Natural Shared Nucleus under z/OS and z/VSE

This document refers to the Natural shared nucleus under z/OS and z/VSE only.

- Environment-Independent Nucleus
- Creating a Shared Nucleus
- Installing a Shared Nucleus
- Linking Subproducts to the Nucleus
- Single-Environment Shared Nucleus
- Environment-Dependent Nucleus
- Statically Linked Non-Natural Programs
- Dynamically Called Non-Natural Programs

Environment-Independent Nucleus

Natural can be split into two functional parts: an environment-independent nucleus and an environment-dependent nucleus.

The environment-independent part of the shared nucleus can reside in the shared area of the operating system; that is,

- in z/OS environments: the link pack area (LPA) or extended link pack area (ELPA),
- in z/VSE environments: the shared virtual area (SVA).

By executing from these special areas of the operating system, the independent nucleus can be commonly accessed (shared) by multiple address spaces (that is, regions or partitions), for example, CICS, Com-plete, TSO and batch mode, within the same operating system.

Components of the Shared Nucleus

The following basic modules must be linked together to build the independent (shared) Natural nucleus:

Module	Function
NATSTUB	Natural stub module.
NATURAL	Natural compiler and runtime.
NATCONFG	Natural configuration module.
NATCMOD	Bundling module of C routines (server calls).
NATBPMGR	Natural buffer pool manager.
NAT2SORT	Natural Sort for all systems (if you wish to use a sort program, either Natural's internal one or an external one). It is also possible to place NAT2SORT in a load library from where it can be loaded dynamically at runtime; this requires that NAT2SORT is specified with the profile parameter RCA.
NATRPC*vr* or NTRPC	Natural RPC runtime. **Note:** If more than one version of this module is available, see the current *Natural Release Notes* for the available Natural RPC versions. If only a single version of this module is available, see the installation job to link a shared nucleus in the *Installation* documentation for the actual module name.
NATEDIT	Natural program editor and map editor.
NATTEXT	Natural syntax.
NATTXT2	Natural keywords.
NATTXT3	Substitution fragments for Natural error messages.
NATPM	Natural print mode.
INPL	INPL module.
NATEDT	Software AG Editor module.
NATLAST	Final include.

Terminal Drivers and Batch Mode Modules

The following modules are optional:

Module	Function
NATTTY	Natural Teletype Support
NAT3270	3270 Terminal Support
NAT3279	3279 Terminal Support
NATWEB	Web I/O Terminal Converter; see *Unicode Input/Output Handling in Natural Applications*, *Web I/O Interface*, in the *Unicode and Code Page Support* documentation.
NATBTCH	Natural Batch Module

Modules Required for Unicode and Code Page Support

For a list of the mainframe-specific modules to be linked for Unicode and Code Page Support, refer to *Configuration and Administration of the Unicode/Code Page Environment*, *ICU Library* in the *Unicode and Code Page Support* documentation.

Module Required for REQUEST DOCUMENT and PARSE XML Statement Support

Module	Function
`NATXML`	Nucleus Routine Module

For further information, see *Installation Steps for REQUEST DOCUMENT and PARSE XML* in the Natural *Installation* documentation.

Linking Additional Modules

Linking of additional modules may be required, for example, common user exits or user-defined programs used by all Natural regions. The entry name of the linked module must be `CMSTUB`.

Benefits of a Shared Nucleus

The benefits of a shared nucleus are:

- virtual storage relief;
- less paging activity, as there is only one copy of the nucleus in the system;
- less maintenance, as Zaps must be applied only once.

By removing the environment-independent parts of Natural and placing them in the shared nucleus, you achieve a significant reduction of the size of the environment-dependent nucleus, since only the environment-dependent part is loaded into the batch or TP-monitor address space, and the shared nucleus is accessed from the operating system’s link pack area.

Since less storage is required by a Natural batch job, this results in less paging and better overall performance of the operating system. The more batch jobs that concurrently access the shared nucleus, the greater the savings.

As is the case with batch environments, Natural running in an online environment can also access the same common nucleus. In production environments which, for example, run Natural under multiple CICS regions, the savings in virtual storage can be substantial.

There are also benefits when you apply corrective fixes to the Natural nucleus, since you only need to apply these ZAPs once to the shared nucleus, which is then accessed by the multiple environments (for example, CICS, Com-plete, TSO and batch).

Additional benefits are possible if you use products such as Natural for VSAM, Natural for DB2, Natural for DL/I or Natural Advanced Facilities, since these products are all eligible to execute from the shared nucleus. When installing these products, you would simply place the `INCLUDE` statements specific to these products into the link-edit of the shared nucleus.

Administration Aspects

In any module installed in the LPA/ELPA or SVA, Zaps cannot be applied online, because the LPA/ELPA or SVA is write protected. Under z/OS, you can use the operator command `SETPROG` to load a new copy of the shared nucleus into the LPA/ELPA. However, to test corrective fixes in a specific environment, it is recommended that you use one of the following methods which can be adapted to suit your site-dependent needs:

Environment	Requirement
Batch Mode	Link-edit the shared nucleus to a load library which you add to the `STEPLIB` concatenation. The operating system will access this copy of the shared nucleus instead of the copy in the shared area.
CICS	Link-edit the shared nucleus to a load library which you add to the `DFHRPL` concatenation in the CICS startup procedure. This allows CICS to load the shared nucleus from your `DFHRPL` library instead of from the shared area. You need to modify the `ALT` (alternate load table) entry for the shared nucleus to read `SHR=NO` so that CICS will access the `DFHRPL` libraries instead of the shared area. Users of CICS Version 3.3.0 and above make this change to the PPT entry for the shared nucleus instead, since the `ALT` has been eliminated in these releases of CICS: Specify `USELPACOPY (NO)` in z/OS and `USESVACOPY (NO)` in z/VSE, respectively, for this program definition.
Com-plete/TPF	Link the shared nucleus to your Com-plete user program library and add the shared nucleus to the list of `RESIDENTPAGE` programs in your Com-plete `SYSPARMs` or load the shared nucleus dynamically as `RESIDENTPAGE`.
TSO	Link-edit the shared nucleus to the same load library that contains the TP-dependent nucleus for Natural under TSO. When the `CLIST` is executed, the operating system will access this copy of the shared nucleus instead of the copy in the shared area.
IMS TM	Link-edit the shared nucleus to a load library which you add to the `STEPLIB` concatenation in your procedure used for executing the IMS TM application region. When Natural is started, the operating system will access the shared nucleus from `STEPLIB` instead of from the shared area.

Creating a Shared Nucleus

The shared nucleus is created via the linkage editor in the SMA Job NATI060 as an optional part of the base Natural installation.

When setting up the linkage editor `INCLUDE` statements for the shared nucleus, it is important to carefully follow the installation instructions outlined in the Natural *Installation* documentation.

A common error is to omit or add link-edit `INCLUDE` statements to the shared and/or non-shared nucleus, which can cause unpredictable results when you attempt to start a Natural session. If this happens, please review the installation instructions and if necessary, call Software AG support for assistance.

Installing a Shared Nucleus

The installation of the shared nucleus is described in the Natural *Installation* documentation in the installation sections for the various Natural TP monitor interfaces included in the *TP Monitor Interfaces* documentation. The following points should be noted in general:

- The shared nucleus is created by an additional link step. The target library for this link can be any library, in which the operating system loader searches for executable modules. For test purposes, it may be easier to first link the shared nucleus in one of the libraries in your `STEPLIB` (or `SEARCH` chain) and later into an `LPA` (or `SVA`) library for production. To avoid confusion, you should delete the module in the `STEPLIB` library when linking it into the `LPA` library.

- The name of the shared nucleus to be used is specified with the profile parameter `NUCNAME` in the Natural parameter module when installing the environment-dependent part. It is possible to specify `NUCNAME` as a dynamic parameter in the primary parameter input, but not in the `PROFILE` or `SYS` parameter strings.

Linking Subproducts to the Nucleus

Most Software AG subproducts can be linked either to the environment-independent Natural nucleus or to the environment-dependent part. Refer to the installation instructions of your subproducts.

The following Natural subproduct, however, must be linked to the environment-dependent part and cannot be linked to the shared nucleus:

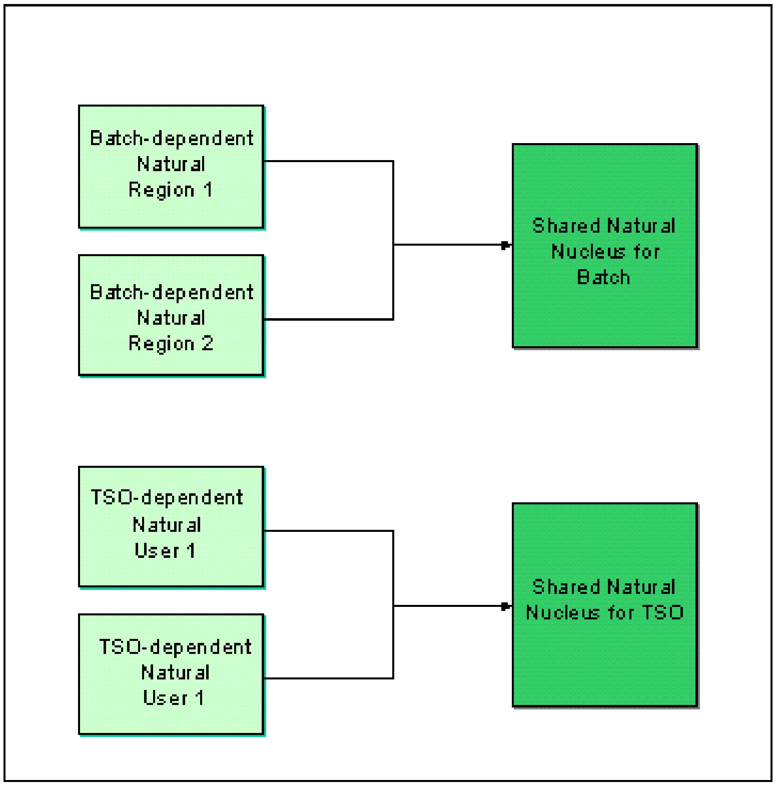
- The Adabas link routine (`ADALNK` or `ADAUSER`)

For a few other products, separate portions need to be linked to the shared nucleus as well as to the environment-dependent part. This is documented in detail with the respective subproducts.

Single-Environment Shared Nucleus

Some subproducts of Natural require that TP-specific modules be included in the Natural nucleus. In this case, you need to create one Single-Environment Shared Nucleus for each operating environment (for example, one for batch mode, one for TSO and one for CICS.) The advantage is still that all batch regions or all TSO users share their own Natural nucleus.

The following diagram shows an example for this situation:


Batch-dependent Natural Region 1
Batch-dependent Natural Region 2
Shared Natural Nucleus for Batch
TSO-dependent Natural User 1
TSO-dependent Natural User 1
Shared Natural Nucleus for TSO

As this concept of Single-Environment Shared Nuclei can always be installed, Software AG's System Maintenance Aid (SMA) generates this type of shared nucleus if the parameter `SHARED-NUC` is set to `Y`.

If all your single-environment shared nuclei are identical and do not contain TP-monitor-specific modules, you can then go from a single-environment shared nucleus to a multi-environment shared nucleus.

Environment-Dependent Nucleus

In addition to the environment-independent part of the shared Natural nucleus, every single Natural region runs one or more environment-dependent module(s), which differ(s) according to the actual environment; that is, Com-plete, CICS, IMS TM, TSO, or batch mode. The environment-dependent part receives control at the beginning of a session and checks whether the Natural nucleus is linked. If not, the shared nucleus is loaded or located and communication is established.

The following modules must be linked together to build the dependent part of Natural specific to each environment:

- the Natural environment-specific interface (that is, `NCFNUC`, `NATCICS`, `NATIMS`, `NATTSO` or `NATOS/NATVSE`) together with other interface-related modules;
- the environment-specific Natural parameter module `NATPARM`;
- Natural subproduct modules with entries defined in the internal `CSTATIC` list via macro `NTINV`, or specified as `CSTATIC` in the Natural parameter module;

- non-Natural programs defined as CSTATIC in the Natural parameter module.
- work-file and print-file modules for Com-plete, TSO or batch mode.

Statically Linked Non-Natural Programs

The Natural parameter module NATPARM contains the list of all non-Natural programs to be statically linked. This list consists of an internal part defined by the macro NTINV and an external part defined by the CSTATIC parameter. Each entry of the list consists of a program name and a V-constant which must be resolved by linking the corresponding module to the Natural parameter module.

The internal list is permanently present in the NATCONFG module of the independent nucleus and is used if no parameter module is linked to the independent module. If there are non-Natural programs statically linked to the independent nucleus, a parameter module must be linked, too, where all these programs are defined.

Optionally, an alternative parameter module can be specified via the PARM parameter. An alternative parameter module has precedence over a linked parameter module. At session initialization time, up to three lists of statically linked programs are merged together. The base list for this merge is that of the actual parameter module, which means that only its entries are used. V-constants not resolved in this list are tried to be satisfied by the environment parameter module if an alternative parameter module is used. V-constants not resolved in the environment parameter module are tried to be satisfied by the environment-independent nucleus.

If a non-Natural program is to be statically linked to the independent nucleus, it must be specified in a parameter module linked to the independent nucleus as well as in the parameter module actually used for the session.

Additionally, "dynamic" linking of non-Natural programs defined for being statically linked is possible during initialization of a Natural session. Refer to the description of the RCA profile parameter for further details.

Dynamically Called Non-Natural Programs

Instead of statically linking a non-Natural program, you can also call it dynamically at execution time by using the Natural CALL statement. In this case, however, the program must not be defined as statically linked.

When the CALL statement is executed, Natural tries a dynamic load and call operation with the help of the environment (sub)system (for example, with EXEC CICS LINK under CICS).