

Scenarios and Programmer Information

This chapter covers the following topics:

- COBOL Scenarios
 - Aborting RPC Server Customer Code and Returning Error to RPC Client
 - Automatic Syncpoint Handling
-

COBOL Scenarios

- Scenario I: Calling an Existing COBOL Server
- Scenario II: Writing a New COBOL Server

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, also a server mapping file. See *When is a Server Mapping File Required?* in the EntireX Workbench documentation.
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* in the XML/SOAP Wrapper documentation

See *Client and Server Examples for z/VSE CICS* in the COBOL Wrapper documentation 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, also a server mapping file. See *When is a Server Mapping File Required?* in the EntireX Workbench documentation. 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* in the XML/SOAP Wrapper documentation

See *Client and Server Examples for z/VSE CICS* in the COBOL Wrapper documentation for COBOL RPC Server examples.

Aborting RPC Server Customer Code and Returning Error to RPC Client

Using EXEC CICS ABEND ABCODE

This approach applies to all CICS scenarios (all programming languages and all interface types); see *Supported Interface Types*.

The CICS feature EXEC CICS ABEND ABCODE(*myabend*) may be used to indicate application error codes. According to IBM CICS standards, ABEND codes starting with the letter A are reserved for CICS itself and should not be used in your RPC server.

The CICS RPC Server follows these IBM CICS standards and sends back the RPC protocol message

1. 10010018 Abnormal termination during program execution. This is returned when an ABEND code starting with the letter "A" is received from CICS, which is a CICS ABEND.
2. 10010045 CICS ABEND *myabend* was issued. This is returned when an ABEND code starting with a letter other than "A" is received from CICS, which is an application error situation forced by your RPC server.

Automatic Syncpoint Handling

The CICS RPC Server issues a SYNCPOINT command under the following circumstances:

- After a successful non-conversational request or an end-of-conversation, the server issues a SYNCPOINT COMMIT command. If you are running under CICS with impersonation, this SYNCPOINT command is not executed by the server, but by CICS when the user task is terminated. See *Impersonation*.
- After abnormal termination of a non-conversational request or a conversation due to an error, the server performs a SYNCPOINT ROLLBACK command to back out any pending database modifications.