Using Security

This section covers the following topics:

- Using Natural RPC with Natural Security
- Impersonation (z/OS Batch Mode)
- Impersonation (CICS)
- Using Natural RPC with EntireX Security
- Using the Integrated Authentication Framework

Using Natural RPC with Natural Security

Natural RPC also supports Natural Security in client/server environments, where security may be active on either (or both) sides.

For general information, refer to the *Natural Security* documentation.

For information on how to control the use of Natural remote procedure calls in a client/server environment, see *Protecting Natural RPC Servers and Services* in the *Natural Security* documentation.

Client Side

The client must send logon data together with the RPC request. The logon data consist of user ID, password and library.

- User ID and password are used to perform the authentication of the client on the Natural RPC server side.
- The library is used to perform a Natural Security protected logon to the requested library.

The following applies to Natural RPC clients only. For EntireX RPC clients that access a Natural Security protected Natural RPC server, refer to the EntireX Developer's Kit documentation.

To send logon data to the Natural RPC server, the Logon option must be used. See *Operating a Natural RPC Environment*, *Using the Logon Option*. The logon data parts are established as follows:

1. The user ID and password:

   If the client runs under Natural Security

   The user ID and password from the Natural Security logon on the client are used and passed to the Natural RPC server.

   If you want to use a different user ID and/or password for the Natural Security logon on the server side, you may use the application programming interface `USR1071N` (see below).

Note:
You may disallow the use of USR1071N in the Natural RPC restrictions part of the *Session Parameters* restrictions of the Natural Security library profile.

If the client does not run under Natural Security

To specify the user ID and password that are passed to the Natural RPC server, the client must call application programming interface USR1071N (see below) before the first RPC request is sent.

2. The library:

   By default, the name of the library to which the client is currently logged on is used. If you want to pass another library name to the Natural RPC server, you may use the application programming interface USR4008N.

If impersonation without password check is active for the Natural RPC server (field Impersonation described in the section *Components of an RPC Server Profile* in the *Natural Security* documentation is set to A), the client may optionally pass an ETID to the Natural RPC server. This ETID will be used by the Natural RPC server to access Adabase on behalf of the client. To specify an ETID on the Natural RPC client side, you may use the application programming interface USR4371N.

USR1071N

The application programming interface USR1071N is provided in the library SYSEXT. It is used to specify the user ID and password that are passed to the Natural RPC server.

To make use of USR1071N

1. Copy the subprogram USR1071N and the program USR1071P from libray SYSEXT to the library SYSTEM in the system file FNAT in the server environment; see *Using a Natural API* in the *SYSEXT Utility* documentation.

2. Using a DEFINE DATA statement, specify the following parameters:

Parameter	I/O	Format	Description	
USERID	I	A08	User ID to be used.	
PASSWORD	I	A08	Password to validate the user ID. This password is not validated on the client side.	
MIXEDCASE	I	A01	Mixed case option for password (optional).	
			Y	Allow mixed case password.
			N	Convert passwords to upper case.

3. In the calling program on the client side, specify the following statement:

```
FETCH RETURN 'USR1071P' USERID PASSWORD [MIXEDCASE]
```

You may alternatively invoke USR1071P from the command line and enter user ID and password in the displayed window.

For a more detailed description, see the USR1071T member in library SYSEXT.

Note:
Two samples are provided to call USR1071N: USR1071P, which is passing just user ID and password, and USR1071X (extended version), which in addition enables the user to set/retrieve various data.

USR4371N

The application programming interface USR4371N is provided in the library SYSEXT. It is used to specify the user ID and the ETID that are passed to the Natural RPC server.

To make use of USR4371N

1. 1. Copy the subprogram USR4371N and the program USR4371P from libray SYSEXT to the library SYSTEM in the system file FNAT in the client environment; see Using a Natural API
2. Using a DEFINE DATA statement, specify the following parameters:

Parameter	I/O	Format	Description
USERID	I	A08	User ID to be used.
ETID	I	A08	ETID to be used.

3. In the calling program on the client side, specify the following statement:

```
FETCH RETURN 'USR4371P' USERID ETID
```

Alternatively, you may invoke USR4371P from the command line, and enter user ID and ETID in the displayed window.

For a more detailed description, see the text member USR4371T in the library SYSEXT.

Server Side

If Natural Security is installed on the server side and AUTO=ON is not specified, a Natural logon with user ID and password is required. It is recommended to use the Natural profile parameter STACK to pass the Natural system command LOGON. If AUTO=ON is specified the contents of *INIT-USER is used for an internal logon as usual.

To enforce the Logon option - that is, if you want a server to accept only requests from clients where the Logon option is set - set the LOGONRQ profile parameter to ON for the server. If the Logon option is not enforced, client request without logon data are accepted and executed in the server library or one of its steplibs. This allows you to provide public as well as secured services.

If the client passes logon data, the user ID and password from the client are verified against the corresponding user security profile on the server, and the logon to the requested library and the execution of the subprogram are performed according to the corresponding Natural Security library and user profile definitions on the server.

After the execution of the subprogram, the library used before the CALLNAT request is updated again on the server. In the case of a conversational RPC, the first CALLNAT request within the conversation sets the library ID on the server, and the CLOSE CONVERSATION statement resets the library ID on the server to the one used before the conversation was opened.

As part of the *Natural RPC Restrictions* in the library profiles of Natural Security, a server session option `Close all databases` is provided. It causes all databases which have been opened by remote subprograms contained in the library to be closed when a Natural logon/logoff to/from the libraries is performed. This means that each client uses its own database session.

If the `Close all databases` option is set, it is also possible to use a client specific ETID for all Adabas accesses which are executed by the server for this client. In this case, you should start the Natural RPC server with `ETID=OFF` and define an appropriate ETID in the user profile for each client that needs an ETID, forexample, by specifying the ETID `*USER`. Please note that in this case two clients with the same name cannot issue two concurrent requests with Adabas calls.

Changing Password

It is possible to change the Natural Security password on the Natural RPC server via a Natural RPC service request. For this purpose, the application programming interface `USR2074N` is provided in the library `SYSEXT`.

To make use of USR2074N

1. Copy the subprogram `USR2074N`, and optionally program `USR2074P`, from libray `SYSEXT` to the library `SYSTEM` or to the steplib library or to any application in the server environment.
2. Using a `DEFINE DATA` statement, specify the following parameters:

Parameter	I/O	Format	Description
USERID	I	A08	User ID to be used.
PASSWORD	I	A08	Password to validate the user ID. This password is not validated on the client side.
NEWPASSWORD	I	A08	New password for the user ID. This password is not validated on the client side.
NODE-NAME	I	A192	Name of the server node to be addressed. The node name may have up to 32 characters for physical node names and up to 192 characters for logical node names. See *Using EntireX Location Transparency* in *Operating a Natural RPC Environment*. The sample USR2074P provided in library SYSEXT supports up to 32 characters.
SERVER-NAME	I	A192	Name of the server to be addressed. The server name may have up to 32 characters for physical server names and up to 192 characters for logical service names. See *Using EntireX Location Transparency* in *Operating a Natural RPC Environment*. The sample USR2074P provided in library SYSEXT supports up to 32 characters.
PROTOCOL	I	A1	The transport protocol to address the server node. Valid value: B: EntireX Broker
RC	O	I2	Return value: 0: OK, MESSAGE contains a confirmation message. 1: Error from RPC or server node, MESSAGE contains the error message. 2: Error from the interface, MESSAGE contains the error message. 3: Natural Security error, MESSAGE# contains the Natural error number and MESSAGE contains the corresponding message text.
MESSAGE#	O	N4	Message number returned.
MESSAGE	O	A80	Message text returned.

3. In the calling program on the client side, specify the following statement:

```
CALLNAT 'USR2074N' user-id password newpassword node-name
server-name protocol rc message# message
```

You may alternatively use program USR2074P from library SYSEXT. Invoke USR2074P from the command line and enter the required data in the displayed window. In this case, all input except for the passwords are translated into upper case. For the passwords, you have the option to enter them in mixed case or not.

Impersonation (z/OS Batch Mode)

- Purpose of Impersonation
- Steps to Activate Impersonation (Server Side)
- Steps to Use Impersonation (Client Side)
- Rules for Impersonation

Purpose of Impersonation

Impersonation is an optional feature on the Natural RPC server side and is only available if the Natural RPC server runs under Natural Security. The impersonation feature is controlled by the *Security Profiles for Natural RPC Servers*. See the field Impersonation described under the heading *Components of an RPC Server Profile* in the section *Protecting Natural RPC Servers and Services* in the *Natural Security* documentation.

Impersonation in z/OS batch mode requires the use of the Natural RPC server front-end under z/OS and uses the SAF interface provided by z/OS.

If impersonation is active for the Natural RPC server, a client request that uses the *Logon Option* is from the perspective of the operating system executed under the user ID that the client passes in the LOGON data (called Natural RPC user ID). Impersonation assumes that access to the operating system on which a Natural RPC server is running is controlled by an SAF-compliant external security system. User authentication (verification of the Natural RPC user ID and password) is performed by this external security system. After successful authentication, the user's identity is established for the operating system (that is, an ACEE is created and linked to the TCB under which the current client request is executed). Any subsequent authorization checks will be performed based on this identity. This means that all accesses to resources that are controlled by the SAF compliant external security system are authorized for this identity. This applies especially to accesses to work files and to data bases.

Impersonation does not turn off Natural Security. After successful authentication of the user's identity by the external security system, a Natural Security logon takes place using the same LOGON data but without password verification.

To start a Natural RPC server using impersonation, see *Starting a Natural Server Using the RPC Server Front-End* in *Starting a Natural RPC Server*.

Note:
Without impersonation, a client request that uses the Logon option is from the perspective of the operating system executed with the user ID under which the Natural RPC server has been started.

Steps to Activate Impersonation (Server Side)

1. **Install RPC server front-end**

   Proceed as described in the corresponding steps of the Natural for Mainframes installation documentation; see *Installing Natural on z/OS*.

   If you choose to use the recommended APF-authorized `LINKLIST` library, you must ensure that the resulting load module does not exist in the `STEPLIB` or `JOBLIB` concatenation.

2. **Link Natural z/OS batch nucleus with DB2 interface `DSNRLI`**

   This step applies to Natural for DB2 users only.

3. **Use reentrant Adabas batch link routine `ADALNKR` instead of `ADALNK`**

   Refer to *Considerations for Mainframe Natural RPC Servers with Replicas* in *Starting a Natural RPC Server*.

4. **Use EntireX Broker stub `BKIMBTSO` instead of `NATETB23`**

   See *Provide Access to the EntireX Broker Stub* in *Setting Up a Natural RPC Environment*.

5. **Define all required RPC server-specific Natural profile parameters**

   Refer to *Set the RPC Server-Specific Natural Parameters* in *Setting Up a Natural RPC Environment*. The parameters are either defined in the `NATPARM` parameter module or in the `CMPRMIN` dataset. The parameter `PARM=` of the `JCL EXEC` statement is not used to provide Natural profile parameters.

6. **Define an RPC server profile in Natural Security**

   Define an RPC server profile in Natural Security (NSC) for the server name that is used by the RPC server (`SRVNAME`) and activate the impersonation.

   Refer to *Security Profiles for Natural RPC Servers* in *Protecting Natural RPC Servers and Services* of the *Natural Security* documentation.

7. **Check SAF definitions**

   (This step applies to Natural for DB2 users only.)

   If the SAF resource class `DSNR` is active, you must check whether you need the following SAF definitions:

   `RDEFINE DSNR (`*`subsys`*`.RRSAF) OWNER(`*`DB2owner`*`)`

   `PERMIT `*`subsys`*`.RRSAF CLASS(DSNR) ID(`*`DB2group`*`) ACCESS(READ)`

   where *`subsys`* is your DB2 subsystem ID.

   Each user who wants to access DB2 must be a member of group *`DB2group`*.

For further information, refer to the relevant DB2 documentation of IBM.

8. **Create user exit `NATRPC02`**

   (This step applies to Natural for DB2 users only.)

   Create the Natural RPC user exit `NATRPC02` with a call to `NATPLAN` to set the required DB2 plan.

   Make sure that you use a `NATPLAN` of your current Natural for DB2 version.

   Sample `NATRPC02`:

```
DEFINE DATA PARAMETER
  1 SUBPROGRAM (A8) BY VALUE END-DEFINE
FETCH RETURN 'NATPLAN' 'planname'
```

9. **Start Roll Server**

   Start the Roll Server for the *`subsystem-id`* used by the Natural RPC server.

10. **Start Natural RPC server front-end**

    Start the Natural RPC server front-end.

    Refer to *Starting a Natural RPC Server Using the RPC Server Front-End* in *Starting a Natural RPC Server*.

    Make sure you have added all required load libaries to your `STEPLIB` concatenation. You will especially need the following:

    - Natural load library
    - EntireX load library
    - Adabas load library (if you use the Adabas link routine `ADAUSER`)
    - DB2 load library (if you want to access DB2)

    The impersonation is successfully activated if you see the following messages:

    - In the job log:

      ```
      RPC0010 Authorized environment for impersonation established
      ```

    - In the RPC trace file:

      ```
      M *** Server is running under NSC with impersonation
      ```

Steps to Use Impersonation (Client Side)

The client must send logon data together with the RPC request as it is already done for a standard Natural Security (NSC) protected Natural RPC server. In contrast to a standard Natural RPC server, the user ID must also be a valid SAF user ID and the password must be the corresponding SAF password. User ID and password are validated by the Natural RPC server against the external security system on the z/OS system under which the server is executing. After successful authentication of the client's identity by the

external security system, the user ID is validated by NSC according to the defined rules. The password is ignored. Therefore, it is not required to set the NSC password to your SAF password.

When the field `Impersonation` described in the section *Components of an RPC Server Profile* in the *Natural Security* documentation is set to `A`, no password is used to authenticate the client against the external security system. This setting may be appropriate if the client has already been authenticated by the EntireX Broker.

Depending on the kind of client, the logon data are set differently:

Natural Clients

1. Turn on the logon option in the *Service Directory Maintenance* function or in the `DFS` keyword subparameter

   Alternatively, you can use the `USR2007N` to turn it on.

   Refer to *Using the Logon Option* in *Operating a Natural RPC Environment*.

2. Set the SAF user ID and the SAF password, using application programming interface `USR1071P`.

   If your client runs under Natural Security (NSC) and the user ID and password of NSC are identical to the SAF user ID and the SAF password, then `USR1071P` is not required.

EntireX RPC Clients

1. Turn on the Natural logon option according to your application environment.
2. Set the RPC user ID and the RPC password to the SAF user ID and SAF password according to your application environment.

Rules for Impersonation

- Impersonation takes place at the start of each non-conversational `CALLNAT` and at the start of each conversation.
- The authentication of the Natural RPC user ID and password is performed by the external security system. The password on the `FSEC` system file is not used.
- After successful authentication, the Natural RPC user ID is established for the operating system (user is impersonated).
- After successful impersonation:

  1. A Natural security logon is performed for the Natural RPC user ID without password check.
  2. All work files with a `DDNAME` that does not start with CM are opened with the Natural RPC user ID.
  3. All Adabas databases are opened with the Natural RPC user ID (applies to Adabas external security only).

4. If an ETID is specified in the NSC user profile, this ETID is used in the Adabas open request.
5. The DB2 connection is opened with the Natural RPC user ID (applies to Natural for DB2 users only).

- At the end of each non-conversational CALLNAT and at the end of each conversation, the Natural RPC user ID is logged off from the operating system.
- After log off:

1. All work files with a DDNAME that does not start with CM are closed.
2. All Adabas databases are closed.

Impersonation (CICS)

The following topics are covered below:

- Purpose of Impersonation
- Steps to Activate Impersonation (Server Side)
- Steps to Use Impersonation (Client Side)
- Rules for Impersonation

Purpose of Impersonation

Impersonation is an optional feature on the Natural RPC server side and is only available if the Natural RPC server runs under Natural Security. The impersonation feature is controlled by the *Security Profiles for Natural RPC Servers*. See the field Impersonation described under the heading *Components of an RPC Server Profile* in the section *Protecting Natural RPC Servers and Services* in the *Natural Security* documentation.

Impersonation under CICS requires the use of the Natural RPC server front-end under CICS and uses the interface provided by CICS.

If impersonation is active for the Natural RPC server, a client request that uses the Logon Option is from the perspective of CICS executed under the user ID that the client passes in the LOGON data (called Natural RPC user ID). Impersonation under CICS uses the CICS option to start a CICS task under a given user ID. After a client request has arrived the Natural RPC server front-end starts a new CICS task using the USERID() option of the EXEC CICS START TRANSID() command, where USERID is the Natural RPC user ID. The User authentication (verification of the Natural RPC user ID) is performed by CICS, typically by using the underlying external security system. After successful authentication, the user's identity is established for the CICS task. Any subsequent authorization checks will be performed based on this identity. This means that all accesses to resources that are controlled by CICS are authorized for this identity. This applies especially to accesses to CICS resources and to data bases.

Impersonation does not turn off Natural Security. After successful authentication of the user's identity by CICS, a Natural Security logon takes place using the same LOGON data with password verification.

To start a Natural RPC server using impersonation, see Starting a Natural Server Using the RPC Server Front-End (CICS only) in Starting a Natural RPC Server.

Note:
Without impersonation, a client request that uses the Logon option is from the perspective of the operating system executed with the user ID under which the Natural RPC server has been started.

Steps to Activate Impersonation (Server Side)

1. **Install the RPC server front-end under CICS**

   Proceed as described in the corresponding steps of the Natural for Mainframes installation documentation; see *Installing the Natural CICS Interface on z/OS*.

2. **Install the Adabas link routine for Adabas external security**

   For further information, refer to the relevant Adabas documentation (applies to Adabas external security users only).

3. **Use EntireX Broker stub `CICSETB` instead of `NATETB23`**

   See *Providing Access to the EntireX Broker Stub on Mainframe* in *Setting Up a Natural RPC Environment*.

   You must link `CICSETB` to your Natural CICS interface nucleus.

4. **Define all required RPC server-specific Natural profile parameters**

   Refer to *Set the RPC Server-Specific Natural Parameters* in *Setting Up a Natural RPC Environment*.

   The parameters are either defined in the parameter module `NATPARM` or in the dataset `CMPRMIN`.

5. **Define an RPC Server Profile in Natural Security**

   Define an RPC Server Profile in Natural Security (NSC) for the server name that is used by the RPC server (`SRVNAME`) and activate the impersonation.

   Refer to*Security Profiles for Natural RPC Servers* in *Protecting Natural RPC Servers and Services* of the *Natural Security* documentation.

6. **If CICS startup parameter `XUSER=YES`**

   If the CICS startup parameter `XUSER=YES` is specified you must define surrogate users for each client user:

   ```
   RDEFINE SURROGATE userid1.DFHSTART UACC(NONE) OWNER(userid1) PERMIT
   userid1.DFHSTART CLASS(SURROGATE) ID(userid2) ACCESS(READ)
   ```

   where

   `userid1` is the user ID of the client,

userid2 is the user ID under which the Natural RPC server front-end is started.

For further information, refer to the relevant CICS documentation of IBM.

7. **Define a CICS `PROGRAM` entry for the RPC server front-end**

   Refer to the corresponding step in *Installing the Natural CICS Interface on z/OS*.

8. Define a CICS `TRANSACTION` entry for the transaction ID that invokes the RPC server front-end.

   Refer to the corresponding step in *Installing the Natural CICS Interface on z/OS*.

9. **Define a `DB2TRAN` and `DB2ENTRY` entry**

   (This step applies to Natural for DB2 users only.)

   Define a `DB2TRAN` and `DB2ENTRY` entry for the transaction ID that invokes the RPC server front-end.

10. **Start the Roll Server**

    Start the *Roll Server* for the subsystem used by the Natural RPC server.

    (This step applies only if the `NCMDIR` macro parameter `ROLLSRV` is set to `YES`.)

11. **Start the Natural RPC server front-end under CICS**

    Refer to *Starting a Natural RPC Server Using the RPC Server Front-End (CICS only)* in *Starting a Natural RPC Server*.

    The impersonation is successfully activated if you see the following message in the RPC trace file:

    ```
    M *** Server is running under NSC with impersonation
    ```

Steps to Use Impersonation (Client Side)

The client must send logon data together with the RPC request as it is already done for a standard Natural Security (NSC) protected Natural RPC server. In contrast to a standard Natural RPC server, the user ID must also be a valid CICS user ID. The user ID is validated by CICS against the external security system on the z/OS system under which CICS is executing. After successful authentication of the client's identity by the external security system, user ID and password are validated by Natural Security according to the defined rules.

When the field `Impersonation` described in the section *Components of an RPC Server Profile* in the *Natural Security* documentation is set to `A`, no password is used to validate the client by Natural Security. This setting may be appropriate if the client has already been authenticated by the EntireX Broker.

Depending on the kind of client, the logon data are set differently:

Natural Clients

1. **Turn on the logon option**

   Turn on the logon option in the *Service Directory Maintenance* function or in the `DFS` keyword subparameter of profile parameter `RPC` or parameter macro `NTRPC`.

   Alternatively, you can use the application programming interface `USR2007N` to turn it on.

   Refer to *Using the Logon Option* in *Operating a Natural RPC Environment*.

2. **Set user ID and password**

   Set the user ID and the password, using application programming interface `USR1071P`.

   If your client runs under Natural Security (NSC) and the user ID and password of NSC are identical to the user ID and password on the server side, then `USR1071P` is not required.

EntireX RPC Clients

1. **Turn on the Natural logon option**

   Turn on the Natural logon option according to your application environment.

2. **Set RPC user ID and password**

   Set the RPC user ID and the RPC password according to your application environment.

Rules for Impersonation

- Impersonation takes place at the start of each non-conversational `CALLNAT` and at the start of each conversation.
- The authentication of the Natural RPC user ID is performed by CICS. The password is not used.
- After successful authentication, the Natural RPC user ID is established for CICS (user is impersonated).
- After successful impersonation:

  1. A Natural security logon is performed for the Natural RPC user ID with password check according to the defined rules.
  2. All CICS resources are accessed with the Natural RPC user ID.
  3. All Adabas databases are opened with the Natural RPC user ID (applies to Adabas external security only).
  4. If an `ETID` is specified in the NSC user profile, this `ETID` is used in the Adabas open request.
  5. The DB2 connection is opened with the Natural RPC user ID (applies to Natural for DB2 users only).

- At the end of each non-conversational CALLNAT and at the end of each conversation, the Natural RPC user ID is logged off from CICS.
- After log off:
  1. All CICS resources are closed.
  2. All Adabas databases are closed.
  3. The connection to DB2 is closed (applies to Natural for DB2 users only).

Using Natural RPC with EntireX Security

Natural RPC fully supports EntireX Security on the client side and on the server side.

- EntireX Security on the Client Side
- EntireX Security on the Server Side

EntireX Security on the Client Side

To logon to and logoff from the EntireX Broker, the Natural Application Programming Interface USR2071N is provided. To logon to EntireX Broker, you use the logon function of USR2071N and pass your user ID and password to the selected EntireX Broker. After a successful logon, the security token returned is saved by Natural and passed to the EntireX Broker on each subsequent call. The Logon option is fully transparent to the Natural application.

If EntireX Security is installed or if AUTOLOGON=NO is specified in the EntireX Broker attribute file, you must invoke USR2071N with the logon function before the very first remote CALLNAT execution.

You are recommended to invoke USR2071N with the logoff function as soon as you no longer intend to use a remote CALLNAT.

To make use of USR2071N

1. Copy the subprogram USR2071N from libray SYSEXT to the library SYSTEM or to the steplib library or to any application in the server environment.
2. Using a DEFINE DATA statement, specify the following parameters:

Parameter	I/O	Format	Description	
function	I	A08	Function code; possible values are:	
			LOGON	Logon to EntireX Broker
			LOGOFF	Logoff from EntireX Broker
broker-id	I	A192	Broker ID The *broker-id* may have up to 32 characters for physical node names and up to 192 characters for logical node names or logical service names. See *Operating a Natural RPC Environment, Using EntireX Location Transparency*. **Note:** For compatibility reasons *broker-id* is defined with BY VALUE RESULT to support existing callers which pass an A8 field for the *broker-id*. The sample USR2071P provided in library SYSEXT supports up to 32 characters.	
user-id	I	A08	User ID.	
password	I	A08	User ID’s password.	
newpassw	I	A08	User ID’s new password.	
rc	O	N04	Return value:	
			0	OK
			1	invalid function code
			9999	EntireX Broker error (see *message*)
message	O	A80	Message text returned by EntireX Broker.	

3. In the calling program on the client side, specify the following statement:

   ```
   CALLNAT 'USR2071N' function broker-id user-id password newpassword rc message
   ```

   See also the *Syntax Description* of the CALLNAT statement.

You may alternatively invoke USR2071P from the command line and enter user ID and password in the displayed window. In this case, all input except for the passwords is translated into upper case. For the passwords, you have the option to enter them in mixed case or not.

Functionality:

LOGON	An EntireX Broker LOGON function is executed to the named *broker-id* with the *user-id* and the *password* passed. After a successful LOGON call, the client can communicate with the EntireX Broker *broker-id* as usual. With *newpassw* the client user can change her/his password via the EntireX Security features. **Notes:** • If a successful logon has been performed, the user ID used in this LOGON will be passed to the named EntireX Broker on all subsequent remote procedure CALLNATs which are routed via this EntireX Broker. Without an explicit LOGON, the current contents of system variable *USER is used. The same applies if you have issued a LOGON to EntireX Broker 1, but your remote procedure CALLNAT is routed via EntireX Broker 2. • It is possible to concurrently log on to multiple EntireX Brokers. For each LOGON, a different user ID may be used. • The user ID used for the LOGON to the EntireX Broker may be different from the Natural user ID under which the client application runs. • An internal re-logon is done after an EntireX Broker timeout has occurred, if the original LOGON was done without a password (the password used in the LOGON is not saved). If no internal re-logon is possible after a timeout has occurred, the client has to explicitly reissue the LOGON. • At the end of the Natural session, an implicit LOGOFF is executed to all EntireX Brokers to which a logon has been performed.
LOGOFF	An EntireX Broker LOGOFF function is executed to the *broker-id* named.

Special Considerations when using Location Transparency:

If you want to LOGON using a logical node name, you have to use the LOGBROKER keyword.

```
BROKER-ID := 'LOGBROKER=my_logical_node,my_set'
```

If you want to LOGON using a logical service name, you have to use the LOGSERVICE keyword.

```
BROKER-ID := 'LOGSERVICE=my_logical_service,my_set'
```

Special Considerations when the Client Request is executed on the Server Side:

If an RPC client request is executed on the Natural RPC server side, a logon to the EntireX Broker, using the Application Programming Interface USR2071N, must also be performed before executing the RPC client request. The logon data of the Natural RPC server itself are not used for RPC client requests.

If the RPC client request is sent to the same EntireX Broker where the Natural RPC server is registered, the user ID must be different from the value of the Natural profile parameter SRVUSER.

EntireX Security on the Server Side

If the value of profile parameter ACIVERS is 2 or higher, the server will log on to the EntireX Broker at the session start using the LOGON function. The user ID is the same as the user ID defined by SRVUSER.

If EntireX Security has been installed and if the EntireX trusted user ID feature is not available, there are two alternative ways to specify the required password:

- Setting SRVUSER=*NSC
- Using application programming interface USR2072N

These alternatives are described below.

Setting SRVUSER=*NSC

If Natural Security is installed on the server, you can set profile parameter SRVUSER to *NSC to specify that the current Natural Security user ID which was used when the server was started is used for the LOGON in conjunction with the accompanying Natural Security password. In this case, the value set for ACIVERS must be at least 4.

Using Application Programming Interface USR2072N to Specify a Password

The Application Programming Interface USR2072N enables you to specify a password which is used for the LOGON in conjunction with profile parameter SRVUSER.

To make use of USR2072N

1. Copy the subprogram USR2072N and optionally program USR2072P from libray SYSEXT to the library SYSTEM or to the steplib library or to any application in the server environment.
2. Using a DEFINE DATA statement, specify the following parameter:

Parameter	I/O	Format	Description
password	I	A08	User ID’s password.

3. In the calling program on the client side, specify the following statement:

   ```
   CALLNAT 'USR2072' password
   ```

   See also the *Syntax Description* of the CALLNAT statement.

4. The calling program must be executed before the Natural RPC server has started its initialization. To accomplish this, put the name of the calling program on the Natural stack when starting the server. For this purpose, you may also use the program USR2072P from library SYSEXT. In this case, the password is translated into upper case by default. You have the option to enter the password in mixed case by passing the mixed case option Y as second parameter.

   ```
   STACK=(LOGON server-library;USR2072P password [Y])
   ```

Using the Integrated Authentication Framework

The Integrated Authentication Framework (IAF) is an optional feature that can be used on the Natural RPC server side.

- Purpose of the Integrated Authentication Framework
- Steps to Use the Integrated Authentication Framework (Client Side)
- Steps to Activate the Integrated Authentication Framework (Server Side)

Purpose of the Integrated Authentication Framework

The Integrated Authentication Framework is available under the following conditions:

1. The Natural RPC server runs under Natural Security.
2. The EntireX Broker is protected by an IAF server.
3. The Natural RPC server and the EntireX Broker use the same IAF server.
4. The Software AG Security eXtension (SSX) must have been installed by EntireX.
5. The Natural RPC server runs under TSO, z/OS batch mode, UNIX or Windows.

The IAF feature is controlled by the Natural Security profiles for Natural RPC servers. See *Protecting Natural RPC Servers and Services* in the *Natural Security* documentation.

If a Natural RPC server is configured to use the Integrated Authentication framework, the Natural RPC server will no longer authenticate a client request by the user ID and password passed in the logon data. Instead, the user ID with which the client has logged on to the EntireX Broker is used as a trusted user ID without authentication. The following steps take place:

1. The client request is authenticated by the EntireX Broker using the IAF server.
2. The IAF server returns an encrypted and signed token that contains the user ID.
3. The EntireX Broker passes the IAF token together with the RPC request to the Natural RPC server.
4. The Natural RPC server validates and decrypts the IAF token and treats the user ID that is contained in the IAF token as trusted.
5. Natural Security validates the user ID and performs a logon to the requested library using the defined rules for authorization. No password is used.

As a consequence, after a successful Natural Security logon, the Natural user ID in the Natural system variable `*USER` and the EntireX user ID are identical.

Steps to Use the Integrated Authentication Framework (Client Side)

The client must logon to the EntireX Broker as it is done within a standard EntireX Security environment. It is transparent to the client that User ID and password are authenticated by the IAF server.

The client must also send logon data together with the RPC request as it is done for a standard Natural Security protected Natural RPC server.

In contrast to a standard Natural Security protected Natural RPC server the user ID and password provided in the logon data are ignored and no authentication takes place (see above). Only the Natural library is evaluated by the Natural RPC server. User ID and password may therefore be omitted in the logon data.

Steps to Activate the Integrated Authentication Framework (Server Side)

To activate the Integrated Authentication Framework on the server side, perform the following steps according to your environment:

- Under TSO and in z/OS Batch Mode
- Under Windows and UNIX

Under TSO and in z/OS Batch Mode

1. Define the IAF service in Natural Security Refer to *IAF Support* in the section *Protecting Natural RPC Servers and Services* of the *Natural Security* documentation.
2. Add the CA certificate that is used by your IAF server to the RACF keyring that is referenced in the Trust store field of *IAF Support*.
3. Permit access to keyrings to the RACF user ID under which the Natural RPC server will be started.
4. Define an RPC Server Profile in Natural Security for the server name that is used by the RPC server (`SRVNAME`) and activate the IAF support. Refer to Security Profiles for Natural RPC Servers in the section *Protecting Natural RPC Servers and Services* of the Natural Security documentation.
5. Generate the Natural z/OS batch nucleus that is used by the Natural RPC server with LE support (`LE370=YES`).
6. Set `ACIVERS=9` or above in the Natural profile parameters that are used by your Natural RPC server.
7. Turn on POSIX in your batch JCL that executes the Natural RPC server by using the `CEEOPTS` input data set. See *Starting a Batch Server under z/OS*.
8. Add the Software AG Security eXtension (SSX) load library to your JCL. See *Starting a Batch Server under z/OS*.

Under Windows and UNIX

1. Define the IAF service in Natural Security. Refer to *IAF Support* in the section *Protecting Natural RPC Servers and Services* of the *Natural Security* documentation.
2. Define an RPC Server Profile in Natural Security for the server name that is used by the RPC server (`SRVNAME`) and activate the IAF support. Refer to *Security Profiles for Natural RPC Servers* in the section *Protecting Natural RPC Servers and Services* of the Natural Security documentation.
3. Set `ACIVERS=9` or above in the Natural profile parameters that are used by your Natural RPC server.

4. Modify the environment variable:

UNIX:	Add the `$SAG/iaf/v`*XX*`/lib/` directory to the `$LD_LIBRARY_PATH` (`$SHLIB_PATH` on HP-UX systems) environment variable. *XX* stands for the current version number.
Windows:	Add the `%ProgramFiles%\Software AG\EntireX\Bin` directory to the `%PATH%` environment variable.