Conversational RPC

RPC conversations are supported when communicating with an RPC server.

It is assumed that you are familiar with the concepts of conversational RPC and non-conversational RPC. Open and closing conversations are provided through the *Generic RPC Services Module*.

This chapter covers the following topics:

- Using Conversational RPC
- Terminating a Conversational RPC Communication
- Closing and Committing a Conversational RPC Communication

Using Conversational RPC

To use conversational RPC

1. Open a conversation with the function Open Conversation `OC` (see `COM_FUNCTION` under *RPC Communication Area*) from Generic RPC Services module:

   With the Call Interface:

```
...
ERXCOM.COM_FUNCTION = ’OC’; /* Open Conversation */
ERXCOM. COM_SERVER_LIBRARY = ’MYLIB’;
call xxxSRVI(ERXCOM); /* see (1) below */
IF SUBSTR(ERXCOM.COM_ERROR,8) ^= ERX_S_SUCCESS then
DO;
/* error handling */
        /* ... */
END;
/* begin of application logic including calls to interface objects */
...
```

   With the `EXEC CICS LINK` Interface:

```
ERXCOM.COM_FUNCTION = ’OC’; /* Open Conversation */
ERXCOM. COM_SERVER_LIBRARY = ’MYLIB’;
CICS_LEN = STORAGE(ERXCOM);
CICS_RESP1 = DFHRESP(NORMAL);
CICS_RESP2 = DFHRESP(NORMAL);
/* called CICS program name depends on PP switch ERXFCTPRE */
EXEC CICS LINK PROGRAM  (’xxxSRVI’) /* see (1) below */
               RESP     (CICS_RESP1)
               RESP2    (CICS_RESP2)
               COMMAREA (ERXCOM)
               LENGTH   (CICS_LEN);
IF SUBSTR(ERXCOM.COM_ERROR,8) ^= ERX_S_SUCCESS then
DO;
/* error handling */
        /* ... */
END;
/* begin of application logic including calls to interface objects */
...
```

[1] The prefix of the program name (xxxSRVI) can be customized, see *PL/I Preprocessor Settings*. The default is PLISRVI.

- The Open Conversation requires a library to be set in the RPC communication area field COM_SERVER_LIBRARY. See *The RPC Communication Area (Reference)*.
- After a successful Open Conversation, the broker's conversation ID is stored within the RPC communication area field COM_SERVER_CONVID. See *The RPC Communication Area (Reference)*. The conversation ID
  - is used during calls to interface objects and also needed for closing the conversation.
  - is cleared if the end of conversation is forced by the broker or the RPC server. This happens if an error with message class 0003 occurs. See *Message Class 0003 - EntireX ACI - Conversation Ended*.
  - is not cleared and remains for any other error returned to be able to continue the conversation.

2. Issue your RPC requests as is done within non-conversational mode, using the generated interface objects.

   - Different interface objects can participate in the same RPC conversation.
   - RPC conversations and simple non-conversational RPC requests can *not* be handled in parallel using the same RPC communication area without saving and restoring some fields.
   - If you need to handle RPC conversations in parallel, or simple non-conversational RPC requests within an ongoing RPC conversation, use multiple RPC communication areas or save and restore the following fields:
     - COM_BROKER_ID (if another broker)
     - COM_SERVER_CLASS (if another class)
     - COM_SERVER_NAME (if another name)
     - COM_SERVER (if another service)
     - COM_SERVER_LIBRARY (if another library)
     - COM_SERVER_CONVID
     - and possibly others, for example user ID, token and password if needed

Terminating a Conversational RPC Communication

Terminate an RPC conversation unsuccessfully with the function Close Conversation CB (see COM_FUNCTION under *RPC Communication Area*) from Generic RPC Services module:

With the Call Interface:

```
...
ERXCOM.COM_FUNCTION = 'CB'; /* Close Conversation */
call xxxSRVI(ERXCOM); /* see (1) below */
IF SUBSTR(ERXCOM.COM_ERROR,8) ^= ERX_S_SUCCESS then
DO;
/* error handling */
        /* ... */
END;
/* begin of application logic including calls to interface objects */
...
```

[1] The prefix of the program name (`xxxSRVI`) can be customized, see *PL/I Preprocessor Settings*. The default is `PLISRVI`.

With the `EXEC CICS LINK` Interface:

See *Using Conversational RPC* above.

Closing and Committing a Conversational RPC Communication

Close the RPC conversation successfully with the function Close Conversation and Commit `CE` (see `COM_FUNCTION` under *RPC Communication Area*) from Generic RPC Services module:

With the Call Interface:

```
...
ERXCOM.COM_FUNCTION = 'CE'; /* Close Conversation and Commit */
call xxxSRVI(ERXCOM); /* see (1) below */
IF SUBSTR(ERXCOM.COM_ERROR,8) ^= ERX_S_SUCCESS then
DO;
/* error handling */
        /* ... */
END;
/* begin of application logic including calls to interface objects */
...
```

[1] The prefix of the program name (`xxxSRVI`) can be customized, see *PL/I Preprocessor Settings*. The default is `PLISRVI`.

With the `EXEC CICS LINK` Interface:

See *Open Conversation* above.