

Natural CICS Interface Functionality

This part of the Natural CICS Interface documentation describes the functionality of the Natural CICS interface. It covers the following topics:

- NCISTART - Natural CICS Interface
- Natural Nucleus under CICS
- System Control under CICS
- OSCOR/GETVIS - Natural Components in CICS Dynamic or Operating System Storage
- Natural Storage Threads under CICS
- Natural Roll Facilities under CICS
- CICS Roll Facilities
- Natural Local Buffer Pool under CICS
- Natural Swap Pool under CICS
- NCITIDEX Terminal ID Exit Interface
- NCIUIDEX - User ID Exit Interface
- Natural CICS Interface Debugging Facilities

Related Documents

- **Installation** - refer to Installing the Natural CICS Interface in the Natural Installation Guide for Mainframes.
- **Utility** - refer to the Natural utility SYSTP which provides various TP-monitor-specific functions
- **Operation, Individual Components** - for information on operation and the individual components of Natural in a CICS environment, see also:
Node Error Program Considerations for Natural | CICS 3270 Bridge Considerations | Special Natural CICS Functionality | Natural CICS Sample Programs | NCIUIDEX User ID Exit Interface | Invoking Natural from User Programs | Asynchronous Natural Processing under CICS | Logging Natural Sessions under CICS | Performance Considerations | Natural CICS Interface Debugging Facilities | Natural Work Files Under CICS

NCISTART - Natural CICS Interface

The Natural CICS Interface NCISTART is implemented in command level Assembler, thus allowing Natural to be compatible with the CICS Multiple Region Option and the debugging facility CEDF.

NCISTART controls session initialization, roll-in restart (in pseudo-conversational mode), terminal I/O, database access, ABEND processing, Natural local buffer pool calls and the loading, linking to and releasing of external subroutines. In addition, all roll I/O operations are made from NCISTART.

Natural Nucleus under CICS

The Natural nucleus is a combination of the reentrant Natural module and various support routines, which are delivered as source programs requiring site-dependent assemblies and as load modules.

The CICS-related components of the Natural nucleus are:

- the Natural CICS Interface NCISTART,
- the Natural CICS parameter module NCIPARM,
- the NaturalCICS interface object-only part NCINUC.

System Control under CICS

Natural features specific to CICS include the organization of dynamic storage in threads and the additional capability of handling these threads so that the Natural CICS System Control Program can more efficiently handle dynamic storage.

The Natural CICS System Control Program was initially developed to overcome the 64 KB GETMAIN limit under CICS. It provides complete storage allocation and management functions, including roll file I/O operations and relocation functions for pseudo-conversational users.

In order to enhance the pseudo-conversational processing capabilities of Natural with CICS, the System Control Program uses threads, a contiguous amount of storage which is set up for each user. This structure allows Natural to manage dynamic storage with minimal CICS involvement.

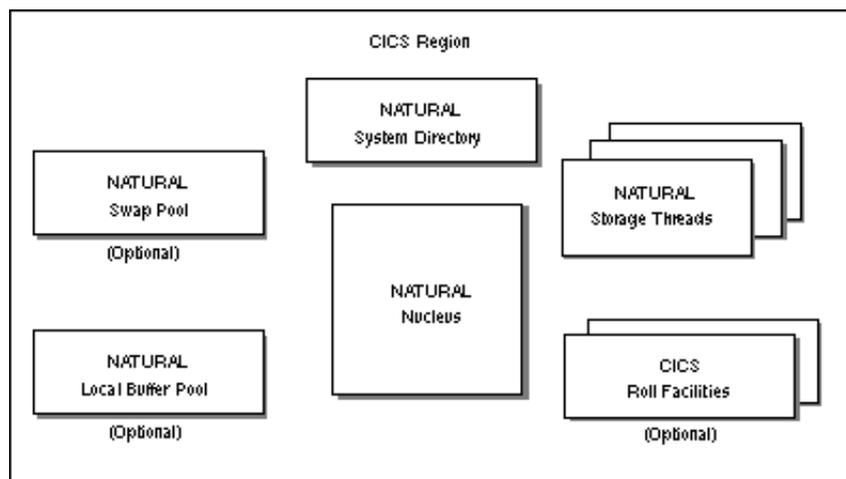
A complete understanding of system control can be attained from the following discussion of its structure and operation. Ensure that you understand this mechanism before starting the installation procedure of Natural under CICS.

OSCOR/GETVIS - Natural Components in CICS Dynamic or Operating System Storage

Scenario 1: Single CICS Region

The diagram below shows the components of the Natural system that reside in CICS dynamic storage. The components are explained under the following headings:

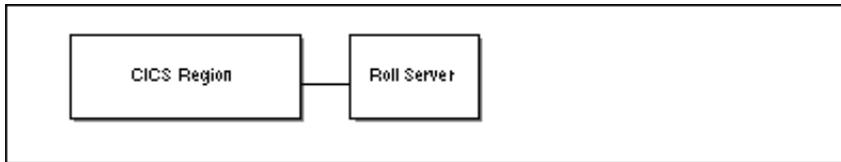
- Natural Storage Threads under CICS
- Natural Local Buffer Pool under CICS
- Natural Swap Pool under CICS
- Natural Roll Facilities



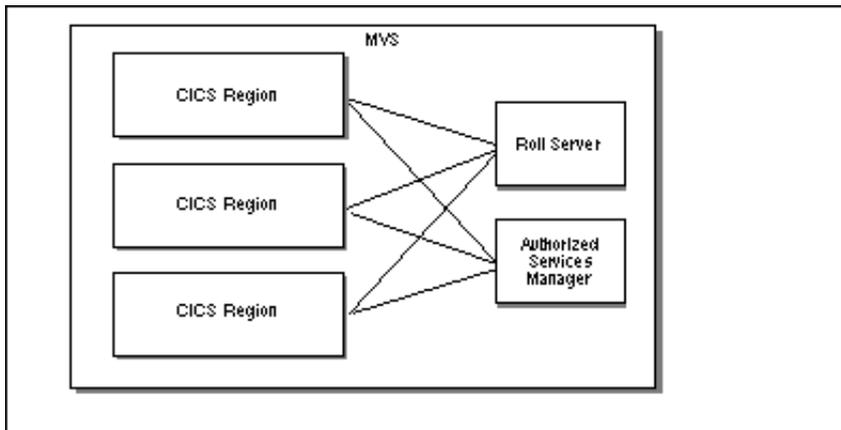
Scenario 1 applies when running Natural locally in a single CICS application region under OS/390 or VSE/ESA.

Platform:	Requirement:
OS/390 only	Additional scenarios are possible. The following three diagrams show combinations of OS/390 systems, CICS regions, the Natural Roll Server and the Natural Authorized Services Manager.

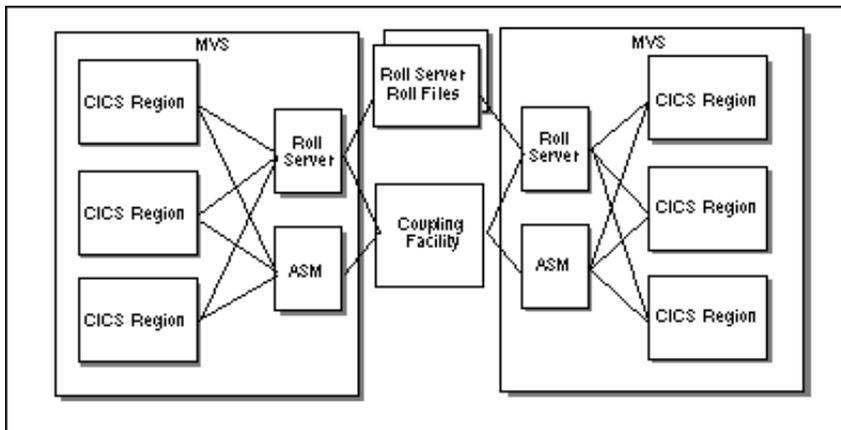
Scenario 2: Single OS/390 With Single CICS Region, Single Roll Server



Scenario 3: Single OS/390 With Multiple CICS Regions, Single Roll Server and (Optional) Authorized Services Manager



Scenario 4: Multiple OS/390 With Multiple CICS Regions, Multiple Roll Servers/Authorized Services Managers



In this scenario, you have to use the ADASVC Version 6.2.n and the Adabas link routine of the Adabas/CICS interface (ACI) Version 6.2.n.

Parameter Settings Required for the Above Scenarios

Module	Scenario 1	Scenario 2	Scenario 3	Scenario 4
(NT)BPI	TYPE=SWAP,SIZE= <i>nnn</i>	n/a	n/a	n/a
NCMDIR CICSPLX	NO	NO	YES/MODE	YES/MODE
NCMDIR SIPSERV	NO	NO/YES	YES	YES
NCMDIR ROLLSRV	NO	YES	YES	YES
Roll Server CF structure name	n/a	none	none	<i>name</i>
Authorized Services Manager/SIP	n/a	n/a	SIP slot number/size	XCF group name/ CF structure name

The Natural CICS interface requires a SIP slot size of 256 bytes.

Note:

For the scenarios 2, 3 and 4, the very first Natural session initializing the NCI environment must have the SUBSID parameter set to the value of the corresponding Roll Server and/or Authorized Services Manager.

Natural Storage Threads under CICS

A thread is a contiguous storage area from where Natural requests all its required storage. It can either be storage shared by several Natural users or, in 31-bit mode environments, CICS user storage above the 16 MB line dedicated to a specific task.

Each storage thread can be seen as the "address space" for a Natural user. Each memory allocation request issued by the Natural nucleus is transferred to the system control program to be satisfied from the storage thread.

Storage threads are allocated when the Natural CICS interface is initialized. They are allocated in a CICS region or partition, in which case they are permanent (shared) threads or they are allocated during the start of a Natural CICS task, in which case they are exclusive threads (task-dependent user storage).

The technique of storage threads was implemented with Natural for the following reasons:

- To overcome the 64 KB limitation of CICS for user storage in non-31-bit mode systems.
- To be able to optimize rolling (formerly, each piece of user storage had to be written to the roll medium; now, as there is a contiguous storage area, this area is compressed by making the relevant portions contiguous to each other before rolling out).
- The Natural CICS interface tries to satisfy all GETMAIN requests of a Natural session from its thread. This is faster than GETMAIN requests by means of CICS service calls. This is particularly true for CICS command level calls, as the CICS EXEC Interface Program (EIP) is involved, too.

A thread is released by the owning task with every screen I/O. This is true for both conversational and pseudo-conversational tasks. When a session is resumed, its storage is rolled into a thread again, unless its storage is still there; that is, no other task used the thread in between.

The Natural thread selection algorithm balances thread usage to minimize roll I/Os. This means that the more threads there are, the better is the chance of finding the old data thus preventing a roll-in. However, the more threads there are, the more paging the operating system must perform to keep all threads efficiently in real storage.

Threads are grouped together depending on their size and their type; that is, whether they have been pre-allocated as permanently shared storage or via a GETMAIN request. The decision on which kind of thread group to use, is controlled by the CICS transaction code at session initialization time. All storage threads belonging to the same group have the same size.

The thread should be defined as small as possible; see also the Buffer Usage Statistics function of the Natural utility SYSTP (described in the Natural Utilities documentation). However, the thread must still be large enough to hold the session with the largest sizes.

If you have separate Natural development and production environments, the rule is to have more smaller threads in the production environment (to serve production requests as soon as possible) and fewer larger threads in the development environment (as Natural programmers normally need larger Natural sizes and have longer "think times").

The very first Natural session allocates all permanent (shared) threads.

Natural Roll Facilities under CICS

As permanent storage threads are shared by several users and as larger threads allocated via GETMAIN should not be kept for too much time, a Natural task releases its thread with each terminal I/O. Previously, however, the user data have to be saved to be able to restart the Natural session after the terminal I/O has been performed.

Session data can be saved by using

- the Natural Roll Server with its local roll buffer and roll files;
- the CICS Roll Facilities;
- the Natural swap pool.

See also the various component scenarios. For more information, see Roll Server (described in the Natural Operations for Mainframes documentation).

CICS Roll Facilities

CICS Roll Facilities are local CICS storage facilities. They can be either CICS main or auxiliary temporary storage or VSAM relative record datasets (RRDS) which the user has previously defined to CICS. These files allow Natural to store a user's compressed dynamic storage when a roll-out occurs.

When a swap pool is used, the CICS roll facilities only serve as backup for the swap pool. The choice of the roll medium is of greater importance when no swap pool is used, since it affects Natural performance and throughput.

Every CICS service request causes CICS system overhead. So, the larger the CISIZE/record size for the roll facility is, the less CPU overhead occurs due to fewer CICS service calls to roll a Natural session. On the other hand, larger CISIZE/record size also means more VSAM buffer space allocated for the roll facility.

See Performance Considerations for further information on roll facilities.



When using the Roll Server, the swap pool and the CICS Roll Facilities are not available.

Natural Local Buffer Pool under CICS

The Natural local buffer pool contains all Natural modules during execution and copies of Natural modules once they have been loaded from the Adabas or VSAM system file.

The local buffer pool must be large enough to minimize the number of Natural program loads. However, if the local buffer pool is too large, this means wasted storage and may introduce paging overhead.

The local buffer pool is allocated as GETMAIN storage, i.e., EXEC CICS GETMAIN SHARED with all CICS Transaction Server versions or a GETVIS request with CICS/VSE in VSE/ESA. Sufficient storage must be available in the partition or in the relevant CICS DSA.

A local buffer pool is optional, as Natural can also run with a global buffer pool, which can be shared with other Natural environments like batch Natural (OS/390 and VSE/ESA) or Natural/TSO or Natural/IMS (under OS/390 only).

Natural Swap Pool under CICS

The Natural swap pool offers the possibility to "swap" a compressed Natural session from the thread into a main storage area instead of doing expensive roll I/Os.

The swap pool is allocated as GETMAIN storage, i.e., EXEC CICS GETMAIN SHARED with all CICS Transaction Server versions or a GETVIS request with CICS/VSE in VSE/ESA. Sufficient storage must be available in the partition or in the relevant CICS DSA.

The options for the swap management are set in the Natural CICS source module "NCISPCB" and by using the Natural profile parameter BPI.

The size, name and cache size of the swap pool are specified using profile parameter BPI or the corresponding macro NTBPI in the Natural parameter module NATPARM, that is, the (NT)BPI settings in effect for the Natural session initializing the NCI environment are taken.

For further details on the swap pool, see Natural Swap Pool (described in the Natural Operations for Mainframes documentation) and Using the Natural Swap Pool under CICS.

OS/390 systems:

The swap pool can only be used when running Natural under CICS locally in a single CICS region. However, even in such a scenario, you should consider using the Roll Server instead, because it runs asynchronously to the CICS region and because it can provide more roll buffers in its data space than the swap pool.

When using the Roll Server, the swap pool and the Roll Facilities are not available under CICS.

NCITIDEX Terminal ID Exit Interface

While the NCIUIDEX/NATUEX1 user exit interface can be used to set the internal terminal ID system variable *INIT-ID, the NCITIDEX terminal ID exit interface can be used to set the terminal ID which is then used by the Natural CICS interface NCI further on, i.e., this "logical" terminal ID will be used as part of the real/physical terminal ID to register the Natural session (refer to the SYSTP utility, User Sessions) and instead of the temporary storage key for that session.

Then this logical terminal ID is also the default value for the *INIT-ID system variable.

The NCITIDEX terminal ID exit interface gets control earlier than the NCIUIDEX/NATUEX1 user exit interface, therefore the system variable *INIT-ID set by the NCITIDEX exit interface can be modified once more by the NCIUIDEX/NATUEX1 user exit interface.

This is important to know when you are running Natural transactions under a CICS session manager.

Restrictions

1. The Natural terminal ID fields are 8 characters long. So the NCITIDEX terminal ID exit may set an 8-character terminal ID. However, the CICS terminal IDs are just 4 characters long and they are unique per CICS region. Therefore as the Natural CICS interface NCI takes only the first 4 characters of the logical terminal ID for its purposes, it lies in the user's responsibility to make sure that all logical terminal IDs are unique regarding the first 4 characters.

2. Certain Natural CICS interface functions cannot work if the first 4 characters of the logical terminal ID do not match the physical terminal.

As a consequence,

- you cannot send a message by way of message switching to a logical terminal,
- you cannot use the SYSTP utility or NEP to flush a session at a logical terminal.

NCIUIDEX - User ID Exit Interface

Natural provides a user exit interface to determine whether or not a user is authorized to use Natural. The name of this user exit is NATUEX1.

NATUEX1 - User Exit

NATUEX1 is called using standard calling conventions (registers 13, 14, 15 and 1) whenever a Natural user session is activated (see the following section).

For the CICS environment, the standard calling conventions are not sufficient to issue CICS request calls and to obtain addressability of CICS control blocks. Therefore, the load module NCIUEX1 is delivered as an interface. This module calls the user exit NCIUIDEX (formerly CMUIDEX) using the standard linkage conventions, but in addition passing CICS related addresses into other registers: R6 (TCTTE), R4 (EIB), R5 (EISTG).

Thus, if you want to issue requests requiring addressability of the CICS environment, the NCIUIDEX user ID exit interface should be used rather than the standard NATUEX1 interface. Source module XNCIUIDX contains a sample user ID exit.

Important: With each installation of a new CICS release, the NCIUIDEX interface must be reassembled and linked.

Natural CICS Interface Debugging Facilities

The following topics are covered:

- Using the TPF Parameter
- Using Asynchronous Natural Sessions

Using the TPF Parameter

The dynamic parameter TPF=(TPF1,TPF2,TPF3,TPF4,TPF5,TPF6,TPF7,TPF8) can be set for driver-specific options by specifying "1" for the corresponding option.

Supported options are:

TPF1	Invoke Adabas linkage module via EXEC CICS LINK with Adabas parameter in TWA and CICS COMMAREA rather than via DCI. Enables debugging of Adabas-related problems via CEDF.
TPF3	Dump the whole Natural buffer pool. With this parameter setting, the entire Natural buffer pool is included in a CICS transaction dump. Note: Usually the Natural buffer pool is not required in a dump, as all objects from the buffer pool relevant to a session are dumped anyway; so this option may only be required in the case of a buffer pool problem.
TPF4	Dump the whole EDITOR buffer pool. With this parameter setting, the EDITOR buffer pool is included in a CICS transaction dump.
TPF6	Handle terminal I/O errors by NCI With this parameter setting, NCI will not pass control back to Natural for terminal I/O errors, but will handle it by itself, which results in one of the error messages NT06 - NT13.
TPF7	Force abend in case of NCI system errors With this parameter setting, a program check is forced in case of NSxx, NLxx, NRxx or NUSnnnn error messages. This is particularly helpful when a debugging tool intercepting abends is active. Then the error can be analyzed directly online.

When specifying "0" (which can also be omitted), the corresponding option is not set, for example:

TPF=(0,0,0,1) which is equivalent to TPF=(,,1)

Using Asynchronous Natural Sessions

If the first 5 characters in the dynamic parameter string for starting Natural are "ASYN,", the Natural CICS interface will always setup an asynchronous Natural session, regardless of whether the session is terminal-bound or not.

This may be helpful for testing purposes, particularly with EDF or with other debugging tools installed.