

Natural under Com-plete/SMARTS

This document describes how to operate Natural in a Com-plete environment.

The following topics are covered:

- Use of the Abend Exits
- Storage Usage
- Support of Back-end Programs
- Com-plete Support in Natural Batch Runs
- Support of Asynchronous Natural Processing
- Invoking Natural from User Programs
- Storage Thread Key Handling
- Support of User Exit Handling during Session Initialization
- Use of the SMARTS Server Environment

See also:

- For further details of the Com-plete product, refer to the Com-plete documentation set.
 - For details concerning the following topics, refer to the Natural Installation Guide for Mainframes:
 - Structure and Functionality of the Natural Com-plete/SMARTS Interface
 - Prerequisites
 - Installation Tape for the Natural Com-plete/SMARTS Interface
 - Installation Procedure for the Natural Com-plete/SMARTS Interface
 - Installation of the Natural Server under Com-plete
 - Using the Com-plete *ULIB Function
 - Installation Verification
 - Customizing a Natural Com-plete Environment
 - The Natural utility SYSTP provides various TP-monitor-specific functions (see SYSTP Utility).
 - See also Natural under Com-plete Abend Codes in the Natural Codes and Messages documentation.
-

Use of the Abend Exits

The ABEXIT exits can generally be deactivated by setting SPIEA=NO in NCFPARM.

The ABEXIT exit is called during Com-plete's EOJ handling for an abnormal program termination processing.

By default, an 0CX abend is interpreted by the ABEXIT exit routine.

- Running with DU=ON/SNAP/ABEND, the Natural session is dumped and correctly terminated with error message NAT9974.
- Running with DU=FORCE, the ABEXIT exit routine is disabled, an immediate dump during Com-plete is produced.

If DU=OFF, Natural responds with error message NAT0954, NAT0955 or NAT0956, and the entire abend PSW and the Registers 0 to 15 are contained in the IOCB at offset x'290'.

Storage Usage

At session initialization, the amount of space defined with parameter NTHSIZE in NCFPARM is allocated as thread GETMAIN above or below the 16 MB line, depending on the parameter THABOVE, for usage by Natural.

The WPSIZE profile parameter determines the sizes of below and above work pools. By default, the size of the below subpool is set to 32 KB.

Therefore, you must catalog the Natural Com-plete front part with the Com-plete utility ULIB, RG size = 36 KB or larger.

The remaining areas within the Com-plete thread parts below and/or above (Com-plete ULIB RG= *specification* and/or THABOVESIZE= *specification*) are used by Com-plete for the following things:

- user subroutines,
- increasing of variable buffers inside the Natural thread,
- subproducts doing "physical" GETMAIN requests, this enforces the Natural work pool allocation.

For more details concerning the ULIB RG and THABOVESIZE parameters, refer to the *Com-plete Utilities* documentation.

Support of Back-end Programs

Natural passes the following string to a back-end program:

- the Natural return code (fullword),
- the Natural termination message (A72),
- the length of the termination area (fullword),
- the termination data.

This string is mapped by the NAMBCKP macro.

The XNCFBACK source module is an example of a Natural back-end program in a Com-plete environment. It is written as reentrant code and can be loaded as RESIDENTPAGE program or once per user.

Com-plete Support in Natural Batch Runs

If you use the Com-plete services in a Natural batch run, the batch user ID remains logged on at the end of the batch run.

To avoid this situation, include the module COMPBTCH from the Com-plete distribution library in the batch Natural nucleus. This resolves the entry point for module EOJ, which is called at the end of the Natural batch job for termination clean-up.

The module NCFAM is used to access Com-plete print/work files. It has to be included in the linking of the Natural nucleus, together with the module COMPBTCH from the Com-plete distribution library.

Support of Asynchronous Natural Processing

Asynchronous Natural processing is discussed in the section Asynchronous Processing in the Natural Operations for Mainframes documentation; however, some additional considerations apply when running Natural under Com-plete.

Make sure that appropriate SENDER and OUTDEST destinations are specified for an asynchronous Natural session; otherwise, any output will lead to an abnormal termination.

An example to start an asynchronous Natural transaction under Com-plete can be found in the library SYSEXTP, program ASYNCOMP.

Invoking Natural from User Programs

The Com-plete FETCH function is used to invoke Natural from a user front-end program under Com-plete; see the *Com-plete Application Programmer's Manual* for details.

Storage Thread Key Handling

If you want to use protection mode between Com-plete and your application program, you must set the profile parameter SKEY=OFF in the Natural parameter module NATPARM. The application program runs in the corresponding thread key. For any Natural or Editor buffer pool call, the front-end driver switches into the appropriate key and back to the thread key after the call.

You can improve the performance of the application program dramatically under Com-plete by activating the Storage-Protection Override facility on your machine.

Set the thread key = 9 in the Com-plete startup parameter THREAD-GROUP for your Natural sub-group.

The front-end driver sets the Natural application automatically to the privileged mode if the thread key is 9, and uses the SPKA instruction for the key switch handling instead of using the Com-plete function MODIFY with function codes THRD/TCS.

Support of User Exit Handling during Session Initialization

During session initialization, it is possible to pass user-specific session information about the activation of a user exit to Natural. The exit is called before Natural has been initialized, after the driver/IOCB initialization is complete.

The driver passes as a parameter the address of the IOCB in Register 1, whereas the exit is activated/deactivated by the Com-plete functions COLOAD/CODEL; see the *Com-plete Application Programmer's Manual* for details.

The NCFUEXIT source module is an example of a user exit. The user exit can be defined in the parameter module NCFPARAM.

Use of the SMARTS Server Environment

With the SMARTS Server Environment, it is possible to use the SMARTS portable file system as a container for input and output files as well as datasets on the native file system. It depends on the setting of the SMARTS parameters `CDI_DRIVER` and `MOUNT_FS` whether the environment variable refers to a the portable file system or to a native file system. For more information, see the *SMARTS Installation and Operations Manual*.

If environment variables are not defined, the normal datasets are accessed as described in the section *Datasets Used by Natural under OS/390 Batch* in the *Natural Operations for Mainframes* documentation.

The following topics are covered below:

- Input/Output
- Print File/Work File

Input/Output

Input/output in the SMARTS Server Environment is performed by DLL `NCFBTIO`.

`NCFBTIO` must be loaded into the resident area. If `NCFBTIO` is loaded into the application program thread, the Natural session is terminated with `NAT9980`.

Supported environment variables:

- `CMPRINT` - Primary Report Output
- `CMSYNIN` - Primary Command Input
- `CMOBJIN` - Input for Natural `INPUT` Statements

These environment variables are described below.

CMPRINT - Primary Report Output

Syntax:

```
CMPRINT=/pathname /filename[ / ], [mode]
```

Where

pathname specifies the location of the output file.

If *pathname* refers to a portable file system, the path will be created; if it refers to a native dataset, it must be available.

filename specifies the name of the output file.

An asterisk (*) as the file name means that the file name is generated from the actual user ID.

If *pathname* refers to the native file system and *filename* is terminated with the slash character "/", the sequential dataset *pathname/filename* will be accessed; if it is not terminated with "/", the member *filename* in dataset *pathname* will be accessed.

Example:

Assume `/fs/` is mapped to the native file system and `/pfs/` is mapped to a portable file system.

CMPRINT = /fs/natural/test/print Member print in dataset natural.test is accessed.

CMPRINT = /fs/natural/test/print/ Sequential dataset natural.test.print is accessed.

CMPRINT = /pfs/natural/test/print Member print in /natural/test of the portable file system is accessed.

CMSYNIN - Primary Command Input

Syntax:

```
CMSTYNIN=/pathname/filename[ / ]
```

Specifies the *pathname* and *filename* of the appropriate command input file.

If *pathname* refers to the native file system and *filename* is terminated with the "/" character, the sequential dataset *pathname/filename* will be accessed; if it is not terminated with "/", the member *filename* in dataset *pathname* will be accessed.

CMOBJIN - Input for Natural INPUT Statements

Syntax:

```
CMOBJIN=/pathname/filename[ / ]
```

Specifies the *pathname* and *filename* of the appropriate data input file.

If *pathname* refers to the native file system and *filename* is terminated with the "/" character, the sequential *dataset pathname/filename* will be accessed; if it is not terminated with "/", the member *filename* in dataset *pathname* will be accessed.

Print File/Work File

Print file and work file access in the SMARTS Server Environment is performed by DLL NCFWFAPS.

NCFWFAPS must be loaded into the resident area. If NCFWFAPS is loaded into the application program thread, the Natural session is terminated with NAT9980.

Supported environment variables:

- NAT_PRINT_ROOT - Path to the printer files on a PFS or native file system.
- NAT_WORK_ROOT - Path to the work files on a PFS or native file system.

Syntax Example:

```
NAT_WORK_ROOT=/qualifier/path1/path2
```

Where

qualifier determines whether a SMARTS portable file system or a native, OS-managed file system will be accessed.

path1/path2 is the path to the location of the file in the appropriate file system.