

# Administering the EntireX RPC Server

The EntireX z/VSE CICS® RPC Server allows standard RPC clients to communicate with RPC servers on the operating system z/VSE under CICS. It supports the programming language COBOL. This chapter covers the following topics:

- Customizing the RPC Server
  - Configuring the RPC Server
  - Locating and Calling the Target Server
  - User Exit COBUEX02
  - Multiple RPC Servers in the same CICS
- 

## Customizing the RPC Server

The following elements are used for setting up the CICS RPC Server:

- ERXMAIN Control Block
- ERXMAIN Macro
- RPC Online Maintenance Facility

### ERXMAIN Control Block

- defines a setup of the CICS RPC Server that is persistent over CICS restarts
- is defined with parameters of the *ERXMAIN Macro*; see column 1 in the table under *Configuring the RPC Server*
- contains the following important settings:
  - connection information such as broker ID, see BKRN, server address, see CLZN, SRVN and SVCN
  - location and usage of server-side mapping container, see SVM and *Usage of Server Mapping Files*
  - scalability parameters such as *endworker*, *minworker* and *maxworker*, see ENDW, MINW and MAXW
  - etc.

### ERXMAIN Macro

- creates an *ERXMAIN Control Block*, a persistent setup of the CICS RPC Server

- needs to be assembled to define a setup
- is defined in Assembler program EMAINGEN (in sublibrary EXP960) - use this for assembling; see *Build the ERXMAIN Control Block* under *Installing EntireX RPC Servers under CICS*

## RPC Online Maintenance Facility

- provides commands (see column 2 in the table below) to vary most of the permanently defined parameters in the *ERXMAIN Control Block* currently in use. All modifications are lost if CICS is restarted. Use *ERXMAIN Macro* for permanent modifications
- allows you to try out new setups of the CICS RPC Server easily without the need to reassemble the *ERXMAIN Control Block*.
- supports
  - starting
  - stopping
  - pinging
  - monitoring
  - activating trace

of the CICS RPC Server. See *RPC Online Maintenance Facility*.

## Configuring the RPC Server

The following rules apply for the *ERXMAIN Macro* syntax (column 1 in table below):

- keywords are given in uppercase
- there are no abbreviations for keywords

The following rules apply for the RPC Online Maintenance Facility commands (column 2 in table below):

- Underscored letters in a command indicate the minimum number of letters that can be used for abbreviation.

For example, in brokerid=localhost, brok is the minimum number of letters that can be used as an abbreviation, i.e. the commands brokerid=localhost and brok=localhost are equivalents.

ERXMAIN Macro Syntax	RPC Online Maintenance Facility Commands	Default	Values	Req/Opt
BKRN	<u>brokerid</u>	ETB001	Broker ID used by the server. See <i>Using the Broker ID in Applications</i> .  Example: BKRN=myhost.com:1971	R
CLZN	<u>class</u>	RPC	Server class part of the server address used by the server. The server address must be defined as a service in the broker attribute file (see <i>Service-specific Attributes</i> under <i>Broker Attributes</i> ). Case-sensitive, up to 32 characters. Corresponds to CLASS attribute of the broker attribute file.  Example: CLZN=MyRPC	R

ERXMAIN Macro Syntax	RPC Online Maintenance Facility Commands	Default	Values	Req/Opt
SRVN	<u>server</u> name	SRV1	<p>Server name part of the server address used by the server. The server address must be defined as a service in the broker attribute file. See <i>Service-specific Attributes</i> under <i>Broker Attributes</i>. Case-sensitive, up to 32 characters. Corresponds to SERVER of the broker attribute file.</p> <p>Example: SRVN=mysrv</p>	R
SVCN	<u>service</u>	CALLNAT	<p>Service part of the server address used by the server. The server address must be defined as a service in the broker attribute file. See <i>Service-specific Attributes</i> under <i>Broker Attributes</i>. Case-sensitive, up to 32 characters. Corresponds to SERVICE attribute of the broker attribute file.</p> <p>Example: SVCN=MYSERVICE</p>	R
CODE	<u>codepage</u>	no codepage transferred	<p>Depending on the internationalization approach, the codepage (locale string) where incoming data is provided to the COBOL server. Conversely, the COBOL server must provide outgoing data in the given codepage, otherwise unpredictable results occur. See <i>What is the Best Internationalization Approach to use?</i> under <i>Internationalization with EntireX</i> for information on which internationalization approach requires a codepage (locale string).</p> <p>By default, no codepage is transferred to the broker. For the most popular internationalization approach, <i>ICU Conversion</i>, the correct codepage (locale string) must be provided. This means it must:</p> <ul style="list-style-type: none"> <li>• follow the rules described under <i>Locale String Mapping</i></li> <li>• be a codepage supported by the broker</li> <li>• be the codepage used in your environment for file and terminal IO, otherwise unpredictable results may occur.</li> </ul> <p>Example: CODE=ibm-273</p>	O
COMP	<u>compress</u> level	N	<p>Enforce compression when data is transferred between broker and server. See <i>Data Compression in EntireX Broker</i>.</p> <p>compresslevel= 0   1   2   3   4   5   6   7   8   9   Y   N</p> <p>0-9 0=no compression 9=max. compression N No compression. Y Compression level 6.</p> <p>Example: COMP=6</p>	O
CYCL	<u>restart</u> cycles	15	<p>Number of restart attempts if the broker is not available. This can be used to keep the CICS RPC Server running while the broker is down for a short time. A restart cycle will be repeated at an interval which is calculated as follows:</p> <p>timeout + ETB_TIMEOUT + 60 seconds</p> <p>where timeout is the RPC server parameter (see this table), and</p> <p>ETB_TIMEOUT is the environment variable (see <i>Environment Variables in EntireX</i>)</p> <p>When the number of cycles is reached and a connection to the broker is not possible, the RPC server stops.</p> <p>Example: CYCL=30</p>	O
DPLY	<u>deployment</u>	NO	<p>Activates the deployment service, see <i>Deployment Service</i>. Required to use the <i>Server Mapping Deployment Wizard</i>. See <i>Server Mapping Deployment Wizard</i> in the <i>EntireX Workbench</i> documentation.</p> <p><b>YES</b> Activates the deployment service. The RPC server registers the deployment service in the broker.</p> <p><b>NO</b> The deployment service is deactivated. The RPC server does not register the deployment service in the broker.</p> <p>Example: DPLY=YES</p>	O

ERXMAIN Macro Syntax	RPC Online Maintenance Facility Commands	Default	Values	Req/Opt
ENCR	encryptionlevel	0	<p>Enforce encryption when data is transferred between client and server. Requires EntireX Security. See ENCRYPTION-LEVEL under <i>Broker ACI Fields</i>.</p> <p>0 Encryption is enforced.                      1 Encryption is enforced between server and broker kernel.                      2 Encryption is enforced between server and broker kernel, and also between client and broker.</p> <p>Example: ENCR=2</p>	O
ENDW	endworker	TIMEOUT	<p><b>NEVER</b> Defines worker model <b>FIXED</b> with a fixed number of worker threads. The number of active workers is defined with ERXMAIN macro parameter <b>MINW</b>.</p> <p><b>TIMEOUT</b> Defines slow-shrinking worker model <b>SCALE</b>, where the number of worker threads is adjusted to the current number of client requests. With value <b>TIMEOUT</b>, all worker threads not used are stopped in the time specified by the ERXMAIN macro parameter <b>TOUT</b>, except for the minimum number of active workers specified with ERXMAIN macro parameter <b>MINW</b>. The upper limit of workers parallel active is restricted with ERXMAIN macro parameter <b>MAXW</b>.</p> <p><b>IMMEDIATE</b> Defines fast-shrinking worker model <b>SCALE</b>, where the number of worker threads is adjusted to the current number of client requests. With value <b>IMMEDIATE</b>, worker threads not used are stopped immediately as soon as they have finished their conversation, except for the minimum number of active workers defined with ERXMAIN macro parameter <b>MINW</b>. The upper limit of workers active in parallel is restricted with ERXMAIN macro parameter <b>MAXW</b>.</p> <p>This parameter is forced to value <b>TIMEOUT</b> if impersonation is switched on, see <i>Impersonation</i> and ERXMAIN macro parameter <b>IMPS</b>.</p> <p>Example: ENDW=IMMEDIATE, MINW=2, MAXW=6</p>	O
MINW	minworker	1	<p>Minimum limit of tasks active in parallel.</p> <ul style="list-style-type: none"> <li>For worker model <b>SCALE</b>: minimum number of workers active in parallel. Do not set a value higher than ERXMAIN macro parameter <b>MAXW</b>.</li> <li>For worker model <b>FIXED</b>: number of workers active in parallel. Do not set a value higher than 31 without adjusting ERXMAIN macro parameter <b>SIZE</b>.</li> </ul> <p>See also ERXMAIN macro parameter <b>ENDW</b>.</p> <p>Example: MINW=2</p>	O
MAXW	maxworker	10	<p>Upper limit of tasks active in parallel.</p> <ul style="list-style-type: none"> <li>For worker model <b>SCALE</b>: workers active in parallel. Do not set a value higher than 31 without adjusting ERXMAIN macro parameter <b>SIZE</b>. See also ERXMAIN macro parameter <b>ENDW</b>.</li> <li>For <i>Impersonation</i>: workers and impersonated user tasks active in parallel. Do not set a value higher than 15 without adjusting ERXMAIN macro parameter <b>SIZE</b>. See also ERXMAIN macro parameter <b>IMPS</b>.</li> </ul> <p>Example: MAXW=2</p>	O
ETBL	etblnk	BKIMC	<p>Define the broker stub to be used. See <i>Administration of Broker Stubs under z/VSE</i> for available stubs.</p> <p>Example: ETBL=BKIMC</p>	O
EXIT	n/a		<p>At startup, the CICS RPC Server will call the user exit to synchronize its version. If successful, the CICS RPC Server will continue and call the user exit for the implemented events. See <i>User Exit COBUEXO2</i>.</p>	O

ERXMAIN Macro Syntax	RPC Online Maintenance Facility Commands	Default	Values	Req/ Opt
IMPS	impersonation	NO	<p>Defines if RPC requests are executed under the user ID of the RPC client. Depending on settings, different levels of checks are done prior to RPC server execution. See also <i>Impersonation</i>.</p> <p>impersonation= <u>NO</u>   YES                        AUTO [, <u>sameuser</u>   , anyuser                      ]</p> <p><b>NO</b> The RPC request is executed anonymously, which means the user ID of the RPC client is not used. RPC requests are executed under the user ID of the RPC server.</p> <p><b>YES</b> The RPC request runs impersonated under the supplied <i>RPC client user ID</i>. For execution of the RPC request, the CICS RPC Server starts a separate impersonated user task, that is, the client must be known to CICS and the supplied password is validated against CICS. The worker model <i>SCALE</i> is forced; for details see <i>Impersonation</i>.</p> <p><b>AUTO</b> Same as option <b>YES</b> above, except that no password validation is performed, that is, the client is treated as already authenticated. For this setting, make sure the RPC client is correctly authenticated; use either</p> <ul style="list-style-type: none"> <li>• a secure broker (validation must be against the correct mainframe security repository where the user IDs are defined) and option <code>sameuser</code> or</li> <li>• your own security implementation (option <code>anyuser</code> is supported for compatibility reasons if you need different broker and server user IDs - the customer-written security implementation must validate the RPC client using the <i>RPC client user ID</i>)</li> </ul> <p><b>sameuser</b> The CICS RPC Server checks whether the <i>broker client user ID</i> matches the <i>RPC client user ID</i>. This is the default if <b>AUTO</b> is used.</p> <p><b>anyuser</b> The <i>RPC client user ID</i> is used for impersonation. The <i>broker client user ID</i> is ignored.</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. EntireX supports two user ID/password pairs: a <i>broker client user ID/password</i> pair and an (optional) <i>RPC user ID/password</i> pair sent from RPC clients to the RPC server.</li> <li>2. With EntireX Security, the <i>broker client user ID/password</i> pair is checked. The <i>RPC user ID/password</i> pair is designed to be checked by the target RPC server. Thus it is possible to use different user IDs in the broker and target RPC server.</li> <li>3. RPC clients send the (optional) <i>RPC user ID/password</i> pair in the same way as specifying the Natural user ID/password pair for a Natural RPC Server. See for example <i>Using Natural Security</i> for applications under C   COBOL   PL1   Web Services   SOAP/XML   Java.</li> <li>4. If the RPC client does not specify the optional <i>RPC user ID/password</i> pair, the <i>broker client user ID</i> is inherited to the <i>RPC user ID</i> and thus used for impersonation by the CICS RPC Server.</li> </ol> <p>Example:                      IMPS=auto</p>	O

ERXMAIN Macro Syntax	RPC Online Maintenance Facility Commands	Default	Values	Req/Opt
LOGN	<u>logon</u>	YES	Execute broker functions LOGON/LOGOFF in worker threads. Must match the setting of the broker attribute AUTOLOGON. Reliable RPC requires logon set to YES. See <i>Reliable RPC</i> .  <b>NO</b> No logon/logoff functions are executed. <b>YES</b> Logon/logoff functions are executed.  Example: LOGN=no	O
n/a	<u>mapname</u>		Alias for command <i>memory</i> .	O
n/a	<u>memory</u>		Command to load an ERXMAIN Control Block. See <i>Modifying Parameters of the RPC Server</i> .	O
OPTS	<u>runoption</u>	0	This parameter is for special purposes. It provides the CICS RPC Server with additional information. The runoptions are normally set to meet the platform's requirements. Set this parameter only if a support representative provides you with an option and asks you to do so.  Syntax: OPTS=( <option-list> ) <option-list> = [ <option-list> , ] <option>  Example: OPTS=( RUNOPT1 , RUNOPT2 )	O
PSWD	<u>password</u>		Password for broker logon. Case-sensitive, up to 32 characters. For more information see broker ACI control block field PASSWORD.  Example: PSWD=MyPwd	O
PRELOAD	<u>preload</u>	YES	Enable to call CICS RPC Server with AMODE=24  <b>YES</b> Enable to call RPC server with AMODE 24 or 31. Internally the CICS RPC Server preloads the called RPC server before execution to check the AMODE and releases the RPC server after this. The disadvantage of this approach is the CICS USECOUNT of the called RPC server program is increased by 2 for every executed RPC call.  <b>NO</b> The CICS RPC Server does not preload the called RPC server to check its AMODE. All RPC servers are called as running in AMODE 31. This option is useful for customers who require the CICS USECOUNT in their accounting (increased by 1 for every executed RPC call) but prevents usage of calling RPC Server with AMODE 24.	O
REPL	<u>replicatename</u>	ESRV	CICS transaction ID (uppercase, up to 4 characters) assigned to worker tasks and as default for user tasks if <i>Impersonation</i> is set. In the START-USER event of the user exit (see <i>User Exit COBUEX02</i> ) the CICS transaction ID for user tasks can be overridden. See also <i>Inside the RPC Server</i> .	O
SIZE	n/a	32768	Size in bytes to hold work memory for worker tasks and impersonated user tasks if impersonation is used. Each task (worker and user) requires the same amount of memory. The following rules apply when calculating the ERXMAIN macro parameter MAXW:  1. The theoretical maximum number of tasks can be calculated using the formula: maximum = integer part of ((SIZE-2036)/864-1).  2. For tasks in intermediate states (starting or ending), the theoretical maximum number must be reduced. We recommend reserving at least 10% for this purpose.  3. If impersonation is used, the theoretical maximum number must be halved.  This means: <ul style="list-style-type: none"><li>• For the default SIZE value of 32768, the theoretical maximum number of tasks (see rule 1 above) is 34 ((32768-2036)/864-1).</li><li>• Reducing this value by at least 10% (see rule 2 above) gives 31 for MAXW if no impersonation is used.</li><li>• If impersonation is used, MAXW should be no more than 15 (see rule 3 above).</li></ul>	O
SMH	<u>smhport</u>	0	The port where the server listens for commands from the System Management Hub (SMH). If this port is 0 (default), no port is used and management by the SMH is disabled.  See <i>SMH Listener Service</i> for more information.  Example: SMH=3001	O

ERXMAIN Macro Syntax	RPC Online Maintenance Facility Commands	Default	Values	Req/Opt
SVM	<a href="#">svmfile</a>		<p>Usage and location of server-side mapping files. See <i>Server-side Mapping Files in the RPC Server</i>. If no <i>SVM</i> parameter is given, the RPC server tries to open the server-side mapping container, using CICS file with name <i>ERXSVM</i>. If this CICS file is not available, no server-side mapping files are used. If you use server-side mapping files, the server-side mapping container must be installed and configured; see <i>Step 1: Define a Server-side Mapping Container - VSAMDEF.J (Optional)</i> under <i>Installing the z/VSSE EntireX RPC Servers</i>. There are also client-side mapping files that do not require configuration here; see <i>Server Mapping Files for COBOL</i>.</p> <p>Syntax:  <i>SVM</i>=NO   <i>cicsname</i></p> <p><b>cicsname</b> The RPC server tries to open the server-side mapping container using the CICS file with name <i>cicsname</i>.</p> <p><b>no</b> No server-side mapping files are used.</p> <p>Example:  <i>SVM</i>=MYSVM</p> <p>See also <i>Usage of Server Mapping Files</i>.</p>	O
TOUT	<a href="#">ttimeout</a>	600	<p>Timeout in seconds, used by the server to wait for broker requests. See broker ACI control block field <i>WAIT</i> for more information. Also influences <i>restartcycles</i>.</p> <p>See worker model <i>SCALE</i> to define the lifetime of worker threads in slow-shrinking worker model <i>SCALE</i>.</p> <p>Example:  <i>TOUT</i>=300</p>	O
TRCL	<a href="#">tracedestination</a>	CSSL	<p>Name of the destination for trace output. A valid CICS transient data queue.</p>	O
TRLV	<a href="#">tracellevel</a>	0	<p>Trace level for the server. See also <i>Activating Tracing for the RPC Server</i>.</p> <p>Syntax:  <i>TRLV</i>= <b>None</b>   <b>Standard</b>   <b>Advanced</b>   <b>Support</b></p> <p><b>None</b> No trace output.  <b>Standard</b> For minimal trace output.  <b>Advanced</b> For detailed trace output.  <b>Support</b> This trace level is for support diagnostics and should only be switched on when requested by Software AG support.</p> <p>Example:  <i>TRLV</i>=standard</p>	O
USER	<a href="#">userid</a>	ERXSRV1	<p>Used to identify the server to the broker. See broker ACI control block field <i>USER-ID</i>. Case-sensitive, up to 32 characters.</p> <p>Example:  <i>USER</i>=MyUId</p>	R

## Locating and Calling the Target Server

The IDL library and IDL program names that come from RPC client are used to locate the RPC server. See *library-definition* and *program-definition*. This two-level concept (library and program) has to be mapped to the CICS RPC Server environment.

The approach used to derive the CICS program name for the RPC server depends on whether server mapping is used or not. See *Usage of Server Mapping Files* for an introduction.

1. If the RPC client sends a client-side type of server mapping with the RPC request, this server mapping is used first.
2. If no server mapping is available from step 1 above, and if server-side type of server mapping is used, the IDL library and IDL program names are used to form a key to locate the server mapping in the server-side mapping container. If a server mapping is found, this is then used.

3. If a server mapping is available from step 1 or 2 above, the CICS program name of the RPC server is derived from this mapping. In this case the IDL program name can be different to the CICS program name if it is renamed during wrapping process (see *Customize Automatically Generated Server Names*) or during the extraction process in the *COBOL Mapping Editor*.
4. If no server mapping is used at all, the IDL program name is used as the CICS program name of the RPC server (the IDL library name is ignored).

### ➤ To use the CICS RPC Server with COBOL

1. Make sure that all CICS programs called as RPC servers
  - use an interface type supported by the CICS RPC Server for target language COBOL; see *Supported Interface Types*.
  - can be called with an EXEC CICS LINK PROGRAM
  - are accessible through the CICS RPL chain or accessible remotely using CICS DPL
2. Configure the ERXMAIN macro parameter SVM depending on whether server-side mapping files are used or not. See also *Usage of Server Mapping Files*.

See also *Scenario I: Calling an Existing COBOL Server* or *Scenario II: Writing a New COBOL Server*.

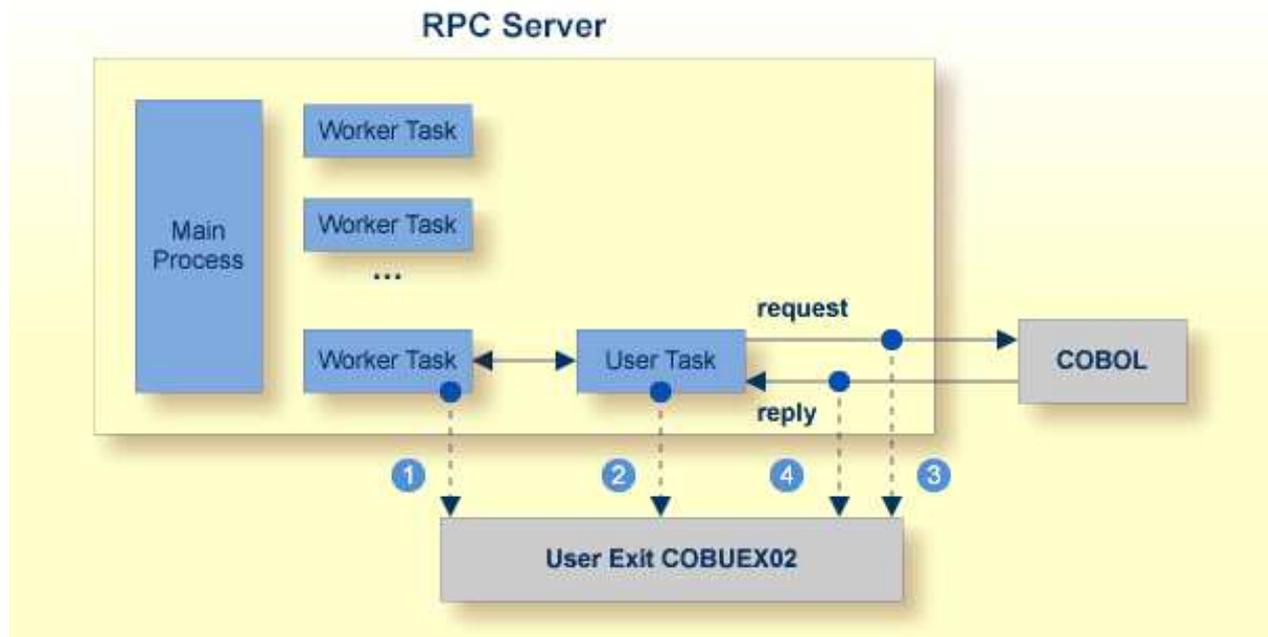
## User Exit COBUEx02

The CICS RPC Server provides a user exit COBUEx02 to influence/control the RPC logic. This section covers the following topics:

- User Exit Events
- Writing the User Exit
- Configuring the User Exit

### User Exit Events

The user exit is called on the following events:



- 1 **START-WORKER** event before a CICS worker task is started. This allows you to programmatically set the CICS transaction ID. You can terminate an RPC request by specifying an *ERROR-CODE* and optional *ERROR-TEXT*.
- 2 **START-USER** event. Before an impersonated CICS transaction (worker task) is started, the user exit may change the user ID and CICS transaction ID of the new impersonated worker. See *Impersonation*. You can terminate an RPC request by specifying an *ERROR-CODE* and optional *ERROR-TEXT*.
- 3 **CALL-START** event. The RPC request (payload data from the RPC client to the RPC server) may be inspected and modified. You can terminate an RPC request by specifying *ERROR-CODE* and optional *ERROR-TEXT*.
- 4 **CALL-END** event. The RPC reply (payload data from the RPC server to the RPC client) may be inspected and modified. If an *ERROR-CODE* and optional *ERROR-TEXT* is given in the API, this error is returned to the RPC client instead of the payload.

## Writing the User Exit

The Developer's Kit RPC source data set EXP960.SRCE of the EntireX CICS installation provides the user exit skeleton COBUEx02 for COBOL. Copy this skeleton so you have your own user exit source for modifications.

Accordingly, a COBOL copybook COBUEx02 is provided in EXP960.INCL. Please add this library to your COBOL compiler SYSLIB DD chain.

The sublibrary EXP960 of the EntireX CICS installation provides the user exit skeleton COBUEx02.C for COBOL. Copy this skeleton so you have your own user exit source for modifications.

Accordingly, a COBOL copybook COBUEx02.CPY is provided in EXP960.

Hint: The copybook extension CPY might not be understood by the z/VSE COBOL compiler. In this case, copy COBUEX02.CPY into your copybook library and rename it to COBUEX02.C. Add this library to your COBOL compiler LIBDEF chain.

The most important API parameters of the user exit are described below. Other parameters are informational and are described in the source code. The user exit program must comply with the EXEC CICS LINK PROGRAM COMMAREA conventions.

Parameter	Description
VERSION	Required for future changes. Do not change the skeleton code.
ERROR-CODE	You can terminate the current request: Any number between 1 and 9999 will cause the CICS RPC Server to stop execution of the current RPC request and pass back the given error code with message class 1022 to the RPC client. See <i>Message Class 1022 - CICS RPC Server User Exit Messages</i> . With error code 0000, the CICS RPC Server continues as normal.
ERROR-TEXT	If the error code is not zero, an error text of up to 256 characters may be applied. This is passed to the RPC client.
CICS-TRANSID	Can be applied in the event START-USER, otherwise it is informational. Apply the TRANSID that your business logic requires.
CICS-TERMID	Can be applied in the event START-USER, otherwise it is informational. In some (rare) cases, RPC server routines require a terminal ID. Apply the TERMID that your business logic requires.
USERID	Can be applied in the event START-USER otherwise it is informational. Under some circumstances, it might be necessary to change the original RPC-USERID from the calling RPC client.
DATA-POINTER	This pointer refers to the payload data for the events CALL-START and CALL-END. The payload to which this pointer is pointing may be inspected as well as modified. The pointer itself must not be changed.

## Configuring the User Exit

Apply the name of your exit routine to the EntireX RPC server ERXMAIN macro parameter EXIT. See *Configuring the RPC Server*.

At startup, the CICS RPC Server will call the named user exit to synchronize its version. If successful, the *RPC Online Maintenance Facility* will display the user exit as map field "parameter opts". See *To display the Server parameters (PF06) under RPC Online Maintenance Facility*. The CICS RPC Server will continue and call the user exit for the implemented events.

## Multiple RPC Servers in the same CICS

If you need to install multiple instances in the same CICS region, see *Installing Multiple EntireX RPC Servers in the same CICS (Optional)* under *Installing EntireX RPC Servers under CICS* in the z/VSE installation documentation.