Starting a Natural RPC Server

This section describes how to start a Natural RPC server on the different platforms.

The following topics are covered:

- Preliminaries before Starting a Natural RPC Server

- Starting a Natural RPC Server in a Mainframe Online Environment (all TP Monitors)

- Starting a Natural RPC Server in a Mainframe Online Environment (Com-plete and CICS only)

- Starting a Batch Server in a Mainframe Environment

- Starting a Natural RPC Server in a Windows Environment

- Starting a Natural RPC Server in a UNIX Environment

- Starting a Natural RPC Server in an OpenVMS Environment

- Considerations for Mainframe Natural RPC Servers with Replicas

- Starting a Natural RPC Server Using the RPC Server Front-End (z/OS Batch Mode only)

- Starting a Natural RPC Server Using the RPC Server Front-End (CICS only)

Preliminaries before Starting a Natural RPC Server

Any kind of Natural session can be used as a Natural RPC server, but typically a Natural server is a Natural session which is started as an asynchronous or as a background task.

On Mainframes:

For the purpose of starting a server, you have the following options:

- Create an RPC-specific Natural parameter module.

  For a list of the relevant parameters, refer to the section *Setting Up a Natural RPC Environment*, *Set the RPC Server-Specific Natural Parameters*.

  This parameter module is either specified dynamically with `PARM=`*serverparm*, where *serverparm* is the name of the parameter module linked to your Natural.

- Alternatively, you can also specify the profile parameters dynamically.

  The RPC-specific Natural profile parameters may be specified in a profile created with the `SYSPARM` utility. Natural would then be started with

  `PROFILE=`*serverprofile*

where *serverprofile* is the name of the profile.

On Windows, UNIX or OpenVMS:

For the purpose of starting a server, you have the following options:

- Create an RPC-specific Natural parameter file.

  For a list of the relevant parameters, refer to the section *Setting Up a Natural RPC Environment*, *Set the RPC Server-Specific Natural Parameters*.

  This parameter file is specified dynamically with PARM=*serverparm*, where *serverparm* is the name of the parameter file.

- Alternatively, you can specify the profile parameters dynamically.

How a Natural server is started depends on the environment and is described in the corresponding paragraphs below.

Starting a Natural RPC Server in a Mainframe Online Environment (all TP Monitors)

To start a Natural server in a mainframe online environment, enter the following command in your TP monitor environment:

```
<natural>
  RPC=(SERVER=ON, SRVNAME=servername, SRVNODE=nodename,
               RPCSIZE=n, MAXBUFF=n)
```

Where `<natural>` is the name with which you start your Natural (transaction code, transaction ID, environment-dependent nucleus name).

Starting a Natural RPC Server in a Mainframe Online Environment (Com-plete and CICS only)

In a Com-plete or CICS environment, you have the following options to start a Natural RPC server:

- You can use the same command as described above for all TP monitors.

- You can use the Natural program STARTSRV in library SYSRPC to start a Natural server in asynchronous mode.

- You can start a Natural RPC server in asynchronous mode during startup of your TP monitor.

Starting a Natural RPC Server in Asynchronous Mode with STARTSRV

STARTSRV is a sample front-end for RPCSSRV that starts the asynchronous Natural session.

By default, the asynchronous Natural is started with the same Natural name in the same library as the current session.

If Natural Security Security (NSC) is used, the user ID of the current Natural session is propagated, too. You may adapt the input to your requirements.

Note that some Natural profile parameters are implicitly added by RPCSSRV. This applies especially to the keyword subparameter RPCSIZE of profile parameter RPC or parameter macro NTRPC. RPCSIZE defaults to MAXBUFF+4, where MAXBUFF is the value entered in the map field **Receiving buffer**. You may overwrite the default value for RPCSIZE by entering RPC=(RPCSIZE=n) in the map field **Session parameter**.

If you want to execute a Natural program during startup of the Natural session, you may use the map field **User Stack**. The content of **User Stack** is put on the Natural STACK and executed before the Natural server is activated.

To show the Natural profile parameters involved, enter *SHOW* in the **Transaction ID** field. STARTSRV will show you all dynamic profile parameters that will be used by RPCSSRV to start the asynchronous Natural session.

See also *Note Concerning CICS* below.

Starting a Natural RPC Server Session in Asynchronous Mode during Startup of TP Monitor

To start a Natural RPC server session in asynchronous mode during startup of your TP monitor, proceed as follows:

- **Under Com-plete:**

  Use the startup option (sysparms) STARTUPPGM to start a Natural session with all required RPC specific Natural profile parameters.

- **Under CICS:**

  Use the PLTPI to start a program that uses EXEC CICS START to start a Natural session with all required RPC specific Natural profile parameters. You may adapt the sample PLTPI program XNCIFRNP that is provided in the Natural CICS source library.

Please note that in both cases the Natural session is started under a user ID that is assigned by the TP monitor:

- Under Com-plete, this user ID is the user ID under which Com-plete is started.

- Under CICS, this user ID is the CICS default user ID (whose default is CICSUSER).

In both cases, this user ID is assigned to the Natural system variables *INIT-USER and *USER. If your Natural session is running under Natural Security, you may therefore have to put a Natural LOGON command on the Natural STACK.

Note Concerning CICS:

It is recommended to use the following settings of the Natural profile parameters TTYPE and SENDER when starting the asynchronous Natural RPC server:

```
TTYPE=ASYL,SENDER=CSSL
```

This will cause each output to the primary output destination to be written in line mode rather than in 3270 mode. Instead of `CSSL`, you may use any other CICS output destination.

The use of `TTYPE=ASYL` requires that `NATBTCH` is linked to your Natural nucleus.

Starting a Batch Server in a Mainframe Environment

A batch server is a standard Natural batch session that is started with the RPC parameters described in the section *Setting Up a Natural RPC Environment*, *Set the RPC Server-Specific Natural Parameters*.

The following topics are covered below:

- Starting a Batch Server under z/OS

- Starting a Batch Server under z/VSE

- Starting a Batch Server under BS2000/OSD

Note:
For a sample JCL using the trace facility, refer to *Operating a Natural RPC Environment*, *Using the Server Trace Facility*.

Starting a Batch Server under z/OS

Sample JCL for z/OS

```
//NATRPC   JOB  CLASS=K,MSGCLASS=X
//         EXEC PGM=NATOS,REGION=8M
//STEPLIB  DD   DISP=SHR,DSN=SAG.NAT.LOAD
//         DD   DISP=SHR,DSN=SAG.EXX.LOAD
//         DD   DISP=SHR,DSN=SAG.ADA.LOAD          <== Note 1
//         DD   DISP=SHR,DSN=DB2_load_library      <== Note 2
//         DD   DISP=SHR,DSN=SAG.SSX.LOAD          <== Note 3
//CMPRMIN  DD   *
IM=D,MADIO=0,MT=0,OBJIN=R,AUTO=OFF,MAXCL=0,ID=',',INTENS=1,
RPC=(SERVER=ON,SRVNAME=servername,SRVNODE=nodename)
RPC=(RPCSIZE=m,MAXBUFF=n),
STACK=(LOGON serverlibrary,userID,password)
/*
//CEEOPTS  DD *                                    <== Note 4
POSIX(ON)
/*
//SYSUDUMP DD SYSOUT=X
//CMPRINT  DD SYSOUT=X
/*
```

Notes:

1. Applies only if the Adabas link routine `ADAUSER` or the Natural profile parameter `ADANAME` is used.
2. Applies to DB2 users only.
3. Applies only if the Integrated Authentication Framework (IAF) is used.
4. Applies only if SSL is used.

Sample JCL for a Started Task

A sample JCL for a started task is provided in the Natural for mainframes *Installation* documentation, part *Installing Natural under z/OS*, *Create Sample JCL for Natural RPC Server*.

Running a Batch Server with Replicas

You can also run a batch server with replicas by setting the keyword subparameter NTASKS of profile parameter RPC or parameter macro NTRPC to a value greater than 1.

Replicas are attached to a Natural main task as additional server tasks. They enable you to start several identical servers in the same region.

Starting a Batch Server under z/VSE

Sample JCL for z/VSE

```
// LIBDEF PHASE,SEARCH=(SAGLIB.NATvrs,SAGLIB.EXXvrs,SAGLIB.ADAvrs),TEMP
// ASSGN SYS000,READER
// ASSGN SYSLST,FEE
// EXEC NATVSE,SIZE=AUTO,PARM='SYSRDR'
IM=D,MADIO=0,MT=0,OBJIN=R,AUTO=OFF,MAXCL=0,ID=',',INTENS=1,
RPC=(SERVER=ON,SRVNAME=servername,SRVNODE=nodename)
RPC=(RPCSIZE=m,MAXBUFF=n),
STACK=(LOGON serverlibrary,userID,password)
/*
```

Running a Batch Server with Replicas

You can also run a batch server with replicas by setting the keyword subparameter NTASKS of profile parameter RPC or parameter macro NTRPC to a value greater than 1.

Replicas are attached to a Natural main task as additional server tasks. They enable you to start several identical servers in the same region.

Starting a Batch Server under BS2000/OSD

Sample JCL for BS2000/OSD

```
/.NATRPC        LOGON
/               SYSFILE   SYSOUT=output-file
/               SYSFILE   SYSDTA=(SYSCMD)
/               SYSFILE   SYSIPT=(SYSCMD)
/               STEP
/               SETSW     ON=2
/               EXEC NATBS2
IM=D,MADIO=0,MT=0,OBJIN=R,AUTO=OFF,MAXCL=0,ID=',',INTENS=1,
RPC=(SERVER=ON,SRVNAME=servername,SRVNODE=nodename)
RPC=(RPCSIZE=m,MAXBUFF=n),
STACK=(LOGON serverlibrary,userID,password)
/               EOF
```

Starting a Natural RPC Server in a Windows Environment

To start a Natural RPC server under Windows, proceed as follows:

1. Create a shortcut for Natural.

2. Enter the shortcut properties.

3. Create a Natural parameter file (see *Invoking Natural with an Alternative Parameter File*) with the *RPC server parameters* set.

4. In the **Target** text box, edit the Natural path and append:

```
parm=serverparm batch
```

where *serverparm* is the name of the parameter file,

or

```
server=on,srvname=servername,srvnode=nodename,maxbuff=n batch
```

Starting a Natural RPC Server in a UNIX Environment

To start a Natural RPC server under UNIX, enter the following command:

```
natural parm=serverparm >/dev/null </dev/null &
```

where `serverparm` is the name of the parameter file,

or

```
natural server=on,srvname=servername,srvnode=nodename,maxbuff=n >/dev/null </dev/null &
```

Starting a Natural RPC Server in an OpenVMS Environment

To start a Natural RPC server under OpenVMS, enter the following commands in the DCL command procedure `myserver.`com:

```
$ DEFINE NATOUTPUT NLA0:
$ NAT parm=serverparm
```

Then submit `myserver.`com to a batch queue:

```
$ SUBMIT myserver.com
```

Considerations for Mainframe Natural RPC Servers with Replicas

This section applies to mainframe Natural servers under z/OS and z/VSE.

- Natural RPC Batch Server with `NTASKS >1`

- Running a Batch Server with Replicas

Natural RPC Batch Server with NTASKS >1

The main task and all replicas run in the same z/OS region or z/VSE partition.

1. Use the reentrant batch link routine `ADALNKR` instead of `ADALNK`.

   If you want to use `ADAUSER`, you must not link `ADAUSER` with your front-end, because `ADAUSER` is non-reentrant (see Item 5). Instead, use the Natural profile parameter `ADANAME` and set `ADANAME=ADAUSER`. This will cause Natural to load `ADAUSER` dynamically at runtime.

   Note for z/VSE: If you use `ADAUSER`, you must rename `ADALNKR` to `ADALNK`.

2. In the `NATPARM` module:

   - Set the keyword subparameter `NTASKS=`*n* of profile parameter `RPC` or parameter macro `NTRPC`, where *n* is the number of parallel servers (< 100) to be started, including the main task.

     Note for z/VSE: The number of subtasks is restricted by the operating system. Ask your system administrator.

   - Use the Natural profile parameter `ETID` to specify the Adabas user identification as a blank character. This is necessary to prevent a NAT3048 error (ETID not unique in Adabas nucleus) when the subtask is started.

3. When using dynamic Natural profile parameters:

   Use the dynamic parameter dataset `CMPRMIN` to pass the dynamic Natural profile parameters to Natural. Do *not* use the `PARM` card or the primary command input dataset `CMSYNIN`.

4. When using a local buffer pool (z/OS only):

   Each subtask allocates its own local buffer pool unless you specify a shared local buffer pool. See the parameter `LBPNAME` in the section *Natural z/OS Generation Parameters* (in the Natural *Operations* documentation).

5. In the Natural front-end link job (z/OS only):

   Link the front-end reentrant by using the `RENT` option of the linkage editor.

   If the front-end were not linked with the `RENT` option, only the main task would start the communication with the EntireX Broker. All subtasks would be set to a `WAIT` status by z/OS, until the main task would have been terminated. If you would terminate the RPC server lateron, the address space would hang and would have to be cancelled.

6. Make sure that any other modules that are additionally linked to the Natural nucleus are reentrant. Any dynamically loaded programs must also be reentrant.

Note for z/OS: If you cannot make a module reentrant, link the module as non-reusable; this means, you should *not* specify the link option `RENT` or `REUS`. This is to ensure that each subtask will get its own copy.

Running a Batch Server with Replicas

For a sample JCL, see *Using the Server Trace Facility*.

Starting a Natural RPC Server Using the RPC Server Front-End (z/OS Batch Mode only)

In z/OS batch mode, a Natural RPC server may alternatively be started using the RPC server front-end. This approach is required with impersonation and is optional in other cases.

If you use the RPC server front-end without impersonation, you are recommended to set the keyword subparameter `NTASKS` of profile parameter `RPC` or parameter macro `NTRPC` to a value greater than 1. Otherwise, there will be no benefit. The *Considerations for Mainframe Natural RPC Servers with Replicas* apply also when you are using the RPC server front-end.

The RPC server front-end uses the Natural Server functionality; see *Natural as a Server under z/OS* (in the Natural *Operations* documentation). It is characterized by the following features:

- The Natural RPC server front-end starts a number of Natural RPC server sessions as specified by the keyword subparameter `NTASKS` of profile parameter `RPC` or parameter macro `NTRPC`.

- All Natural RPC server sessions run with the same Natural profile parameter settings.

  The Natural profile parameter settings are taken from the Natural parameter module `NATPARM` and the dynamic parameter dataset `CMPRMIN` (if available). The parameter `PARM=` of the JCL statement `EXEC` is not used to provide Natural profile parameters.

- If all Natural RPC server sessions are currently in use by clients (executing a client request or waiting for the next request within a conversation) and if the keyword subparameter `NTASKS` of profile parameter `RPC` or parameter macro `NTRPC` is set to a value greater than one, auxiliary Natural RPC server sessions are started. These Natural RPC server sessions are automatically terminated on the first EntireX Broker timeout, provided that there is at least one other Natural RPC server session not in use by a client. If all other Natural RPC server sessions are being used by clients, the auxiliary RPC server session will stay up until the next EntireX Broker timeout. This makes sure that there is always a Natural RPC server available to process a new client request.

- The Natural RPC server sessions are executed in a thread environment that is similar to Natural sessions executing in a TP monitor system.

- All inactive Natural RPC server sessions (sessions that wait for a client request) are rolled out to the Natural Roll Server.

- **With impersonation:**
  At the end of a non-conversational `CALLNAT` and at the end of a conversation, all database sessions and all work files are closed. This ensures that the next client request will open the database and the work files with its own user ID.

- **Without impersonation:**
  After the first EntireX Broker timeout, all database sessions and all work files are closed. This ensures that no resources are blocked during wait times.

Startup Parameters:

The required startup parameters are passed in the `PARM=` parameter of the `EXEC` statement in the JCL. These parameters are:

- The name of the Natural z/OS batch nucleus.

- The size of a storage thread.

- The number of storage threads to be allocated.

  The number of storage threads determines the number of Natural RPC server sessions that can be concurrently executed and should not be smaller than the value of the keyword subparameter `NTASKS=`*n* of profile parameter `RPC` or parameter macro `NTRPC`.

- The optional keyword `UCTRAN`.

  `UCTRAN` indicates that all messages of the RPC Server front-end are translated into upper case.

These parameters must be separated by commas and must be entered without leading or trailing blanks:

> PARM='*Natural-z/OS-batch-nucleus,size-of-thread,number-of-threads*[,*UCTRAN*]'

See also the *Sample JCL* below.

Execution Notes:

- The Natural Roll Server is required.

  You must start a Natural Roll Server for the used `subsystem-id` (as defined by Natural profile parameter `SUBSID`) before the Natural RPC server front-end is started.

- The job name, or the name of the started task, is used as high level qualifier for all Natural sessions rolled out to the roll file.

  It is strongly recommended not to start more than one Natural RPC server front-end with the same job name, or with the same started task, and the same `subsystem-id`.

- The Natural z/OS batch nucleus is dynamically loaded.

  The load library containing the z/OS batch nucleus must be available in the steplib concatenation.

- The EntireX Broker stub `NATETB23` must *not* be used.

  The EntireX Broker is accessed `outside` the Natural context. Therefore you must use the EntireX Broker stub `BKIMBTSO`.

With impersonation only:

When the impersonation feature is used, the RPC server front end must be executed from an Authorized Program Facility (APF) library. You are recommended to execute the RPC server front-end from an APF-authorized LINKLIST library. This eliminates the need of providing the whole steplib concatenation APF authorized.

With Natural Security only:

If the Natural RPC server front-end is started with profile parameter AUTO=OFF, you must provide a Natural LOGON command with library ID, user ID and password on the Natural stack: STACK=(LOGON *library-id;user-id;password*).

Sample JCL:

```
//NATRPC   JOB  CLASS=K,MSGCLASS=X
//         EXEC PGM=RPC-FRONT,REGION=8M
//         PARM='Natural-z/OS-interface-module,1000,5'
//STEPLIB  DD   DISP=SHR,DSN=SAG.NAT.LOAD
//         DD   DISP=SHR,DSN=SAG.EXX.LOAD
//         DD   DISP=SHR,DSN=SAG.ADA.LOAD          <== Note 1
//         DD   DISP=SHR,DSN=DB2_load_library    <== Note 2
//         DD   DISP=SHR,DSN=SAG.SSX.LOAD          <== Note 3
//CMPRMIN  DD   *
IM=D,MADIO=0,MT=0,OBJIN=R,AUTO=OFF,MAXCL=0,ID=',',INTENS=1,
PRINT=((10),AM=STD)
RPC=(SERVER=ON,SRVNAME=servername,SRVNODE=nodename,NTASKS=3)
RPC=(RPCSIZE=m,MAXBUFF=n,TRACE=2),
RCA=BROKER,RCALIAS=(BROKER,BKIMBTSO)
STACK=(LOGON serverlibrary,userID,password)
/*
//CEEOPTS  DD *                                    <== Note 4
POSIX(ON)
*
//SYSUDUMP DD SYSOUT=X
//CMPRT10  DD SYSOUT=X
//CMPRT101 DD SYSOUT=X
//CMPRT102 DD SYSOUT=X
//CMPRT199 DD SYSOUT=X
//CMPRINT  DD SYSOUT=X
//CMPRINT1 DD SYSOUT=X
//CMPRINT2 DD SYSOUT=X
//CMPRIN99 DD SYSOUT=X
/*
```

Notes:

1. Applies only if the Adabas link routine ADAUSER or the Natural profile parameter ADANAME is used.

2. Applies to DB2 users only.

3. Applies only if the Integrated Authentication Framework (IAF) is used.

4. Applies only if SSL is used.

Starting a Natural RPC Server Using the RPC Server Front-End (CICS only)

In CICS, a Natural RPC server may alternatively be started using the RPC server front-end. This approach is required when impersonation is used and is optional in other cases.

If you use the RPC server front-end without impersonation, you are recommended to set the keyword subparameter NTASKS of profile parameter RPC or parameter macro NTRPC to a value greater than 1. Otherwise, there will be no benefit.

The RPC server front-end uses the Natural Server functionality. It is characterized by the following features:

- The Natural RPC server front-end is started via the transaction ID defined in the *Customize CICS* step for the Natural RPC server front-end;

  see *Installing the Natural CICS Interface*, *Customize CICS*.

  The transaction ID may either be entered at a terminal or you may use the Natural program STARTSFE in library SYSRPC to start the Natural RPC server front-end in asynchronous mode.

  The Natural RPC server front-end requires the name of the Natural CICS interface nucleus as startup parameter. This startup parameter is passed with the transaction ID.

- The Natural RPC server front-end starts a number of Natural RPC server sessions as specified by the keyword subparameter NTASKS of profile parameter RPC or parameter macro NTRPC.

- All Natural RPC server sessions run with the same Natural profile parameter settings.

  The Natural profile parameter settings are taken from the Natural parameter module NATPARM and the dynamic parameter dataset CMPRMIN (if available) and may be overwritten by dynamically specified profile parameters that are passed with the transaction ID. The dynamically specified profile parameters must follow the startup parameter.

- If all Natural RPC server sessions are currently in use by clients (executing a client request or waiting for the next request within a conversation) and if the keyword subparameter NTASKS of profile parameter RPC or parameter macro NTRPC is set to a value greater than one, auxiliary Natural RPC server sessions are started. These Natural RPC server sessions are automatically terminated on the first EntireX Broker timeout, provided that there is at least one other Natural RPC server session not in use by a client. If all other Natural RPC server sessions are being used by clients, the auxiliary RPC server session will stay up until the next EntireX Broker timeout. This will ensure that there is always a Natural RPC server available to process a new client request.

- The Natural RPC server sessions are executed in a thread environment that is similar to Natural sessions executing in a TP monitor system.

- All inactive Natural RPC server sessions (sessions that wait for a client request) are rolled out to the Natural Roll Server.

- **With impersonation:**

At the start of a non-conversational `CALLNAT` and at the start of a conversation, a new CICS worker task is started under the user ID of the client by using the `USERID()` option of the `EXEC CICS START TRANSID()` command. The client request is executed by Natural in this worker task. While the client request is executed, the Natural RPC server session waits for the worker task to finish.

At the end of a non-conversational `CALLNAT` and at the end of a conversation, the worker task is terminated and all databases are closed and all CICS resources are freed. This ensures that the next client request will open the database and access the CICS resources with its own user ID.

- **Without impersonation:**

  The client request is executed by the Natural RPC server session itself.

  After the first EntireX Broker timeout, all databases are closed and all CICS resources are freed. This will ensure that no resources are blocked during wait times.

Startup Parameters:

The required startup parameters are passed with the transaction ID. These parameters are:

- The name of the Natural CICS interface nucleus *<ncistart>*.

- The optional keyword `UCTRAN`.

  `UCTRAN` indicates that all messages of the RPC Server front-end are translated into upper case.

- An optional Natural profile parameter string.

Sample Start at a Terminal:

```
<natural> <ncistart>[,UCTRAN]
RPC=(SERVER=ON,SRVNAME=servername,SRVNODE=nodename,RPCSIZE=n, MAXBUFF=n)
RCA=BROKER,RCALIAS=(BROKER,CICSETB)
```

Where `<natural>` is the transaction ID with which you start your Natural RPC server front-end and `<ncistart>` is the name of your Natural CICS interface nucleus.

Execution Notes:

- The Natural Roll Server is required if the `NCMDIR` parameter `ROLLSRV` is set to `YES`.

  You must start a Natural Roll Server for the used subsystem-id (as defined by Natural profile parameter `SUBSID`) before the Natural RPC server front-end is started.

- The transaction ID of the RPC server front-end is used to identify the RPC server environment.

  Do not to start more than one Natural RPC server front-end with the same transaction ID.

- The executable NCI module is dynamically loaded.

  The load library containing the executable NCI module must be available in the `DFHRPL` concatenation.

- The EntireX Broker stub `NATETB23` must not be used.

  The EntireX Broker is accessed outside the Natural context. Therefore, you must use the EntireX Broker stub `CICSETB`.

  Depending on the version of `CICSETB`, you cannot use the profile parameter `RCA`. In that case, you must link `CICSETB` to your Natural CICS interface nucleus instead.

With impersonation only:
When the impersonation feature is used, the RPC server front-end starts worker tasks. Ensure that the setting of the CICS configuration option `MAXTASKS` of your CICS installation is high enough.

With Natural Security only:
If the Natural RPC server front-end is started with profile parameter `AUTO=OFF`, you must provide a Natural `LOGON` command with library ID, user ID and password on the Natural stack: `STACK=(LOGON library-id;user-id;password)`.