOL servers with this interface type.

Compatibility between COBOL Interface Types and RPC Server

To call a server successfully, the RPC server used must support the interface type of the COBOL server. The table below gives an overview of possible combinations of an interface type and a supporting RPC server:

Interface Type	Supported by EntireX Adapter	Supported by RPC Server									
		z/OS			UNIX/Windows			BS2000/OSD	z/VSE		
		CICS	Batch	IMS	CICS ECI	Micro Focus	IMS Connect	Batch	CICS	Batch	
CICS with DFHCOMMAREA Calling Convention (Extractor Wrapper)	x	x			x					x	
CICS with DFHCOMMAREA Large Buffer Interface (Extractor Wrapper)		x								x	
CICS with Channel Container Calling Convention (Extractor Wrapper)		x									
Batch with Standard Linkage Calling Convention (Extractor Wrapper)			x	x					x		x
Micro Focus with Standard Linkage Calling Convention (Extractor Wrapper)						x					
IMS BMP with Standard Linkage Calling Convention (Extractor Wrapper)				x							
IMS MPP Message Interface (IMS Connect) (Extractor)	x							x			