Setting Up a Natural RPC Environment

To set up a Natural RPC environment, you must perform the steps described below for all client Naturals and server Naturals and read the platform-specific notes and considerations.

- Setting Up a Natural Client

- Setting Up a Natural Server

- Setting Up an EntireX Broker Access

- Setting Up an EntireX Broker Environment

Setting Up a Natural Client

Unless otherwise noted, this instruction applies to all environments.

To set up a Natural client, perform the following steps:

- Define the Server Name

- Generate a Stub Subprogram

- Set the RPC Client-Specific Natural Parameters

Define the Server Name

Use the *Service Directory Maintenance* function of the `SYSRPC` utility to define the name of the server to be used for each `CALLNAT` statement to be executed remotely.

For details and example screens, refer to *Service Directory Concept* (in the `SYSRPC` utility documentation).

The generated directory subprogram `NATCLTGS` must be made available to the Natural client application. If you have not generated `NATCLTGS` in your client library, move `NATCLTGS` to this library or to one of the steplibs.

Optionally, you can use one of the following server selection techniques:

- **Address a default server**

  See *Specifying a Default Server Address within a Natural Session*, or profile parameter `DFS`.

- **Use a remote directory server**

  See Using a Remote Directory Server, or profile parameter `RDS`.

Note for Windows, UNIX and OpenVMS Environments:

Predict servers are not maintained in the SYSRPC utility.

For information on how to connect to a Predict server, see the profile parameter USEDIC or the *Dictionary Server Assignments* function in the Global Configuration File.

Generate a Client Stub Subprogram

This step applies only if you do not want to or cannot work with automatic Natural RPC execution (see *Operating a Natural RPC Environment, Working with Automatic Natural RPC Execution*).

For each CALLNAT statement to be executed remotely, use the Stub Generation function of the SYSRPC utility; see *Creating Stub Subprograms*.

Note that the generated stub must be made available to the Natural client environment. If you did not generate the stub subprogram (interface object) in your client library, move the stub subprogram to this library or to one of the steplibs.

Set the RPC Client-Specific Natural Parameters

Set the Natural profile parameters which are relevant to the client-specific handling of remote procedure calls.

Mandatory Parameters:

Parameter	Function
MAXBUFF	Maximum buffer size (for automatic RPC execution only)
RPCSIZE	Size of buffer used by Natural RPC (for mainframe clients only)

Optional Parameters:

Parameter	Function
ACIVERS	Define ACI version for use with EntireX Broker ACI
AUTORPC	Automatic Natural RPC execution
COMPR	Set RPC buffer compression See also *Operating a Natural RPC Environment*, *Using Compression*.
CPRPC	Define code page name
DFS	Specify RPC client's default server address
RDS	Define remote directory server
RPCSDIR	Specify name of Natural library in which the Service Directory is located (for mainframe, UNIX and OpenVMS servers only)
TIMEOUT	Wait time for RPC server response
TRYALT	Try alternative server address

The following notes apply to the use of the EntireX Broker.

Notes:

1. 1. The names specified with the `DFS` parameter must identify an active EntireX Broker and must match a server definition in the EntireX Broker Attribute File, see *Setting Up an EntireX Broker Environment*.

2. 2. The wait time specified with `TIMEOUT` is used to set the `WAIT` field of the EntireX Broker ACI. If `TIMEOUT` is set to zero, `WAIT=YES` is set and the client will wait for the `CLIENT-NONACT` time. If the wait time has elapsed, the remote procedure call is terminated with a corresponding error message. The use of `TIMEOUT` enables you to take advantage of the transport timeout mechanism provided by the EntireX Broker stubs.

Setting Up a Natural Server

A Natural server is a Natural task (server task) that can execute Natural subprograms (services). This Natural task is typically an asynchronous or background task (detached process). The EntireX Broker and the client identify it by using a *nodename* and a *servername*.

To set up a Natural server perform the steps described below:

- Set the RPC Server-Specific Natural Parameters

- Ensure Command Mode Usage in Server Session

- Ensure Unique Adabas ETID Usage

- Start a Natural Server

Set the RPC Server-Specific Natural Parameters

Set the platform-dependent Natural parameters which are relevant to the general and server-specific handling of remote procedure calls for the server Natural.

For Mainframe Servers:

1. Create an RPC-specific Natural parameter module.

2. Set the keyword subparameters of profile parameter `RPC` or parameter macro `NTRPC` (see table below) as desired.

For Windows, UNIX or OpenVMS Servers:

1. Create an RPC-specific Natural parameter file.

2. Set the Natural profile parameters (see table below) as desired.

Mandatory Parameters:

Parameter	Function
MAXBUFF	Maximum buffer size
RPCSIZE	Size of buffer used by Natural RPC (for mainframe servers only)
SERVER	Start Natural session as an RPC server session
SRVNAME	Name of RPC server; see *Note for EntireX Broker* below.
SRVNODE	Name of node; see *Note for EntireX Broker* below.

Optional Parameters:

Parameter	Function
ACIVERS	Define ACI version for use with EntireX Broker ACI
CPRPC	Define code page name
LOGONRQ	Logon for RPC server request required
NTASKS	Minimum and maximum of the number or server replicas (for mainframe servers only)
SRVCMIT	Time at which a Natural RPC server automatically commits an RPC conversation or a non-conversational RPC request
SRVTERM	Server termination event
SRVUSER	User ID for RPC server registry
SRVWAIT	Wait time of RPC server for client request
TRACE	Define trace level for Natural RPC servers
TRANSP	Server transport protocol (no longer required)

The following notes apply to the use of the EntireX Broker.

Notes:

1. The name specified with SRVNODE must identify an active EntireX Broker and the name specified with SRVNAME must match a server definition in the EntireX Broker Attribute File, see *Setting Up an EntireX Broker Environment*.
2. The wait time specified with the SRVWAIT parameter is used to set the WAIT field of the EntireX Broker ACI. If SRVWAIT is not specified or set to zero, WAIT=YES is set and the server will wait for the SERVER-NONACT time. If the wait time has elapsed, a corresponding message is written to the RPC server trace file and the RPC server continues to wait for the next client request. The use of the SRVWAIT parameter enables you to take advantage of the transport timeout mechanism provided by the EntireX Broker stubs.

Ensure Command Mode Usage in Server Session

▶ **To ensure that your Natural server session will enter command mode**

- disable Natural menu mode by setting the Natural profile parameter MENU=OFF (applies to mainframe servers only).

⚠️ **Do not:**

- put a program onto the Natural stack which never terminates.

- use a `STARTUP` program which never terminates.

- disallow `NEXT` mode in Natural Security for your server library.

Ensure Unique Adabas ETID Usage

Ensure that the Adabas `ETID` used by the Natural server session is unique within a certain Adabas nucleus.

Start a Natural Server

To start a Natural server, proceed as described in the section *Starting a Natural RPC Server*.

This server then waits for remote `CALLNAT` requests from a client.

Note for Natural in Batch Mode on z/OS or on z/VSE:

For information about servers using the keyword subparameter `NTASKS` of profile parameter `RPC` or parameter macro `NTRPC`, refer to *Considerations for Mainframe Natural RPC Servers with Replicas*.

Setting Up an EntireX Broker Access

To set up an EntireX Broker interface, perform the steps described below:

- Provide Access to the EntireX Broker Stub

- Set the ACI Version

- Using TCP/IP as Transport Method

Provide Access to the EntireX Broker Stub

Make the EntireX Broker stub accessible to your Natural environment. This step depends on the platform used.

- Providing Access to the EntireX Broker Stub on Mainframe
- Providing Access to the EntireX Broker Stub on UNIX
- Providing Access to the EntireX Broker Stub on Windows

Providing Access to the EntireX Broker Stub on Mainframe

Link the EntireX Broker stub `NATETB23` to your Natural or specify the profile parameter `RCA=BROKER` to load `NATETB23` dynamically at run-time.

In the following cases `NATETB23` cannot be used and you must use a different Broker stub:

- If you want to use the TCP/IP protocol under BS2000/OSD, you must use BKIMBTIA instead.

- If you want to use impersonation in z/OS batch mode, you must use BKIMBTSO instead.

- If you want to use impersonation under CICS, you must use CICSETB instead.

     Note:
     You must link CICSETB to your Natural CICS interface nucleus.

To load BKIMBTIA or BKIMBTSO dynamically at run-time, specify RCA=BROKER
RCALIAS=(BROKER,*stubname*).

It is currently not possible to load CICSETB dynamically at run-time.

Refer to the EntireX Communicator documentation for further details.

Providing Access to the EntireX Broker Stub on UNIX

Copy the Natural-specific broker stub natetb.so / natetb.sl from the directory
$EXXDIR/$EXX/VERS/lib to the directory specified with NATEXTLIB in the Local Configuration
File *NATURAL.INI*.

Providing Access to the EntireX Broker Stub on Windows

The EntireX Broker stub is made available automatically in the course of the EntireX installation.

Set the ACI Version

Set the profile parameter ACIVERS according to your requirements.

Note:
The ACIVERS value set in the Natural parameter module (mainframe) or parameter file (Windows, UNIX
or OpenVMS) can only work if also the EntireX Broker and the EntireX Broker stub support this version.

The table below contains only those ACIVERS values which are associated with a feature that is relevant
for the Natural RPC.

Setting	Function
ACIVERS=2	(Default) Support of the EntireX Broker functions LOGON and LOGOFF. The server performs a LOGON to the EntireX Broker before executing the REGISTER, and a LOGOFF after the DEREGISTER. This does not imply any security checks, but it is a pure EntireX Broker management function, see EntireX Broker function LOGON in the EntireX Broker documentation.

Setting	Function
ACIVERS=3	Support of EntireX Broker non-numeric conversation IDs and data volume > 30 KB. When ACIVERS is set to 3 or higher, the EntireX Broker will also assign non-numeric conversation IDs. If a Natural client issues an OPEN CONVERSATION statement and the client's ACIVERS is 3 or higher, the EntireX Broker will be able to automatically assign non-numeric conversation IDs. It will not check whether the associated server does accept non-numeric conversation IDs, but only the ACIVERS of the requestor (a Natural client in this case) will be decisive. Therefore, make sure that both the Natural client and the server support the corresponding ACI version. In addition, with EntireX Broker ACI Version 3 or higher, the data volume which can be exchanged between client and server in a single request may exceed 30 KB if transport method TCP/IP is used. **Notes:** 1. With EntireX Broker ACI Version 1 or 2, the data volume is limited to 30 KB. 2. With transport method NET, the EntireX Broker attribute EXTENDED-ACB-SUPPORT must be set to YES in order to support more than 30 KB.
ACIVERS=4	Support of code pages and (for servers only) Natural Security. With EntireX Broker ACI Version 4 or higher, the Natural RPC supports code pages. For this, the name of the code page can be specified in the profile parameter CPRPC for clients and servers. The evaluation of the code page is done by the EntireX Broker. The EntireX Broker translates the RPC data sent according to the code page of client and server to the corresponding target code page. The profile parameter CPRPC can be set on the client and/or on the server. It applies for the current process. This means that the client code page does not need to be identical with the server code page. The server is enabled to logon to the EntireX Broker using a qualified user ID. If the profile parameter SRVUSER is set to *NSC and the server is running under Natural Security, the Natural RPC will automatically pass the current Natural user ID (as contained in system variable *USER) and the password defined in Natural Security to the EntireX Broker, where they are checked for conformity with the EntireX Broker security data.
ACIVERS=6	Support of EntireX Encryption (ENCRYPTION-LEVEL). With EntireX Broker ACI Version 6 or higher, the application programming interface USR4009N may be used to set the ACI field ENCRYPTION-LEVEL.

Setting	Function
`ACIVERS=7`	Support of EntireX compression (`COMPRESSLEVEL`). With EntireX Broker ACI Version 7 or higher, the application programming interface `USR4009N` may be used to set the ACI field `COMPRESSLEVEL`.
`ACIVERS=8`	Support of EntireX Security without stub exits (mainframe only). With EntireX Broker ACI Version 8 or higher, the Natural RPC server issues a `KERNELVERS` call to get the correct value for the ACI field `KERNELSECURITY`. In this case it is no longer required to link the EntireX Security exits to the EntireX Broker stubs.
`ACIVERS=9`	Support of EntireX application identification of the client and server environment and (for servers only) of the Integrated Authentication Framework (IAF). With EntireX Broker ACI Version 9 or higher, the EntireX Broker stubs send environmental information about client and server (for example, job name) to the EntireX Broker. With EntireX Broker ACI Version 9 or higher, the Natural RPC server can optionally use IAF for client authentication. For details, refer to the current EntireX documentation.

Using TCP/IP as Transport Method

If TCP/IP is used as transport method and you use a host name to address the server node, you have the following alternatives:

- Define the server node in the Hosts and Services Directory of your TCP/IP installation.

- Use a Domain Name System (DNS) for domain name resolution.

Setting Up an EntireX Broker Environment

In the EntireX Broker Attribute File, add the following:

1. For each Natural RPC server, a service definition must be specified as follows:

   `CLASS=RPC, SERVICE=CALLNAT, SERVER=servername.`

2. If you want to use the conversion services, set `CONVERSION=userexit`. In this case, you must set the profile parameter `CPRPC` accordingly.

If you want to use the reliable RPC, additional settings are required for each Natural RPC server that should support reliable RPC:

- The EntireX Broker attribute `MAX-UOWS` must be set to value greater zero.

- The EntireX Broker attribute `DEFERRED` must be set to `YES` if the client should be able to send reliable RPC messages to an RPC server that is known to the EntireX Broker but has not yet been started.

- The EntireX Broker attribute `STORE` must be set to `BROKER` if recovery of reliable RPC messages after a system failure should be possible. In addition, the EntireX Broker persistent store must be enabled.

- The lifetime of the reliable RPC message itself (EntireX Broker attribute `UWTIME`) and the lifetime of its status (EntireX Broker attribute `UWSTAT-LIFETIME`) must be adapted to your needs.

Note:
If `AUTOLOGON=NO` or `SECURITY=YES` is set in the EntireX Attribute File, you must set `ACIVERS=2` or higher.