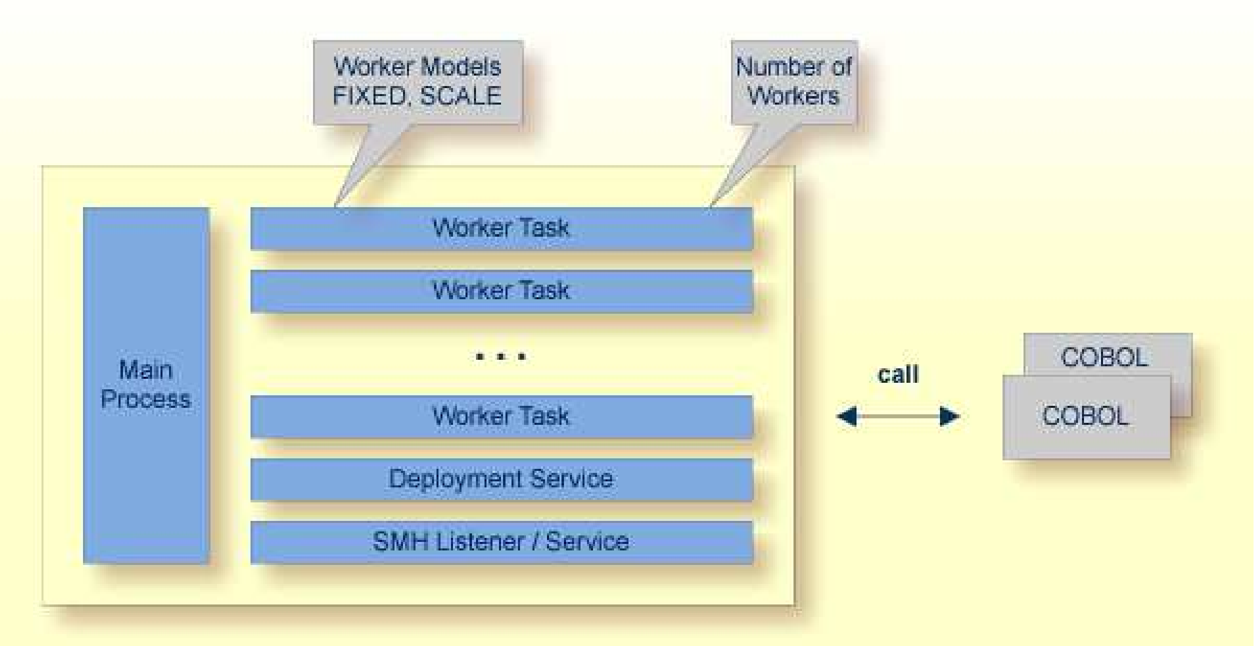
Inside the RPC Server

The EntireX z/OS CICS® RPC Server allows standard RPC clients to communicate with RPC servers on the operating system z/OS under CICS. It supports the programming languages COBOL and PL/I.

This chapter covers the following topics:

- Worker Models
- Inbuilt Services
- User Exit COBUEX02
- Impersonation
- Usage of Server Mapping Files
- Supported Interface Types

Worker Models



RPC requests are worked off inside the RPC server in worker tasks, which are controlled by a main task. Every RPC request occupies during its processing a worker task. If you are using RPC conversations, each RPC conversation requires its own task during the lifetime of the conversation. The CICS RPC Server provides two worker models:

- **FIXED**
  The *fixed* model creates a fixed number of worker tasks. The number of worker tasks (defined with `ERXMAIN` macro parameter `MINW`) does not increase or decrease during the lifetime of an RPC server instance. It is configured by setting the `ERXMAIN` macro parameter `ENDW` to value `"NEVER"`.
  Example:

```
ENDW=NEVER, MINW=4
```

- **SCALE**
  The *scale* model creates worker tasks depending on the incoming load of RPC requests.

  A maximum number (`ERXMAIN` macro parameter `MAXW`) of the worker tasks created can be set to restrict the system load. The minimum number (`ERXMAIN` macro parameter `MINW`), allows you to define a certain number of tasks - not used by the currently executing RPC request - to wait for new RPC client requests to process. In this way the RPC server is ready to handle many RPC client requests arriving at the same time. It is configured by setting the `ERXMAIN` macro parameter `ENDW` to value `"TIMEOUT"` or `"IMMEDIATE"`.

  - With value `IMMEDIATE`, worker tasks shrink fast, that is, worker tasks not used are stopped immediately as soon as it has finished its conversation, except for the number of workers specified as minimum being active.

  - With value `TIMEOUT`, worker tasks shrink slowly, that is, all worker tasks not used are stopped in the time specified by the `ERXMAIN` macro parameter `TOUT`, except for the number of workers specified as minimum being active.

  Example:

  ```
  ENDW=IMMEDIATE, MINW=2,MAXW=6
  ```

Inbuilt Services

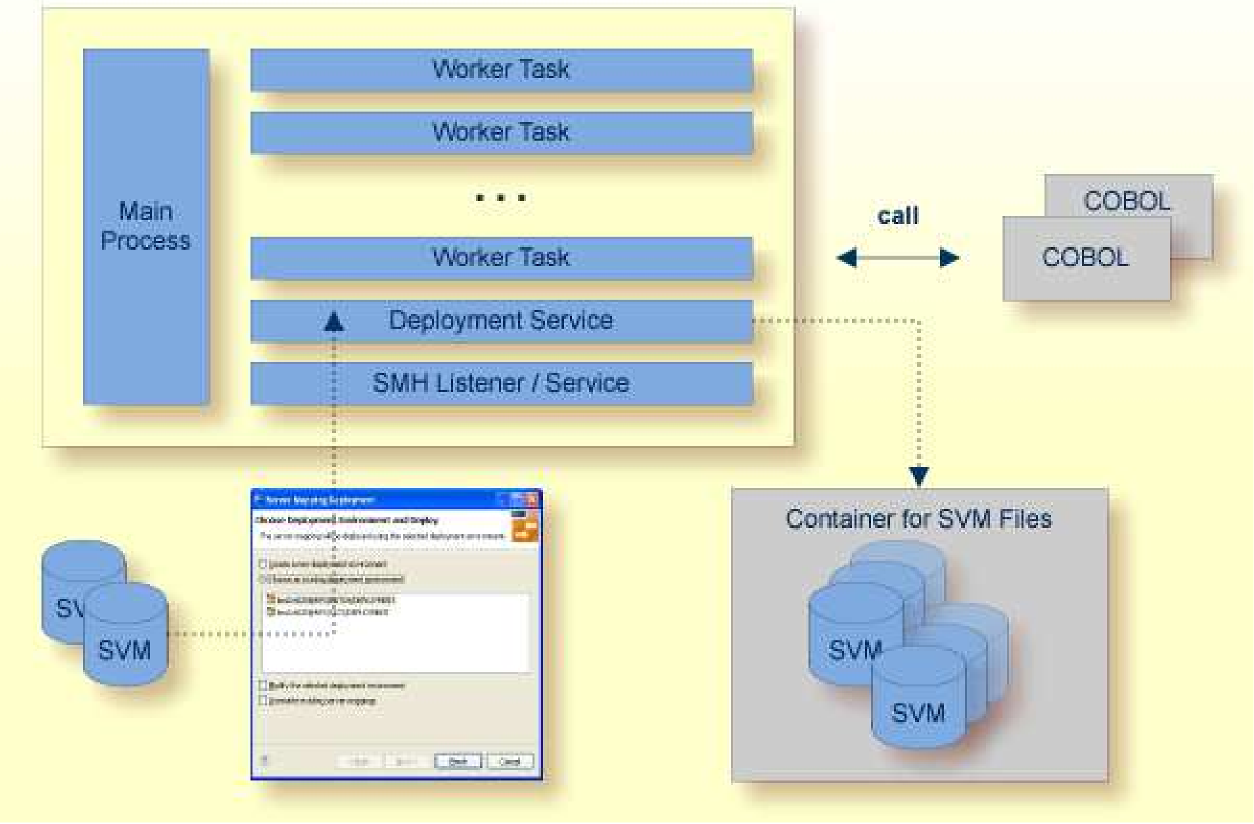
CICS RPC Server provides the following services for ease-of-use:

- Deployment Service

- SMH Listener Service

Deployment Service

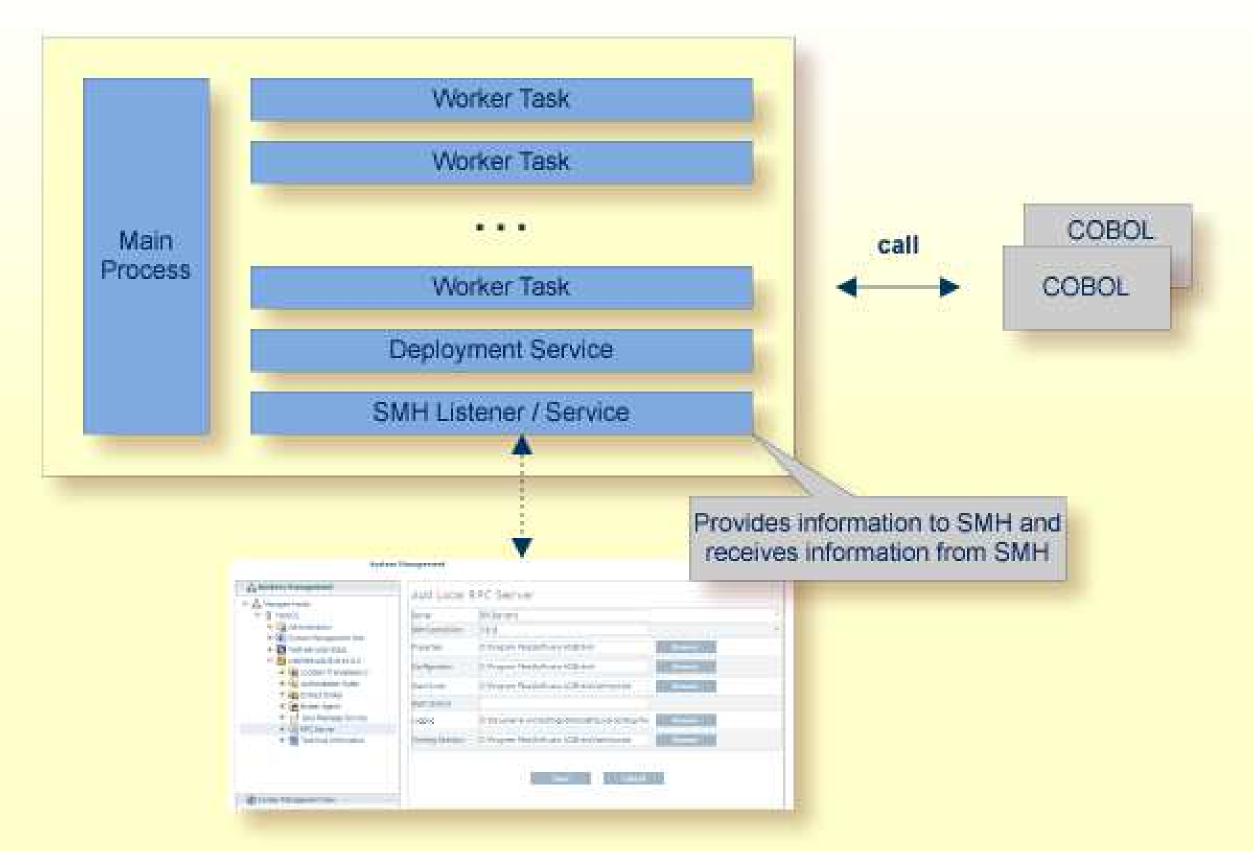
The Deployment Service allows you to deploy server-side mapping files (EntireX Workbench files with extension .svm) interactively using the *Server Mapping Deployment Wizard*. On the RPC server side, the server-side mapping files are stored in a server-side mapping container (VSAM file). See *Server-side Mapping Files in the RPC Server* and *Deployment Service* for configuration information.



SMH Listener Service

With the SMH Listener Service you use the System Management Hub to monitor the RPC server. See *Administering the EntireX RPC Servers using System Management Hub* under UNIX | Windows.

The SMH Listener Service is switched on if the SMH port number is set. See the `ERXMAIN` macro parameter `SMH` under *Configuring the RPC Server*.



User Exit COBUEX02

The CICS RPC Server provides a user exit `COBUEX02`to influence/control the RPC logic. The exit is called on the events `START-WORKER`, `START-USER`, `CALL-START` and `CALL-END`. The following tasks can be performed:

① `WHICH-VERSION` event. Tells the CICS RPC Server the version to use.

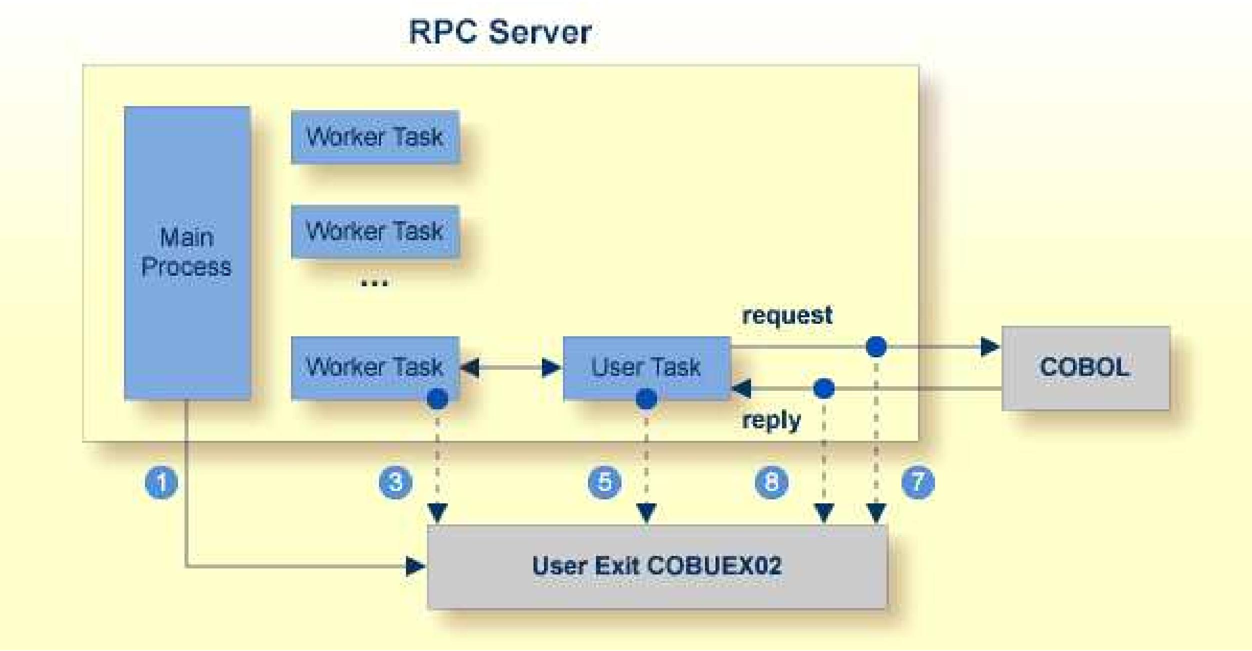
③ `START-WORKER` event. Allows you to set the CICS transaction ID.

⑤ `START-USER` event. Apply user ID, CICS transaction ID and CICS terminal ID to impersonated user tasks. See *Impersonation*.

⑦ `CALL-START` event. Inspect, modify or terminate the RPC request (payload) from the RPC client.

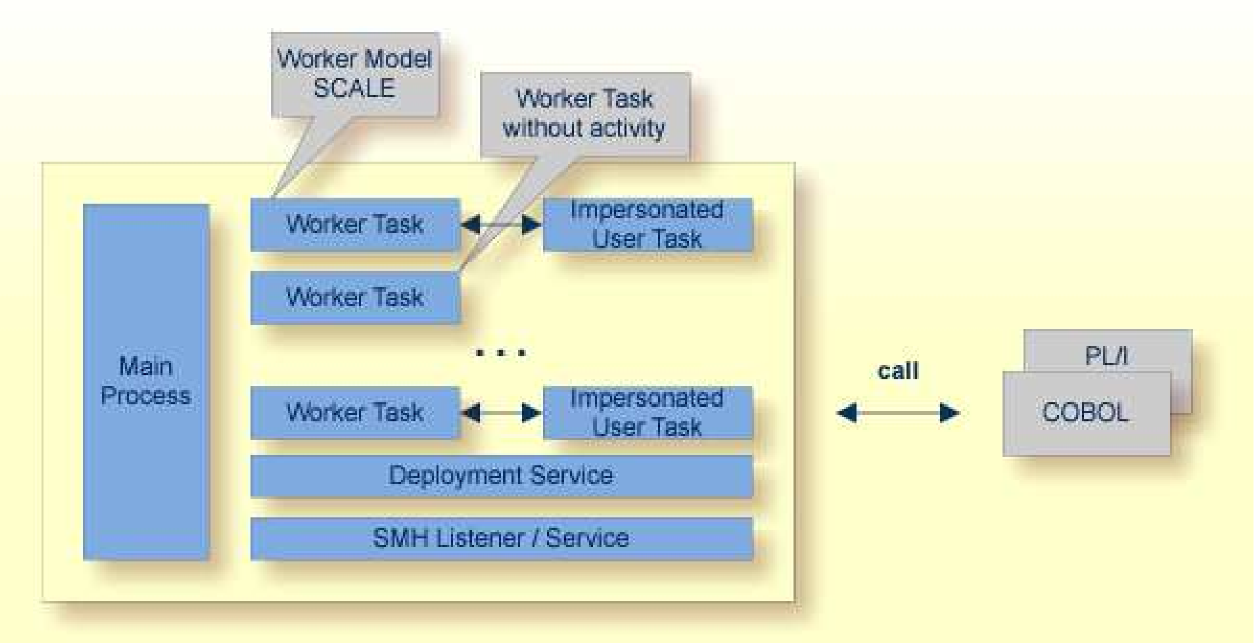
⑧ `CALL-END` event. Inspect or modify the RPC reply (payload) or give an error to the RPC client.

The numbers in the graphic correspond to the event numbers in the user exit.



See also *User Exit `COBUEX02`* under *Administering the EntireX RPC Server*.

Impersonation



The CICS RPC Server can be configured to execute the RPC request impersonated under the RPC client user ID. For this, worker tasks start additional impersonated user tasks. This can be useful, for example for accounting. Impersonation is controlled by the `ERXMAIN` macro parameter `IMPS`.

- For `IMPS` value `AUTO`, the CICS RPC Server does not validate RPC passwords, so you have to take care the RPC client is correctly authenticated, either by using a secure EntireX Broker (validation must be against the correct mainframe security repository where CICS user IDs are defined) or with your own security implementation.

- For `IMPS` value `YES`, the CICS RPC Server uses the RPC user ID and password supplied by the RPC client for authentication and impersonation of the client. This means that the RPC server validates the password.

The picture above shows the configuration `IMPS=YES`.

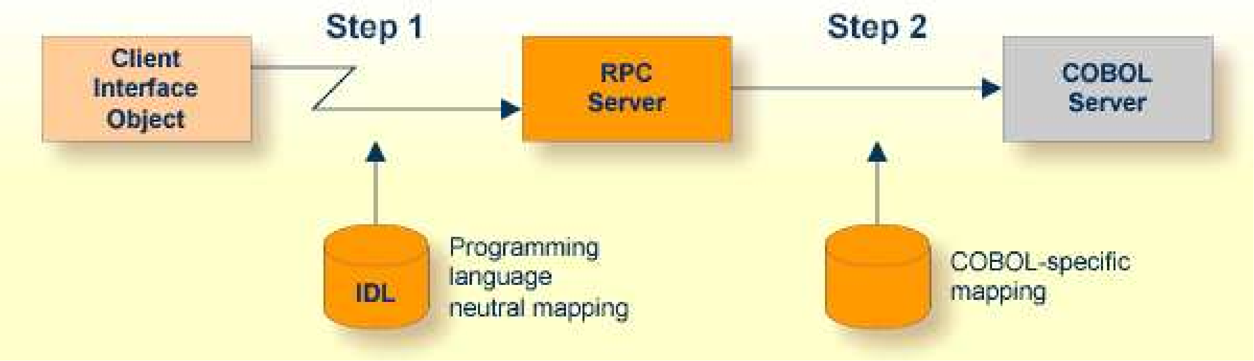
The lifetime of an impersonated user task starts when an open request for an RPC conversation or a non-conversational RPC request is received. It ends when the RPC conversation stops (after a commit operation or timeout) or when the non-conversational RPC request has been performed.

For worker tasks, the slow-shrinking worker model `SCALE` is used - value `TIMEOUT` is forced internally - any value given in the `ERXMAIN` macro parameter `ENDW` is ignored. The lifetime of worker tasks can be controlled with `ERXMAIN` macro parameter `TOUT` as well as the number of workers with macro parameters `MINW` and `MAXW`.

Usage of Server Mapping Files

There are many situations where the CICS RPC Server requires a server mapping file to correctly support special COBOL syntax such as `REDEFINES`, `SIGN LEADING` and `OCCURS DEPENDING ON` clauses, `LEVEL-88` fields, etc.

Server mapping files contain COBOL-specific mapping information that is not included in the IDL file, but is needed to successfully call the COBOL server program.



The RPC server marshals the data in a two-step process: the RPC request coming from the RPC client (Step 1) is completed with COBOL-specific mapping information taken from the server mapping file (Step 2). In this way the COBOL server can be called as expected.

The server mapping files are retrieved as a result of the *IDL Extractor for COBOL* extraction process and the *COBOL Wrapper* if a COBOL server is generated. See *When is a Server Mapping File Required?*.

There are *server*-side mapping files (*EntireX Workbench* files with extension .svm) and *client*-side mapping files (Workbench files with extension .cvm). See *Server Mapping Files for COBOL* and *How to Set the Type of Server Mapping Files*.

If you are using server-side mapping files, you need to customize the server-side mapping container with `ERXMAIN` macro parameter `SVM`. See *Configuring the RPC Server*.

Note:
Server mapping files are used for COBOL only.

Supported Interface Types

The interface types supported by the CICS RPC Server vary depending on the target programming language. See also *Locating and Calling the Target Server*.

COBOL

- *CICS with `DFHCOMMAREA` Calling Convention* (COBOL Wrapper | Extractor)
- *CICS with Channel Container Calling Convention* (COBOL Wrapper | Extractor)
- *CICS with `DFHCOMMAREA` Large Buffer Interface* (COBOL Wrapper | Extractor)

PL/I

- *CICS with `DFHCOMMAREA` Calling Convention* (PL/I Wrapper | Extractor)