Appendix C: Natural for UNIX Installation

- Installing APXcomponent for Natural UNIX Protocol

- Installing Natural for ApplinX on OpenVMS Hosts

- Closing the Natural Application and Natural in Error Situations

- Restrictions

Installing APXcomponent for Natural UNIX Protocol

When the host system on which you run your Natural applications is a UNIX system, additional software for the Natural UNIX protocol has to be installed on the host. The Natural UNIX protocol modules are shipped together with Natural for UNIX.

The protocol enables:

- The option to change the host password via the ApplinX Base Object.

- Support for a non standard screen size.

- User permissions are now received directly from the host. Note that currently it is not possible to log in to the host without providing a user name and password unless you use a customized User Exit authentication function.

- Host keys are automatically recognized by ApplinX and do not need to be identified according to their pattern.

- It is now possible to define the authentication method and authentication localization texts.

- The configuration has been improved. It is possible to specify the Natural parameter file in ApplinX Designer, using the same port for a number of applications in parallel, without making changes in the host.

The following topics are covered below:

- Prerequisites

- Setting up the APX Component

- Directories

- Configuration Files

- Working with the APX UNIX Component

Prerequisites

Supported Operating Systems: The same platforms as those supported by Natural for UNIX.

Other Software Products: Natural Version 6.3.2 or above for UNIX.

Linker: A linker (for example, lD or cc) and the command make must be available in the system.

Setting up the APX Component

> **Setting up APX on UNIX consists of the following steps:**

1. **Stop the NAT UNIX apxsrvd Daemons**

   Note:
   This step is only required for an upgrade installation. It is not required when you install Natural for UNIX for the first time.

   Stop the apxsrvd process using one of the following commands: `apxsrvd.sh servicename stop`

   or

   `apxsrvd.sh portnumber stop`

   Repeat this command for each service that has been started.

2. **Establish the Environment**

   Ensure that the environment definitions, as described in setup.txt in the root directory of the Natural CD, are correct and set.

   Note:
   Special note for Red Hat Enterprise Linux AS: On Red Hat Enterprise Linux AS, only PAM authentication, working with local-machine users is supported (for example it is not possible to use PAM with WinBind for active directory authentication. Implement UserExit1 to use PAM this way). Therefore you have to create a file */etc/pam.d/apxsrv* containing the following two lines:

   `#%PAM-1.0 auth required /lib/security/pam_unix.so nullok`

3. **Install Natural with APX Component**

   ApplinX Natural UNIX protocol is installed during the Natural installation.

   Important:
   The Natural installation provides a APX option which must be activated. For more details, see your Natural installation documentation.

   When you install Natural with APX, the directory $NATDIR/$APXNODE is created. The template files located in $NATDIR/$NATVERS/apx/node-name are then copied to this new directory. The APX daemon $NATDIR/$NATVERS/apx/bin/apxsrvd requires a TCL shared library which is delivered in the $NATDIR/$NATVERS/lib directory. It is linked to the runpath /opt/softwareag/nat/$NATVERS/lib (for Natural Version 6.3) or /opt/softwareag/Natural/v*<version>*/lib (as of Natural Version 8.3), and is installed with permissions

6755 (s-bit). As the s-bit is used, $LD_LIBRARY_PATH is not searched. Therefore, ensure that the apxsrvd daemon can locate the TCL shared library. This can be done by installing Natural in /opt/softwareag and setting a symbolic link from /opt/softwareag to your current $SAG directory, or making the TCL shared library available from a system directory.

4. **Check the Environment Variables for APX**

   The APX-specific settings are shown below:

Environment Variable	Description
APPLINX_ROOT	Home directory.
APXNODE	Name of the node on which APX is installed.
APXSERV	Name of the path to the apxservice file.
APXTIMEOUT	Number of seconds that the daemon waits for an input from the ApplinX server side (SO_TIMEOUT).

Directories

The following directories are created when Natural is installed together with ApplinX on a UNIX system:

Directory	Description
$NATDIR	Top-level Natural directory.
$NATDIR/$NATVERS	Directory with all components for the current Natural version.
$NATDIR/$NATVERS/apx	Directory with the APX components.
$NATDIR/$NATVERS/INSTALL	Shell scripts and environment files to required to install the product.
$NATDIR/$NATVERS/apx/bin	APX executable files.
$NATDIR/$NATVERS/apx/node-name	Contains the template files (services.dat, apxservice, etc.).
$NATDIR/$NATVERS/apx/samples/userexit	Contains the files for building the sample user exit.
$NATDIR/$NATVERS/bin/build	Contains the library (libapx.a) to link with Natural.
$NATDIR/$NATVERS/bin/build.tr	Contains the trace library (libapx.a) to link a trace version with Natural.
$NATDIR/$APXNODE	Contains the configuration files (services.dat, apxservice, etc.).

Note:
The above table lists only the most important directories and files.

Configuration Files

When the APX installation finishes, the directory $NATDIR/$APXNODE will contain the following configuration files:

Configuration File	Description
apx.sh	Shell script to start the Natural application.
apxsrvd.sh	Shell script to start and to stop the daemon.
apxsrvd.conf	Contains the parameters required to configure the authentication method and the localization text.
Apxenv	Default environment script file for bash.
apxenv.csh	Default environment script file for cshell.

Working with the APX UNIX Component

Any Natural application can be used with ApplinX.

Starting a New Natural Application

> **To start a new Natural application, proceed as follows:**

1. Create a new parameter file using the Natural Configuration Utility (see the Natural documentation) and modify the STACK command as follows: `logon library; startprogram; fin`

2. Locate a service/port number which is not in use.

3. When necessary configure the apx.sh shell script. This script is called from the APX daemon in order to start a Natural session. It has the following content:

```sh
#!/bin/sh

# Extract the arguments
IP_ADDR=$1
CLIENT_ID=$2
PARAMETERS=$3
CUSTOM=$4

# Trace
#PT_TRACELEV=i6000
#SAGTMP=$HOME/tmp
#export PT_TRACELEV
#export SAGTMP

# Natural Data Transfer
NSWUCI_FD=$UNIUCI_FD
export NSWUCI_FD
#

$NATDIR/$NATVERS/bin/natapx parm=$PARAMETERS etid=$$ >/dev/null 2>&1
```

apx.sh Script Arguments

The shell script will receive the following arguments:

IP_ADDR

The client IP address from where the session is opened.

Note:
If there is a proxy, this will not be the IP address of the ApplinX server workstation. Instead, it will be the IP address of the proxy.

CLIENT_ID

When connecting with the ApplinX server, the value of this argument will always be ApplinX.

PARAMETERS

The Natural parameters file as it appears in Natural, without the file extension. This file is passed to the parm argument in the Natural command line. The value of the PARAMETERS argument is taken from the ApplinX server configuration (as defined in the ApplinX Designer, in the Host Configuration dialog box).

CUSTOM

For future use.

4. When necessary configure the apxsrvd.conf file:

The configuration file apxsrvd.conf contains information that the user exits need for the APX daemon. It includes the following content:

```
[UserExits]
; UserExit1=/FS/sag/nat/apxexuex/userexit1/libapxuserexit1.so

[PasswdArguments]
Parameters=

[PasswdMessages]
EnterOldPassword=Enter existing login password:
NewPassword=New Password:
ReEnterNewPassword=Re-enter new Password:
PasswordSuccessful=passwd: password successfully changed for*
```

Sections in Configuration file

[UserExits]

The following user exit can be defined:

UserExit1

The library that is defined by UserExit1 contains the following function:

```
int apx_CheckUsernameAndPassword(const char *pUsername, const
char *pPassword, const char *pNewPassword, char *pErrorMessage)
```

If the key UserExit1 is defined in the configuration file, the function `apx_CheckUsernameAndPassword` is responsible for checking the user name and password. If a new password is received, user exit 1 is also responsible for changing the password.

When there is an error, the return code of the function must be "0"; in this case, the `pErrorMessage` is returned to the client. When the user name and password are correct, the return code must be a value other than "0".

[PasswdArguments]

The key Parameter is used to define any additional parameter(s) that have to be passed to the `passwd` command. For example:

```
passwd -r ldap
```

[PasswdMessages]

The keys in this section define the messages that are to be returned by the system (passwd command) when a user changes the password. If any of these messages are not identified by the daemon, an error will be returned to the client.

Password Mechanism

The password and new password are encrypted on the client side and decrypted on the UNIX side. A maximum of 8 characters is allowed.

When user exit 1 is active, the user name, password and new password parameters are passed to the user exit. When user exit 1 is not active, the daemon checks whether the user name and password are correct for the system. If a new password is sent, the daemon changes the password by calling the UNIX command passwd.

Starting and Stopping the ApplinX Daemon

The APX daemon is responsible for accepting new sessions. This daemon can be started and stopped using one of the following commands:

```
apxsrvd.sh servicename [start|stop]
```

```
apxsrvd.sh portnumber [start|stop]
```

Note:
The daemon must be started on a service/port which is not yet in use.

Troubleshooting

To troubleshoot problems in the APX component's functionality it is possible to define tracing its activity. The daemon and the Natural application with the APX lib can be traced separately. In order to determine which of the two components' activity should be traced follow these guidelines: when the problem seems to be to do with connecting the ApplinX server to the APX daemon (including authentication problems), trace the daemon component. For any other problems trace the Natural application. Note that it is not recommended to use the trace file in the production environment on a regular basis.

> **Tracing the APX daemon:**

1. Go to the $NATDIR/$APXNODE directory.

2. Edit the apxsrvd.sh file.

   Find EXECUTABLE=apxsrvd, and change it to EXECUTABLE=apxsrvd.tr

3. Define the following environment variables (the following is an example of how these should be defined in sh/bash):

   ```
   > PT_TRACELEV=i6000
   > SAGTMP=$HOME/tmp
   > export PT_TRACELEV
   > export SAGTMP
   ```

   The trace files are created in the $SAGTMP directory. In the example above this is $HOME/tmp.

4. Restart the daemon:

   ```
   > apxsrvd.sh servicename stop
   > apxsrvd.sh servicename start
   ```

   or

   ```
   > apxsrvd.sh portnumber stop
   > apxsrvd.sh portnumber start
   ```

> **Tracing the Natural application with the APX lib:**

1. Go to the $NATDIR/$APXNODE directory.

2. Edit the apx.sh file.

   $NATDIR/$NATVERS/bin/natapx parm=$PARAMETERS etid=$$ >/dev/null 2>&1

   Change natapx to natapx.tr

   Uncomment (by removing the # from the beginning of the rows) the TRACE section.

The trace files are created in the $SAGTMP directory.

By default it is $HOME/tmp. It will take effect when next connecting to the daemon.

Installing Natural for ApplinX on OpenVMS Hosts

If the host system on which you run your Natural applications is an OpenVMS system, additional software for ApplinX has to be installed on the host. The ApplinX OpenVMS modules are shipped on the Natural OpenVMS CD. In general, ApplinX uses the default system parameter values provided with the OpenVMS system.

- Prerequisites

- Setting Up the ApplinX Components

- Directories

- Configuration File

- Setting Up and Activating the APXSRVD Daemon

Prerequisites

Supported Operating Systems: The same platforms as those supported by Natural for OpenVMS.

Other Software Products: Natural for OpenVMS 6.3.8 or above.

Setting Up the ApplinX Components

Setting up ApplinX on OpenVMS consists of the following steps:

1. Stop the ApplinX Daemons: This step is only required for an upgrade installation. It is not required when you install ApplinX for the first time.

    Stop the apxsrvd process using the following command:

    ```
    stop servicename
    ```

    Repeat this command for each ApplinX service that has been started.

2. Establish the Environment: Ensure that the environment definitions, as described in readme.txt in the root directory of the Natural CD, are correct and set.

3. Install Natural and ApplinX: ApplinX is automatically installed during the Natural installation. When you install Natural and ApplinX, the directory NATDIR:[apxnode] is created, where apxnode contains the system name. The template files located in NATDIR:[natvers.APX] are then copied to this new directory.

4. Check the Environment for ApplinX: Besides the logical names NATDIR and natvers as defined by Natural, ApplinX needs the following logical names which are created during the installation of Natural:

Logical Name	Description
apxnode	Contains the system name.
VAXC$PATH	Contains the physical device specification of NATDIR:[natvers.BIN]

    Example:

    Define VAXC$PATH="ALF9$DKB500:[NATURAL.V41212.BIN]"

    In addition, the logical names NATOW and NATFE are redefined during the start of the daemon process to point to the ApplinX images NATAPXnatvers.EXE and NATFEAPXnatvers.EXE.

Optional. If function keys and message lines are to be displayed in their native format (i.e. as normal text), set the environment variable APX_PF_MSG_LINES_NATIVE_FORMAT to "YES":

Define APX_PF_MSG_LINES_NATIVE_FORMAT="YES"

If APX_PF_MSG_LINES_NATIVE_FORMAT is not set or if its value is not "YES", function keys and message lines are detected automatically (default). If they are to be treated in a special way, you have to define the basic rules Function Keys and Message Line in the same way as for a mainframe screen.

5. Check the OpenVMS UAF Parameters: Check the OpenVMS UAF (user authorization file) parameters listed in the table below and, if necessary, modify them. The size values of these parameters for the APXSRVD daemon are important. The sizes depend on the number of sessions that are started by the daemon. Natural error messages will occur if the size limits are reached or if no more quotas are available. In these cases, you have to increase the values so that they meet your needs.

Parameter	Recommended Value
Fillm	Approximately 50 for each session with ApplinX.
Prclm	50
Bytlm	Approximately 30000 for each session with ApplinX.
BIOlm	10000

6. Define the TCP Port Number: The UCX service with the TCP port number must be defined in the system as follows:

$ UCX SET SERVICE APXDEMO /PORT=22370 /FILE="" /USER="" /PROC=""

Instead of APXDEMO and the above port number, you can also specify other values. For example, you can create or define the TCP service name APXAPPL1 with the port number 25000.

Directories

The following directories are created when Natural is installed together with ApplinX on an OpenVMS system:

Directory	Description
NATDIR	Top-level Natural directory.
NATDIR:[natvers]	Directory with all components for the current Natural version.
NATDIR:[natvers.INSTALL]	Shell scripts and environment files to install the Natural product.
NATDIR:[natvers.BIN]	ApplinX executable files NATFEAPXnatvers.EXE, NATAPXnatvers.EXE and APXSRVDnatvers.EXE.
NATDIR:[natvers.FNAT]	Contains the old Natural demo application SYSEXAPX for ApplinX.
NATDIR:[apxnode]	Contains the configuration files APXSRVD_servicename.COM, APXSRVD_servicename.LOG, SERVICES.DAT and START_APXSRVD.COM.

The files APXSRVD_servicename.COM and APXSRVD_servicename.LOG are created when the ApplinX daemon APXSRVDnatvers.EXE is started with the procedure START_APXSRVD.COM.

servicename is the UCX service as defined in the file SERVICES.DAT.

natvers indicates the version number and patch level of the corresponding Natural version.

Configuration File

The configuration file SERVICES.DAT is located in the directory NATDIR:[apxnode], where the apxnode contains the node name (for example, NATDIR:[ALF9]SERVICES.DAT).

The content of this configuration file is one line for each defined TCP service:

servicename username natural parm1 ... parmn

servicename	Must be the same name as used in the TCP port number definition (see above).
username	Not used.
natural	This is the program name which must not be changed.
parm1 … parmn	Dynamic Natural parameters.

Example:

```
apxdemo sag natural parm=mypar bp=bp1
apxapp1 sag natural parm=app1 bp=bp1
apxapp2 sag natural parm=app2 bp=bp2
```

Note:
If the APXSRVDnatvers daemon does not detect Natural's dynamic parameter ETID, the daemon automatically adds the ETID to the list of the dynamic parameters to be passed to Natural. The ETID added by the daemon has the format ETID=number_ username. It is truncated if the string exceeds 8 characters.

Setting Up and Activating the APXSRVD Daemon

The BYPASS privilege must be authorized for the account to start the ApplinX daemon. The BYPASS privilege must also be set for the daemon process.

When TCP port number and service have been defined (UCX) and the SERVICES.DAT template file has been modified according to your requirements, you can start the APXSRVD daemon to use ApplinX.

To start the daemon, invoke the DCL procedure START_APXSRVD.COM as follows:

@START_APXSRVD.COM service natvers

service contains the name of the service as defined with UCX.

natvers defines the Natural version and patch level.

If both parameters service and natvers are omitted, the defaults APXDEMO and the current Natural version are used. The command procedure creates the temporary file APXSRVD_ servicename.COM which sets up the environment and creates all logicals for ApplinX and starts the daemon.

Once the daemon has been started, the file APXSRVD_ servicename.LOG is created. This file contains information (including the errors) about the daemon.

APXSRVD_ servicename.COM and APXSRVD_servicename.LOG are located in the directory NATDIR:[apxnode].

Note:
The account which starts the daemon must hold the privilege IMPERSONATE as the default privilege. It is not sufficient to have an authorized privilege.

Closing the Natural Application and Natural in Error Situations

There are error situations in ApplinX which force the termination of the Natural session. For example, when the user chooses the close button in the upper right corner of the ApplinX window, the communication with the Natural application is disconnected immediately. Then also the Natural process finishes execution immediately without running through the normal close down code of the application. This may lead to inconsistencies in the system, for example, if the corresponding entry for the Natural process remains in the database user queue.

ApplinX returns error codes to Natural which can be used in a Natural error handling routine. To make sure that the close down code of your application is always executed, write an error handling routine which identifies the ApplinX fatal errors or add this code to your existing error handling routine. For critical errors, move the command FIN to the top of the Natural stack in order to finish and close everything correctly.

The ApplinX critical errors are:

| 6296 | Fatal error in the communication. The communication will be disconnected immediately. |
| 6297 | Fatal error allocating memory. |

Example: Application Program

This Natural program reads the employees file stored in Adabas. The program moves ERRGEN to *ERROR-TA to check any Natural error.

```
DEFINE DATA LOCAL
1 VIS VIEW OF EMP
   2 AA-1
END-DEFINE
MOVE "ERRGEN" TO *ERROR-TA
READ VIS
  DISPLAY AA-1
END-READ
END
```

Example: Error Handling Routine ERRGEN

This Natural program with the name ERRGEN checks the Natural error number. If the error number is 6296 (fatal error in the communication), the program stacks the command FIN on top of the Natural stack. The Natural execution finishes immediately and all databases are closed.

```
DEFINE DATA LOCAL
1 ERRNUMBER (N4)
1 LINENUMBER (N4)
1 STATUS (A1)
1 LEVEL (A2)
1 GNPGACTU (A8)
END-DEFINE
INPUT (SG=OFF) ERRNUMBER LINENUMBER STATUS GNPGACTU LEVEL
IF ERRNUMBER=6296
  STACK TOP COMMAND "FIN"
END-IF
END
```

Restrictions

There are several restrictions when using the presentation clients with Natural applications on UNIX hosts:

- **Runtime errors in Natural applications**

  Runtime errors in Natural UNIX applications are handled by the default NSWUCIET error transaction. The user can define another error transaction by setting the *ERROR-TA Natural system variable. See the Natural documentation for details. Sample Natural error transaction:

  ```
  DEFINE DATA
  LOCAL
  1 ERR_INFO
    2 ERR_NR(N5)
    2 ERR_LINE(N4)
    2 ERR_STAT(A1)
    2 ERR_PNAM(A8)
    2 ERR_LEVEL(N2)
  END-DEFINE
  INPUT ERR_INFO
  DISPLAY ERR_INFO
  STACK TOP COMMAND 'E'
  END
  ```

  Starting with Natural 6.1, a default error transaction named NSWUCIET will be used if Natural is running with ApplinX and an error transaction has not been set by the user.

- **Return to the Natural main screen**

  You cannot use Natural applications that return to the Natural main screen. This always leads to wrong screen display and a loss of the session.

- **Natural editors and utilities**

You cannot use the Natural utilities such as SYSMAIN, SYSDDM and editors such as NATEDIT. This always leads to wrong screen display and a loss of the session.

- **Natural system commands**

  You cannot use any Natural system command such as CATALL, FIND, GLOBALS, HELP, KEY, LIST, SCAN or XREF. This always leads to wrong screen display and a loss of the session.

- **Natural commands SETUP and RETURN**

  You should not use the Natural commands SETUP and RETURN as this may lead to a loss of the session. • Terminal commands Terminal (%) commands are not supported. They do not work when entered in a presentation client.

- **Internal REINPUT**

  The error messages of an internal REINPUT are not displayed in the presentation clients (for example, if you enter the number 500 into a I1 field: NAT1142). Examples of such messages:

NAT1142	Input results in integer value overflow.
NAT1125	Too many significant digits in numeric input value.
NAT1143	Input does not correspond to input edit mask.
NAT1011	Requested function key not allocated.

- **Natural system variable *INIT-ID**

  When using the presentation clients with Natural applications on UNIX hosts, the Natural system variable *INIT-ID will not be filled with a value for the terminal type. Instead, it will contain the value notty.