Natural

Installation for z/VSE

Version 8.2.7

October 2017

ADABAS & NATURAL

This document applies to Natural Version 8.2.7 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: NATMF-INSTALL-VSE-827-20180201

Table of Contents

Preface

This documentation describes the installation of base Natural and Natural add-on products on z/VSE.

Basic Information: ■ Installation Process and Major Natural Features
Base Natural: ■ Installing Natural
Base Natural - Optional Components: ■ Installing International Components for Unicode for Software AG ■ Installation for REQUEST DOCUMENT and PARSE XML Statements ■ Installing Natural Net Data Interface ■ Installing Entire System Server Interface ■ Installing Software AG Editor
TP Monitor Interfaces: ■ Installing Natural CICS Interface ■ Installing Natural Com-plete/SMARTS Interface
Database Management System Interfaces: ■ Installing Natural for DL/I ■ Installing Natural for SQL/DS ■ Installing Natural for VSAM
Other Natural Add-On Products: ■ Installing Natural Security ■ Installing Natural Advanced Facilities ■ Installing Natural Optimizer Compiler ■ Installing Natural Connection ■ Installing Natural Review ■ Installing Entire Transaction Propagator

I Installation Process and Major Natural Features on z/VSE

1 Installation Process and Major Natural Features on z/VSE

This document provides general information on the prerequisites and processes required to install base Natural and Natural add-on products. In addition, it describes installation tools and major Natural components required for installation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

General Prerequisites and System Support

Before beginning the installation process, consider the following:

- Be sure to read the current Natural *Release Notes* for Mainframes for information on software and hardware requirements, known issues and changes to the documentation. These *Release Notes* apply to base Natural and Natural add-on products.
- A supported version of the operating system on which Natural is to run must be installed. For the supported operating systems and versions, refer the **Product Version Availability** section of Software AG's Empower web site at ***https://empower.softwareag.com/***.
- A supported version of the TP monitor/online interface used with Natural must be installed. For the supported versions, refer to *TP Monitors/Online Interfaces* in the current Natural *Release Notes* for Mainframes.
- A supported version of Adabas must be installed to store the Natural system files. See also ***Natural System Files***.

  For the supported versions, refer to *Database Management Systems* in the current Natural *Release Notes* for Mainframes.
- A supported version of each database management or file system used to store the user data processed with Natural must be installed.

  For the supported versions, refer to *Database Management Systems* in the current Natural *Release Notes* for Mainframes.

Note: For information regarding Software AG product compatibility with IBM platforms and any IBM requirements for Software AG products, review the ***Software AG IBM Product Availability*** web page.

Installation Medium

The installation medium (for example, tape or CD-ROM) distributed for Software AG mainframe products contains all data sets required to install base Natural and the Natural add-on products.

The software required for the optional Natural components are contained in the data sets supplied for base Natural. The software required for the Natural add-on products are contained in separate product data sets which are listed in the product-specific sections of the *Installation for z/VSE* documentation. In addition to the product data sets, the installation medium can contain the latest fix updates for the supplied products.

The names of the product data sets begin with a product code that identifies each product, as in the following table:

Product Code	Product Name
ETP	Entire Transaction Propagator
NAF	Natural Advanced Facilities
NAT	Natural
NCF	Natural Com-plete/SMARTS Interface (corresponds to Natural Com-plete Interface)
NCI	Natural CICS Interface
NCJ	Natural Japanese Language Pack
NDL	Natural for DL/I
NOC	Natural Optimizer Compiler
NSC	Natural Security
NSQ	Natural for SQL/DS
NTC	Natural Connection
NVS	Natural for VSAM
RNM	Natural Review

Software AG Product Delivery Report

Each installation medium is delivered with a Software AG Product Delivery Report providing the following information:

- A list of all data sets contained on the medium.
- The sequence in which the data sets are located on the medium.
- Attribute descriptions of each data set.

Installation Method

The installation of Software AG products on z/VSE is performed by installation jobs that contain the JCL required to identify the job to the operating system and run the job.

There are two methods for creating and running the installation jobs:

- using the jobs generated by System Maintenance Aid (SMA), or
- using the jobs created from the sample installation jobs provided.

The *Installation for z/VSE* documentation solely describes the installation procedure for the jobs generated by SMA. If you do not use SMA for installation, refer to the example installation jobs supplied on the installation medium.

SMA is supplied with base Natural.

System Maintenance Aid (SMA)

For each step of the installation procedure, System Maintenance Aid (SMA) generates an installation job according to your specifications in SMA. You then submit and run the generated job.

Before you can start generating the jobs, you have to load the `SMT111.TABS` data set from the installation medium into the SMA system file. `SMT111.TABS` contains the tables SMA requires to build the jobs.

SMA is supplied with base Natural. For instructions on loading the data set and using SMA, refer to the *System Maintenance Aid* documentation.

Readme File

For installation guidance and information on new or changed SMA parameters and Natural features, you can view the product-specific Readme files by using the appropriate SMA function.

To view a product-specific Readme

- From the product list on an SMA **Maintenance** screen, execute the `RM` (**Show Readme File**) command for the required product(s).

Sample Installation Jobs

The sample installation jobs that can be used as an alternative to SMA are provided in a PDS library contained on the data set *product-code-vrs*.LIBJ (for example, NAT824.LIBJ) shipped on the installation medium. All sample installation jobs provided are listed and described in the README document that accompanies the shipment.

You need to adapt the sample installation jobs to your requirements.

Installation Job Identification

Each installation job indicates the **product code** and version (for example, NAT824) of the corresponding product (for example, Natural).

Each step of the installation procedure is identified by a job name (for example, I050) and one or more steps (for example, Steps 0100 and 0101 for Job I050) that indicate the tasks performed by the job. The job name can have a prefix such as a **product code** (for example, NATI050). The prefix can be specified with the SMA parameter JOB-PREFIX (the default prefix is SMA).

A sample installation job from the PDS library can also have a suffix letter which indicates a variant of the job. For example: Job I060L is a variant of Job I060 and used if support of the IBM Language Environment (LE) is required. In SMA, the same variant is executed with Job I060 and the appropriate SMA parameter setting.

Overall Installation Procedure

The installation process comprises the following:

1. Creating the Natural system files.
2. Creating the Natural parameter module.
3. Creating the Natural nucleus.
4. Loading the Natural objects.
5. Installing the optional Natural components.
6. Installing the Natural add-on products.

Installation Verification

Verify the successful completion of the installation by starting Natural and testing the system functions as described in the relevant sections of the *Installation for z/VSE* documentation.

Note: If Natural Security is installed, certain Natural functions and libraries can be restricted to specific users.

INPL Utility

The installation instructions frequently refer to the Natural INPL utility which is used to load the data sets (for example, `NAT`*`vrs`*`.INPL`) contained on the Natural installation medium into the **Natural system files**. The INPL utility is invoked with the Natural system command `INPL`. For detailed information on the INPL utility, refer to the *Utilities* documentation.

Natural Nucleus Components

The Natural nucleus consists of two functional parts: the environment-independent nucleus and the environment-dependent nucleus.

Note: If you maintain different versions of Natural, you must use distinctive names for the nuclei to identify each version.

This section covers the following topics:

- Environment-Independent Nucleus
- Environment-Dependent Nucleus
- Modules for Static Linking
- Modules for Dynamic Loading

- Modules Called Dynamically

Environment-Independent Nucleus

The environment-independent nucleus contains components that are independent of the operating system or TP system (online interface) being used. The same instance of the environment-independent nucleus can be used in different online and batch environments in different partitions. The environment-independent nucleus is reentrant.

The environment-independent nucleus can reside in the shared virtual area (SVA) where it can be shared between different partitions.

A module (such as the environment-independent nucleus) loaded into the SVA is protected against modification. Therefore, tests for modifications of the environment-independent nucleus should be performed in a separate environment.

If the environment-independent nucleus resides in the SVA, multiple batch jobs or TP regions (for example, CICS) share the same instance of the environment-independent nucleus. This results in a significant reduction of paging activities and virtual storage consumption.

Modules for Linking

The following modules must be linked to the environment-independent nucleus:

- Modules for base Natural
- Environment-independent modules of Natural add-on products
- Environment-independent user-supplied modules

When using System Maintenance Aid (SMA), the required modules are linked to the environment-independent nucleus during the appropriate installation job/step. Modules that can optionally be linked are mentioned in the *Installation Procedure*.

The installation of the environment-independent nucleus is described in *Link the Nucleus* in the *Installation Procedure*.

Specifying the Nucleus Name

The name of the environment-independent nucleus to be used is specified with the Natural profile parameter `NUCNAME` in the Natural parameter module during the installation of the environment-dependent nucleus. You can specify `NUCNAME` as a dynamic parameter in the primary parameter input, but you cannot specify `NUCNAME` in the input strings of the Natural profile parameter `PROFILE` or `SYS`.

The Natural parameter module is described in *Building a Natural Parameter Module* in the *Operations* documentation. `NUCNAME`, `PROFILE` and `SYS` are described in the *Parameter Reference* documentation.

If you maintain different versions of Natural, we recommend that you use distinctive names for the nucleus to clearly identify each version and environment, for example: `NAT824` for the

environment-independent nucleus, NAT824C for the environment-dependent nucleus for a CICS interface, and NAT824B for the batch environment.

Environment-Dependent Nucleus

The environment-dependent nucleus contains components that depend on the operating or TP system being used.

In addition to the environment-independent nucleus, every single partition in which Natural runs requires an environment-dependent nucleus containing modules that perform actions specific to the operating or TP system. The environment-dependent nucleus assumes control from the operating or TP system at the start of a Natural session, loads the environment-independent nucleus and passes control to it.

Modules for Linking

The following modules must be linked to the environment-dependent nucleus:

- Environment-specific Natural interface modules
- Environment-specific work file and print file modules
- Environment-specific Natural parameter module (see also *Building a Natural Parameter Module* in the *Operations* documentation)
- Environment-dependent modules of Natural add-on products
- Adabas link routine (ADALNK or ADAUSER)
- Environment-dependent user-supplied modules defined as CSTATIC in the Natural parameter module. The Natural profile parameter CSTATIC is described in the *Parameter Reference* documentation.

When using System Maintenance Aid (SMA), the required modules are linked to the environment-dependent nucleus during the appropriate installation job/step. Modules that can optionally be linked are mentioned in the *Installation Procedure*.

The installation of the environment-dependent nucleus is described in *Link the Nucleus* in the *Installation Procedure*.

Modules for Static Linking

Both the Natural configuration module NATCONFG (described in the *Operations* documentation) and the Natural parameter module contain the Natural-supplied list of additional modules to be statically linked to the nucleus.

The Natural parameter module also contains the user-supplied list of additional modules to be statically linked to the nucleus as specified with the Natural profile parameter CSTATIC.

Each entry of these lists consists of a program name and a V-type address constant which must be resolved by linking the corresponding module to the Natural parameter module.

The Natural-supplied list provided with NATCONFG is used if the Natural parameter module is not linked to the environment-independent nucleus. If modules are statically linked to the environment-independent nucleus, a Natural parameter module that defines all these modules must also be linked to the environment-independent nucleus.

Optionally, you can specify an alternative Natural parameter module by using the Natural profile parameter PARM (described in the *Parameter Reference* documentation). An alternative parameter module takes precedence over a parameter module that is linked to either the environment-independent or the environment-dependent nucleus.

Merging Module Lists

During initialization of a Natural session, up to three lists of statically-linked modules (specified with the Natural profile parameter CSTATIC) are merged:

- Base list for the merge is the list of the Natural parameter module specified with the Natural profile parameter PARM;
- V-type address constants not resolved in this list are resolved using the Natural parameter module linked to the environment-dependent nucleus;
- V-type address constants not yet resolved are resolved using the Natural parameter module linked to the environment-independent nucleus.

If a user-supplied module is to be statically linked to the environment-independent nucleus, it must be specified in the Natural parameter module linked to the environment-independent nucleus as well as in the Natural parameter module specified with the Natural profile parameter PARM.

Modules for Dynamic Loading

When initializing a Natural session, you can also dynamically load the modules (supplied by Software AG or user-defined) that have been defined for static linking. For information on whether the module of a Natural add-on product is suitable for dynamic loading, read the documentation for your specific Natural add-on product.

For information on defining external names for static non-Natural programs and dynamic linking and controlling these programs, see the Natural profile parameters `RCA` and `RCALIAS` described in the *Parameter Reference* documentation.

Modules Called Dynamically

If a module is not defined for static linking, Natural attempts to load and execute the module using environment-dependent functions (for example, `EXEC CICS LINK` under CICS) when the corresponding Natural `CALL` statement is executed.

Natural System Files

The Natural system files are stored in an Adabas database.

The table below lists and describes the Natural system files that are usually available in a Natural environment. The availability of the system files and the data contained in the files depends on the Software AG products installed in addition to base Natural.

The settings for the system files are defined with Natural profile parameters of the same names (exception: scratch-pad file). You can follow the hyperlinks in the table below to read details about these parameters in the *Parameter Reference* documentation.

System File	Supplied with	File Contents
`FNAT`	Base Natural	All objects required for Natural system applications.
`FUSER`	Base Natural	User-specific objects required for user-defined applications.
`FPROF`	Base Natural	Parameter profiles specified by the profile parameter `PROFILE`, provided no database information is supplied as subparameter of `PROFILE`.
Scratch-pad file	Base Natural	Data that is not stored explicitly as a Natural object in another system file. See also *Natural Scratch-Pad File* in the *Operations* documentation.
`FDIC`	Base Natural	Natural Data Definition Modules (DDMs). If Predict is installed, `FDIC` also contains data for the Predict dictionary system.

System File	Supplied with	File Contents
		If the Natural Development Server is installed, FDIC also contains application data and holds object locking information.
FREG	Base Natural	Registry data that is not stored explicitly in another system file.
FSEC	Natural Security	Control information required for security definitions.
FSPOOL	Natural Advanced Facilities	Control and spooling information required to output a report on a screen or printer and obtain print statistics.

It is also possible to store Natural system files in a VSAM file system if **Natural for VSAM** is installed. The *Installation for z/VSE* documentation describes the installation steps that apply when using an Adabas database for storage.

Defining a Scratch-Pad File

Like all other system files of Software AG products, the scratch-pad file is a logical file. The logical file number of the scratch-pad file is 212.

Since there is no mnemonic for the scratch-pad file such as FNAT and FUSER or FDIC, it has to be defined:

- either statically by using the macro NTLFILE in the Natural parameter module or
- dynamically by using the Natural profile parameter LFILE.

Examples of NTLFILE and LFILE Definitions:

LFILE Parameter:

```
LFILE=(212,physical-dbid,physical-fnr,password,cipher-key)
```

NTLFILE Macro:

```
NTLFILE 212,physical-dbid,physical-fnr,password,cipher-key
```

II Installing Natural on z/VSE

2 Installing Natural on z/VSE

This document describes the steps for installing Natural (product code NAT) on z/VSE.

Related Topic:

For information on how to run Natural in a z/VSE environment, see the relevant section in the *Operations* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support***.

Installation Medium

The **installation medium** contains the following data sets required for product installation:

Data Set Name	Contents
ICS*vrs*.LIBR	Source modules, macros, phases and object modules for International Components for Unicode for Software AG (ICS)
MLC*vrs*.LIBJ	Sample installation jobs for Software AG's mainframe license check software The *vrs* in the library name represents the version of the license check software, which is not necessarily the same as the version of Natural. For detailed information on the license check software, see *Software AG Mainframe Product Licensing*.
MLC*vrs*.LIBR	Source modules, macros, phases and object modules for Software AG's mainframe license check software including the LICUTIL license utility The *vrs* in the library name represents the version of the license check software, which is not necessarily the same as the version of Natural. For detailed information on the license check software and the LICUTIL utility, see *Software AG Mainframe Product Licensing*.
NAT*vrs*.LICS	Product license file For information on the license file and product licensing, see *Software AG Mainframe Product Licensing*.
NAT*vrs*.SYSF	Natural system file definitions

Data Set Name	Contents
NAT*vrs*.LIBJ	Sample installation jobs
NAT*vrs*.LIBR	Source modules, macros, phases and object modules
NAT*vrs*.INPL	Natural objects
NAT*vrs*.EXPL	Natural example objects

The data set type and the space each data set requires on disk is shown in the *Software AG Product Delivery Report*.

Copying Data Sets to a z/VSE Disk

Copy the data sets from the supplied installation medium to your disk before you perform the individual installation procedure for each component to be installed.

The way you copy the data sets depends on the installation method and the medium used:

- If you use System Maintenance Aid (SMA), refer to the copy job instructions provided in the *System Maintenance Aid* documentation.
- If you are not using SMA and want to copy the data sets from CD-ROM, refer to the README.TXT file on the CD-ROM.
- If you are not using SMA and want to copy the data sets from tape, follow the instructions in this section.

This section explains how to copy the data sets .LIBJ, .LIBR and .LICS from tape to disk. All other data sets can be installed directly from the tape.

- Step 1: Copy Data Set COPYTAPE.JOB to Disk
- Step 2: Modify COPYTAPE.JOB on Your Disk
- Step 3: Submit COPYTAPE.JOB

Step 1: Copy Data Set COPYTAPE.JOB to Disk

- Modify the following sample job according to your requirements:

```
* $$ JOB JNM=LIBRCAT,CLASS=0,                                          +
* $$ DISP=D,LDEST=(*,UID),SYSID=1
* $$ LST CLASS=A,DISP=D
// JOB LIBRCAT
* ***************************************
*     STORE COPYTAPE.JOB IN LIBRARY
* ***************************************
// ASSGN SYS004,nnn
// MTC REW,SYS004
// MTC FSF,SYS004,4
ASSGN SYSIPT,SYS004
// TLBL IJSYSIN,'COPYTAPE.JOB'
```

```
// EXEC LIBR,PARM='MSHP; ACC S=lib.sublib'
/*
// MTC REW,SYS004
ASSGN SYSIPT,FEC
/*
/&
* $$ EOJ
```

where:

nnn is the tape address, and
lib.sublib is the library and sublibrary in which the data set COPYTAPE.JOB is to be stored.

- Execute the job to copy the data set COPYTAPE.JOB to disk.

  COPYTAPE.JOB contains the JCL required to copy the data sets .LIBJ, .LIBR and .LICS from tape to disk.

Step 2: Modify COPYTAPE.JOB on Your Disk

- Modify COPYTAPE.JOB according to your requirements and set the disk space parameters as appropriate.

Step 3: Submit COPYTAPE.JOB

- Execute COPYTAPE.JOB to copy the data sets .LIBJ, .LIBR and .LICS to your disk.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Step 1: Prepare, Convert, Assemble and Link the License File

(Job I006, Step 0104 and Job I007, Steps 0101, 0102)

You must install a valid Natural license file.

For detailed information on the license file and product licensing, see *Software AG Mainframe Product Licensing*.

1. Copy the license file from the supplied installation medium to disk or transfer it from the PC as described in *Transferring a License File from PC to a z/VSE Host Using FTP* in *Software AG Mainframe Product Licensing*.
2. Catalog, check, convert, assemble and link the license file:

Job I006, Step 0104	Catalog license file `NAT`*`vrs`*`.LICS` in the sublibrary.
Job I007, Step 0101	Check license file `NAT`*`vrs`*`.LICS`. This job runs the CHECK function of the LICUTIL license utility (see below).
Job I007, Step 0102	Convert license file into an assembler source. This job runs the MAKE function of the LICUTIL license utility (see below). Assemble and link the assembler source to generate load module `NATLIC`. This module is then linked to the nucleus in **Job I060**.

The functions and option settings provided by LICUTIL are described in *Using the License Utility: LICUTIL* in *Software AG Mainframe Product Licensing*.

Step 2: Load the FNAT System File Definition

(Job I050, Step 0100)

Skip this step if you want to use an existing Natural `FNAT` system file.

Load the new Natural `FNAT` system file definition:

1. Specify the database ID and file number of the Adabas file where to load the new `FNAT` system file definition by using the Adabas ADALOD utility.

   In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 10: Build the Natural Parameter Module***.
2. Load the `FNAT` system file definition contained in the `NAT`*`vrs`*`.SYSF` data set by using the Adabas ADALOD utility.

   The following ADALOD utility parameter must *not* be changed:

```
ISNREUSE=YES
```

The following ADALOD utility parameter setting is recommended:

```
USERISN=YES
```

This avoids the Natural errors NAT9988 and NAT7397 after reorganizing the FNAT system file using the Adabas utility ADAULD or ADALOD.

Step 3: Load the FUSER System File Definition

(Job I050, Step 0101)

Skip this step if you want to use an existing Natural FUSER system file.

Load the new Natural FUSER system file definition:

1. Specify the database ID and file number of the Adabas file where to load the new FUSER system file definition by using the Adabas ADALOD utility.

   In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 10: Build the Natural Parameter Module***.
2. Load the FUSER system file definition contained in the NAT*vrs*.SYSF data set by using the Adabas ADALOD utility.

   The following ADALOD utility parameter must *not* be changed:

   ```
   ISNREUSE=YES
   ```

3. If you want to use existing Natural applications, copy all user-written objects to the empty FUSER.
4. If you want to use Natural Application Programming Interfaces (APIs), see *Using a Natural API* in the *SYSEXT Utility* documentation for further guidance.

Step 4: Load the Scratch-Pad File Definition

(Job I050, Step 0102)

This step is only required if you want to use read-only system files. See also *Natural Scratch-Pad File* in the *Operations* documentation.

You can skip this step if you want to use an existing Natural scratch-pad file.

Load the new Natural scratch-pad system file definition:

1. Set the System Maintenance Aid (SMA) parameter NAT-SCRF to Y (Yes).

2. Specify the database ID and file number of the Adabas file where to load the scratch-pad file by using the Adabas ADALOD utility.

   In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 10: Build the Natural Parameter Module***.
3. Load the scratch-pad system file definition contained in the `NAT`*`vrs`*`.SYSF` data set by using the Adabas ADALOD utility.

   The following ADALOD utility parameter must *not* be changed:

```
ISNREUSE=YES
```

Step 5: Load the FREG System File Definition

(Job I050, Step 0104)

This step is only required if registry information must be available to control concurrent user sessions limited with the Natural profile parameter `UCONMAX` (see the *Parameter Reference* documentation).

You can skip this step if you want to use an existing Natural `FREG` system file.

Load the new Natural `FREG` system file definition:

1. Set the System Maintenance Aid (SMA) parameter `NAT-FREG` to `Y` (Yes).
2. Specify the database ID and file number of the Adabas file where to load the `FREG` system file by using the Adabas ADALOD utility.

   In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 10: Build the Natural Parameter Module***.
3. Load the `FREG` system file definition contained in the `NAT`*`vrs`*`.SYSF` data set by using the Adabas ADALOD utility.

   The following ADALOD utility parameter must *not* be changed:

```
ISNREUSE=YES
```

Step 6: Load the FDIC System File Definition

(Job I050, Step 0103)

Skip this step:

- if you want to install Predict. In this case, use the corresponding installation step described in the Predict *Installation* documentation.
- if you want to use an existing Natural `FDIC` system file.

Load the new Natural `FDIC` system file definition:

1. Specify the database ID and file number of the Adabas file where to load the new `FDIC` system file definition by using the Adabas ADALOD utility.

   In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 10: Build the Natural Parameter Module***.
2. Load the `FDIC` system file definition contained in the `NAT`*`vrs`*`.SYSF` data set by using the Adabas ADALOD utility.

   The following ADALOD utility parameter must *not* be changed:

   ```
   ISNREUSE=YES
   ```

Step 7: Load the FSEC System File Definition

(Job I050, Step 9900)

Skip this step, if you do not use Natural Security.

- If you use Natural Security, refer to ***Installing Natural Security***.

Step 8: Build the Natural Configuration Module

(Job I055, Step 0110)

This step is only required if you need to change the delivered `NATCONFG` module, for example, to adapt the `NTDVCE` macro definition to your requirements.

1. Change and assemble the source contained in the `NAT`*`vrs`* sublibrary.
2. Link the resulting Natural configuration module (`NATCONFG`) to the nucleus when you link the nucleus in ***Step 11: Link the Nucleus***.

   For information on the configuration tables in `NATCONFG`, refer to *Natural Configuration Tables* in the *Operations* documentation.

Step 9: Build the Natural-Specific IBM Language Environment Options

(Job I055, Step 0120)

Build the Natural-specific runtime options for the IBM Language Environment (LE).

This step is only required if you need to adapt the LE options at the `.VSEDEF` label in the delivered `NATLEOPT` module to your requirements.

1. Set the System Maintenance Aid (SMA) parameter `NAT-LEOPT` to `Y` (Yes); default is `N` (No).
2. Change the required LE options in the `NATLEOPT` source module contained in the `NAT`*`vrs`* sublibrary at the `.VSEDEF` label.
3. Assemble the `NATLEOPT` source contained in the `NAT`*`vrs`* sublibrary.
4. Link the resulting `NATLEOPT` module to the environment-dependent nucleus (see ***Step 11: Link the Nucleus***).

Step 10: Build the Natural Parameter Module

(Job I060, Step 0010)

Build the Natural parameter module for batch mode.

1. Modify the settings of the Natural profile parameters supplied with this job, if required. The parameters and corresponding macros (if applicable) are described in the *Parameter Reference* documentation. The most important parameter/macro settings are described below.
   - Configure the z/VSE batch interface: Modify the settings of the parameters supplied with the `NTVSEP` macro to meet your requirements.
   - Adapt the following parameters:

     ```
     FNAT=(database-id,file-number)
     FUSER=(database-id,file-number)
     FDIC=(database-id,file-number)
     ```

     where *`database-id`* and *`file-number`* are either the database ID and file number you specified when loading the new `FNAT`, `FUSER` and `FSEC` system files (see ***Step 2***, ***Step 3*** and ***Step 6***, respectively), or the database ID and file number of your existing Natural system files.

     These parameters are supplied with the `NTPRM` macro described in the *Operations* documentation.
   - If you want to to limit the number of concurrent users with the Natural profile parameter `UCONMAX`, proceed as follows:

     Supply the following parameter with the `NTPRM` macro:

```
FREG=(database-id,file-number)
```

where *database-id* and *file-number* are either the database ID and file number you specified when loading the new Natural FREG system file (see ***Step 5***), or the database ID and file number of your existing Natural FREG system file.

- If you want to use read-only system files, proceed as follows:

  Supply the following parameter with the NTPRM macro:

  ```
  ROSY=ON
  ```

  Specify the NTLFILE macro (see the parameter LFILE):

  ```
  NTLFILE 212,database-id,file-number
  ```

  where *database-id* and *file-number* are the database ID and file number you specified when loading the new Natural scratch-pad file (see ***Step 4***), or the database ID and file number of your existing Natural scratch-pad file.

- If you want to use a Natural global buffer pool, perform the following steps:

  Specify the NTBPI macro (see the BPI parameter):

  ```
  NTBPI TYPE=NAT,NAME=gbp-name
  ```

  where *gbp-name* is the name of the Natural global buffer pool to be used.

  Supply the following parameter with the NTPRM macro:

  ```
  SUBSID=subsystem-name
  ```

  where *subsystem-name* is the name of the Natural subsystem specified when creating the global buffer pool.

  Make sure that the **System Maintenance Aid (SMA)** parameter NAT-GLOBAL-BP is set to Y (Yes). This is the default setting.

  For detailed information on the Natural global buffer pool, see *Natural Global Buffer Pool under z/VSE* in the *Operations* documentation.

2. Assemble and catalog the Natural parameter module.

Step 11: Link the Nucleus

(Job I060, Steps 0020, 0105)

1. If you want Natural to run in the IBM Language Environment (LE), set the **System Maintenance Aid (SMA)** parameter `NAT-LE` to `Y` (Yes). The default setting is `N` (No).
2. Link the **environment-dependent nucleus** (Step 0020) for batch Natural.

   The list of modules to be linked for the environment-dependent nucleus is supplied with Step 0020.

   Link the environment-dependent nucleus with the linkage editor option `RMODE(24)`.

   If you want Natural to run in the IBM Language Environment (LE), include the object modules `NATVSEL`, `NATLEOPT` and `NATVSE`, and specify `ENTRY LESTART` instead of `ENTRY CMSTART`.
3. Link the **environment-independent nucleus** (Step 0105).

   The list of modules to be linked for the environment-independent nucleus is supplied with Step 0105.

   Ensure that the Natural profile parameter `NUCNAME` (see the *Parameter Reference* documentation) specified in the Natural parameter module contains the name of the module resulting from this link step.

Step 12: Link the Global Buffer Pool and Catalog the Start and Stop Jobs

(Job I060, Steps 0120, 0126, 0127, 0128, 0129)

These steps are only required if you want to use a global buffer pool. For further information on the global buffer pool, see *Global Buffer Pool under z/VSE* in the *Operations* documentation.

- Create and catalog the jobs `NGBPSTRT`, `NGBPSTOP`, `EGBPSTRT` and `EGPSTOP`:

```
Step 0120 LINK GLOBAL BUFFER POOL MANAGER NATGBP
     0126 SAMPLE JOB TO START NATURAL GLOBAL BUFFER POOL
     0127 SAMPLE JOB TO STOP NATURAL GLOBAL BUFFER POOL
     0128 SAMPLE JOB TO START EDITOR GLOBAL BUFFER POOL
     0129 SAMPLE JOB TO STOP EDITOR GLOBAL BUFFER POOL
```

Step 13: Link the Editor Buffer Pool Program

(Job I060, Step 1900)

Link the NATEDFMV program to initialize the editor buffer pool work file.

Step 14: Load New Natural Objects and Natural Error Messages

(Job I061, Step 0100)

- Load the Natural objects and Natural error messages from the NAT*vrs*.INPL data set into the Natural system files by using the Natural **INPL utility**.

  The Natural error messages comprise short and long message texts and the German (ULANG=2) short message texts. You can use the ERRUPPER program of the Natural SYSERR utility to convert the message texts to upper case.

  For details on the ULANG profile parameter and ERRUPPER, see the *Parameter Reference* and the *Utilities* documentation, respectively.

Step 15: Load the Japanese Messages

(Job I061, Step 8300, 8302)

This step is optional and only required if you want to replace the English long message texts by their Japanese equivalents or if you want to load the Japanese short message texts. The Japanese message texts are contained in the Natural Japanese Language Pack (product code NCJ), a separate product that can be loaded optionally. If you do not load the Japanese message texts, the English message texts are used instead.

1. Replace the English long message texts (Step 8300) by loading the Japanese long message texts from the NCJ*vrs*.INPL data set into the Natural system file with the Natural INPL utility.
2. Load the Japanese (ULANG=59) short message texts (Step 8302) from the NCJ*vrs*.ERRN data set into the Natural system file by using the ERRLODUS program of the Natural SYSERR utility (described in the *Utilities* documentation).

   For details on the ULANG profile parameter and ERRLODUS, see the *Parameter Reference* and the *Utilities* documentation, respectively.

Step 16: Load the Natural Example Objects

(Job I061, Step 0103)

- Load the Natural example objects from the NAT*vrs*.EXPL data set into the Natural system file by using the Natural INPL utility.

Step 17: Link and Start the Optimize Monitor Buffer Pool

(Job I009, Step 1230 and Job I200, Step 0105)

These steps are only required if you want to use the Optimize Monitor Buffer Pool described in *Optimize Monitor Buffer Pool* in the *Operations* documentation.

1. Link the NAT04182 program for the Optimize Monitor Buffer Pool (Job I009).
2. Create and start the job SAGEOI (supplied with Job I200) before using Natural. See *Starting the Optimize Monitor Buffer Pool* in the *Operations* documentation.

Installation Verification

This section provides instructions for verifying the successful installation of Natural.

Test Batch Natural

You can use the following sample JCL to invoke Natural in batch mode and check whether the Natural system files are available:

```
* $$ JOB JNM=JOBNAME,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB JOBNAME
// LIBDEF PHASE,SEARCH=(SAGLIB.USERLIB,SAGLIB.NATvrs,SAGLIB.ADAvrs),TEMP
// EXEC NATvrsBA,SIZE=NATvrsBA,PARM='SYSRDR'
DC='.',IM=D,MADIO=0,MT=0,OBJIN=R,AUTO=OFF,MAXCL=0,ID=',',INTENS=1
/*
ADARUN DB=001,DE=3390,SVC=045,MODE=MULTI
/*
EDIT
WRITE 'TESTBAT'
END
.E
RUN
SAVE TESTBAT
FIN
```

```
/*
/&
* $$ EOJ
```

This job executes a log on to the Natural library SYSTEM, creates the example program TESTBAT, and executes the program with the system command RUN.

Test Online Natural

You can use the following verification procedure to test Natural system functions in online mode and check whether the Natural system files are available.

1. Log on to the Natural user library SYSTEM:

   ```
   LOGON SYSTEM
   ```

2. Enter the following Natural system command:

   ```
   MAINMENU
   ```

3. Select **Development Functions** and enter the following:

   C in the **Code** field,
   P in the **Type** field, and
   TEST in the **Name** field.

4. In the editing area of the program editor, type the following:

   ```
   WRITE 'HELLO'
   END
   ```

5. Save the source code and exit the program editor.
6. In the **Development Functions** menu, enter the following:

   L in the **Code** field and
   TES* in the **Name** field.

7. On the **LIST Objects in a Library** screen, enter the RU line command for the TEST program.
8. Enter the following Natural system command:

   ```
   SYSDDM
   ```

9. In the SYSDDM utility menu, enter the following:

   R in the **Code** field and
   EMPLOYEES in the **DDM Name** field.

10. After pressing ENTER, enter the following:

`C` in the **Code** field,
the appropriate Natural system file number in the **FNR** field,
the appropriate database ID in the **DBID** field, and
`Y` in the **Replace** field.

The specified DDM has been adapted to your environment.

11. Repeat Steps 9 and 10 for the `VEHICLES` DDM.
12. You can check whether the DDMs `EMPLOYEES` and `VEHICLES` are now available in your environment by logging on to the Natural system library `SYSEXSYN` and executing the example programs (for example, `AEDEX1R`) with the system command `RUN`.

III Installing International Components for Unicode for Software AG on z/VSE

3 Installing International Components for Unicode for Software AG on z/VSE

This document describes the steps for installing International Components for Unicode for Software AG (ICS) on z/VSE which allows Natural to convert code pages and support Unicode.

International Components for Unicode for Software AG (ICS) requires the use of an ICS module and an ICU data library. In addition, you can use ICU data items to load ICU components that are not contained in the ICU data library. The ICS module does not have to be linked to the Natural nucleus if neither code page conversion nor Unicode support are required.

The use of ICU functionality increases the required Natural thread size.

Note: For increased flexibility, it is also possible to load the ICS module during initialization of the Natural session. A dynamically loaded ICS module overrides the statically linked ICS module. You can also load an ICU data library during initialization of the Natural session. A dynamically loaded ICU data library overrides any statically linked ICU data library.

The ICS module, the ICU data libraries and the ICU data items are explained in *Enabling Unicode and Code Page Support* in the *Unicode and Code Page Support* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The ICS module, the ICU data libraries and the ICU data items are contained on the ICS*vrs*.LIBR data set supplied on the **installation medium** for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

- Step 1: Make Phase ICSNTTAB Available for SMARTS
- Step 2: Load the ICS Module at Session Start
- Step 3: Load an ICU Data Library at Session Start

Step 1: Make Phase ICSNTTAB Available for SMARTS

Phase ICSNTTAB is required under SMARTS for reentrancy reasons.

1. Make sure that the ICS library is available at runtime of a Natural session:

   Specify the ICS library in the job control statement LIBDEF PHASE for SMARTS.
2. Define ICSNTTAB as RESIDENTPAGE to make it available for execution under SMARTS.

Step 2: Load the ICS Module at Session Start

1. Make sure that the ICS library is available at runtime of a Natural session:
   - Specify the ICS library in the job control statement LIBDEF PHASE for your Natural batch or SMARTS jobs, or for the TP monitor jobs under which Natural runs.
2. At the start of a Natural session, set the Natural profile parameter RCA as follows:

```
RCA=SAGICU
```

   RCA is described in the *Parameter Reference* documentation.

   The ICS module SAGICU is described in the *Unicode and Code Page Support* documentation.

   Instead of the ICS module SAGICU, you can also load the ICS module SAGICUA9. In addition to SAGICU, this module supports IBM architecture level 9: see alternative ICS modules in the *Unicode and Code Page Support* documentation.

   If you want to load SAGICUA9, use the following parameter setting:

```
RCA=SAGICU RCALIAS=(SAGICU,SAGICUA9)
```

   The Natural profile parameter RCALIAS is described in the *Parameter Reference* documentation.

Step 3: Load an ICU Data Library at Session Start

1. Make sure that the ICS library is available at runtime of a Natural session:
   - Specify the ICS library in the job control statement LIBDEF PHASE for your Natural batch or SMARTS jobs, or for the TP monitor jobs under which Natural runs.
2. At the start of a Natural session, set the Natural profile parameters RCA and CFICU for the ICU data library (ICSDT58E) to be used:

```
RCA=ICSDT58E  CFICU=(DATFILE=ICSDT58E)
```

`RCA` and `CFICU` are described in the *Parameter Reference* documentation.

Step 4: Load ICU Data Items on Request in a Session

This step is optional and only required if you want to dynamically load an ICU data item on request during a Natural session instead of an entire ICU data library. For more information, see the *Unicode and Code Page Support* documentation.

1. Make sure that the ICS library is available at runtime of a Natural session:
   - Specify the ICS library in the job control statement `LIBDEF PHASE` for your Natural batch or SMARTS jobs, or for the TP monitor jobs under which Natural runs.
2. Depending on your TP environment and the setting of the Natural `CFICU` profile parameter, perform one of the following options:
   - Under CICS, with `CFICU=(DATITEM=NONE)` set:

     Add one PPT entry for each ICU data item.

     See also the **corresponding step** in *Installing Natural CICS Interface*.
   - Under Com-plete and Natural Development Server, with `CFICU=(DATITEM=NONE)` set:

     Add `THREAD-ESQA-SIZE=15K` as a keyword parameter to the startup options for your Com-plete or Natural Development Server.

     Add one `RESIDENTPAGE` entry for each ICU data item that represents a converter (refer to the `ICSCCOMP` source member in the `ICS`*`vrs`*`.LIBR` data set).

Installation Verification

After the last step of the installation procedure has been completed, proceed as follows:

1. Configure and activate your Unicode and code page environment by following the instructions in *Configuration and Administration of the Unicode/Code Page Environment* and *Profile Parameters and Macros* in the *Unicode and Code Page Support* documentation.

   For information on the code pages and ICU data files available in your current Natural environment, you can use the SYSCP utility (described in the *Utilities* documentation).
2. After successful activation, you can execute the example programs described in the *Unicode and Code Page Support* documentation.

IV Installation for REQUEST DOCUMENT and PARSE XML Statements on z/VSE

4 Installation for REQUEST DOCUMENT and PARSE XML Statements on z/VSE

This document describes the installation steps for enabling the use of the Natural statements `REQUEST DOCUMENT` and `PARSE XML` on z/VSE.

Related Topics:

For information on the functions provided by `REQUEST DOCUMENT` and `PARSE XML`, see the following documents:

- *Statements for Internet and XML Access* in the *Programming Guide*
- `REQUEST DOCUMENT` in the *Statements* documentation
- `PARSE XML` in the *Statements* documentation

Prerequisites

The following requirements must be met to execute the `REQUEST DOCUMENT` and `PARSE XML` statements:

- The ICU library must be installed to convert data from one encoding to another (at least internally). For details, see the relevant section in the *Unicode and Code Page Support* documentation.
- A TCP/IP stack must be available and enabled for the execution environment.
- A DNS (Domain Name System) server or DNS services must be available in the execution environment to resolve internet addresses (`gethostbyname` function).

For Internet Protocol Version 6 (IPv6) support, the following additional prerequisites apply:

- An activated IPv6 stack must be available on the local host.
- The local network must support IPv6.
- An accessible and IPv6-capable DNS server must be available.
- For IPv6 internet communication, an IPv6 connection from the service provider must be available.
- If both IPv4 and IPv6 are used, a dual stack must be supported.
- IPv6 support must be configured with the appropriate keyword subparameters of the Natural profile parameter `XML` described in the *Parameter Reference* documentation.

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Procedure

Be sure to read *Installation Process and Major Natural Features* before you start the installation procedure.

The installation procedure comprises the following:

- Step 1: Link the Module NATXML or NATXMLA9 to the Nucleus
- Step 2: Enable the Environment-Dependent Nucleus for LE Execution
- Step 3: Link the Required Modules to the Nucleus

Step 1: Link the Module NATXML or NATXMLA9 to the Nucleus

The module `NATXML` is required to execute the statements `REQUEST DOCUMENT` and `PARSE XML`. If you want the `PARSE XML` statement to support IBM architecture level 9, alternatively, you can use the module `NATXMLA9`. See also *PARSE XML Support for Architecture Levels*.

1. Set the **System Maintenance Aid (SMA)** parameter `NATXML` to `Y` (default is `N`).
2. The `NATXML` module is then linked to the **environment-independent nucleus** by using the following `INCLUDE` statement:

```
INCLUDE NATXML
```

Or:

For support of IBM architecture level 9:

1. Set the **SMA** parameters `NATXML` and `NAT-ARCHLEVEL9` to `Y` (default is `N`).
2. The alternative `NATXMLA9` module is then linked to the **environment-independent nucleus** by using the following `INCLUDE` statement:

```
INCLUDE NATXMLA9
```

Step 2: Enable the Environment-Dependent Nucleus for LE Execution

The `REQUEST DOCUMENT` and `PARSE XML` statements require the IBM Language Environment (LE) for execution. If you want Natural to run in the IBM Language Environment (LE), perform the following steps:

1. Set the **System Maintenance Aid (SMA)** parameter `NAT-LE` to `Y` (Yes). The default setting is `N` (No).
2. Additionally, if you want to modify the LE options in the `NATLEOPT` source module or if you use non-Natural programs running in 24-bit mode, set the appropriate SMA parameter as described in ***Build the Natural-Specific IBM Language Environment Options* in *Installing Natural***.
3. Link the environment-dependent nucleus to support LE.

This applies to all batch and TP monitor system environments except Com-plete.

Step 3: Link the Required Modules to the Nucleus

Link the modules indicated in this section to the nucleus depending on the environment to be used.

- Batch and CICS
- Com-plete and SMARTS

Batch and CICS

1. Add the LE library (usually `PRD2.SCEEBASE`) to the `OBJ-SEARCH` definition of the link job to resolve the references to LE functions.
2. Link the following LE and TCP/IP access modules to the **environment-dependent nucleus** by using the appropriate `INCLUDE` statement:

   ```
   INCLUDE NAT2LE
   ```

   Do *not* specify the `NOAUTO` parameter in the `ACTION` statement for the link step.
3. If you run z/VSE Version 5 Release 1 (or above) and use a BSI TCP/IP Version 2.5.2 stack (or above), use the following statement:

   ```
   INCLUDE NAT2TCP
   ```

   If no such z/VSE versions and BSI TCP/IP stack versions are available, use the following statement:

```
INCLUDE NAT2TCP4
```

Com-plete and SMARTS

1. Link the following LE access module to the **environment-dependent nucleus** by using the appropriate `INCLUDE` statement:

```
INCLUDE NAT2LE
```

2. Copy the `NCFTCPvr` phase to the Com-plete/SMARTS load library.
3. For support of the IBM TCP/IP stack, define the CDI (Communication Driver Interface) as described in *Standard CDI Definitions* in the *Com-plete* documentation.
4. Add the `POSIX SERVER` statement to the Com-plete parameter module `SYSPARM`.

PARSE XML Support for Architecture Levels

The architecture level (the higher the better) you can use depends on the IBM hardware facility installed at your site.

The following architecture levels can be specified with the `PARSE XML` statement, International Components for Unicode for Software (ICS) and/or the Natural Optimizer Compiler (NOC):

Level Value	Supported By	IBM Hardware Facility Required
0	All	Specifies that no architecture level is used. This is the default setting for compatibility with all mainframe platforms supported by Natural.
1 to 4	All	These values are not evaluated and treated as `ARCH=0`.
5 to 6	NOC only	■ z800 or z900 Extended-Translation Facility 2 ■ z890 or z990 HFP Multiply-and-Add/Subtract Facility
7	NOC only	■ z9 to z109 Extended-Immediate Facility
8	NOC only	■ z10 General-Instructions-Extension Facility ■ z10 Execute-Extensions Facility
9	PARSE XML and ICS only	■ zEnterprise 196 Load/Store-on Condition Facility

Level Value	Supported By	IBM Hardware Facility Required
		Floating-Point-Extension-Facility Distinct-Operands Facility High-Word-Facility
10	NOC only	■ zEnterprise EC12 (zEC12) Decimal Floating-Point Facility Decimal Floating-Point Zoned-Conversion Facility
11	NOC only	■ zEnterprise z13 Decimal Floating-Point Packed-Conversion Facility

Caution: An operation exception error (abend code S0C1) can occur if code generated with an architecture level greater than 0 is executed on a machine where the corresponding hardware facility is not installed.

Installation Verification

After the last step of the installation procedure has been completed, proceed as follows:

1. Activate the statements in the runtime environment; see *Activation/Deactivation* in the section *Statements for Internet and XML Access* in the *Programming Guide*.

   For information on the profile settings that enable the support of the REQUEST DOCUMENT and/or PARSE XML statement, see the following documents:

   - *Profile Settings* in the section *Statements for Internet and XML Access* in the *Programming Guide*
   - Profile parameter XML in the *Parameter Reference* documentation
2. Try the example programs contained in the Natural system library SYSEXV.

V Installing Natural Net Data Interface on z/VSE

5 Installing Natural Net Data Interface on z/VSE

This document describes the steps for installing the Natural Net Data Interface NATNETTO on z/VSE.

Related Topic:

For operational information, see *Natural Net Data Interface NATNETTO* in the *Operations* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Device Configuration in NATCONFG

Logical net data devices are configured with the IONET flag byte, all other flag and value settings must be made as in the sample definition given below. The module entry is VCNETTO, additional device entries must be specified with WXTRN=OFF. The settings of FLAG1, FLAG2 and RTAL must be according to the example below.

If the data delimited option (NEDLM) is set, the delimiter character which separates the fields in the value buffer can be set by using the BEL keyword subparameter of the NTDVCE macro.

IONET Settings

IONET	DS	XL1	NETDATA CONTROL FLAG
NECUFNR	EQU	X'01' 1	CURSOR POSITION = FIELDNR
NEMSG	EQU	X'02' .... ..1.	SEND MESSAGE LINE (if not set, message line will be skipped)
NEABO	EQU	X'04' .... .1..	ATTRIBUTE BUFFER OPTION

NEFBO	EQU	X'08' .... 1...	FORMAT BUFFER OPTION
NEFLG	EQU	X'10' ...1	FIELD LENGTH OPTION
NEDLM	EQU	X'20' ..1.	DATA DELIMITED OPTION
NEFIX	EQU	X'40' .1..	FIXED FORMAT OPTION
NEFBOPTE	EQU	X'80' 1...	EXTENDED FORMAT BUFF. OPT.

NATCONFG already has a device entry for NATNETTO.

The TTYPE is NETF. The protocol options are set as follows:

- The value buffer structure is fixed (without delimitation between the fields).
- Format buffer, extended format buffer and attribute buffer options are set.
- The cursor position is in the field number notation.

Message line and PF-key line are suppressed.

Example:

```
NTDVCE TYP=NETF,NAME=NETTF,ENTRY=VCNETTO,MSG=BOT,          HSO6-
       FLAG1=CMNIXD,FLAG2=CMTNOPT,RTAL=255,                HSO7-
       FLAGS=(IONET,-,CO,IONET,+,NEFIX+NEFBO+NEABO+NECUFNR+NEFB-
       OPTE,WINDTITI,+,PFKNDISP)                           HSO6
```

VI Installing Entire System Server Interface on z/VSE

6 Installing Entire System Server Interface on z/VSE

The Entire System Server Interface is required if the Entire System Server or Natural ISPF is to be used.

This document describes the steps for installing the Entire System Server Interface on z/VSE. You can choose between default installation (recommended) and customized installation.

Related Topic:

For information on installing and using the Entire System Server or Natural ISPF, refer to the relevant product documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

A supported version of either of the following products must be installed before you can install the Entire System Server Interface:

- Entire System Server or Natural ISPF, version as specified under *Software AG Product Versions Required with Natural* in the current Natural *Release Notes* for Mainframes.

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Default or Customized Installation

If you want to use the default value settings in the modules `ESYNODTB` and `NATPNIP` (used by the Entire System Server and Natural ISPF), proceed with ***Link the Entire System Server Interface to the Nucleus***.

If you do *not* want to use the default value settings, edit the modules `NATPNIP` and `ESYNODTB` described in this section.

- NATPNIP

- ESYNODTB

NATPNIP

The NATPNIP module contains the following parameters and default values:

```
NAMVIEWP BUFLEN=12288,NUMREQ=5,MAXCBL=3000,MAXEDL=6000,EXTUSER=INIT-USER
```

The parameters are explained below:

BUFLEN	Length of all Adabas buffers in bytes
NUMREQ	Number of possible nested FIND loops in Natural calling the Entire System Server
MAXCBL	Complex FIND buffer length
MAXEDL	Editor session buffer length MAXEDL is used by the NSPF editor and incore database. The default value is 6000, which should be sufficient for an NSPF editor session and typical incore database applications. However, for large layouts within an incore database file that value might not be large enough and the following message is issued: NAT3077: Not enough space for extent. DB/FNR/Subcode :1:/:2:/:3:. (see the *Messages and Codes* documentation). In this case, the value of MAXEDL has to be increased.
EXTUSER	External user ID passed to the Entire System Server for security checks See also EXTUSER in the following section.

EXTUSER

The parameter EXTUSER describes how to inherit security definitions from an external security system such as RACF, ACF2 and TOP-SECRET.

The appropriate parameter setting depends on whether a multi-user address space or a single-user address space is used in your environment:

- A multi-user address space provides the option to maintain different user security definitions, for example, one for a CICS and one for a Com-plete user).
- A single-user address space supports a subsystem, for example, a Natural subtask for Entire Output Management or Entire Operations Managment, or a Natural RPC, Natural Web/IO Interface or batch server.

  Recommended values for EXTUSER are:

EXTUSER=INIT-USER	Recommended for a multi-user address space. The contents of the Natural system variable *INIT-USER must be identical to the user definiton in the external security system (for example, RACF). The Entire System Server transfers the value of *INIT-USER to the external security system, and all calls to security restricted resources are handled under this user ID. In this case, the security definition from a CICS or Com-plete user (for example) is inherited by Entire System Server, and a new logon is not required. If the value of *INIT-USER is not found in RACF (for example), an error occurs indicating that a logon is required. (*INIT-USER is described in the *System Variables* documentation.)
EXTUSER=USER	Recommended for a multi-user address space in a Natural Security environment. Processing is similar to EXTUSER=INIT-USER except that the Natural system variable *USER (described in the *System Variables* documentation) is used. (*USER is described in the *System Variables* documentation.)
EXTUSER=ADDRESS-SPACE	Recommended for a batch or server environment. The security description of this address space is inherited for security evaluation.

ESYNODTB

The ESYNODTB module contains the following parameters and default values:

```
NAMXNOD ID=148,NAME=PRODUCTION-1
NAMXNOD ID=149,NAME=PRODUCTION-2
NAMXNOD ID=1490,NAME=DBID-ABOVE-255,LAST=Y
END
```

The parameters and default values are explained below:

ID	Entire System Server node number (also known as DBID)
NAME	Entire System Server node name
LAST	Indicator for last entry in table

Optional Node Name for Entire System Server Calls

Calls to the Entire System Server from Natural are usually handled with the NODE parameter which specifies the node number to be used for the call, for example:

```
FIND ACTIVE-JOBS WITH JOB-NAME = 'ADA*' AND NODE = 148
```

If the node number is defined in the ESYNODTB module, alternatively, you can specify the logical name of the required Entire System Server with the NODE-NAME parameter, for example:

```
FIND ACTIVE-JOBS WITH JOB-NAME = 'ADA*' AND NODE-NAME = 'PRODUCTION-2'
```

Assemble the Parameter Module for the Entire System Server Interface Component

Natural ISPF

If Natural ISPF is used as the INCORE database:

- (Job I055, Step 1106)

  Link the parameter module NATPNIP. In this case, the module ESYNODTB is not required.

Entire System Server

If the Entire System Server is used:

- (Job I055, Steps 1106, 1107)

  Assemble and link the modules NATPNIP (Step 1106), and, optionally ESYNODTB (Step 1107).

Link the Entire System Server Interface to the Nucleus

(Job I060, Step 3720)

- Link the following Entire System Server Interface modules to either the **environment-independent nucleus** or the **environment-dependent nucleus** by using the corresponding INCLUDE statements:

`INCLUDE NATPNIP`	Entire System Server Interface parameters
`INCLUDE ESXNUC`	Entire System Server Interface module
`INCLUDE ESYNODTB`	Optional, node table

Installing and Activating the Write-to-Spool Feature

If you want to use the Write-to-Spool feature, either link the access method to your nucleus or load the method dynamically. See also the Natural parameters `RCA` and `RCALIAS`.

You can define and assemble the defaults for your nucleus by using the source member `NATPWSDF` before linking the defaults to the nucleus. You can find the source member `NATPWSDF` in the source library of Natural.

The default settings of source member `NATPWSDF` are as follows:

```
NAMPWSPL NODE=148,
PROGRAM=,
CLASS=A,
HOLD=YES,
CNTL=A,
FORM=,
RMT=,
FORMDEF=,
PAGEDEF=
```

The table below shows the modifiable parameters of source member `NATPWSDF` and a detailed description of these parameters:

Parameter	Description
`Node`	NPR target node. The node number can consist of up to 5 digits. It addresses the destination started task of the Entire System Server and where the output is written.
`Program`	JES Writer which can contain up to 8 characters. JES provides control to the Writer program. If JES does not find it, it is ignored. Possible value: `*OUTPUT` means that the input from the Natural statement `DEFINE PRINTER` is used to be interpreted as JES Writer.

Parameter	Description
`Class`	`SYSOUT` class within JES where the output has to be written. It can contain only one character or digit. It is a descriptor for further software (for example, Entire Output Management) to detect the output stream for processing.
`Hold = yes/no`	Specifies whether the output stream is to be held within the JES spool in case the task previously started by the Entire System Server terminates.
`CNTL`	Represents the control character for the `SYSOUT` data set. `CNTL` contains one character: `A` — ASA control character `M` — Machine control character
`Form`*`RMT`*	Describes the form control buffer for JES. This value is transferred to JES which handles the processing. *`RMT`* represents the JES remote user ID if `SYSOUT` has to be routed to a different JES system. You can find the name of the JES system in the destination field within the `DEFINE PRINTER` statement (for example, `DEFINE PRINTER OUTPUT='DAEM'`).
`Formdef` *`Pagedef`*	Can contain up to 6 characters.

After editing the `NATPWSDF` source member with customized values, you can assemble and link it. If you want to use the default settings, you can omit this step.

If you want to use the Write-to-Spool feature with statically linked access method at your site, relink the Natural module as follows:

```
INCLUDE NATPWSPL    The Write-to-Spool access method for Natural
INCLUDE NATPWSDF    The Write-to-Spool defaults (your adapted parameter module)
```

If you want to use dynamic load, you can either use the delivered module `NATPWSAM` with default parameters or you can link your adapted parameter module.

The output is written into the `POWER` spool under a new job number. The name of the printed spool data set will be the name of the original batch job or the user ID of the TP monitor.

For further information, see System Spool Access in the *Operations* documentation.

VII Installing Software AG Editor on z/VSE

7 Installing Software AG Editor on z/VSE

The Software AG Editor is an optional Natural component that is required by several Natural utilities (for example, SYSRPC and SYSBPM), Natural add-on products (for example, Natural ISPF) and other Software AG products (for example, Predict).

This document describes the steps for installing the Software AG Editor on z/VSE.

Related Topics:

- For operational information, see *Operating the Software AG Editor* in the *Operations* documentation.
- For information on the features and functions of the *Software AG Editor*, see the relevant section in the *Editors* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

The **System Maintenance Aid (SMA)** parameter `SAG-EDITOR` is set to `Y` (Yes) by default to allow installation of the Software AG Editor.

- Step 7: Link the Software AG Editor to the Nucleus

Step 1: Adapt the Editor Buffer Pool Parameter Macro NTEDBP

1. Modify the editor buffer pool settings supplied with the `NTEDBP` macro in the Natural parameter module to meet your requirements. For a description of this macro, see the corresponding dynamic profile parameter `EDBP`.
2. Assemble the Natural parameter module and link it to the Software AG Editor work file formatting utility (`NATEDFMV`) contained in the Natural load library. The Software AG Editor work file formatting utility is described in *Editor Work File* in the *Operations* documentation.

   The editor buffer pool parameters contained in the Natural parameter module are stored in the editor buffer pool work file control record during formatting (see ***Step 3: Format the Editor Work File***).

For the Initial Installation:

You can leave the defaults. In this case, it is not necessary to assemble and link the Natural parameter module.

Step 2: Allocate the Editor Work File

(Job I008, Steps 1900, 1901, 1903, 1905)

Under Com-plete, this step is only required if you want to use a global editor buffer pool.

A VSAM RRDS is used as the editor work file. To best exploit the VSAM data set space, the record length should be defined 8 bytes less than the control interval length; see also *Editor Work File* in the *Operations* documentation.

- Depending on your TP monitor environment, execute the following steps to allocate the data set:

```
Step 1900 ALLOCATE EDITOR WORK FILE BATCH
     1901 ALLOCATE EDITOR WORK FILE CICS
     1903 ALLOCATE EDITOR WORK FILE TSO
     1905 ALLOCATE EDITOR WORK FILE GLOBAL
```

Step 3: Format the Editor Work File

(Job I081, Steps 1900, 1901, 1903, 1905)

Under Com-plete, this step is only required if you want to use a global editor buffer pool.

1. Use the Software AG Editor work file formatting utility (`NATEDFMV`) to format and load the control record in the editor work file.
2. Depending on your TP monitor environment, execute the following steps to format the data set:

```
Step 1900 FORMAT EDITOR WORK FILE BATCH
     1901 FORMAT EDITOR WORK FILE CICS
     1903 FORMAT EDITOR WORK FILE TSO
     1905 FORMAT EDITOR WORK FILE GLOBAL
```

Step 4: Modify the Startup JCL and Subsystem Definitions

You can specify the data set to be used for the editor work file by either adding the appropriate DLBL statement or using the keyword subparameter `DDNAME` of the parameter macro `NTEDBP` (see *EDBP - Software AG Editor Buffer Pool Definitions* in the *Parameter Reference* documentation).

The data set name specified with `DDNAME` must correspond to the data set definition in your JCL.

- **In Batch Mode:**

  Add a DLBL statement for the editor work file:

  ```
  // DLBL CMEDIT,'data-set-name',,VSAM,CAT=catalog-name
  ```

  where:
  data-set-name is the name of the data set to be used for the work file,
  catalog-name is the name of the VSAM catalog that holds the information about the data set.

  Instead of using an actual VSAM file, you can also define a dummy editor work file by using JCL or standard labels (see the *Example* below). Such a label definition enables you to use the editor and the editor buffer pool. However, any write access to the editor work file will lead to an error, and you cannot use the recovery feature.

Example:

```
// DLBL CMEDIT,'SAG.EDITOR.WORK.FILE',,VSAM,CAT=catalog-name
// EXTENT SYSnnn
// ASSGN SYSnnn,IN
```

where *nnn* is a valid SYS number.

- **Under Com-plete:**
  - If you want to use a global editor buffer pool, add a DLBL statement for the editor work file:

    ```
    // DLBL dlbl-name,'data-set-name',,VSAM,CAT=catalog-name
    ```

    where:
    dlbl-name is the name of the global editor buffer pool,
    data-set-name is the name of the data set to be used for the work file,
    catalog-name is the name of the VSAM catalog that holds the information about the data set.

    In addition, the DLBL name has to be defined to Com-plete by using the UUTIL utility function FM (described in the relevant section of the *Com-plete* documentation).
  - If you want to use a local editor buffer pool, define an SD file as the editor work file. The name of the SD file is indicated in the keyword subparameter `DDNAME` of the `NTEDBP` macro.

    The number of work file records is set with the keyword subparameter `RECNUM` and the work file record length is set with the keyword subparameter `LRECL` of the `NTEDBP` macro.

    For a local editor buffer pool, add the definitions of the editor buffer pool to the `SERVER` parameter of your startup parameters as indicated in ***Define the Natural Com-plete/SMARTS Interface Server*** in the section *Installing Natural Com-plete/SMARTS Interface*.

    For explanations of the keyword subparameters and macros mentioned above, see *EDBP - Software AG Editor Buffer Pool Definitions* in the *Parameter Reference* documentation.
- **Under CICS:**

  Add an entry in the CICS File Control Table (Job I005).

Step 5: Build the Natural Parameter Module

(Job I080)

1. Add the following parameter to your Natural parameter module to specify the size of the editor area:

   ```
   SSIZE=nn
   ```

   where *nn* must be set to at least 54 (the default is 64).
2. The Software AG Editor requires either a local or a global editor buffer pool (see also *Natural Buffer Pools* in the *Operations* documentation).
   - If you want to use a local editor buffer pool, specify the NTBPI macro as follows:

     ```
     NTBPI TYPE=EDIT
     ```

     The size of the region must be large enough to allocate the local buffer pool.

     For more information on NTBPI, see *NTBPI Macro Syntax* in the *Parameter Reference* documentation.
   - If you want to use a global editor buffer pool, proceed as follows:

     Specify the NTBPI macro as follows:

     ```
     NTBPI TYPE=EDIT,NAME=gbp-name
     ```

     where *gbp-name* is the name of the global editor buffer pool to be used.

     Supply the following parameter with the NTPRM macro (described in the *Operations* documentation):

     ```
     SUBSID=subsystem-name
     ```

     where *subsystem-name* is the name of the Natural subsystem specified when creating the global buffer pool.

     For detailed information on SUBSID, see *SUBSID - Subsystem ID under z/OS and z/VSE* in the *Parameter Reference* documentation.
3. Assemble and link the Natural parameter module.

Step 6: Define the Global Editor Buffer Pool

The global editor buffer pool can be shared by several regions. It is defined and started using the same procedure as for Natural global buffer pools; see *Natural Global Buffer Pool under z/VSE* in the *Operations* documentation. The parameter setting `TYPE=EDIT` identifies the buffer pool as an editor buffer pool.

All users of the same global editor buffer pool must share the same editor work file; otherwise, an error occurs.

Step 7: Link the Software AG Editor to the Nucleus

(Job I080)

- Link the following Software AG Editor module to either the **environment-independent nucleus** or the **environment-dependent nucleus** by using the corresponding `INCLUDE` statement:

```
INCLUDE NATEDT
```

- Instead of linking the Software AG Editor module to the nucleus, you can dynamically load it during initialization of a Natural session. In this case, the `NATEDT` editor module must be linked as `EDITOR` object module and the Natural session must be started with the profile parameter setting `RCA=EDITOR` (see also *RCA - Resolve Addresses of Statically Linked Modules* described in the *Parameter Reference* documentation).

Installation Verification

You can verify the successful installation of the Software AG Editor by performing the following:

- Invoke Natural and enter the following system command:

```
SYSEDT
```

  The SYSEDT Utility Main Menu appears, which can be used to display all buffer pool parameters and usage statistics; see *SYSEDT Utility - Editor Buffer Pool Administration* in the *Utilities* documentation.

You can only test the full operation of the Software AG Editor if another Software AG product (for example, Natural ISPF or Predict) that uses Software AG Editor functionality is installed.

VIII Installing Natural CICS Interface on z/VSE

Installing Natural CICS Interface on z/VSE

8 Installing Natural CICS Interface on z/VSE

This document describes the steps for installing Natural CICS Interface Version 8.2.7 (product code NCI) on z/VSE.

Related Topics:

For information on how to operate Natural in a CICS environment, see *Using Natural with TP Monitors* and *Natural under CICS* in the *TP Monitor Interfaces* documentation and the following topics:

- *NCISCPCB Generation Parameters*
- *NCMPRM Macro Parameters*
- *Customization of VSAM RRDS Roll Files*
- *NCISCPRI Warnings and Error Messages*

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

A supported version of the following product must be installed before you can install the Natural CICS Interface:

- Adabas CICS Interface (product code ACI), version as specified under *Software AG Product Versions Required with Natural* in the current Natural *Release Notes* for Mainframes.

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following data set required for product installation:

Data Set Name	Contents
NCI*vrs*.LIBR	Source modules, macros, phases and object modules

Copy the data set into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the NAT*vrs*.LIBJ data set and are prefixed with the product code. The data set is provided on the installation medium supplied for base Natural.

Prefix Used for Natural CICS Interface Components

When used in this document, *prefix* denotes a common Natural CICS Interface prefix of 1 to 5 characters, for example, NCI43. This prefix is determined by the value of the parameter PREFIX in the Natural CICS Interface parameter module, for example in **Step 4** of the *Installation Procedure*. PREFIX is described in the *TP Monitor Interfaces* documentation.

prefix is followed by specific characters to make up the names of the following objects:

*prefix*CB	Natural CICS Interface system directory, for example, NCI43CB
*prefix*R1 to *prefix*R9	Natural CICS Interface VSAM RRDS roll files (optional)
*prefix*XFA	Natural CICS Interface 3270 Bridge XFAINTU exit

Installation Procedure

Before you start the installation procedure for the Natural CICS Interface, be sure to read the following:

- *System Control under CICS* in the *TP Monitor Interfaces* documentation
- ***Installation Process and Major Natural Features***

The installation procedure comprises the following:

- Step 1: Customize CICS
- Step 2: Allocate the VSAM RRDS Roll Files for the Natural CICS Interface
- Step 3: Build the Roll-File Initialization Module
- Step 4: Build the Natural CICS Interface Parameter Module
- Step 5: Build the Natural CICS Interface Starter Module NCISTART
- Step 6: Build the Natural CICS Interface Root Module NCIROOT
- Step 7: Build the Natural CICS Interface System Directory Module
- Step 8: Build the Natural CICS Interface External CALLNAT Interface Module
- Step 9: Build the Natural CICS Interface Node Error Program
- Step 10: Build the Natural CICS Interface XFAINTU Exit
- Step 11: Build the Natural CICS Interface Front-End Driver NCISFED
- Step 12: Build the Natural Parameter Module
- Step 13: Link the Environment-Dependent Nucleus

Step 1: Customize CICS

(Job I005, Steps 2211, 2212, 2230)

1. Steps 2211 and 2212:

   Create the CICS tables and RDO entries as described in ***CICS Resource Definitions***.

2. Step 2230 (optional):

   This step is only required if you want to dynamically load an ICU data item without using the SVC instruction on request during a Natural session instead of an entire ICU data library:

   Set `CFICU=(DATITEM=NONE)` and add one PPT entry for each ICU data item.

   See also the **corresponding step** in *Installing International Components for Unicode for Software AG*.

   ICU data items and ICU data libraries are described in the *Unicode and Code Page Support* documentation.

Step 2: Allocate the VSAM RRDS Roll Files for the Natural CICS Interface

(Job I008, Step 2200)

This step must be performed only if VSAM roll files are used as CICS roll facility.

- Allocate the VSAM RRDS roll files for the Natural CICS Interface.

  The Natural CICS Interface uses VSAM RRDS roll files for optimum performance, which means without CI/CA splits.

Step 3: Build the Roll-File Initialization Module

(Job I070, Step 2205)

This step must be performed only if VSAM roll files are used as roll facility.

- Assemble and catalog the `NCISCPRI` batch program of the Natural CICS Interface to initialize a roll file.

  This step creates an executable batch module which is used in ***Step 15: Link the VSAM Roll-File Initialization Module***.

Step 4: Build the Natural CICS Interface Parameter Module

(Job I070, Steps 2220, 2225)

- Edit, assemble and catalog the Natural CICS Interface parameter module `NCIPARM`.

  The Natural CICS Interface parameter module `NCIPARM` contains a macro named `NCMPRM` which contains parameters specific to the Natural CICS Interface.

  You can generally use the default values for all parameters. Modify only the values of those parameters whose default values do not suit your requirements. The only mandatory parameter without a default value is the common Natural CICS Interface prefix.

  To simplify the Natural CICS Interface parameter module installation process, the source module `NCIPARM` contains the `NCMPRM` macro request with parameter `PREFIX=&SYSPARM`. Thus, when generating a Natural CICS Interface parameter module, assemble the `NCIPARM` source module with assembler option `SYSPARM=`*`prefix`* rather than editing the source module.

  The individual parameters are described in the section *NCMPRM Macro Parameters* in the *TP Monitor Interfaces* documentation.

Step 5: Build the Natural CICS Interface Starter Module NCISTART

(Job I070, Step 2230)

- Make sure that the IBM Language Environment macro library is available in the `SYSLIB` chain in the assembler step.
- Translate, assemble and catalog the Natural CICS Interface starter module `NCISTART`. Repeat these steps after you have installed a new CICS version.

  If you are not using the most recent CICS version, the translate step may result in a non-zero return code (4 - 16, depending on your CICS version) because of CICS commands being used that are unknown to your CICS translator. This return code can be ignored as long as the subsequent assembly step will end with a return code of 0 by using conditional assemblies to bypass potential CICS translator messages.

Step 6: Build the Natural CICS Interface Root Module NCIROOT

(Job I070, Step 2235)

- Translate, assemble and catalog the Natural CICS Interface module NCIROOT. Repeat these steps after you have installed a new CICS version.

  If you are not using the most recent CICS version, the translate step may result in a non-zero return code (4 - 16, depending on your CICS version) because of CICS commands being used that are unknown to your CICS translator. This return code can be ignored as long as the subsequent assembly step will end with a return code of 0 by using conditional assemblies to bypass potential CICS translator messages.

Step 7: Build the Natural CICS Interface System Directory Module

(Job I070, Steps 2245, 2250)

- Edit, assemble and catalog the NCISCPCB module.

  The Natural CICS Interface system directory is generated by assembling and cataloging the source module NCISCPCB.

  A sample job is contained in the NAT*vrs*.LIBJ data set and a comprehensive sample source in the NCI*vrs*.LIBR data set.

  For descriptions of the individual macros and parameters contained in NCISCPCB, see *NCISCPCB Generation Parameters* in the *TP Monitor Interfaces* documentation.

Step 8: Build the Natural CICS Interface External CALLNAT Interface Module

(Job I070, Step 2270)

This step must be performed only if you want to use the external CALLNAT interface module of the Natural CICS Interface. See also *Environment Dependencies* in *Natural 3GL CALLNAT Interface - Purpose, Prerequisites, Restrictions* in the *Operations* documentation.

- Translate, assemble and catalog the external CALLNAT interface module NCIXCALL. See also NCIXCALL as described for the CALLNAT in *Environment Dependencies* in the *Operations* documentation.

  Repeat these steps after you have installed a new CICS version.

  If you have an NCIXCALL module from a previous Natural version, specify a new name for this module (for example, NCIXCIOV) by using the SYSPARM parameter contained in NCIXCALL. The NCIXCALL module from the previous Natural version must be linked (see ***Step 16: Link the Natural CICS Interface External CALLNAT Interface Module***) to assign it the new name.

If you are not using the most recent CICS version, the translate step may result in a non-zero return code (4 - 16, depending on your CICS version) because of CICS commands being used that are unknown to your CICS translator. This return code can be ignored as long as the subsequent assembly step will end with a return code of 0 by using conditional assemblies to bypass potential CICS translator messages.

Step 9: Build the Natural CICS Interface Node Error Program

(Job I070, Step 2275)

This step must be performed only if you want to use the node error program (NEP) of the Natural CICS Interface.

- Translate, assemble and catalog the Natural CICS Interface module `NCIZNEP`. Repeat these steps after you have installed a new CICS version.

  For information on `NCIZNEP`, see *NCIZNEP Functionality* in the *TP Monitor Interfaces* documentation.

  If you are not using the most recent CICS version, the translate step may result in a non-zero return code (4 - 16, depending on your CICS version) because of CICS commands being used that are unknown to your CICS translator. This return code can be ignored as long as the subsequent assembly step will end with a return code of 0 by using conditional assemblies to bypass potential CICS translator messages.

Step 10: Build the Natural CICS Interface XFAINTU Exit

(Job I070, Step 2280)

This step must be performed only if you want to use Natural via the CICS 3270 Bridge.

- Translate, assemble and catalog the Natural CICS Interface module `NCIXFATU`. Repeat these steps after you have installed a new CICS version.

  If you are not using the most recent CICS version, the translate step may result in a non-zero return code (4 - 16, depending on your CICS version) because of CICS commands being used that are unknown to your CICS translator. This return code can be ignored as long as the subsequent assembly step will end with a return code of 0 by using conditional assemblies to bypass potential CICS translator messages.

Step 11: Build the Natural CICS Interface Front-End Driver NCISFED

(Job I070, Step 2290)

This step must be performed only if you want to use the Natural RPC server front-end under CICS. This step is only generated if the **System Maintenance Aid (SMA)** parameter `NAT-RPC` is set to `Y` (Yes). `NAT-RPC` is used in connection with the SMA parameter `NCI-RPC-FRONT`.

-
- Make sure that the IBM Language Environment (LE) macro library is available in the `SYSLIB` chain in the assembler step.
- Translate, assemble and catalog the Natural CICS Interface front-end driver `NCISFED`. Repeat these steps after you have installed a new CICS version.

  If you are not using the most recent CICS version, the translate step may result in a non-zero return code (4 - 16, depending on your CICS version) because of CICS commands being used that are unknown to your CICS translator. This return code can be ignored as long as the subsequent assembly step will end with a return code of 0 by using conditional assemblies to bypass potential CICS translator messages.

Step 12: Build the Natural Parameter Module

(Job I080, Step 2220)

Build the Natural parameter module for the Natural CICS Interface:

1. Modify the settings of the supplied Natural profile parameters as required and in accordance with the settings you specified when building the Natural parameter module for batch mode during the ***Installation Procedure*** for base Natural.

   Make sure that the profile parameters `FNAT` and `FUSER` are set to the same values you specified when loading the system file. The parameters and corresponding macros (if applicable) are described in the *Parameter Reference* documentation.
2. Assemble and catalog the Natural parameter module.

Step 13: Link the Environment-Dependent Nucleus

(Job I080, Step 2230)

- Link the **environment-dependent nucleus** for the Natural CICS Interface with the following modules:

  `NCIPARM` built in **Step 4**,
  `NCISTART` built in **Step 5**,
  `NCIROOT` built in **Step 6**, and
  the Natural parameter module built in **Step 12**.

Include the CICS stub module DFHELII (do not use DFHEAI).

All components of the environment-dependent nucleus are reentrant. Therefore, you can link it with the shared virtual area (SVA) option to take advantage of the CICS (E)RDSA storage area.

Step 14: Link the Natural CICS Interface System Directory

(Job I080, Step 2250)

- Link the Natural CICS Interface system directory into your CICS user library under the module name *prefix*CB (see ***Prefix Used for Natural CICS Interface Components***).

  The Natural CICS Interface system directory is not reentrant. Therefore, it cannot be linked with the shared virtual area (SVA) option.

Step 15: Link the VSAM Roll-File Initialization Module

(Job I080, Step 2265)

This step must be performed only if VSAM roll files are used as CICS roll facility.

- Link the VSAM roll-file initialization module NCISCPRI into your CICS user library.

Step 16: Link the Natural CICS Interface External CALLNAT Interface Module

(Job I080, Steps 2270, 2271)

This step must be performed only if you want to use the external CALLNAT interface module of the Natural CICS Interface.

- Link the external CALLNAT interface module NCIXCALL.

  Step 2270 is needed if the NCIXCALL module has been installed in the same CICS partition with a previous Natural version. The previous Natural version NCIXCALL module must then be assigned a new name, for example NCIXCIOV. Thus, you can preserve the name of the NCIXCALL module used in the previous version and need not link it to all your 3GL programs using it.

  Step 2271 links the module NCIXCALL for the current Natural version.

  The external CALLNAT interface module of the Natural CICS Interface is reentrant. Therefore, you can link it with the shared virtual area (SVA) option to take advantage of the CICS (E)RDSA storage area.

Step 17: Link the Natural CICS Interface Node Error Program

(Job I080, Step 2275)

This step must be performed only if you want to use the node error program (NEP) of the Natural CICS Interface. See also *CICS Node Error Program Considerations for Natural* in the *TP Monitor Interfaces* documentation.

- Link the Natural CICS Interface module `NCIZNEP`.

  The node error program of the Natural CICS Interface is reentrant. Therefore, you can link it with the shared virtual area (SVA) option to take advantage of the CICS (E)RDSA storage area.

Step 18: Link the Natural CICS Interface XFAINTU Exit

(Job I080, Step 2280)

This step must be performed only if you want to use Natural with the CICS 3270 Bridge. See also *CICS 3270 Bridge Support* in the *TP Monitor Interfaces* documentation.

- Link the Natural CICS Interface module `NCIXFATU` under the name *`prefix`*`XFA`; see ***Prefix Used for Natural CICS Interface Components***.

  The Natural CICS Interface module `NCIXFATU` is reentrant. Therefore, you can link it with the shared virtual area (SVA) option to take advantage of the CICS (E)RDSA storage area.

Step 19: Link the Natural RPC Server Front-End

(Job I080, Step 2290)

This step must be performed only if you want to use the Natural RPC server front-end under CICS. This step is only generated if the **System Maintenance Aid (SMA)** parameter `NAT-RPC` is set to `Y` (Yes). `NAT-RPC` is used in connection with the SMA parameter `NCI-RPC-FRONT`.

- Define the name of the Natural RPC server front-end with the SMA parameter `NCI-RPC-FRONT`.

  The Natural RPC server front-end under CICS is reentrant. Therefore, you can link it with the shared virtual area (SVA) option to take advantage of the CICS (E)RDSA storage area.

Step 20: Initialize the VSAM Roll Files

(Job I081, Step 2200)

This step must be performed only if VSAM roll files are used as CICS roll facility.

- Initialize the VSAM roll files.

  This step must be repeated for all roll files used if roll files are the primary roll facility.

  A VSAM RRDS file is a direct (random) access type file that must be formatted.

  For the Natural CICS Interface VSAM roll files, formatting is done by the `NCISCPRI` batch program. To execute `NCISCPRI`, the Natural roll file to be initialized has to be assigned the file name `ROLL` in the JCL DLBL statement. No other parameter input is required for `NCISCPRI`; all data required for file initialization is obtained by `SHOWCB` VSAM macro calls.

  When running the Natural CICS Interface, supply an end-of-data (`/*`) statement in the JCL for compatibility reasons, although no parameter input is required for `NCISCPRI`.

For descriptions of the messages that can be output during this step, see *NCISCPRI Warnings and Error Messages* in the *TP Monitor Interfaces* documentation.

CICS Resource Definitions

This section describes resource definitions required or recommended for customizing your CICS system.

Note: We generally recommend that you keep all Natural version-dependent components such as programs, transactions and files in a separate resource group. Such a group is represented by *natgroup* in this section.

- Program Definitions
- Transaction Definitions
- File Definitions

- Transient Data Destinations

Program Definitions

- Environment-Dependent Nucleus
- Environment-Independent Nucleus
- Natural CICS Interface System Directory
- External CALLNAT Interface Module
- Node Error Program
- Global User Exit
- Natural RPC Server Front-End

Environment-Dependent Nucleus

- Add a program definition for the environment-dependent nucleus:

```
DEFINE PROGRAM(dep-nuc) GROUP(natgroup) LANGUAGE(ASSEMBLER)          *
         DESCRIPTION(ENVIRONMENT-DEPENDENT NUCLEUS)
```

where *dep-nuc* is the Natural CICS Interface module built during the **link step**.

We recommend that you set the following parameter value in the CICS program definition under CICS TS:

```
DATALOCATION(ANY)
```

Important Note for CICS TS:

Program attributes (for example, `EXECKEY` or `DATALOCATION`) are inherited from `NCISTART` because standard linkage conventions (BALR 14,15) are used for the program calls. Examples are calls for the environment-independent nucleus, the Adabas CICS Interface or Natural calls (`SET CONTROL 'P=S'`) for external programs.

Environment-Independent Nucleus

This definition is optional.

- Add a program definition for the **environment-independent nucleus**:

```
DEFINE PROGRAM(ind-nuc) GROUP(natgroup) LANGUAGE(ASSEMBLER)            *
         DESCRIPTION(NATURAL ENVIRONMENT-INDEPENDENT NUCLEUS)
```

where *ind-nuc* is the name of the **environment-independent nucleus** specified with the Natural profile parameter `NUCNAME`. The default name is `INDNUC`*vr*. You need not specify `EXECKEY` or `DATALOCATION` for the environment-independent nucleus as all attributes of the `NCISTART` program are inherited since standard linkage conventions (BALR 14,15) are used.

To access the environment-independent nucleus in the SVA, specify `USESVACOPY(YES)` for this program definition and `SVA=YES` in the CICS startup parameters.

Natural CICS Interface System Directory

- Add a program definition for the Natural CICS Interface system directory:

```
DEFINE PROGRAM(prefixCB) GROUP(natgroup) LANGUAGE(ASSEMBLER)            *
         DESCRIPTION(NATURAL CICS INTERFACE SYSTEM DIRECTORY)
```

External CALLNAT Interface Module

This definition is optional.

- Add a program definition for the external `CALLNAT` interface module:

```
DEFINE PROGRAM(ncixcall) GROUP(natgroup) LANGUAGE(ASSEMBLER)            *
         DESCRIPTION(NATURAL CICS INTERFACE EXTERNAL CALLNAT MODULE)
```

We recommend that you set the following parameter value in the CICS program definition under CICS TS:

```
DATALOCATION(ANY)
```

Node Error Program

This definition is optional.

- Add a program definition for the node error program (NEP) of the Natural CICS Interface:

```
DEFINE PROGRAM(nciznep) GROUP(natgroup) LANGUAGE(ASSEMBLER)             *
         EXECKEY(CICS)                                                  *
         DESCRIPTION(NATURAL CICS INTERFACE NODE ERROR PROGRAM)
```

where *nciznep* is the NEP name specified in ***Step 9: Build the Natural CICS Interface Node Error Program***.

Global User Exit

This definition is optional.

- Add a program definition for the `XFAINTU` global user exit:

```
DEFINE PROGRAM(prefixXFA) GROUP(natgroup) LANGUAGE(ASSEMBLER)          *
         EXECKEY(CICS)                                                 *
         DESCRIPTION(NATURAL CICS INTERFACE XFAINTU GLUE)
```

Natural RPC Server Front-End

This definition is optional.

This definition is only required if you want to use the Natural RPC server front-end *ncisfe*. This definition is only generated if the **System Maintenance Aid (SMA)** parameter `NCI-RPC-FRONT` is set.

- Add a program definition for the Natural RPC server front-end:

```
DEFINE PROGRAM(ncisfe) GROUP(natgroup) LANGUAGE(ASSEMBLER)            *
         DESCRIPTION(NATURAL RPC SERVER FRONT-END)
```

We recommend that you set the following parameter value in the CICS program definition under CICS TS:

```
DATALOCATION(ANY)
```

The name of the Natural RPC server front-end *ncisfe* must be identical to the name specified in ***Step 19: Link the Natural RPC Server Front-End***.

Transaction Definitions

We recommend that you define or choose a CICS profile for the Natural transactions similar to the following:

```
DEFINE PROFILE(natprof) GROUP(natgroup)                                *
         DESCRIPTION(CICS PROFILE FOR NATURAL TRANSACTIONS)            *
         SCRNSIZE(ALTERNATE) INBFMH(ALL)
```

where *natprof* is the name of the CICS profile assigned to the Natural transactions.

We also recommend that you define a CICS transaction class for the Natural transactions similar to the following:

```
DEFINE TRANCLASS(natclass) GROUP(natgroup) MAXACTIVE(999)                *
         DESCRIPTION(CLASS FOR NATURAL TRANSACTIONS) ↩
```

where *natclass* is the name of the CICS transaction class assigned to the Natural transactions.

A CICS transaction class dedicated to Natural helps control storage usage by Natural (see also *Controlling Storage Usage* in the *TP Monitor Interfaces* documentation). Assign this transaction class to the definitions of all transactions that directly or indirectly call Natural.

You can define the following:

- Natural Transaction
- Natural Message Switching Transaction
- Node Error Program
- Natural RPC Server Front-End

Natural Transaction

- Add a definition for the Natural transaction:

```
DEFINE TRANSACTION(ncitransact) GROUP(natgroup)                          *
         PROGRAM(dep-nuc) TWASIZE(128) DUMP(NO) SPURGE (YES)             *
         PROFILE(natprof) TRANCLASS(natclass)
```

  where:

  ncitransact is the name of the Natural CICS Interface user transaction ID.
  dep-nuc is the Natural CICS Interface module built during the **link step**.

  We recommend that you set the following parameter value in the CICS transaction definitions under CICS TS:

```
TASKDATALOC(ANY)
```

  `TASKDATALOC(ANY)` can have an impact on non-Natural programs called by Natural; for details, see the relevant IBM literature on CICS.

Natural Message Switching Transaction

- Add a definition for the Natural internal message switching transaction:

```
DEFINE TRANSACTION(nmsg) GROUP(natgroup)                                *
         PROGRAM(dep-nuc) TWASIZE(128) DUMP(NO) SPURGE (YES)            *
         PROFILE(natprof) TRANCLASS(natclass)
```

where:

dep-nuc is the Natural CICS Interface module built during the **link step**.

nmsg is the name of the Natural CICS Interface message switching transaction ID as defined with the MSGTRAN parameter described in *Parameters in Macro NCMPRM* in the *TP Monitor Interfaces* documentation. The default name is NMSG.

We recommend that you set the following parameter value in the CICS transaction definitions under CICS TS:

```
TASKDATALOC(ANY)
```

TASKDATALOC(ANY) can have an impact on non-Natural programs called by Natural; for details, see the relevant IBM literature on CICS.

Node Error Program

This definition is optional and applies to CICS TS only.

- Add a definition for the node error program (NEP) of the Natural CICS Interface:

```
DEFINE TRANSACTION(neptran) GROUP(natgroup)                             *
         PROGRAM(nciznep) DUMP(NO) PRIORITY(255)                        *
         TASKDATAKEY(CICS) TASKDATALOC(ANY)                             *
         PROFILE(natprof)
```

where:

neptran is the NEP transaction code defined with the NEPTRAN generation parameter in the SYSPARM specification in the NCIZNEP module built in ***Step 9: Build the Natural CICS Interface Node Error Program***.

nciznep is the NEP name specified in ***Step 9: Build the Natural CICS Interface Node Error Program***.

NCIZNEP and NEPTRAN are described in the *TP Monitor Interfaces* documentation.

Natural RPC Server Front-End

- Add a definition for the Natural RPC server front-end transaction:

```
DEFINE TRANSACTION(ncisfetransact) GROUP(natgroup)                    *
         PROGRAM(ncisfe) TWASIZE(128) DUMP(NO) SPURGE (YES)           *
         PROFILE(natprof) TRANCLASS(natclass)
```

  where:

  ncisfetransact is the name of the Natural RPC server front-end transaction ID.
  ncisfe is the name of the Natural RPC server front-end specified in ***Step 19: Link the Natural RPC Server Front-End***.

File Definitions

These definitions are only required if VSAM roll files are to be used.

- Add one entry in the FCT for each Natural CICS Interface VSAM roll file:

```
DFHFCT TYPE=FILE,                                                     *
       FILE=prefixRn,                                                 *
       ACCMETH=VSAM,                                                  *
       RECFORM=(FIXED,BLOCKED),                                       *
       SERVREQ=(UPDATE,DELETE,ADD),                                   *
       FILSTAT=(ENABLED,OPENED),                                      *
       BUFND=5,STRNO=3
```

  Local shared resources (LSR) should be used whenever possible. If multiple LSR pools are supported, one pool should be dedicated exclusively to Natural CICS Interface roll files.

Transient Data Destinations

- Error Messages
- Natural CICS Interface Session Statistics
- Natural CICS Interface Profile Parameter File

Error Messages

This definition is optional but highly recommended to log Natural CICS Interface informational messages and Natural abend codes and corresponding error messages.

Add entries in the DCT for the Natural CICS Interface error message logging facility. For Natural error messages, you can use:

- A destination that is already defined in CICS (for example, CSSL); in this case, no extra DCT entry is required.

- An extra partition destination as a synonym for an existing CICS message destination:

```
DFHDCT TYPE=INDIRECT,                                                  *
       DESTID=nerr,                                                    *
       INDDEST=name
```

  where:

  nerr is the name of the Natural CICS Interface error message destination as defined with the MSGDEST parameter described in *NCMPRM Macro Parameters* in the *TP Monitor Interfaces* documentation. The default name is NERR.

  name is the name of the corresponding indirect destination.

- An extra file:

```
DFHDCT TYPE=SDSCI,                                                     *
       DSCNAME=NATMSG,                                                 *
       RECFORM=VARUNB,                                                 *
       RECSIZE=nnn,                                                    *
       TYPEFLE=OUTPUT

DFHDCT TYPE=EXTRA,                                                     *
       DSCNAME=NATMSG,                                                 *
       DESTID=nerr,                                                    *
       OPEN=INITIAL
```

  where *nerr* is the name of the Natural CICS Interface error message destination as defined with the MSGDEST parameter described in *NCMPRM Macro Parameters* in the *TP Monitor Interfaces* documentation. The default name is NERR.

  You can, for example, change the RECFORM format from VARUNB (variable unblocked) to VARBLK (variable blocked). Natural and the Natural CICS Interface messages have a length of up to 120 bytes. Therefore, the record size (RECSIZE=*nnn*) should be at least 124 bytes for variable record format or 120 bytes for fixed record format.

  When using a disk file:

  Sufficient disk space must be reserved for this data set; a DLBL statement must be added to the CICS startup JCL.

Natural CICS Interface Session Statistics

This definition is optional.

- Add two entries in the DCT for the Natural CICS Interface session statistics:

```
DFHDCT TYPE=SDSCI,                                                     *
       DSCNAME=NATLOG,                                                 *
       RECFORM=VARBLK,                                                 *
       BLKSIZE=4628,                                                   *
       RECSIZE=4624,                                                   *
       DEVICE=DISK

DFHDCT TYPE=EXTRA,                                                     *
       TYPEFLE=OUTPUT,                                                 *
       DSCNAME=NATLOG,                                                 *
       DESTID=nlog,                                                    *
       OPEN=INITIAL
```

  where *nlog* is the name of the Natural CICS Interface logging destination as defined with the parameter `LOGDEST` described in the *TP Monitor Interfaces* documentation. The default name is `NLOG`.

  Sufficient disk space must be reserved for this data set; a DLBL statement must be added to the CICS startup JCL.

Natural CICS Interface Profile Parameter File

This definition is optional.

- Add two entries in the DCT for the Natural CICS Interface profile parameter file:

```
DFHDCT TYPE=SDSCI,                                                     *
       RECSIZE=80,           card image                                *
       BLKSIZE=nnn,                                                    *
       BUFNO=1,                                                        *
       DSCNAME=cmprmin,                                                *
       RECFORM=FIXBLK,                                                 *
       TYPEFLE=INPUT

DFHDCT TYPE=EXTRA,                                                     *
       DESTID=nprm,                                                    *
       DSCNAME=cmprmin,                                                *
       OPEN=DEFERRED
```

  where:

nprm is the name of the Natural CICS Interface profile parameter input destination as defined with the PRMDEST parameter described in *NCMPRM Macro Parameters* in the *TP Monitor Interfaces* documentation. The default name is NPRM.

nnn is a multiple of 80.

A DLBL statement must be added to the CICS startup JCL.

Installation Verification

You can verify the successful installation of the Natural CICS Interface by performing the following steps:

1. From a CICS session, type in the Natural transaction ID to start a Natural session.
2. Proceed with the steps described in the section ***Test Online Natural***.

IX Installing Natural Com-plete/SMARTS Interface on z/VSE

9 Installing Natural Com-plete/SMARTS Interface Version 8.2.7 on z/VSE

This document describes the steps for installing Natural Com-plete/SMARTS Interface Version 8.2.7 which corresponds to the Natural Com-plete Interface (product code NCF) on z/VSE.

Note: For support of Natural zIIP Enabler under Com-plete, Natural Com-plete/SMARTS Interface Version 8.3.3 (and above) must be installed.

Related Topics:

For information on how to operate Natural in a Com-plete/SMARTS environment, see the following topics:

- *Using Natural with TP Monitors* in the *TP Monitor Interfaces* documentation.
- *Natural under Com-plete/SMARTS* in the *TP Monitor Interfaces* documentation.
- *Natural under Com-plete/SMARTS User Abend Codes* in the *Messages and Codes* documentation.

For information on installing and using Com-plete, see the *Com-plete* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

A supported version of the following product must be installed before you can install the Natural Com-plete/SMARTS Interface:

- Com-plete

  See the Com-plete *Installation* documentation.

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following data set required for product installation:

Data Set Name	Contents
NCF*vrs*.LIBR	Source modules, macros, phases and object modules

Copy the data set into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the NAT*vrs*.LIBJ data set and are prefixed with the product code. The data set is provided on the installation medium supplied for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Step 1: Build the Natural Parameter Module

(Job I080, Step 2300)

The Natural profile parameters and parameter macros mentioned in this section are described in the *Parameter Reference* documentation unless otherwise noted.

1. Modify the Natural parameter module for Com-plete/SMARTS:
   - Configure the Com-plete/SMARTS batch interface: Modify the settings of the parameters supplied with the macro NTCOMP to meet your requirements. For descriptions of these parameters, see the corresponding profile parameter COMP.
   - Modify the following parameters:

     ```
     FNAT=(database-id,file-number)
     FUSER=(database-id,file-number)
     ```

     where *database-id* and *file-number* are the values you specified when loading the system files during the ***Installation Procedure*** for base Natural.
2. Define a Natural local buffer pool under Com-plete by modifying the values of the keyword parameters supplied with the parameter macro NTBPI (see the Natural profile parameter BPI) as required.

A local buffer pool is allocated during initialization of the first Natural session after Com-plete startup.

The status of the local buffer pools can be displayed on the operator console by issuing the following Com-plete operator command:

```
SERV,server-name,BPSTAT
```

where *server-name* is the name of the server as specified with the Com-plete SERVER startup option.

Step 2: Link the Nucleus

(Job I080, Step 2320)

Link the **environment-dependent nucleus** for the Natural Com-plete/SMARTS Interface.

- Link the **environment-dependent nucleus** into your Com-plete user program library.

  The list of the modules to be linked for the environment-dependent nucleus is supplied with Step 2320.

  If you want Natural to run in the IBM Language Environment (LE), set the **System Maintenance Aid (SMA)** `NAT-LE` to `Y` (Yes). The default setting is `N` (No).
- Specify the **environment-dependent nucleus** as a Com-plete startup option by setting the following Com-plete keyword parameter:

  ```
  RESIDENTPAGE=name
  ```

  where *name* is the name of the environment-dependent nucleus for the Natural Com-plete/SMARTS Interface.

Step 3: Link the Natural Com-plete/SMARTS Interface Server

(Job I080, Step 2350)

- Link the Natural Com-plete/SMARTS Interface server.

The Natural Com-plete/SMARTS Interface server is used to maintain common storage and tables across Natural sessions, for example, the local buffer pool.

Step 4: Define the Natural Com-plete/SMARTS Interface Server

- Specify the Natural Com-plete/SMARTS Interface server as a Com-plete startup option by setting the following Com-plete keyword parameter:

```
SERVER=(server-name,module-name)
```

  where:

  server-name is the name of the server as specified with the keyword subparameter SERVER in the parameter macro NTCOMP (see the *Parameter Reference* documentation).

  module-name is the name of the object module linked in ***Step 3: Link the Natural Com-plete/SMARTS Interface Server***.

The Natural Com-plete/SMARTS Interface server module is loaded during Com-plete initialization. The module must therefore be placed in a load library contained in the COMPINIT load library concatenation (see also the Com-plete *Installation* documentation).

Step 5: Catalog the Natural Com-plete/SMARTS Interface

This step is required if either of the following is true:

You run Natural under Com-plete/SMARTS and use threads below the line (THABOVE=NO setting in the NTCOMP macro).

Or:

You want to use Natural work pools below the 16-MB line.

- Catalog the Natural Com-plete/SMARTS Interface by using the Com-plete ULIB utility.
  - For threads below the line:

    The region size to be specified with the ULIB utility parameter RG depends on the setting of the keyword subparameter NTHSIZE in the parameter macro NTCOMP described in the *Parameter Reference* documentation.
  - For work pools below the 16-MB line:

    The region size to be specified with the ULIB utility parameter RG depends on the setting of the Natural profile parameter WPSIZE (see the *Parameter Reference* documentation) for the parameter macro NTPRM (see the *Operations* documentation).

  See also *Storage Usage* in the section *Natural under Com-plete/SMARTS* in the *TP Monitor Interfaces* documentation.

  After installation, you can use the Natural SYSTP utility (see the *Utilities* documentation) to determine the region size actually used.

Installation Verification

You can verify the successful installation of the Natural Com-plete/SMARTS Interface by performing the following steps:

1. Stop and restart Com-plete.
2. Enter the Com-plete user menu and type in the name of the environment-dependent nucleus for the Natural Com-plete/SMARTS Interface.

   The Natural initial screen should appear.
3. Proceed with the steps described in the section ***Test Online Natural***.

10 Installing Natural Com-plete/SMARTS Interface Version 8.3.5 on z/VSE

This document describes the steps for installing Natural Com-plete/SMARTS Interface Version 8.3.5 which corresponds to the Natural Com-plete Interface (product code NCF) on z/VSE.

Related Topics:

For information on how to operate Natural in a Com-plete/SMARTS environment, see the following topics:

- *Using Natural with TP Monitors* in the *TP Monitor Interfaces* documentation.
- *Natural under Com-plete/SMARTS* in the *TP Monitor Interfaces* documentation.
- *Natural under Com-plete/SMARTS User Abend Codes* in the *Messages and Codes* documentation.

For information on installing and using Com-plete, see the *Com-plete* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

A supported version of the following product must be installed before you can install the Natural Com-plete/SMARTS Interface:

- Com-plete

  See the Com-plete *Installation* documentation.

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following data set required for product installation:

Data Set Name	Contents
NCF*vrs*.LIBR	Source modules, macros, phases and object modules

Copy the data set into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the NAT*vrs*.LIBJ data set and are prefixed with the product code. The data set is provided on the installation medium supplied for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Step 1: Build the Natural Parameter Module

(Job I080, Step 2300)

The Natural profile parameters and parameter macros mentioned in this section are described in the *Parameter Reference* documentation unless otherwise noted.

1. Modify the Natural parameter module for Com-plete/SMARTS:
   - Configure the Com-plete/SMARTS batch interface: Modify the settings of the parameters supplied with the macro NTCOMP to meet your requirements. For descriptions of these parameters, see the corresponding profile parameter COMP.
   - Modify the following parameters:

     ```
     FNAT=(database-id,file-number)
     FUSER=(database-id,file-number)
     ```

     where *database-id* and *file-number* are the values you specified when loading the system files during the ***Installation Procedure*** for base Natural.
2. Define a Natural local buffer pool under Com-plete by modifying the values of the keyword parameters supplied with the parameter macro NTBPI (see the Natural profile parameter BPI) as required.

   A local buffer pool is allocated during initialization of the first Natural session after Com-plete startup.

   The status of the local buffer pools can be displayed on the operator console by issuing the following Com-plete operator command:

```
SERV,server-name,BPSTAT
```

where *server-name* is the name of the server as specified with the Com-plete SERVER startup option.

Step 2: Link the Nucleus

(Job I080, Step 2320)

Link the **environment-dependent nucleus** for the Natural Com-plete/SMARTS Interface.

- Link the **environment-dependent nucleus** into your Com-plete user program library.

  The list of the modules to be linked for the environment-dependent nucleus is supplied with Step 2320.

  If you want Natural to run in the IBM Language Environment (LE), set the **System Maintenance Aid (SMA)** `NAT-LE` to `Y` (Yes). The default setting is `N` (No).
- Specify the **environment-dependent nucleus** as a Com-plete startup option by setting the following Com-plete keyword parameter:

```
RESIDENTPAGE=name
```

where *name* is the name of the environment-dependent nucleus for the Natural Com-plete/SMARTS Interface.

Step 3: Link the Natural Com-plete/SMARTS Interface Server

(Job I080, Step 2350)

- Link the Natural Com-plete/SMARTS Interface server.

The Natural Com-plete/SMARTS Interface server is used to maintain common storage and tables across Natural sessions, for example, the local buffer pool.

Step 4: Define the Natural Com-plete/SMARTS Interface Server

- Specify the Natural Com-plete/SMARTS Interface server as a Com-plete startup option by setting the following Com-plete keyword parameter:

```
SERVER=(server-name,module-name)
```

where:

server-name is the name of the server as specified with the keyword subparameter SERVER in the parameter macro NTCOMP (see the *Parameter Reference* documentation).

module-name is the name of the object module linked in ***Step 3: Link the Natural Com-plete/SMARTS Interface Server***.

The Natural Com-plete/SMARTS Interface server module is loaded during Com-plete initialization. The module must therefore be placed in a load library contained in the COMPINIT load library concatenation (see also the Com-plete *Installation* documentation).

Step 5: Catalog the Natural Com-plete/SMARTS Interface

This step is required if either of the following is true:

You run Natural under Com-plete/SMARTS and use threads below the line (THABOVE=NO setting in the NTCOMP macro).

Or:

You want to use Natural work pools below the 16-MB line.

- Catalog the Natural Com-plete/SMARTS Interface by using the Com-plete ULIB utility.
  - For threads below the line:

    The region size to be specified with the ULIB utility parameter RG depends on the setting of the keyword subparameter NTHSIZE in the parameter macro NTCOMP described in the *Parameter Reference* documentation.
  - For work pools below the 16-MB line:

    The region size to be specified with the ULIB utility parameter RG depends on the setting of the Natural profile parameter WPSIZE (see the *Parameter Reference* documentation) for the parameter macro NTPRM (see the *Operations* documentation).

  See also *Storage Usage* in the section *Natural under Com-plete/SMARTS* in the *TP Monitor Interfaces* documentation.

  After installation, you can use the Natural SYSTP utility (see the *Utilities* documentation) to determine the region size actually used.

Installation Verification

You can verify the successful installation of the Natural Com-plete/SMARTS Interface by performing the following steps:

1. Stop and restart Com-plete.
2. Enter the Com-plete user menu and type in the name of the environment-dependent nucleus for the Natural Com-plete/SMARTS Interface.

   The Natural initial screen should appear.
3. Proceed with the steps described in the section ***Test Online Natural***.

X Installing Natural for DL/I on z/VSE

11 Installing Natural for DL/I on z/VSE

This document describes the steps for installing Natural for DL/I (product code NDL) on z/VSE.

Related Topic:

For information on how to operate Natural in a DL/I environment, see *Natural for DL/I* in the *Database Management System Interfaces* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following data sets required for product installation:

Data Set Name	Contents
NDL*vrs*.LIBJ	Sample installation jobs
NDL*vrs*.LIBR	Source modules, macros, phases and object modules

Copy the data sets into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

The installation procedure comprises the following:

- **Common Installation Steps**
- **Installation Steps Specific to CICS**

Common Installation Steps

This section described the installation steps that apply to all Natural environments where Natural for DL/I can be installed. The steps additionally required for a particular TP monitor are described in the following sections.

Step 1: Build the Natural for DL/I Parameter Module

(Job I055, Step 1500)

1. Modify the Natural for DL/I parameter module `NDLPARM` as described in the section *Natural Parameter Modifications for DL/I* in the *Database Management System Interfaces* documentation.
2. Assemble and catalog `NDLPARM`.

Step 2: Build the Natural Parameter Module

(Job I060, Steps 0010, 0015)

Build the Natural parameter module for batch mode:

1. Add the Natural profile parameter `DLISIZE` and specify `DLISIZE=27`. This value applies if the default values of the parameters contained in the `NDLPARM` module are used. `DLISIZE` is described in the *Parameter Reference* documentation.
2. Add an `NTDB` macro (see the *Parameter Reference* documentation) to the Natural parameter module specifying the database IDs (DBIDs) that relate to DL/I segment types. The numbers specified in this DBID list must be in the range from 1 to 254. They indicate which DBIDs are reserved for DL/I segment types. Up to 254 entries can be specified. All Natural DDMs that refer to a DL/I segment type are cataloged with a DBID from this list. The number with the lowest value in this list is the default DBID for DL/I segment types.

Examples:

```
NTDB DLI,(250,253,252)
NTDB DLI,250
```

Note: Values for DL/I DBIDs above 255 are not possible.

3. Assemble and link the parameter module.

Step 3: Link the Nucleus

(Job I060, Steps 0020, 0105)

1. Link the **environment-dependent nucleus** (Step 0020) for batch Natural.

   Add the following `INCLUDE` statements and the corresponding sublibraries to the search chain for the linkage editor:

   ```
   INCLUDE NDLSIOBA
   INCLUDE NDLPARM
   INCLUDE ASMTDLI
   ```

2. Link the **environment-independent nucleus** (Step 0105).

   Add the following `INCLUDE` statement and the corresponding sublibrary to the search chain for the linkage editor:

   ```
   INCLUDE NDLNUC
   ```

Step 4: Build the Natural for DL/I Sample Database

(Jobs I008, I053, I075)

Build a Natural for DL/I environment with a sample database:

1. Allocate VSAM spaces for the sample database (Job I008, Steps 1500 and 1501).
2. Create the DBDs, PSBs and ACB, and perform the initial load (Job I053, Steps 1500, 1501, 1520, 1521, 1530, 1531, 1550 and 1560).
3. Execute the procedures `NATPSB` and `NATDBD` for the sample database (Job I075, Steps 1500, 1501, 1510 and 1511).

   Additional data must be added to the `FDIC` system file to enable Natural to access DL/I databases. To do so, the procedures `NATPSB` and `NATDBD` (see the relevant sections in the *Database Management System Interfaces* documentation) must be executed for each PSB/DBD to be used.

Installation Steps Specific to CICS

This section describes the additional steps required to install Natural for DL/I in a CICS environment.

- Step 1: Build the Natural Parameter Module for the Natural CICS Interface
- Step 2: Link the Nucleus for the Natural CICS Interface

Step 1: Build the Natural Parameter Module for the Natural CICS Interface

(Job I080, Steps 2210, 2220)

Build the Natural Parameter Module for the Natural CICS Interface:

1. Add the Natural profile parameter `DLISIZE` and specify `DLISIZE=27`. This value applies if the default values of the parameters contained in the `NDLPARM` module are used. `DLISIZE` is described in the *Parameter Reference* documentation.
2. Add an `NTDB` macro (see the *Parameter Reference* documentation) to the Natural parameter module specifying the database IDs (DBIDs) that relate to DL/I segment types. The numbers specified in this DBID list must be in the range from 1 to 254. They indicate which DBIDs are reserved for DL/I segment types. Up to 254 entries can be specified. All Natural DDMs that refer to a DL/I segment type are cataloged with a DBID from this list. The number with the lowest value in this list is the default DBID for DL/I segment types.

   Examples:

   ```
   NTDB DLI,(250,253,252)
   NTDB DLI,250
   ```

   Note: Values for DL/I DBIDs above 255 are not possible.

3. Assemble and link the parameter module.

Step 2: Link the Nucleus for the Natural CICS Interface

(Job I080, Step 2230)

- Link the **environment-dependent nucleus** for the Natural CICS Interface.

  Add the following `INCLUDE` statements and the corresponding sublibraries to the search chain for the linkage editor:

```
INCLUDE NDLSIOCX
INCLUDE NDLPARM
INCLUDE ASMTDLI
```

Alternative Link-Edit Options

Instead of the standard link-edit options, you can also use one of the following link-edit methods:

- Link-edit all Natural for DL/I modules (NDLNUC, NDLPARM and NDLSIOCX), the DL/I module ASMTDLI and an alternate Natural parameter module as a separate module with the mandatory entry name CMPRMTB. The name of the resulting module is optional. This way of link-editing only applies if an alternate parameter module (PARM=) is used. If so, under CICS, an additional CICS PPT entry with PROGRAM=*name* is required.
- Link-edit all Natural for DL/I modules (NDLNUC, NDLPARM and NDLSIOCX) and the DL/I module ASMTDLI as a separate module with the mandatory entry name NATGWDLI. The name of the resulting module is optional. However, if it is different from NATGWDLI, it must be specified as an alias name in an NTALIAS macro entry of the Natural parameter module. This way of link-editing only applies if the Natural Resolve CSTATIC Addresses feature (RCA) is used. If so, under CICS, an additional CICS PPT entry with PROGRAM=*name* is required.

Installation Verification

You can verify the successful installation of Natural for DL/I online by using the **DL/I Services** function (described in the *Database Management System Interfaces* documentation) of the Natural SYSDDM utility:

1. Invoke online Natural.
2. Invoke the SYSDDM utility.
3. In the SYSDDM main menu, enter function code D to invoke the **DL/I Services** function.
4. On the resulting screen, enter function code D to invoke the **NDB Maintenance** function.
5. On the resulting screen, enter function code S to select the NDB which was created in ***Step 4: Build the Natural for DL/I Sample Database*** .

   On the resulting screen, enter function code L to list the NDB segments.

   On the resulting screen, enter function code A to assign DBID and FNR to the segments.
6. On the same screen, enter function code G to generate a DDM from the segment description.
7. Catalog the generated DDM.
8. Only if running under CICS:

Enter `NATPSB ON NATSYSF` in the command line where `NATSYSF` refers to the name used for DBD and PSB in *Step 4: Build the Natural for DL/I Sample Database* .

9. Edit and run the following program:

```
DEFINE DATA LOCAL
01 COURSE VIEW OF NATEXPL-COURSE
  02 COURSEN
  02 TITLE
  02 DESCRIPN
END-DEFINE
READ (100) COURSE BY COURSEN
  DISPLAY COURSEN TITLE DESCRIPN
END
```

where `NATEXPL-COURSE` refers to the name used for DBD and PSB in *Step 4: Build the Natural for DL/I Sample Database* .

XI Installing Natural for SQL/DS on z/VSE

12 Installing Natural for SQL/DS on z/VSE

This document describes the steps for installing Natural for SQL/DS (product code NSQ) on z/VSE.

Related Topic:

For information on how to operate Natural in an SQL/DS environment, see *Natural for SQL/DS* in the *Database Management System Interfaces* documentation.

Note: When used in this document, DB2 for VSE & VM is referred to as SQL/DS.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

The following software must be installed before you can install Natural for SQL/DS:

- **Software AG Editor**

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following data sets required for product installation:

Data Set Name	Contents
NSQ*vrs*.LIBJ	Sample installation jobs
NSQ*vrs*.LIBR	Source modules, macros, phases and object modules
NSQ*vrs*.INPL	Natural objects
NSQ*vrs*.ERRN	Natural error messages

Copy the data sets into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

This section provides step-by-step instructions on how to install Natural for SQL/DS.

Step 1: Generate the Natural for SQL/DS I/O Module NDBIOMO

(Job I055, Step 1600)

By executing a standard Natural batch job, this step generates the assembly source for `NDBIOMO` from the member `NDBIOTM`.

This batch job invokes the Natural program `NDBGENI` which is loaded with the Natural **INPL utility** during the installation of base Natural. `NDBGENI` contains the following two positional parameters which can be modified to meet your specific requirements:

- the first parameter specifies the DB environment and must be set to SQL/DS,
- the second parameter specifies the number of parallel dynamic prepared SQL/DS statements.

`NDBIOMO` performs the dynamic access to SQL/DS and contains all necessary `EXEC` SQL statements (see also *I/O Module NDBIOMO for Dynamic SQL Statement Execution* in the *Database Management System Interfaces* documentation). In addition, it contains some special SQL statements which cannot be executed in dynamic mode.

Check the output report created by this job for successful job completion. In addition, a condition code of 0 indicates normal completion.

Step 2: Build NDBIOMO

(Job I055, Step 1610)

- Precompile (using the SQL precompiler), assemble and link-edit the Natural for SQL/DS I/O module `NDBIOMO`. Ensure that an appropriate SQL/DS user ID and password is specified for precompiling.

Note: Since no precompiler options are specified, the default SQL/DS isolation level Repeatable Read may lead to locking problems, because all SQL/DS locks are held until the end of the transaction. Thus, depending on your application, it may be necessary to specify a different isolation level.

Step 3: Build the Natural Parameter Module

Build the Natural parameter module for batch mode. The macros and parameters mentioned in this section are described in the *Parameter Reference* documentation.

1. Modify the settings of the profile parameters supplied with the Natural parameter module as required:
   - Set the parameters supplied with the `NTVSEP` macro to configure the z/VSE batch interface. For descriptions of these parameters, see the corresponding dynamic profile parameter `VSEP`.
   - Set the parameters specific to Natural for SQL/DS supplied with the `NTDB2` macro. For descriptions of these parameters, see the corresponding dynamic profile parameter `DB2`.

     See also ***Natural Parameter Modifications for Natural for SQL/DS***.
2. Assemble and link the Natural parameter module.

Step 4: Relink your Nucleus

Modify the JCL used to link your nucleus by adding the following `INCLUDE` statements and the corresponding DLBL statements:

`INCLUDE` *`nat-parm-module`*	Natural parameter module, where *`nat-parm-module`* is the module name used in ***Step 3: Build the Natural Parameter Module***
`INCLUDE NDBNUC`	Environment-independent Natural for SQL/DS nucleus
`INCLUDE NDBNSQ`	Environment-independent Natural for SQL/DS interface
`INCLUDE NDBPARM`	Natural for SQL/DS parameter module delivered on the installation medium
`INCLUDE NDBIOMO`	Natural for SQL/DS I/O module created in ***Step 1: Generate the I/O Module NDBIOMO***
`INCLUDE` *`xxxxxxx`*	Environment-dependent SQL/DS interface (see below)

Depending on your environment(s), `INCLUDE` the appropriate environment-specific language interface *`xxxxxxx`* as shown in the following table:

Interface	Environment
ARIPRDID	In batch mode
ARIRRTED	Under CICS

Note: If you want to use Natural for SQL/DS in both environments, repeat this step for each of these environments.

Instead of link-editing your nucleus in the way described above, you have the following alternatives:

1. If you use a shared nucleus, only include NDBNUC and NDBNSQ in the link-edit of this nucleus. All other modules must be included in the link-edit of your Natural environment-dependent nucleus.
2. Remove NDBNUC and NDBNSQ from the link-edit of the nucleus and link-edit them as a separate module with the mandatory entry name NATGWDB2. The name of the resulting phase is arbitrary. However, if you use a name different from NATGWDB2, this name must be specified as an alias name in an NTALIAS macro entry of the Natural parameter module. This way of link-editing only applies if the Natural Resolve CSTATIC Addresses feature (RCA) is used.
3. Include all modules in the link-edit job of a separate Natural parameter module with the mandatory entry name CMPRMTB. The name of the resulting phase is arbitrary. This way of link-editing only applies if an alternative parameter module (see the Natural profile parameter PARM described in the *Parameter Reference* documentation) is used.
4. If link-editing is done in this way, you can install Natural for SQL/DS without having to modify your nucleus or driver.

If link-editing is done according to Step 2 or 3 above, the following applies:

Under CICS:

The resulting module must be defined via a PPT entry or RDO:

```
DFHPPT TYPE=ENTRY,PROGRAM=module-name,PGMLANG=ASSEMBLER
```

Step 5: Load the Natural Objects

(Job I061, Step 1600)

Load the Natural objects specific to Natural for SQL/DS from the NSQ*vrs*.INPL data set into the Natural system file by using the Natural **INPL utility**. The Natural objects are loaded into the Natural system libraries SYSDDM, SYSTEM and SYSSQL in the system file FNAT.

Caution: Ensure that your newly created SYSSQL library contains all necessary Predict interface programs which are loaded into SYSSQL when installing Predict (see the relevant *Predict* documentation).

Step 6: Load the Natural Error Messages

(Job I061, Step 1620)

Load the Natural error messages specific to Natural for SQL/DS from the NSQ*vrs*.ERRN data set by executing the Natural batch job that runs the ERRLODUS load program of the Natural SYSERR utility (see the *Utilities* documentation).

ERRLODUS loads the error messages into the Natural system library SYSERR in the system file FNAT.

Step 7: Generate a Static Assembler Program (Sample)

(Job 1065, Step 1600)

1. Define six Natural work files for output.
2. For the static generation process, define the following libraries as PHASE search libraries:
   - the library that contains the batch modules NDBSTAT and NDBCHNK (in the NDB*vrs*.LOAD data set), and
   - the library where you installed this Natural for SQL/DS version.
3. Define the necessary Natural commands and the Natural input for the static generation procedure.

Natural Parameter Modifications for Natural for SQL/DS

Adapt the Natural parameters described in this section to meet your requirements. For detailed information on the parameters, refer to the *Parameter Reference* documentation.

This section covers the following topics:

- NTDB2 Macro

DB2SIZE Parameter

Add the following Natural profile parameter to your Natural parameter module:

```
DB2SIZE=nn
```

The DB2SIZE parameter can also be specified dynamically. It indicates the size of the SQL/DS buffer area, which should be set to at least 6 KB.

The setting of DB2SIZE can be calculated according to the following formula:

```
((1332 + n1 * 48 + n2 * 120) + 1023) / 1024 KB
```

The variables *n1* and *n2* correspond to the following:

n1	The number of statements for dynamic access as specified as the second parameter in ***Step 1: Generate the I/O Module NDBIOMO***.
n2	The maximum number of nested database loops as specified with the MAXLOOP keyword subparameter in the NTDB2 macro.

Note: Ensure that you have also added the Natural parameters required for the Software AG Editor (see ***Installing Software AG Editor***).

Since DB2SIZE applies to Natural for SQL/DS and Natural for DB2, it should be set to the maximum value if you run more than one of these environments.

Performance Considerations for the DB2SIZE Parameter

During execution of an SQL statement, storage is allocated dynamically to build the SQLDA for passing the host variables to SQL/DS.

In previous Natural for SQL/DS versions, this storage was always obtained from the TP monitor or operating system. For performance reasons, it is now first attempted to meet the storage requirements by free space in the Natural for SQL/DS buffer (DB2SIZE). Only if there is not enough space available in this buffer, the TP monitor or operating system is invoked.

To take advantage of this performance enhancement, you must specify your DB2SIZE larger than calculated according to the **formula**. The additional storage requirements (in bytes) can be calculated as follows:

- With sending fields:

```
64 + n * 56
```

where *n* is the number of sending fields in an SQL statement.

The storage is freed immediately after the execution of the SQL statement.

- With receiving fields (that is, with variables of the INTO clause (see *into-clause*) of a SELECT statement):

```
64 + n * 56 + 24 + n * 2
```

where *n* is the number of receiving fields in an SQL statement.

The storage remains allocated until the loop is terminated.

Example:

If you use the default value 10 for both variables (*n1* and *n2*), the calculated DB2SIZE will be 3012 bytes. However, if you specify a DB2SIZE of 20 KB, the available space for dynamically allocated storage will be 17468 bytes, which means enough space for up to either 310 sending fields or 299 receiving fields.

As space for receiving fields remains allocated until a database loop is terminated, the number of fields that can be used inside such a loop is reduced accordingly: for example, if you retrieve 200 fields, you can update about 102 fields inside the loop.

Note: When using VARCHAR fields (that is, fields with either an accompanying L@ field in the Natural view or an explicit LINDICATOR clause), additional storage is allocated dynamically if the L@ or LINDICATOR field is not specified directly in front of the corresponding base field. Therefore, always specify these fields in front of their base fields.

NTDB Macro

Add an NTDB macro for the database type SQL specifying the list of logical database IDs that relate to SQL/DS tables. All Natural DDMs that refer to an SQL/DS table must be cataloged with a database ID (DBID) from this list.

DBIDs can be any number from 1 to 65535. For most user environments, one DBID (usually 250) is sufficient.

Note: Ensure that all Natural for SQL/DS DDMs used when cataloging a given program have a valid SQL/DS DBID. Also ensure that the DBIDs selected in the NTDB macro for SQL/DS do not conflict with DBIDs selected for other database systems.

The DBID for SQL/DS used when cataloging a Natural program does not have to be in the NTDB list of DBIDs used when executing this program. Therefore, when executing existing Natural programs, DBID 250 is not mandatory.

Two sample NTDB macros follow:

```
NTDB SQL,250
```

```
NTDB SQL,(200,250,251)
```

NTLFILE Macro

Set the LFILE profile parameter in the NTLFILE macro to specify a logical database ID (DBID) that relates to the database type SQL:

```
NTLFILE 101,250,1
```

This is necessary for using ISQL or calling NDBISQL with Natural for DB2.

The LFILE parameter can also be specified dynamically at the start of a Natural session.

NTDB2 Macro

Set the keyword subparameters in the NTDB2 macro according to your requirements. The NTDB2 keyword subparameters can also be specified dynamically at the start of a Natural session by using the DB2 profile parameter.

Installation Verification

You can verify the successful installation of Natural for SQL/DS by following the instructions in this section.

- Prepare your SQL/DS Environment
- Online Verification Methods
- Sample Batch Verification Job

Prepare your SQL/DS Environment

As all dynamic access to SQL/DS is performed by NDBIOMO, all Natural for SQL/DS users must have RUN privilege on NDBIOMO.

Online Verification Methods

You can verify the successful installation of Natural for SQL/DS online by using either **SQL Services** or `DEM2` example programs:

- Using SQL Services
- Using DEM2* Example Programs

Using SQL Services

You can verify the successful installation of Natural for SQL/DS by using the **SQL Services** function (described in the *Database Management System Interfaces* documentation) of the Natural SYSDDM utility:

1. Invoke Natural.
2. Invoke the SYSDDM utility.

   In the SYSDDM main menu, enter the function code `B` to invoke the **SQL Services (NDB/NSQ)** function.

   Enter the function code `S` to select all SQL/DS tables.

   The communication between Natural and SQL/DS works if all existing SQL/DS tables are displayed.
3. For one of the tables, generate a Natural DDM as described in *Generate DDM from an SQL Table* in the *Database Management System Interfaces* documentation.

   To enable SYSDDM to generate a DDM, the Natural administrator requires access to the following SQL/DS tables:

SYSTEM.SYSCATALOG
SYSTEM.SYSCOLUMNS
SYSTEM.SYSINDEXES
SYSTEM.SYSVIEWS
SYSTEM.SYSSYNONYMS
SYSTEM.SYSUSAGE

4. After you have generated a DDM, access the corresponding SQL/DS table with a simple Natural program as indicated in the following example:

```
DEFINE DATA
01 view-name OF ddm-name
02 field
...
END-DEFINE
FIND view-name WITH field = value
  DISPLAY field
END-FIND
END
```

where:

view-name is a view of the DDM *ddm-name*.

field is a DDM field.

value is the search value to be used for the field.

5. If you receive the message SYSFUL 3700, enter the system command SQLERR to display the corresponding SQL return code. SQLERR is described in the *System Commands* documentation.

Using DEM2* Example Programs

To verify and test your installation you can also use the DEM2* example programs in the Natural system library SYSSQL provided on the installation medium.

You can create an SQL/DS table with DEM2CREA, and then create the corresponding DDM by using the Natural SYSDDM utility. You can store data in the created table with DEM2STOR, and retrieve data from the table with DEM2FIND or DEM2SEL. You can also drop the table with DEM2DROP.

Sample Batch Verification Job

To verify the successful installation of the Natural interface to SQL/DS, a sample batch verification job (Job I065) is provided. This step contains sample JCL and example programs to test Natural with Natural for SQL/DS in batch mode.

The example program DEM2CONN performs the connection to the database, which is required before you can run a Natural program that accesses SQL/DS. DEM2CONN calls the DB2SERV module with the function U which in turn calls the database connect services. For details, see *Function U* described in the *Database Management System Interfaces* documentation.

The example program DEM2JOIN performs a JOIN combining information from SQL/DS SYSTEM.SYSDBSPACE and SYSTEM.SYSCATALOG.

XII Installing Natural for VSAM on z/VSE

This document describes the installation of Natural for VSAM (product code NVS) on an Adabas database or in a VSAM (Virtual Storage Access Method) file system on z/VSE.

Related Topic:

For information on how to operate Natural in a VSAM environment, see *Natural for VSAM* in the *Database Management System Interfaces* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

13 Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

For the installation of Natural ISPF in a VSAM file system, see the Natural ISPF *Installation* documentation. Be sure that you use the relevant module (`NVSISPV`) provided on the Natural for VSAM installation medium.

14 Installing Natural for VSAM on Adabas System Files on z/VSE

This document describes the steps for installing Natural for VSAM on an Adabas database on z/VSE.

Installation Medium

The **installation medium** contains the following data sets required for product installation:

Data Set Name	Contents
NVS*vrs*.LIBJ	Sample installation jobs
NVS*vrs*.LIBR	Source modules, macros, phases and object modules
NVS*vrs*.EMPL	EMPLOYEES example data
NVS*vrs*.EXPL	Natural example objects

Copy the data sets into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Step 1: Define the CICS RDO Definitions

(Job I005)

- Define the CICS RDO definitions for the sample VSAM files.

Step 2: Load the Employees Example Data

(Job I008, Steps 1403 - 1407)

- Load the VSAM-specific `EMPLOYEES` file containing employees example data from the `NVS`*`vrs`*`.EMPL` data set and define the alternate index path `EMPLX` for the file `EMPL`.

Step 3: Build the Natural for VSAM I/O Module

- Assemble and link the Natural for VSAM I/O module:
  - Under Com-plete:

    (Job I055, Steps 1410, 1411, 1415, 1416)

    Assemble the I/O module `NVSMISC` by using the parameter `SMARTS=YES` (Steps 1415 and 1416). For detailed information, see *NVSMISC Module* and `SMARTS` described in the *Database Management System Interfaces* documentation.
  - Under CICS:

    (Job I070, Step 1400)

    Use the I/O module `NVSCICS`. See *NVSCICS Module* described in the *Database Management System Interfaces* documentation.

    Note: If you are not using the most recent CICS version, the precompile step may result in a non-zero return code (4 - 16, depending on your CICS version) because of CICS commands being used that are unknown to your CICS translator. This return code can be ignored as long as the subsequent assembly step ends with a return code of 0.
  - In any other environment:

    Use the I/O module `NVSMISC`. See *NVSMISC Module* described in the *Database Management System Interfaces* documentation.

Step 4: Build the Natural Parameter Module for VSAM

(Jobs I060, I080)

Build the Natural parameter module:

1. Modify the appropriate jobs according to the batch modules or TP monitor you are relinking: Job I060 for batch, Job I080 for Com-plete and Job I080 for CICS. This applies also to ***Step 5: Link the Nucleus***.

   Add the following parameter and macro calls to your Natural parameter module:

```
VSIZE=72 NTDB VSAM, vsam-dbid NTVSAM
```

The values for `VSIZE` depend on the values specified in `NTVSAM` (see also the *VSIZE Parameter* in the *Database Management System Interfaces* documentation).

2. Assemble and link the Natural parameter module.

Step 5: Link the Nucleus

(Jobs I060, I080)

- Modify the JCL used to link your **environment-independent nucleus** by adding the following `INCLUDE` statement:

```
INCLUDE NVSNUC
```

- Modify the JCL used to link your **environment-dependent nucleus** for the Natural CICS Interface by adding the following `INCLUDE` statement:

```
INCLUDE NVSCICS
```

- Modify the JCL used to link your **environment-dependent nucleus** for Natural TP monitor interfaces other than the Natural CICS Interface by adding the following `INCLUDE` statement:

```
INCLUDE NVSMISCD
```

- Add the corresponding sublibrary for Natural for VSAM to the search chain for the linkage editor.

For information on the components and structure of Natural for VSAM, see also *Components of Natural for VSAM* and *Structure of the Natural Interface to VSAM* in the *Database Management System Interfaces* documentation.

Step 6: Load the Natural Example Objects

(Job I061, Step 1400)

- Load the Natural example objects specific to Natural for VSAM from the `NVS`*`vrs`*`.EXPL` data set into the Natural system file by using the Natural **INPL utility**.

Step 7: Customize your TP Monitor

- Customize your TP monitor environment:

TP Monitor	Instruction
Com-plete	Catalog all VSAM files to Com-plete using the CA function of the Com-plete UFILE utility (described in the *Com-plete* documentation). If you have specified `PATH=CHECK` in `NTVSAM`: 1. Catalog your front program to Com-plete using the CA function of the Com-plete ULIB utility with a region size of 40 KB if you have not changed the first default value of the `WPSIZE` parameter in the Natural parameter module. You must also catalog the front program as privileged. 2. Load the IBM routine `IGGOCLAO` either in the LPA or as a resident program.
CICS	Add the entries for the VSAM-specific example files `EMPLVS` and `EMPLVX` to your RDO definition as described in ***Step 1: Define the CICS RDO Definitions***; you can find the CICS tables on the `NVS`*`vrs`*`.JOBS` data set as `NVSIO05`.

Installation Verification

You can verify the successful installation of Natural for VSAM by performing the following:

- Log on to the Natural system library `SYSEXNVS` and run the following programs:
  - `NVSINST1`
  - `NVSINST2`
  - `NVSINST3`
  - `NVSINST4`
  - `NVSINST5`
  - `NVSINST6`

  If all these programs can be executed successfully, the installation of Natural for VSAM is completed and verified.

15 Installing Natural for VSAM on VSAM System Files on z/VSE

This document describes the steps for installing Natural for VSAM in a VSAM file system on z/VSE.

The Natural system files FNAT, FUSER, FDIC, FSEC and FSPOOL can also be located on VSAM files.

The installation of Natural for VSAM in a VSAM file system is basically a combination of the installation descriptions for both base Natural and Natural for VSAM, plus some points specific to VSAM.

For support of source object locking, a separate FLOCK file and related paths are required.

Installation Medium

The **installation medium** contains the following data sets required for product installation:

If you want to install Natural in a VSAM file system, you need the data sets for both base Natural and Natural for VSAM. The required data sets are listed in the table below:

Data Set Name	Contents
NAT*vrs*.LIBR	Source modules, macros, phases and object modules
NAT*vrs*.JOBS	Sample installation jobs
NAT*vrs*.INPL	Natural objects
NAT*vrs*.EXPL	Natural example objects
NVS*vrs*.LIBJ	Sample installation jobs
NVS*vrs*.LIBR	Source modules, macros, phases and object modules
NVS*vrs*.JOBS	Sample installation jobs
NVS*vrs*.VINI	FDIC initialization file for Natural for VSAM
NVS*vrs*.LINI	FLOCK initialization file for Natural for VSAM
NVS*vrs*.EMPL	EMPLOYEES example data
NVS*vrs*.EXPL	Natural example objects

Copy the data sets into your environment as described in *Copying Data Sets to a z/VSE Disk*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

The VSAM jobs (for example, VSAMI008) indicated in this section are identical to the jobs generated by **System Maintenance Aid** (for example, I008).

Step 1: Define the CICS RDO Definitions

(Job VSAMI005)

- Define the CICS RDO definitions for the sample VSAM files.

Step 2: Load the Employees Example Data

(Job VSAMI008, Steps 1403 - 1407)

- Load the VSAM-specific `EMPLOYEES` file containing employees example data from the `NVS`*`vrs`*`.EMPL` data set and define the alternate index path `EMPLX` for the file `EMPL`.

Step 3: Prepare the VSAM Clusters for the Natural System Files

(Job VSAMI008, Steps 1420 - 1446)

- Define three VSAM clusters to be used as system files for Natural (`FNAT`, `FUSER` and `FDIC`) and a path for `FDIC`.

  We strongly recommend that you keep these three system files on separate VSAM clusters.

Step 4: Prepare a VSAM Cluster for the Spool File

(Job VSAMI008, Steps 0300 - 0309)

This step must be performed only if you have Natural Advanced Facilities installed and want your spool file to be a VSAM file, too.

- Define an additional VSAM cluster to be used as the spool file (`FSPOOL`) and five alternate indices.

  Note: Path processing is *not* supported for `FSPOOL`.

Step 5: Prepare a VSAM Cluster for the Security File

(Job VSAMI008, Steps 9900 - 9907)

This step must be performed only if you have Natural Security installed and want your security file to be a VSAM file, too.

- Define an additional VSAM cluster to be used as the security file (`FSEC`) and three alternate indices.

  Note: Path processing is *not* supported for `FSEC`.

Step 6: Prepare a VSAM Cluster for the Scratch-Pad File

(Job VSAMI008, Steps 1450, 1451)

This step must be performed only if you want to use a scratch-pad file; that is, if you want to use read-only Natural system files (`ROSY=ON`); see also the Natural profile parameter `ROSY` and the macro `NTLFILE` described in the *Parameter Reference* documentation.

- Define an additional VSAM cluster to be used as the scratch-pad file (Step 1450).
- Initialize the VSAM scratch-pad file (Step 1451).
- Set the following parameters in the Natural parameter module according to your requirements:

```
NTLFILE 212,dbid,nt-file-number,dlbl-name-scratch-pad-file
ROSY=ON
```

- If you want your Natural system file(s) to be opened for input, adapt your Natural parameter module as follows:

```
FNAT=(dbid,fnr,filename,,RO),
FUSER=(dbid,fnr,filename,,RO),
FSEC=(dbid,fnr,filename,,RO),
```

Step 7: Prepare a VSAM Cluster for the Source Locking File FLOCK

(Job VSAMI008, Steps 1460, 1461)

This step must be performed only if you want to lock source objects in a VSAM file system (`SLOCK=PRE`); see also the parameter `SLOCK` and the macro `NTLFILE` (described in the *Parameter Reference* documentation).

- Define an additional VSAM cluster to be used as the source locking file (Step 1460).
- Load and print the example data record contained in the VSAM source locking file (Step 1461).
- Set the following parameters in the Natural parameter module according to your requirements:

```
NTLFILE 002,dbid,nt-file-number,dlbl-name-source-locking-file,,PATH
SLOCK=PRE
```

The default DLBL name (*dlbl-name*) is `FLOCK`, the related default paths are `FLOCKA`, `FLOCKB` and `FLOCKC`.

Step 8: Build the Natural for VSAM I/O Module

(Job VSAMI055, Step 1410)

- Edit, assemble and link the Natural for VSAM I/O module `NVSMISC` with the LSR options:

```
DEFER=YES
COMMIT=NO
READINT=NO
```

For the parameters that can be specified in *NVSMISC Module*, see the relevant section in the *Database Management System Interfaces* documentation.

Step 9: Build the Natural Parameter Module for Batch Mode for VSAM

(Job I060, Step 0010)

1. Modify the settings of the supplied Natural profile parameters as required for batch mode. The parameters and corresponding macros (if applicable) are described in the *Parameter Reference* documentation. The most important parameter/macro settings are described below.

   - Configure the z/VSE batch interface:
     Modify the settings of the parameters supplied with the `NTVSEP` macro to meet your requirements.
   - In addition to the `VSIZE` and `NTDB` specifications, modify the parameters `FNAT`, `FUSER` and `FDIC` as follows:

     ```
     VSIZE=160,
     FNAT=(vsam-dbid,fnr-fnat,dlbl-name-fnat),
     FUSER=(vsam-dbid,fnr-fuser,dlbl-name-fuser),
     FDIC=(vsam-dbid,fnr-fdic,dlbl-name-fdic),
     NTDB VSAM,vsam-dbid
     NTVSAM
     ```

     The *vsam-dbid* must have the same value in all four entries.

     We recommend that you use different files and different file numbers for `FNAT` and `FUSER`. The `FDIC` file must be a file different from `FNAT` and `FUSER`. Therefore, you may *not* omit the `FDIC` parameter.

     The DLBL names (*dlbl-name*) are the logical names of the Natural system files; each DLBL name can be up to seven characters long. The DLBL name for the `FDIC` path is created by appending an `X` to the DLBL name of the `FDIC` file.
   - If you have Natural Advanced Facilities installed and want your spool file to be a VSAM file, modify the `FSPOOL` parameter accordingly:

     ```
     FSPOOL=(vsam-dbid,fnr-fspool,dlbl-name-fspool)
     ```

   - If you have Natural Security installed and want your security file to be a VSAM file, modify the `FSEC` parameter accordingly:

     ```
     FSEC=(vsam-dbid,fnr-fsec,dlbl-name-fsec)
     ```

     The `FSEC` file must be a file different from `FNAT`.
   - For a quick installation, use the Natural for VSAM LSR feature and specify the following `NTVLSR` definitions in the Natural parameter module (see also *NTVLSR Macro* in the *Parameter Reference* documentation):

```
NTVLSR fnat-dlbl-name,1
NTVLSR fuser-dlbl-name,2
NTVLSR fdic-dlbl-name,3
NTVLSR fdicx-dlbl-name,3
```

If you want to use FSEC system files:

```
NTVLSR fsec-dlbl-name,4
NTVLSR fseca-dlbl-name,4
NTVLSR fsecb-dlbl-name,4
NTVLSR fsecc-dlbl-name,4
```

2. Assemble and link the batch parameter module.

Step 10: Link the Nucleus for Batch Natural

(Job I060, Step 0020)

1. Modify the JCL used to link your **environment-dependent nucleus** for batch Natural by adding the following INCLUDE statements:

```
INCLUDE NVSNUC
INCLUDE NVSFNAT
INCLUDE NVSFSPO
INCLUDE NVSFSEC
INCLUDE NVSFLOCK
INCLUDE NVSMISCD
```

   The module NVSFSPO is only required if you have Natural Advanced Facilities installed and want your spool file to be installed in a VSAM file system.

   The module NVSFSEC is only required if you have Natural Security installed and want your security file to be installed in a VSAM file system.

   The module NVSFLOCK is only required if you want to lock source objects contained in an FUSER or FNAT system file in a VSAM file system.

   If your **environment-dependent nucleus** is not linked to your **environment-independent nucleus**, NVSMISCD must be linked to the Natural parameter module instead.

2. Add the corresponding sublibrary for Natural for VSAM to the search chain for the linkage editor.

For information on the components and structure of the Natural interface to VSAM, see also *Components of Natural for VSAM* and *Structure of the Natural Interface to VSAM* in the *Database Management System Interfaces* documentation.

Step 11: Load the Natural Objects

(Job I061, Step 0100)

- Load the Natural objects from the NAT*vrs*.INPL data set into the Natural system file by using the Natural **INPL utility**.

  Ensure that the DLBL names specified in the Natural parameter module (see ***Step 9: Build the Natural Parameter Module for Batch Mode for VSAM***) are also specified for the load function performed with the Natural INPL utility. In addition, a path DLBL name (*dlbl-name-fdic*X) must be specified for FDIC.

Note: If you want to install any other Software AG products that require Natural objects to be loaded with the Natural **INPL utility**, ensure that the corresponding installation steps are adapted according to **Job VSAMI061**.

Step 12: Load the Natural Example Objects

(Job I061, Step 0103, and Job VSAMI061, Step 0103)

- Load the Natural example objects from the NAT*vrs*.EXPL data set into the Natural system file by using the Natural INPL utility (Job I061, Step 0103).
- Load the Natural example objects specific to Natural for VSAM from the NVS*vrs*.EXPL data set into the Natural system file by using the Natural INPL utility (Job VSAMI061, Step 0103).

Ensure that the DLBL names specified in the Natural parameter module (see ***Step 9: Build the Natural Parameter Module for Batch Mode for VSAM***) are also specified for the load function performed with the Natural INPL utility. In addition, a path DLBL name (*dlbl-name-fdic*X) must be specified for FDIC.

Step 13: Reorganize the FNAT System File

- Reorganize the FNAT system file by using the VSAM facility AMS REPRO to unload and reload the file.

Step 14: Build the Natural for VSAM I/O Module for CICS

(Job VSAMI070, Step 1400)

This step must be performed only if you want to install Natural for VSAM under CICS.

- Assemble and link the module NVSCICS.

Step 15: Link the Nucleus for Natural Under a TP Monitor

(Job VSAMI080)

Proceed with the TP monitor-specific installation steps for base Natural described in the relevant sections in *Installation for z/VSE*, taking into account the following additions:

- Modify your VSAM-specific Natural parameter module according to ***Step 9: Build the Natural Parameter Module for Batch Mode for VSAM***.
- Add the following `INCLUDE` statements to all links of the online nucleus:

```
INCLUDE NVSNUC
INCLUDE NVSFNAT
INCLUDE NVSFSPO
INCLUDE NVSFSEC
INCLUDE NVSFLOCK
```

  The module `NVSFSPO` is only required if you have Natural Advanced Facilities installed and want your spool file to be installed in a VSAM file system. The online environment for Natural Advanced Facilities must be a CICS environment, and the spool files installed in a VSAM file system must be defined in the CICS FCT.

  The module `NVSFSEC` is only required if you have Natural Security installed and want your Natural security system file to be installed in a VSAM file system. The VSAM Natural security system files installed in a VSAM file system must be defined in the CICS FCT.

  The module `NVSFLOCK` is only required if you want to lock source objects contained in an FUSER or FNAT system file in a VSAM file system. The locking files installed in a VSAM file system must be defined in the CICS FCT.

- Modify the JCL used to link your **environment-dependent nucleus** for the Natural CICS Interface by adding the following `INCLUDE` statement:

```
INCLUDE NVSCICS
```

- Modify the JCL used to link your **environment-dependent nucleus** for Natural TP monitor interfaces other than the Natural CICS Interface by adding the following `INCLUDE` statement:

```
INCLUDE NVSMISCD
```

- Add the corresponding sublibrary for Natural for VSAM to the search chain for the linkage editor. Before starting Natural, ensure that the DLBL names of the Natural system files in the VSAM file system are known in your batch and online environments.

Step 16: Customize your TP Monitor

Customize your TP monitor environment:

TP Monitor	Instruction

Com-plete	Add the following DLBL statements to your Com-plete startup job:
	
	`// DLBL FNAT,'DSN=SAGLIB.VSAM.FNAT',,VSAM,CAT=xxxx`
	`// DLBL FUSER,'DSN=SAGLIB.VSAM.FUSER',,VSAM,CAT=xxxx`
	`// DLBL FDIC,'DSN=SAGLIB.VSAM.FDIC',,VSAM,CAT=xxxx`
	`// DLBL FDICX,'DSN=SAGLIB.VSAM.FDIC.PATH',,VSAM,CAT=xxxx`
	`// DLBL EMPLVS,'DSN=SAGLIB.VSAM.EMPLVS',,VSAM,CAT=xxxx`
	`// DLBL EMPLVX,'DSN=SAGLIB.VSAM.EMPLVX.PATH',,VSAM,CAT=xxxx`
	
	If Natural Security is installed, add the following DLBL statements to your Com-plete startup job:
	
	`// DLBL FSEC,'DSN=SAGLIB.VSAM.FSEC',,VSAM,CAT=xxxx`
	`// DLBL FSECA,'DSN=SAGLIB.VSAM.FSEC.AIXA',,VSAM,CAT=xxxx`
	`// DLBL FSECB,'DSN=SAGLIB.VSAM.FSEC.AIXB',,VSAM,CAT=xxxx`
	`// DLBL FSECC,'DSN=SAGLIB.VSAM.FSEC.AIXC',,VSAM,CAT=xxxx`
	
	If you want to lock source objects contained in the `FUSER` or `FNAT` system file in the VSAM file system, add the following DLBL statements to your Com-plete startup job:
	
	`// DLBL FLOCK,'DSN=SAGLIB.VSAM.FLOCK',,VSAM,CAT=xxxx`
	`// DLBL FLOCKA,'DSN=SAGLIB.VSAM.FLOCK.PATHA',,VSAM,CAT=xxxx`
	`// DLBL FLOCKB,'DSN=SAGLIB.VSAM.FLOCK.PATHB',,VSAM,CAT=xxxx`
	`// DLBL FLOCKC,'DSN=SAGLIB.VSAM.FLOCK.PATHC',,VSAM,CAT=xxxx`
	
	If you have specified `PATH=CHECK` in `NTVSAM`, catalog your front program to **Com-plete** using the CA function of the Com-plete ULIB utility (described in the *Com-plete* documentation) with a region size of 36 KB if you have not changed the first default value for the `WPSIZE` (described in the *Parameter Reference* documentation) in the Natural parameter module.

CICS	Add the following entries to your FCT: ■ the Natural system files `FNAT`, `FUSER`, `FDIC` and `FDICX` required for VSAM; ■ the Natural example files `EMPLVS` and `EMPLVX` provided for VSAM; ■ the Natural Security files `FSEC`, `FSECA`, `FSECB` and `FSECC` if you have Natural Security installed; ■ the VSAM files `FLOCK`, `FLOCKA`, `FLOCKB` and `FLOCKC` if you want to lock source objects contained in the `FUSER` or `FNAT` system file in the VSAM file system. Refer to Job VSAMI005 for examples. You can add DLBL statements for these data sets to your CICS startup job, too.

Installation Verification

You can verify the successful installation of Natural for VSAM by performing the following:

- Log on to the Natural system library `SYSEXNVS` and run the following programs:
  - `NVSINST1`
  - `NVSINST2`
  - `NVSINST3`
  - `NVSINST4`
  - `NVSINST5`
  - `NVSINST6`

  If all these programs can be executed successfully, the installation of Natural in a VSAM file system is completed and verified.

Note for Batch Mode:

For verification in batch mode, you can run Job VSAMI200 which executes the above programs.

Restrictions

The AIX files that relate to the Natural `FSEC` and `FSPOOL` system files provided for a VSAM file system cannot be accessed using a path definition. The reason is that null values are not suppressed during VSAM upgrade handling for AIX keys. The record length of AIX files related to `FSEC` and `FSPOOL` would be exceeded for AIX keys filled with blanks or binary zeros. This would cause problems under CICS, as the record length supported is limited to 32 K only. Natural for VSAM supports null-value suppression for AIX keys and the upgrade handling for AIX files.

XIII Installing Natural Security on z/VSE

16 Installing Natural Security on z/VSE

This document describes the steps for installing Natural Security (product code NSC) on z/VSE.

Related Topic:

For information on the features and functions provided by Natural Security, see the *Natural Security* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Additional Prerequisites for Natural Security in a Heterogeneous Environment

In addition to the prerequisites referred to above, the following software must be installed to use Natural Security in a heterogeneous environment:

- Entire Net-Work
- Natural Security for Mainframes

The following software must be installed as required:

- Natural Security for UNIX
- Natural Security for Windows
- Natural Security for OpenVMS

For further information, see *Using Natural Security on Multiple Platforms* in the *Natural Security* documentation.

Installation Medium

The **installation medium** contains the following data sets required for product installation:

Data Set Name	Contents
NSC*vrs*.SYSL	Natural Security log file
NSC*vrs*.INPL	Natural objects
NSC*vrs*.VINI	Natural Security FDIC initialization file for VSAM system files

Copy the data sets into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the NAT*vrs*.LIBJ data set and are prefixed with the product code. The data set is provided on the installation medium supplied for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Note for the Reinstallation:

- When you repeat a Natural Security installation, only the Natural objects are replaced; the Natural Security data defined for SYSSEC and DBA is *not* reset to the values defined after the initial installation. You can use the **Natural Security Recover** function of the Natural **INPL utility** to reset the data.

This section describes the actual installation steps:

- Step 1: Load the FSEC System File Definition
- Step 2: Load the Log File
- Step 3: Load the Logon and Error Log File
- Step 4: Build the Natural Parameter Module
- Step 5: Load New Natural Objects

Step 1: Load the FSEC System File Definition

(Job I050, Step 9900)

Skip this step if you want to use an existing Natural FSEC system file.

Skip this step if you want to load the FSEC system file into a VSAM file system. In this case, refer to ***Installing Natural for VSAM***. See also *Natural for VSAM with Natural Security* in the *Database Management System Interfaces* documentation for restrictions on the use of the FSEC system file in a VSAM environment.

- Load the new Natural FSEC system file definition:
  1. Set the System Maintenance Aid (SMA) parameter NSC-FIRST-INSTALL to Y (Yes). This is the default setting.
  2. Specify the database ID and file number of the Adabas file where to load the FSEC system file by using the Adabas ADALOD utility.

     In addition, you must specify this database ID and file number in the Natural parameter module as described in ***Step 4: Build the Natural Parameter Module***.
  3. Load the FSEC system file definition contained in the NAT*vrs*.SYSF data set by using the Adabas ADALOD utility.

     The following ADALOD utility parameter must *not* be changed:

     ```
     ISNREUSE=YES
     ```

Step 2: Load the Log File

(Job I050, Step 9901)

This step only applies if Adabas is installed and if the Natural Security function **Logging of maintenance functions** (see the *Natural Security* documentation) is to be used. It creates the log file to be used by the function.

- Load the log file by using the Adabas ADALOD utility. Input for ADALOD is the NSC*vrs*.SYSL data set.

Step 3: Load the Logon and Error Log File

(Job I050, Step 9902)

This step only applies if Adabas is installed and if the Natural Security function **Store Logon and Error Data on Separate System Files** (see the *Natural Security* documentation) is to be used. It creates the logon and error log file to be used by the function.

- Load the logon and error log file by using the Adabas ADALOD utility. Input for ADALOD is the NAT*vrs*.SYSF data set.

Step 4: Build the Natural Parameter Module

(Jobs I060, I080)

1. Specify the following profile parameter in your Natural parameter module:

   ```
   FSEC=(database-id,file-number)
   ```

   where *database-id* and *file-number* are the database ID and file number of either the new FSEC system file loaded in **Step 1** or your existing FSEC system file.

   The FSEC profile parameter is described in the *Parameter Reference* documentation.

   Repeat Job I080 for all your TP monitors.
2. Assemble and link your Natural parameter module.

Step 5: Load New Natural Objects

(Job I061, Step 0102 or Step 9905)

For the migration installation (Step 0102):

1. Set the System Maintenance Aid (SMA) parameter NSC-FIRST-INSTALL to N (No). The default setting is Y (Yes).
2. Load the Natural objects specific to Natural Security from the NSC*vrs*.INPL data set into the appropriate Natural libraries in your FNAT system file by using the Natural **INPL utility**.

   Once this step has been performed, it is not possible to remove Natural Security from the Natural system file; to remove Natural Security from the system file, you would have to delete the entire contents of the system file and reinstall all Natural components again.

For the initial installation (Step 9905):

1. Set the System Maintenance Aid (SMA) parameter NSC-FIRST-INSTALL to Y (Yes).
2. Load the Natural objects specific to Natural Security from the NSC*vrs*.INPL data set into the appropriate Natural libraries in your FNAT system file by using the Natural **INPL utility**.

   When you load the contents of the NSC*vrs*.INPL data set for the first time, this step creates the following security profiles and relationships:

   - A library security profile with the library ID SYSSEC. The library is people-protected (**People-protected** set to Y and **Terminal-protected** set to N).
   - A user security profile with the user ID DBA, the user type Administrator, and the password set to DBA.

     The user DBA is linked to the library SYSSEC (ordinary link, no special link).

Installation Verification

Natural Security is operational after ***Step 5: Load New Natural Objects*** of the *Installation Procedure* has been completed successfully.

For the initial installation, proceed as described in the section *First Steps After the Installation* in the *Natural Security* documentation. If you upgrade from a previous Natural Security version, you can skip the steps described in this section.

XIV Installing Natural Advanced Facilities on z/VSE

This document describes the installation of Natural Advanced Facilities (product code NAF) on z/VSE.

Installing Natural Advanced Facilities under CICS on z/VSE

Installation Verification for Natural Advanced Facilities under CICS on z/VSE

Related Topic:

For information on the features and functions provided by Natural Advanced Facilities, see the *Natural Advanced Facilities* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

17 Installing Natural Advanced Facilities under CICS on z/VSE

This document describes the steps for installing Natural Advanced Facilities under CICS on z/VSE.

Prerequisites

Supported versions of the following products must be installed before you can install Natural Advanced Facilities:

- **Natural CICS Interface**
- **Natural for VSAM** if a VSAM file is to be used as a spool file

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following data sets required for product installation:

Data Set Name	Contents
NAF*vrs*.SYSF	Natural FSPOOL system file definition
NAF*vrs*.LIBJ	Sample installation jobs
NAF*vrs*.LIBR	Source modules, macros, phases and object modules
NAF*vrs*.INPL	Natural objects
NAF*vrs*.ERRN	Natural error messages

Copy the data sets into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Step 1: Define the CICS RDO Definitions

(Job I005)

This step only applies when using VSAM system files.

- Define the CICS RDO definitions for the sample VSAM files:

  Add the Natural Advanced Facilities spool files (`SPOOL`, `SPOOLA`, `SPOOLB`, `SPOOLC`, `SPOOLD` and `SPOOLE`) to your FCT.

For performance reasons, we strongly recommend you to specify a transaction ID for the spool server, which is different from that of the terminal task. It is then possible to dedicate special threads to the spool server.

If you want to convert an existing VSAM spool file, the FCT must contain the entries for this spool file. The cluster names of the new and the old versions must be different. The VSAM database ID and file number as well as the VSAM DD names must be unique.

Step 2: Prepare a VSAM Cluster for the Spool File

(Job I008, Steps 0300 - 0311)

This step only applies when using VSAM system files.

- Define and initialize a VSAM cluster to be used as the spool file (`FSPOOL`) and five alternate indices.

Step 3: Load the FSPOOL System File Definition for Adabas

(Job I050, Step 0300)

This step only applies when using Adabas system files.

Skip this step if you want to use an existing spool file of Natural Advanced Facilities.

Load the new Natural `FSPOOL` system file definition contained in the `NAF`*`vrs`*`.SYSF` data set by using the Adabas ADALOD utility.

Step 4: Build the NAFPARMC Parameter Module

(Job I055, Step 0305)

The use of the NAFPARMC parameter module is optional. Alternatively, you can set the server options with Function 30 (see the *Natural Advanced Facilities* documentation) of the SYSPOOL application.

- Modify, assemble and link the NAFPARMC module.

Step 5: Build the Natural Parameter Module

(Jobs I060, I080)

1. Modify the parameters FSPOOL, NTPRINT, NAFUPF and NAFSIZE in the Natural parameter module according to your site requirements. For more information on these parameters, see *NATSPOOL Initialization* in the *Natural Advanced Facilities* documentation.
2. Assemble and link the Natural parameter module.

VSAM System Files

The following additional step applies when using VSAM system files:

- Set the FSPOOL parameter as follows:

```
FSPOOL=(vsam-dbid,fnr-fspool,dd-name-fspool)
```

where:
vsam-dbid is the database ID of the VSAM file to be used as the spool file,
fnr-fspool is the file number of the VSAM file to be used as the spool file,
dd-name-fspool is limited to seven characters.

Step 6: Link the Nucleus

(Jobs I060, I080)

- Add the following INCLUDE statements to the link steps for Natural and link-edit the executable module:

```
INCLUDE NAFAF
```

`INCLUDE NAFNUC`
`INCLUDE NAFPARMC` (optional)

The link-edit of the object module containing Natural Advanced Facilities can be done in any of the following ways:

- Include all modules of Natural Advanced Facilities, that is, `NAFNUC`, `NAFAF` and, optionally, `NAFPARMC` in the link-edit of Natural.

  Note: If a shared nucleus is created, the modules can be included in the shared nucleus.

- Link-edit `NAFNUC`, `NAFAF` and, optionally, `NAFPARMC` and an alternative Natural parameter module as a separate module with the mandatory name `CMPRMTB` specified in the `ENTRY` statement. The name of the resulting module is optional.

  Note: This way of link-editing only applies if an alternate parameter module (`PARM=`*`parameter`*) is used. If so, an additional CICS PPT entry with `PROGRAM=`*`name`* is required.

- Link-edit `NAFNUC`, `NAFAF` and, optionally, `NAFPARMC` as a separate module with the mandatory name `NATAM08` specified in the `ENTRY` statement. The name of the resulting module is optional. If it is different from `NATAM08`, however, it must be specified as an alias name in the `NTALIAS` macro of the Natural parameter module.

  Note: This way of link-editing only applies if the `CSTATIC` and `RCA` profile parameters (see the *Parameter Reference* documentation) are used. If so, an additional CICS PPT entry with `PROGRAM=`*`name`* is required.

The following additional step applies when using VSAM system files.

- Add the following `INCLUDE` statement to all links of the nucleus:

```
INCLUDE NVSFSPO
```

Step 7: Load the Natural Objects

(Job I061, Step 0300)

- Load the Natural objects specific to Natural Advanced Facilities from the NAF*vrs*.INPL data set into the Natural system files SYSPOOL and SYSPRINT by using the Natural **INPL utility**.

  Ensure that the INPL load function finishes with the message:

```
Natural Advanced Facilities initialized by INPL
```

  If this initialization fails, various problems will be encountered at execution time.

The maps contained on the data set are provided in source form so that you can modify them according to your requirements (for example, translate the maps from English into another language). If you modify these maps, ensure that all fields have the same format/length/relative position in the map. Failure to abide by this restriction will result in an invalid system.

Step 8: Load the Natural Error Messages

(Job I061, Step 0304)

- Load the Natural error messages specific to Natural Advanced Facilities from the NAF*vrs*.ERRN data set into the Natural FDIC system file by using the ERRLODUS program of the Natural SYSERR utility (described in the *Utilities* documentation).

Step 9: Create a Separate Thread Group for Printer Transaction

(Job I070, Steps 2245, 2250, and Job I080)

We recommend that you establish a separate thread group for the Natural Advanced Facilities printer transaction. To do so, perform these steps:

1. Modify the Natural CICS Interface system directory (Step 2245):

   Include a definition of the Natural Advanced Facilities printer thread group into the Natural CICS Interface system directory.
2. Relink the modified Natural CICS Interface system directory (Step 2250).

   Repeat linking of the Natural CICS Interface system directory.

See also *Build the Natural CICS Interface System Directory Module* in *Installing Natural CICS Interface on z/OS* or *Installing Natural CICS Interface on z/VSE* in the Natural *Installation* documentation.

Step 10: Define Natural Advanced Facilities for Natural Security

This step must be performed only if Natural Advanced Facilities is being installed in a Natural Security environment.

- Define SYSPOOL to Natural Security with the startup program MENU.

  Note: The physical CICS printers and the application SYSPRINT need not be defined to Natural Security. The Natural Security logon processing will identify the NATSPOOL spool server and perform a simplified logon to SYSPRINT, that is, without any further security checks. In this way, maintenance efforts and the number of Adabas calls at the start of the spool server are considerably reduced. Any logon to SYSPRINT attempted by users other than the NATSPOOL spool server will be rejected by Natural Security, regardless of whether SYSPRINT is defined to it or not.

Step 11: Start Natural

- Start Natural and add the user profile, as defined in the NAFUPF parameter of the Natural parameter module, to the SYSPOOL file by using Function 31.1 (described in the *Natural Advanced Facilities* documentation).

  Note: A NAT7201 message is issued at the start of the session indicating that the profile has not yet been added to the SYSPOOL file.

Step 12: Create the NATSPOOL Environment

- Initialize a new NATSPOOL environment as described in *NATSPOOL Initialization* in the *Natural Advanced Facilities* documentation.

Step 13: Define Natural Advanced Facilities for VTAM/SNA

This step must be performed only if Natural Advanced Facilities is to be used in conjunction with VTAM/SNA printers.

- Define devices in the TCT with a RELREQ setting of (YES,YES). (This will ensure that VTAM printers are released at the end of printout time when devices are shared with TSO, BATCH, JES, etc.)
- Define TRMSTAT=INTLOG or CREATESESS=YES for the printer to allow EXEC CICS START requests to create a session.
- Ensure that the devices have the SHARE option generated into the controller VTAM specifications.

Step 14: Define Natural Advanced Facilities for VTAM/non-SNA

This step must be performed only if Natural Advanced Facilities is to be used in conjunction with VTAM/non-SNA printers.

1. Include `TRMSTAT=TRANSCEIVE` in the TCT definition for the device.
2. Set the VTAM definition for the device parameter `ISTATUS` to `ACTIVE`.

18 Installation Verification for Natural Advanced Facilities under CICS on z/VSE

You can verify the successful installation of Natural Advanced Facilities by following the instructions provided in this section.

System Testing

You can perform the following steps to determine whether Natural Advanced Facilities functions correctly.

To print a test report

- In the appropriate logical printer, set the `Disposition` to `K` for testing.

  Log on to the Natural system library `SYSPOOL` and execute the programs `NTEST` and `SPPTEST`. These programs contain `WRITE (1)` statements.

  Or:

  Create test reports with Function 42 described in the *Natural Advanced Facilities* documentation.

To proceed if a test report cannot be printed

1. Check that the CICS printer status is `IN SERVICE`.
2. Use the CICS message switching transaction `CMSG` to route a message to the specified printer:

   ```
   CMSG 'message',ROUTE=term-id,SEND
   ```

   where *term-id* is the terminal identification of the CICS printer as specified in the TCT.
3. If Natural Security is installed, check whether logon error records (of the `NATSPOOL` spool server `NATP`) have been written to the Natural Security system file. The maintenance system for these error records can be invoked by entering `ERROR` in the command line of Natural Security.
4. Check that the standard Natural `LOGON` object has *not* been replaced. The `NATSPOOL` spool server `NATP` must be able to execute the `SVPCIC01` module in the `SYSPRINT` library.

   Also check whether the `LOGON` user exit causes the `LOGON` of the spool server to `SYSPRINT` to terminate.
5. Check that the catalog dates of the objects in `SYSPRINT` and `SYSPOOL` are identical to one another.
6. Check that the parameters for the spool server match the Natural Advanced Facilities environment. To do so, invoke `MENU` in the `SYSPOOL` library and check the **CICS Options** defined with Function 30.5 (see the *Natural Advanced Facilities* documentation).

7 Check that neither the external security (`EXTSEC` operand of CICS PCT) nor the security levels (`TRANSEC` operand of CICS PCT) are defined for the spool server `NATP`. This ensures that `NATP` can be initiated without security violations.

8 Check the TCT and VTAM definitions (see ***Step 10: Define Natural Advanced Facilities for Natural Security***, ***Step 13: Define Natural Advanced Facilities for VTAM/SNA*** and ***Step 14: Define Natural Advanced Facilities for VTAM/non-SNA*** in the section *Installing Natural Advanced Facilities under CICS*).

9 Allocate a screen device instead of a printer device to the `Destination/Form` pairing of the first logical printer, execute `NTEST` in the `SYSPOOL` library, and check for error messages on the screen.

On certain devices, the CICS abend ATNI may occur.

Since the Natural Security logon processing checks whether the device is a printer, this test is only possible if Natural Security is *not* installed.

10 Start the transaction `CEDF` for your terminal device and check whether the command `EXEC CICS START` (to start the spool server) is executed.

11 Start the transaction `CEDF` for your printer device (`CEDF` terminal ID of printer) and check which commands are executed.

12 Check the Natural Advanced Facilities messages on the system operator console and/or in the log file (the destination of messages is defined with Function 30.5 (see the *Natural Advanced Facilities* documentation).

Console messages sent by the spool server start with:

```
NAF SP-SERV:
```

Console messages sent by the terminal task start with:

```
NAF-
```

13 Obtain a `NATSPOOL` trace and check which Natural Advanced Facilities modules are called, which Adabas commands are executed, and which return codes are encountered.

NATSPOOL Trace using SYSRDC

To obtain an online `NATSPOOL` trace by using the Natural SYSRDC utility

1 Start a Natural session with the following dynamic parameters:

```
RDCSIZE=100, TRACE=(NATAM08), ITRACE=ON
```

2 Log on to the SYSRDC library.

3 Execute the following command to select the internal trace type:

```
RDCSET N
```

4 Execute the following command to start the trace:

```
RDCSTART
```

5 Execute a program which creates a report, for example, NTEST in the SYSPOOL library.

6 Log on to the SYSRDC library.

7 Execute the following command to display the trace entries:

```
RDCDISP
```

You will now see when a NATSPOOL module begins (marked as BEG) and ends (marked as END) as well as its return code in decimal representation.

After the execution of an Adabas call (marked as ADA), you will see the command code, the first byte of the command ID and the return code in decimal representation.

For detailed information on the SYSRDC utility, see the *Utilities* documentation.

NATSPOOL Reason Codes

Errors that may occur during the check for printer availability:

Error	Description
INV REQU	Invalid request
INV ID	Invalid ID
INV ADDR	Invalid address
INV DEVC	Invalid logical device code for page status
ATI REQU	ATI required on NON-ATI terminal
RESO PRO	Resource problem for inter-partition session
INV PROG	Invalid program name
UNAB PER	Unable to perform request
INV TYPE	Type is not LUC
RESO QUI	Resource quiesced by TMP
LOCATERR	Any error different from those listed above

Errors that may occur during the start of the spool server:

Error	Description
TERMIDER	Terminal ID error
TRANIDER	Transaction ID error
SYSIDERR	SYSID error
INVREQ	Invalid request
IOERR	I/O error
LENGERR	Length error
ISCINVRE	ISC invalid request
NOTAUTH	Not authorized
STARTERR	Any error different from those listed above

NATSPOOL Initialization Console Messages

Messages that may occur during the initialization of NATSPOOL:

Message	Text
NAF-01C	ADABAS RC*xxx*, DBID*xxx*, FNR*xxx*, AT OPEN
NAF-02C	Not used
NAF-03C	ADABAS RC*xxx*, DBID*xxx*, FNR*xxx*, AT READ REPORT
NAF-04C	ADABAS RC*xxx*, DBID*xxx*, FNR*xxx*, AT READ PRINTER
NAF-05C	ADABAS RC*xxx*, DBID*xxx*, FNR*xxx*, AT CLOSE
NAF-06C	ADABAS RC*xxx*, DBID*xxx*, FNR*xxx*, AT UPDATE REPORT
NAF-07C	ADABAS RC*xxx*, DBID*xxx*, FNR*xxx*, AT UPDATE PRINTER
NAF-08I	REPORT *xxxxxxxxx*, JOB*xx.xxx*, SET -TO BE PRINTED-
NAF-09I	PRINTER *xxxx* SET -FREE-
NAF-10I	PRINTER *xxxx* RESTARTS IN 20 SECONDS, DEST=*xxxxxxxx*, FORM=*x*
NAF-11C	RESTART ERROR NAT *xxxx* ON PRINTER *xxxx*

NATSPOOL Print Server Messages

See the online help for descriptions of the NATSPOOL print server messages.

To invoke online help for print server messages

1 In the **Natural Spool Administration Menu**, press PF1.

2 Enter function code 99 (miscellaneous information).

3 Enter function code 4 (Natural Advanced Facilities SP-SERV messages from spool server).

NATSPOOL Abend Codes

Abend Code	Reason	Action
NAF1 - INVALID LENGTH	The length of the data to be printed is not positive.	Obtain the dump and contact Software AG technical support.
NAF2 - INVALID LENGTH	The length of the data to be printed is greater than the maximum length of the terminal I/O buffer (TIOBM).	Obtain the dump and contact Software AG technical support.
NAF3 - INVALID RETURN CODE	The return code of the task-end routine CMTSKND is not zero.	Obtain the dump and contact Software AG technical support.
NAF4 - INVALID RETURN CODE	The return code of the print routine CMWTERM is not zero.	Obtain the dump and contact Software AG technical support.

XV

19 Installing Natural Optimizer Compiler on z/VSE

20 Installing Natural Optimizer Compiler Version 8.2.7 on z/VSE

This document describes the steps for installing the Natural Optimizer Compiler Version 8.2.7 (product code NOC) on z/VSE.

Related Topic:

For information on the features and functions provided by the Natural Optimizer Compiler, see the *Natural Optimizer Compiler* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following data set required for product installation:

Data Set Name	Contents
NOC*vrs*.LIBR	Source modules, macros, phases and object modules

Copy the data set into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the NAT*vrs*.LIBJ data set and are prefixed with the product code. The data set is provided on the installation medium supplied for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

- Step 1: Build the Natural Parameter Module

Step 1: Build the Natural Parameter Module

(Jobs I060 and I080)

1. Activate the Natural Optimizer Compiler by adding the following macro to your Natural parameter module:

   ```
   NTOPT 'INDX,OVFLW,ZD=OFF'
   ```

   See also *Macro NTOPT* in the *Natural Optimizer Compiler* documentation.
2. Assemble and link the Natural parameter module.

Step 2: Relink the Nucleus

(Jobs I060, I080)

Adapt the link steps for Natural:

1. Add the following `INCLUDE` statement to the link of the nucleus to include the Natural Optimizer Compiler modules:

   ```
   INCLUDE NOCNUC
   ```

2. Relink your nucleus as described in ***Link the Nucleus*** in *Installing Natural*.

Installation Verification

You can verify the successful installation of the Natural Optimizer Compiler by performing the following steps:

1. Recatalog an existing program or write a new program and then catalog it.
2. Check the directory information for the program you have just cataloged, by using the following `LIST` system command:

   ```
   LIST DIR object-name
   ```

   The directory information for the specified object will be displayed, showing the size of the machine code at the bottom of the screen.

21 Installing Natural Optimizer Compiler Version 8.3.5 on z/VSE

This document describes the steps for installing the Natural Optimizer Compiler Version 8.3.5 (product code NOC) on z/VSE.

Related Topic:

For information on the features and functions provided by the Natural Optimizer Compiler, see the *Natural Optimizer Compiler* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

The following software must be installed before you can install the Natural Optimizer Compiler Version 8.3.5:

- **Natural** Version 8.2.7 (or higher)

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following data set required for product installation:

Data Set Name	Contents
NOC*vrs*.LIBR	Source modules, macros, phases and object modules

Copy the data set into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the NAT*vrs*.LIBJ data set and are prefixed with the product code. The data set is provided on the installation medium supplied for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Step 1: Build the Natural Parameter Module

(Jobs I060 and I080)

1. Activate the Natural Optimizer Compiler by adding the following macro to your Natural parameter module:

   ```
   NTOPT 'INDX,OVFLW,ZD=OFF'
   ```

   See also *Macro NTOPT* in the *Natural Optimizer Compiler* documentation.
2. Assemble and link the Natural parameter module.

Step 2: Relink the Nucleus or Dynamically Load Modules

You can either statically link the Natural Optimizer Compiler modules to the nucleus or dynamically load them when initializing a Natural session.

Relink the Nucleus:

(Jobs I060, I080)

Adapt the link steps for Natural:

1. Add the following `INCLUDE` statement to the link of the nucleus to include the Natural Optimizer Compiler modules:

   ```
   INCLUDE NOCNUC
   ```

2. Relink your nucleus as described in ***Link the Nucleus*** in *Installing Natural*.

Dynamically Load Phases

1. Add the library containing the phases statement:

```
// LIBDEF PHASE,SEARCH=(,,,,,SAGLIB.NOCvrs,...),TEMP
```

2. At the start of a Natural session, set the Natural profile parameter RCA as follows:

```
RCA=NOCGW,RCALIAS(NOCGW,NOCNUC)
```

RCA is described in the *Parameter Reference* documentation.

Installation Verification

You can verify the successful installation of the Natural Optimizer Compiler by performing the following steps:

1. Recatalog an existing program or write a new program and then catalog it.
2. Check the directory information for the program you have just cataloged, by using the following LIST system command:

```
LIST DIR object-name
```

The directory information for the specified object will be displayed, showing the size of the machine code at the bottom of the screen.

XVI Installing Natural Connection on z/VSE

22 Installing Natural Connection on z/VSE

This document describes the steps for installing Natural Connection (product code NTC) on z/VSE.

Related Topic:

For information on the features and functions provided by Natural Connection, see the *Natural Connection* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

A supported version of the following product must be installed before you can install Natural Connection:

- Entire Connection

  See the *Installation* section in the *Entire Connection* documentation.

See also ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Installation Medium

The **installation medium** contains the following data set required for product installation:

Data Set Name	Contents
NTC*vrs*.LIBR	Source modules, macros, phases and object modules

Copy the data set into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the NAT*vrs*.LIBJ data set and are prefixed with the product code. The data set is provided on the installation medium supplied for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Step 1: Build the Natural Parameter Module

(Jobs I060, I080)

1. Adapt the Natural parameter module:
   - Specify the keyword subparameter `AM=PC` in the `NTPRINT` macro for all printer files and work files to be used for data transfer between the host and the PC. For example:

     ```
     NTPRINT (7),AM=PC
     NTWORK  (7),AM=PC
     ```

   - Specify the profile parameter `PC=ON`.

   The parameters and the macro are described in the *Parameter Reference* documentation.
2. Assemble and link the Natural parameter module.

Step 2: Adapt the Link Steps

(Job I080)

Adapt the link steps for online Natural:

1. Add the Natural Connection sublibrary into the LIBDEF OBJ search:

   ```
   // LIBDEF OBJ,SEARCH=(.....,SAGLIB.NTCvrs,....),TEMP
   ```

2. Add the following `INCLUDE` statement in the SYSIPT to the Natural link job for the linkage editor:

   ```
   INCLUDE NTCPCAM3
   ```

3. Relink your nucleus as described in ***Link the Nucleus*** in *Installing Natural*.

Installation Verification

You can verify the successful installation of Natural Connection by following the instructions below:

1. Invoke Entire Connection on the PC and start the terminal emulation.
2. Use the terminal emulation to invoke Natural on the mainframe.
3. Enter the terminal command `%+` to activate the PC connection.
4. Use the Natural Object Handler (described in the *Utilities* documentation) to download Natural objects to the PC. For information on transferring Natural objects, see also *Data Transfer* in the section *Terminal Emulation* in the *Entire Connection* documentation.
5. Verify that the downloaded objects are now on your PC.

XVII Installing Natural Review on z/VSE

23 Installing Natural Review on z/VSE

This document describes the steps for installing Natural Review (product code RNM) on z/VSE.

Related Topic:

For information on the features and functions provided by Natural Review, see the *Natural Review* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

Storage Requirements

The Natural Review monitoring system requires CICS shared storage for collecting its monitoring data. For each Natural session, a user account area of 656 bytes is allocated plus space for a Natural call table (NCT). This table is used to track the Natural programs and database calls issued within a transaction. The number of NCT entries is determined by the `NCTSIZE` parameter in the source member `RNMSCB3.A` (see ***Step 7: Build the Natural Review System Control Block***). Each table entry is 48 bytes long and the table contains 32 entries (default). This yields a total amount of 656+32*48=2192 bytes per running session. The storage is reused when the session terminates.

Each active response time report requires a basic control block that is 400 bytes long. Each detail record is 128 bytes long. The maximum number of detail records per report is controlled by the `Number of Records` in the report definition. If the `Transaction Summary` option is set in the report definition, a 64 bytes transaction summary area is allocated for each transaction ID.

After a report has been written to the Natural Review repository file by the history session, all report-depending storage is released.

Installation Medium

The **installation medium** contains the following data sets required for product installation:

Data Set Name	Contents
RNM*vrs*.SYSF	ADALOD loadable RNM system file
RNM*vrs*.LIBR	Source modules, macros, phases and object modules
RNM*vrs*.INPL	Natural objects

In the RNM*vrs*.LIBR, the following sublibrary object types are used:

.A	Assembler source code, examples, and so on
.J	Installation jobs
.OBJ	Object modules

Copy the data sets into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Sample Jobs

Sample installation jobs are contained in the NAT*vrs*.LIBJ data set and are prefixed with the product code. The data set is provided on the installation medium supplied for base Natural.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

- Step 1: Add CICS Control Table Entries for Natural Review
- Step 2: Load the Natural Review Repository File
- Step 3: Build the Natural Parameter Module
- Step 4: Link the Nucleus
- Step 5: Load New Natural Objects
- Step 6: Define the SYSRNM Library in Natural Security
- Step 7: Build the Natural Review System Control Block
- Step 8: Link the Natural Review History Session Startup Module
- Step 9: Activate the Natural Review Modules

Step 1: Add CICS Control Table Entries for Natural Review

(Job I005, Step 2211)

Define the following CICS table entries with RDO. You may have to include additional parameters according to your requirements. Natural Review supports transaction isolation (that is, it can run in user key).

Program Control Table (PCT)

1. Define the Natural Review history session start/stop transaction:

```
CEDA DEFINE TRANSACTION(RVH1) PROGRAM(RNMHIST3)
     GROUP(RNMvr)
```

2. Define the Natural Review asynchronous Natural history session transaction:

```
CEDA COPY TRANSACTION(xxxx) GROUP(yyyy) AS(RVH2)
     TO(RNMvr)
```

   This copies your existing Natural transaction ID *xxxx* from your Natural definition group *yyyy* as an alias transaction for Natural Review. This definition is optional and you may use the online Natural transaction code *xxxx* as well, but it helps the administrator identify the Natural Review asynchronous history session.

   The transaction codes `RVH1` and `RVH2` can be chosen freely (that is, you may change them if required). The Natural session transaction code must be the same as defined with the parameter `NATTRAN` (in this example, `NATTRAN=RVH2`). See ***Step 7: Build the Natural Review System Control Block*** and *Installation Verification* (**Step 3**).

Processing Program Table (PPT)

1. Define the Natural Review history session start/stop program:

```
CEDA DEFINE PROGRAM(RNMHIST3) LANGUAGE(ASSEMBLER)
     GROUP(RNMvr)
```

2. Define the Natural Review system control block:

```
CEDA DEFINE PROGRAM(RNMSCB3) LANGUAGE(ASSEMBLER)
     RESIDENT(YES) GROUP(RNMvr)
```

3. After entering the online definitions, activate them by using `CEDA INSTALL GROUP(RNMvr)`. The new `GROUP` should be added to the `GRPLIST` defined for CICS cold start.

Program List Tables CICS Startup and Shutdown (PLTPI and PLTSD)

This table entry is optional. You can use it for automatic start and termination of the Natural Review history session during CICS startup and shutdown.

1. Define the Natural Review history session start/stop program:

```
DFHPLT TYPE=ENTRY,PROGRAM=RNMHIST3
```

2. Insert the table entry in your assembled PLTPI and/or PLTSD CICS table. This avoids manual starting and stopping of the Natural Review history session (see ***Step 7: Build the Natural Review System Control Block***).

Step 2: Load the Natural Review Repository File

(Job I050, Step 2620)

You can skip this step if you have already installed a Natural Review repository file and want to continue using it. The format of the repository file is compatible with previous versions.

The repository file is an Adabas file used for storing response time reports and history data. It is possible to share the repository file across several CICS regions. Any Adabas file can be used to contain the Natural Review repository file. The corresponding file number must be defined to Natural as a logical system file (see ***Step 3: Build the Natural Parameter Module***). The repository file has to be initialized via Natural (see ***Step 10: Initialize the Natural Review Repository File***).

- Modify the job as follows before submitting it:
  - Change the data set definitions according to your requirements.
  - Change `DB=dbid` in the Adabas `ADARUN` statements to the correct database ID.
  - Change `SVC=nnn` to the correct Adabas SVC number.
  - Change the Adabas `ADALOD` statement to the file number `fnr` of the Adabas file that will contain the Natural Review repository file, and the Natural Review version `vrs` you are running:

```
ADALOD LOAD FILE=fnr,NAME='REPOS-RNMvrs'
```

Step 3: Build the Natural Parameter Module

(Job I060, Steps 0010, 0015)

1. Modify the settings of the parameters supplied with the Natural parameter module as follows:

```
RDCSIZE=2
MADIO=5000
MAXCL=0
RCA=NATGWREV
NTLFILE 180,dbid,fnr
```

   where:

   `RDCSIZE` is the Natural profile parameter that determines whether a session is monitored by Natural Review. If you set `RDCSIZE=0` (this is the default) for a session, it is *not* monitored by Natural Review.

   `RCA` is the Natural profile parameter required if you want to link the Natural Review monitor interface module separate from the nucleus (see ***Step 7: Build the Natural Review System Control Block***) by means of the RCA technique.

   `NTLFILE` (or dynamic `LFILE` parameter) is the macro that determines the Natural Review repository file used by the `SYSRNM` application for retrieving and maintaining report definitions and storing history report data for the history session. The currently accessed repository file can be changed using the `LFILE` command within Natural Review (see the *Natural Review* documentation).

   `dbid` is the database ID and *`fnr`* the file number specified in ***Step 2: Load the Natural Review Repository File***.

   For details about the parameters mentioned above, see the *Parameter Reference* documentation.
2. Assemble and link the Natural parameter module.

Step 4: Link the Nucleus

(Job I060, Step 0020)

- Link the nucleus for Natural Review by including the following module from the Natural Review library:

```
INCLUDE RNMNUC3
```

Natural Review uses the Natural Data Collector exit interface to get data from Natural. For more information on the SYSRDC Data Collector, see the Natural SYSRDC utility described in the *Utilities* documentation.

The Natural Review module RNMNUC3 supports all different ways of statically linked Natural subprograms. For more information about linking the nucleus, see *Linking Natural Objects to the Natural Nucleus* in the Natural *Operations* documentation.

If you run an **environment-independent nucleus** for multiple environments (for example, CICS and batch), link RNMNUC3 to the environment-dependent nucleus, because Natural Review monitoring runs under CICS only. This prevents unnecessary overload in your non-CICS systems.

- If RNMNUC3 is not linked to the **environment-dependent nucleus**, the following CICS assembler command level stub from the CICS library must be linked to RNMNUC3:

```
INCLUDE DFHEAI
```

- Instead of linking RNMNUC3 to the nucleus, you can link it as a separate module defined with the Natural profile parameter RCA (described in the *Parameter Reference* documentation). The following linkage editor statements are then required:

```
PHASE NATGWREV,*
MODE RMODE(ANY)
INCLUDE DFHEAI
INCLUDE RNMNUC3
ENTRY NATGWREV
```

The MODE statement is optional. A CICS PPT entry is required for the module NATGWREV. It must be specified with the profile parameter RCA (see ***Step 3: Build the Natural Parameter Module***). The module can be shared between multiple Natural nuclei.

Step 5: Load New Natural Objects

(Job I061, Step 2661)

1. If you want to continue using a profile text object DEFAULT you modified according to your requirements, rename your DEFAULT object in the SYSRNM library before you start loading the new Natural objects.
2. Load the Natural objects specific to Natural Review from the RNM*vrs*.INPL data set into your FNAT system file by using the Natural INPL utility.

   It is sufficient to load the Natural objects only into one FNAT system file even if you want to monitor other FNAT system files as well.

The Natural Review User Profile Subsystem is initialized when the profile text object DEFAULT is copied to the SYSRNM library.

3. If you renamed the profile text object DEFAULT in the first step, replace the newly loaded DEFAULT by this object.

Step 6: Define the SYSRNM Library in Natural Security

This step only applies if Natural Security is installed.

- Define the Natural Review library SYSRNM in Natural Security.

  You can define a startup menu for the SYSRNM library. If the library is People-protected, each user of this library must be linked to it.

- Define REVHIST as a user of type PERSON with a default application of SYSRNM. REVHIST is used as the user ID by the Natural Review history session.

Step 7: Build the Natural Review System Control Block

(Job I070, Steps 2622, 2623)

The Natural Review System Control Block RNMSCB3 is defined as a program in CICS. RNMSCB3 is not an executable program. Its storage is used by Natural Review as the common anchor and control point for all monitored Natural sessions and reports within one CICS address space. There are some installation-specific generation parameters you can specify in object RNMSCB3 in the Natural Review source library.

- Set the following parameters in RNMSCB3 according to your requirements:

Parameter	Explanation	
NATTRAN=	This is the Natural/CICS transaction code for the Natural Review history session. You must specify this parameter to set the correct Natural transaction code (see *Step 1: Add CICS Control Table Entries for Natural Review*).	
NPARMS=	Additional dynamic Natural parameters for the Natural Review history session. This parameter is optional.	
CLOSE=	This parameter determines whether any started Natural Review reports are closed automatically during the termination of the Natural Review history session. There are two possible values:	
	CLOSE=YES	All started reports are closed. This is the default setting.
	CLOSE=NO	Started reports are not closed.
DATE=	The date format used in the records stored in the Natural Review repository file. There are two possible values:	

Parameter	Explanation	
	DATE=OLD	The date format is *YY/MM/DD*. This is the default setting.
	DATE=NEW	The date format is *YYYYMMDD*.
EMPTY=	This parameter determines whether empty history records are stored in the Natural Review repository file. A record is considered empty if no transactions occurred within the report time interval. There are two possible values:	
	EMPTY=YES	Empty history records are stored.
	EMPTY=NO	Empty history records are not stored. This is the default setting.
NCTSIZE=	This parameter determines the number of entries in the Natural Call Table (NCT) of Natural Review. The NCT is allocated in CICS shared storage and is used to track the usage of the Natural programs per session. Possible values: 0 - 128. The default setting is NCTSIZE=32.	

- Modify and run the job RNMI070 to generate the Natural Review System Control Block. The module must be linked with the NORENT option. The target link library can be any library defined to CICS.

Step 8: Link the Natural Review History Session Startup Module

(Job I070, Step 2625)

Natural Review history data is written to the Natural Review repository file by an asynchronous (that is, not terminal-bound) Natural session.

- Modify and run job RNMLINK in the Natural Review source library.

  It links the CICS-dependant history session startup module RNMHIST3. The target link library can be any library defined to CICS.

  There can be only one history session within one CICS address space. The Natural objects specific to Natural Review (see ***Step 5: Load New Natural Objects***) must be loaded to the FNAT system file running with the history session.

Step 9: Activate the Natural Review Modules

- Use the following `CEMT` transaction to activate the nucleus module to which `RNMNUC3` and the modified the Natural parameter module are linked:

```
CEMT SET PROGRAM(. . . .) NEWCOPY
```

- Restart CICS if you cannot activate the nucleus module.

Step 10: Initialize the Natural Review Repository File

Skip this step if your Natural Review repository file is already initialized.

- Initialize the repository file after loading it in *Step 2: Load the Natural Review Repository File*:

  Logon to the `SYSRNM` library and enter the following at the `NEXT` prompt:

```
INSTALL NM
```

  The following messages will then appear:

```
Now creating sample report system response time
Now creating sample report highest response
Natural Review repository initialization complete.
Press <ENTER> to continue
```

  During the initialization process, two default (`sample`) response time reports are added as shown above.

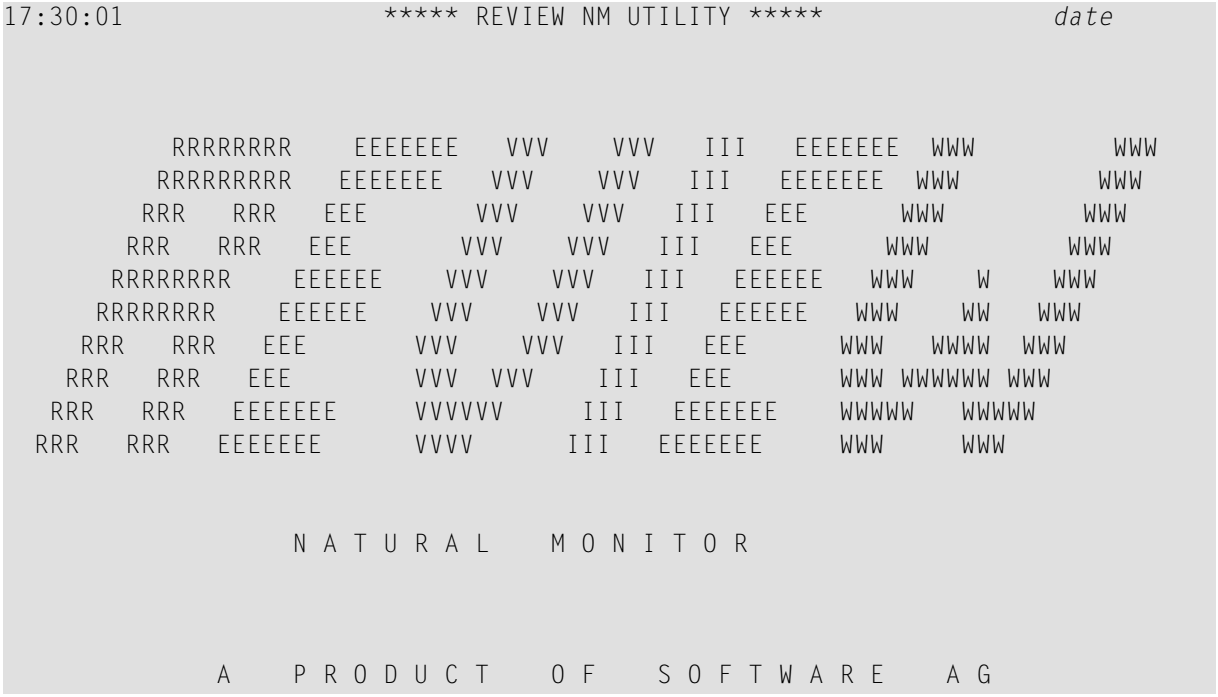
Installation Verification

You can verify the successful installation of Natural Review by performing the following steps:

1. At any Natural command prompt, enter the following:

```
SYSRNM
```

   Natural Review is started and a Natural Review logo screen similar to the example below appears:



(You can suppress this screen by specifying `BANNER=N` in the text object `CONFIG` in the library `SYSRNM`.)

2. Press ENTER to invoke the Natural Review **Main Menu** screen:

```
17:30:17                  ***** REVIEW NM UTILITY *****                date
                                    - Main Menu -



                        Code  Function

                         NM   Natural Monitor System
                         UP   User Profile System









 Command ===>

 REV00001 - Welcome to Review NM running under CICS

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help        Exit                                                  Fin
```

Alternatively, you can invoke this menu by logging on to the Natural library `SYSRNM` and then entering the following command:

```
MENU
```

3. In the command line of the Natural Review **Main Menu** screen, enter either of the following commands:

   ```
   NM
   ```

   (for **Natural Monitor System**)

   or

   ```
   UP
   ```

   (for **User Profile System**)

   Depending on the command entered, the main screen of the requested subsystem appears. You can then check your installation parameters and the status of the history session. For detailed information on the functions provided by the subsystems, see the *Natural Review* documentation.
4. If you want to run history reports, start the history session. It runs as an asynchronous (non-terminal) Natural session and writes the collected report data to the repository file each time a report time interval has expired.

You can start and stop the history session either automatically by using the CICS PLTPI/PLTSD (see ***Step 1: Add CICS Control Table Entries for Natural Review***), or manually outside Natural by using the RVH1 transaction (see ***Step 1: Add CICS Control Table Entries for Natural Review***) in the following ways:

RVH1	Start the Natural Review history session with the transaction code RVH2 as defined with NATTRAN in ***Step 8: Build the Natural Review System Control Block***. The history session can also be started with the START command of Natural Review (see the *Natural Review* documentation).
RVH1 STOP	Terminate the Natural Review history session. The history session can also be terminated with the STOP command of Natural Review (*Natural Review* documentation).
RVH1 TEST	Start the Natural Review history session on the current terminal. This option can be used for debugging purposes, for example, to debug the history session with CEDF (the CICS debugging facility). Note that no Natural terminal I/Os are supported during the normal processing of the history session. Therefore, you have to terminate the history session from another terminal.

5. Check the Natural Review initialization messages on the console log of your CICS system to find out whether the history session has started successfully. The following message should appear: REV20200 - REVIEW NM HISTORY SESSION STARTED.

   For more detailed explanations of the messages in Natural Review, enter MSG followed by the message number (for example, MSG 1) in the command line of Natural Review.

XVIII Installing Entire Transaction Propagator on z/VSE

24 Installing Entire Transaction Propagator on z/VSE

This document describes the steps for installing the Entire Transaction Propagator (product code ETP) on z/VSE.

Related Topic:

For information on the features and functions provided by the Entire Transaction Propagator, see the *Entire Transaction Propagator* documentation.

Notation *vrs* or *vr*:

When used in this document, the notation *vrs* or *vr* represents the relevant product version (see also Version in the *Glossary*).

Entire Transaction Propagator Version 1.5.2 Compatibility with Earlier Releases

It is essential that the ETPNUC module and the Natural programs for Entire Transaction Propagator in the Natural system library SYSETP have the same version. When using a Version 1.5.2 ETPNUC module with Entire Transaction Propagator Version 1.4.1 Natural programs - or vice versa - can cause unpredictable results.

The Natural profile parameter ETPSIZE (see the *Parameter Reference* documentation) is still accepted by Natural, but it is no longer necessary to specify this parameter. The required storage (approx. 10 KB) will be automatically allocated when the first call to an Entire Transaction Propagator database is issued. Any value specified for the ETPSIZE parameter will be ignored. The size of the ETPSIZE parameter should be left at its default (ETPSIZE=0).

As the Natural profile parameter DATSIZE (see the *Parameter Reference* documentation) is automatically adjusted to the required length, it is no longer necessary to adjust the value of the Natural DATSIZE parameter for a Natural environment running a replication task. However, the DATSIZE in such an environment will be approximately 170 KB.

After Entire Transaction Propagator Version 1.5.2 is installed and when the MENU command is invoked for the first time in the Entire Transaction Propagator maintenance utility, the Entire Transaction Propagator administration file is automatically migrated to the Entire Transaction Propagator Version 1.5.2 format and an appropriate message is displayed.

Afterwards, any attempt to access the migrated administration file from an earlier version of the Entire Transaction Propagator maintenance utility is denied.

Before enabling the new Entire Transaction Propagator maintenance utility, all log files must be empty and all log and confirmation files must be accessible for the migration process. Migration of the administration file from the Entire Transaction Propagator Version 1.4 format to the Entire Transaction Propagator Version 1.5 format is performed automatically when the Entire Transaction Propagator Version 1.5 maintenance utility is invoked for the first time. If a previous Entire

Transaction Propagator version is installed, you should use the Natural SYSMAIN utility (see the *Utilities* documentation) to delete the contents of the library `SYSETP` before installing Entire Transaction Propagator Version 1.5.2. Following installation of Entire Transaction Propagator Version 1.5.2, the maintenance utility of any earlier Entire Transaction Propagator version will be denied access to the administration file.

Otherwise, Entire Transaction Propagator Version 1.5.2 is compatible with Entire Transaction Propagator Version 1.4.1. The FDT (Field Definition Table) of the administration, log and confirmation files remains unchanged.

Prerequisites

See ***General Prerequisites and System Support*** in the section *Overview of the Installation Process*.

For using the Entire Transaction Propagator with 3GL programs under CICS, see ***Installing ETP Interface for CICS***.

Installation Medium

The **installation medium** contains the following data sets required for product installation:

Data Set Name	Contents
`ETP`*vrs*`.SYSF`	System file definition for multiple use: administration, logging and confirmation
`ETP`*vrs*`.SYS1`	System file definition for administration
`ETP`*vrs*`.SYS2`	System file definition for logging
`ETP`*vrs*`.SYS3`	System file definition for confirmation
`ETP`*vrs*`.LIBJ`	Sample installation jobs
`ETP`*vrs*`.LIBR`	Source modules, macros, phases and object modules
`ETP`*vrs*`.INPL`	Natural objects
`ETP`*vrs*`.ERRN`	Natural error messages
`ETP`*vrs*`.FDTA`	System file for containing FDT definitions for all Entire Transaction Propagator files

Copy the data sets into your environment as described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Installation Procedure

Be sure to read ***Installation Process and Major Natural Features*** before you start the installation procedure.

Step 1: Load the Entire Transaction Propagator System Files

(Job I050, Step 5300)

The `ETP`*`vrs`*`.SYSF` data set has a Field Definition Table (FDT) that is suitable for containing the administration, confirmation and log files within one physical Adabas file.

Load the `ETP`*`vrs`*`.SYSF` data set from the installation medium by using Job I050.

The following files are suitable for installing individual Entire Transaction Propagator files:

- `ETP`*`vrs`*`.FDTA`

  This file contains the same FDTs as the `ETP`*`vrs`*`.SYSF` data set, but in a format suitable for loading with the Adabas ADACMP utility.

- `ETP`*`vrs`*`.SYS`*`n`*

  This data set contains separate sample Adabas files that can be loaded using the Adabas ADALOD utility. These files have FDTs suitable for defining individual administration, log and confirmation files.

The Adabas utility parameters `ISNREUSE=YES` (for mainframes) and/or `REUSE=ISN` (for UNIX, Windows or OpenVMS systems) can be set to reuse freed ISNs as they become available for Entire Transaction Propagator master, replicate, confirmation, administration and/or log files.

Step 2: Modify the Natural Parameter Module

Before linking the assembler modules as described in the next step, refer to *Step 4: Specify Master File Databases* and *Step 8: Define the Administration File*.

Reassemble and relink the Natural parameter module after you have modified it.

Note: You must relink the modified Natural parameter module to all Natural nuclei that update a master file or start a replication task.

Step 3: Link the Assembler Modules

You must link the ETPNUC module to all Natural nuclei that update a master file or start a replication task. We recommend that you relink Natural with the ETPNUC module.

To avoid the need to relink Natural with ETPNUC, specify the Natural profile parameter RCA=ON or RCA=NATGWETP in the Natural parameter module to allow dynamic loading of the Entire Transaction Propagator during Natural startup. In this case, perform the following steps:

1. Use the following example to create a phase with the name NATGWETP:

```
* $$ JOB JNM=LINK,CLASS=O,DISP=D
* $$ LST CLASS=A,DISP=H
// JOB LINK
// LIBDEF OBJ,SEARCH=(SAGLIB.ETPvrs),TEMP
// LIBDEF PHASE,CATALOG=libname.ssssss,TEMP
// OPTION CATAL,LIST
 PHASE NATGWETP,*,NOAUTO
 INCLUDE ETPNUC
/*
// EXEC LNKEDT
/*
/&
* $$ EOJ
```

2. Place the NATGWETP module in a library from which your TP monitor or batch system can perform dynamic loads. In Com-plete systems, NATGWETP can be loaded as a resident program.

Step 4: Specify Master File Databases

Use the NTDB macro in the Natural parameter module to specify the databases containing the master files:

```
NTDB ADABAS,dbid,ETP
```

Or:

```
NTDB ADABAS,(dbid,dbid,...),ETP
```

where *dbid* specifies one or more databases separated by commas, each containing one or more master files.

You can define the same database as an ENTIRE database and also as an Entire Transaction Propagator database if you specify both options for the NTDB macro:

```
NTDB ADABAS,dbid,ETP,ENTIRE
```

It is also possible to specify the databases containing the master files dynamically at Natural startup using the DB parameter:

```
DB=(ADABAS,(dbid,dbid,...),ETP)
```

The first time that a database which is defined as an Entire Transaction Propagator database is accessed, Entire Transaction Propagator automatically obtains a buffer (ETPSIZE) of the required size (approx. 10 KB). Any value specified for the Natural profile parameter ETPSIZE is ignored. The size of the ETPSIZE parameter should be left at its default (ETPSIZE=0).

Step 5: Install the Maintenance Utility

The Entire Transaction Propagator maintenance utility is a menu-based control facility for defining and managing master and replicate files. Although not required, we recommend that you install the maintenance utility on every node containing a master file.

Step 6: Load the Natural Objects and Natural Error Messages

(Job I061, Steps 5300, 5301)

Load the Natural objects and Natural error messages specific to the Entire Transaction Propagator by performing the following steps:

1. Step 5300: Load the ETP*vrs*.INPL data set (contains the Natural objects) into your Natural FNAT system file by using the Natural **INPL utility**.

2. Step 5301: Load the ETP*vrs*.ERRN data set (contains the Natural error messages) into your Natural FNAT system file by using the ERRLODUS program of the Natural SYSERR utility (described in the *Utilities* documentation).

Caution: The SYSETP library should be protected against general access with Natural Security or an equivalent security facility to prevent uncoordinated changes to the administration file. Such changes can destroy the consistency and integrity of the master and replicate files.

Step 7: Define Natural Security Options

The Entire Transaction Propagator and the maintenance utility support the concept of functional security, meaning that selected functions can be allowed or disallowed under control of Natural Security. When a user is restricted by Natural Security from performing a specific Entire Transaction Propagator function, the function is not displayed on the user's corresponding menu.

Note: Disallowing the general dialog functions - for example, EXIT, CANCEL - can cause unpredictable results.

To install the Entire Transaction Propagator if Natural Security is already installed, perform the following steps:

1. Log on to the library SYSSEC.
2. Issue the command ADD LIBRARY SYSETP.
3. In the **Add Library** menu, enter the appropriate information and select **Additional Options**.
4. In the **ADDITIONAL OPTIONS** window, select **Functional Security**.
5. In the **FUNCTIONAL SECURITY** window:
   - Define the command processor WADNCP1 for the library SYSETP.
   - Enable or disable keywords (for example, DELETE, MASTER ...) or functions (for example, REPLICATE TRANSACTIONS) as required for your site. Note that any restrictions you define here apply to *all* users.

   In a similar way, you can restrict the availability of keywords or functions for every user that has access to SYSETP.

Step 8: Define the Administration File

Each database containing an Entire Transaction Propagator master file should also contain an administration file to hold all master and replicate file definitions. The logical file ID of the administration file must be 200. To define the administration file for your Natural applications, specify the `NTLFILE` macro in the Natural parameter module:

```
NTLFILE 200,dbid,fnr
```

where *fnr* is the physical file number of the administration file and *dbid* is its physical database ID.

The administration file setting can also be changed dynamically at the start of the Natural session, using the Natural profile parameter `LFILE`:

```
LFILE=(200,dbid,fnr)
```

The administration file must always be defined before using the Entire Transaction Propagator; if the Entire Transaction Propagator maintenance utility is used, the utility prompts the user for an administration file if no valid `LFILE` definition is found.

The Entire Transaction Propagator also works correctly, even if the Natural macro `NTTF` or the profile parameter `TF` is used. However, the Entire Transaction Propagator may not work properly if you have an Adabas user exit installed which modifies the database ID or file number in the Adabas control block.

Caution: Do not change any of the information in the administration file while it is being used by a replication task.

To install an administration file in your database, use either Job I050 or the Adabas ADALOD utility to load `ETP`*vrs*`.SYS1`. You can keep the administration file quite small since it contains only a single record for every:

- master file definition
- replicate file definition
- log file
- replicate database
- user profile

Step 9: Define a Master File and its Log File

A master file is normally an existing Adabas file. If the master file is new, you must first create the file as a normal Adabas file. This description assumes that the file to be defined as a master already exists.

Before a master file can be defined, an administration file must first be defined. See *Step 8: Define the Administration File*.

After defining the administration file, perform the following steps to define a master and log file:

1. Stop all updating on the file that is to be defined as a master file.
2. Copy the file to be defined as a master file. We recommend that you use the Adabas ADAULD utility for this purpose; the Adabas ADASAV utility may also be used, but only where the file will be reloaded on the same device type as before. Note the exact date and time of the copy.
3. Using the Entire Transaction Propagator maintenance utility's **Master File Definition Maintenance** menu (see *Master File Task Screens* in the *Entire Transaction Propagator* documentation), specify the master file and log file. If desired, all master files on a database can share the same log file.
4. Restart the database operation to make the master file available again.

From this time on, all changes applied to the master file will be recorded in the log file. Any replicate files of the master file can now be defined without interfering with the master file operation, providing the log file contains all changes applied to the master file since the copy in **Step 2** above was created. For more information, see *Updating the Administration File* in the *Entire Transaction Propagator* documentation.

Transaction logging will start as soon as a master file is defined and a new Natural session with the appropriate administration file is started or the master file is updated from within the Natural session which was used to define the master file. Therefore, the procedure described above might not be applicable, especially when a new master file is to be defined in a running environment. If it cannot be guaranteed that no updates are applied to the to-be-defined master file that is to be defined while the above steps are executed, perform the following steps (note that it is required that the log file specified for a master file is empty when the master file is defined):

1. Stop the transaction replication processes.
2. Define the new master file.
3. Shut down the database containing the new master and log files.
4. Copy the master file as described above.
5. Use the Adabas ADADBM utility with the `REFRESH` function to refresh the log file.
6. Restart the database to make the master file available again; transaction logging will start as soon as a user updates the master file;
7. Copy the master file's contents into the replicate file;

8. Restart transaction replication.

Install a log file in your database by using either Job I050 or the Adabas ADALOD utility to load `ETP`*`vrs`*`.SYS2`.

The number of records in the log file depends on the number of transactions that update master files between two successive invocations of the **Clean up log file** maintenance function (provided that all transactions are replicated), which is described in the *Entire Transaction Propagator* documentation. The approximate number of log file records equals:

```
(transaction count)*(updates per transaction, +1)
```

When loading the log file, the ADALOD utility parameter `PGMREFRESH=YES` is required if you want the **Clean up log file** function to refresh the log file for improved performance.

Note: It is impossible to refresh a log file that is also used as an administration or confirmation file.

Step 10: Define and Initialize the Replicate and Confirmation Files

After the master file is defined, the replicate files can be defined. A confirmation file must also be defined on each database where a replicate file is defined. If desired, multiple replicate files on a database can share the same confirmation file. To define a replicate file, perform the following steps:

1. Using the **Add replicate file definition** screen (described in the *Entire Transaction Propagator* documentation) of the online maintenance utility, specify the database IDs and file numbers for the replicate, master and confirmation files:
2. Load the unloaded copy of the master file into the replicate file using the Adabas ADALOD utility (if the file was unloaded with ADAULD) or ADASAV (if the file was unloaded with ADASAV). Specify the parameter `USERISN=YES` for the related Adabas mainframe utilities. For the replicate files on UNIX, Windows or OpenVMS systems, the option `USERISN` must be specified when loading a non-empty file unless the distribution key is used as the replication criterion. If an empty replicate file is created, the option `USERISN` or parameter setting `USERISN=YES` is not required.

   When specifying the `MAXISN` parameter while defining the replicate file you should remember that, when using the records' ISNs as the replication criterion, the Address Converter (AC) is not automatically extended. This can occur if the Entire Transaction Propagator issues Adabas N2 calls to add new replicate file records and specify an ISN value that exceeds the file's `MAXISN` value; in such a case, a response code 113 is returned. Specify a `MAXISN` value large enough for future extensions of the replicate file.
3. Using the **Display transactions** function of the maintenance utility (described in the *Entire Transaction Propagator* documentation), check for any log file entries that have been made since the master file was copied.

4. If a replicate file contains a subset of the master file records, you should either delete all unneeded records as defined by the specified distribution key ranges or copy only the selected subset of records from the master file. This can be done using one of the following methods:
   - Use a Natural program to delete the unneeded records from the replicate file.
   - Use a Natural program to copy only the selected records from the master file to an intermediate file which is then subsequently copied to the replicate file.
   - Use the `SELCRIT` and `SELVAL` parameters of the Adabas utility function `ADAULD UNLOAD` to select only the subset for unloading. This is the recommended method.

     Caution: After the replicate file has been initialized, it should not be manually changed. Otherwise, the consistency with the master file could be destroyed.

5. To install a confirmation file in your mainframe database, use either Job I050 or the Adabas ADALOD utility to load `ETP`*`vrs`*`.SYS3` from the installation medium. For confirmation files on UNIX, Windows or OpenVMS systems, use the confirmation file field definitions in file `ETP`*`vrs`*`.FDTA` to supply field definitions for the Adabas ADAFDU utility. For every replicate file that uses the confirmation file, the latter file contains a single record.
6. If the log file contains any entries for the master file, start the replication task for the replicate file using the **Replicate transactions** maintenance function described in the *Entire Transaction Propagator* documentation. The task checks the appropriate administration file for master or replicate files in the file range to be processed. The task then synchronizes the master and replicate file by applying all updates to the replicate file that are not already applied. The replicate file is now available for use.

To add other replicate files of an existing master file, perform the steps above after creating an up-to-date copy of the master file. Note that when creating a replicate file, no logged changes to the master file should be removed from the log file. If this rule is followed, a replicate file can be added without affecting the normal mode of operation.

Step 11: Set Related Parameters

- **Natural `WH` Parameter**

  When running multiple Entire Transaction Propagator replication tasks in parallel, specify `WH=ON` to avoid NAT3145 errors (`record already in hold status for another user`) when two tasks attempt to access the same record simultaneously.

- **Reusing ISNs in Entire Transaction Propagator Files**

  The Adabas utility parameters `ISNREUSE=YES` (for mainframes) and/or `REUSE=ISN` (for UNIX, Windows or OpenVMS systems) can be set to reuse freed ISNs as they become available for Entire Transaction Propagator master, replicate, confirmation, administration and/or log files.

- **Transactions with Many Updates**

  If transactions that include a lot of updates are to be logged, increase the value of the Adabas ADARUN parameter LDEUQP. The required size for transaction logging can be computed as:

```
LDEUQP = (updates per transaction * 29)
```

Step 12: Modify the WADUSER2 and/or WADUSER3 User Exits

The two user exits WADUSER2 and WADUSER3 are delivered in source form in the library SYSETP.

- **Optional User Exit** WADUSER2

  The optional user exit WADUSER2 is a Natural subprogram for controlling file replication. WADUSER2 is called after the Entire Transaction Propagator decides whether the record in question is to be replicated or not.

  The WADUSER2 user exit, defined as User Exit 2, is called only if the user exit option is specified when a master file is defined (see *Master File Task Screens* in the *Entire Transaction Propagator* documentation). An example of WADUSER2 is included in the SYSETP library.

- **Message Handler** WADUSER3

  The subprogram WADUSER3 is used to display all messages issued by the replication task. WADUSER3 can be modified to filter the task messages and, if desired, send them directly to the operator console. The WADUSER3 subprogram receives the error number, severity level and the message text from the replication task. This allows the user to select the messages to be displayed. By means of the Natural CMWTO entry (for an example, see the program WTO in the library SYSEXTP), the messages can be sent to the operator console.

  Caution: The user exits WADUSER2 and WADUSER3 should neither issue Adabas calls that update a database file nor should they issue any End Transaction (ET) or Back Out Transaction (BT) commands; otherwise, the results are unpredictable.

Step 13: Run the Entire Transaction Propagator in Batch Mode

(Job I200, Step 5300)

In batch mode, command execution is possible only by means of direct commands. For a list of direct commands and their minimum abbreviations, see *Entering Direct Commands* in the *Entire Transaction Propagator* documentation.

The following example is for a batch file for starting a replication task:

```
LOGON SYSETP                                          (1)*
MENU                                                  (2)*
REPLICATE TRANSACTIONS                                (3)
1,1,65535,65535,1,65535,00:30:00,200,1000,N,EXIT  (4)
EXIT                                                  (5)
FIN                                                   (6)
```

* If Natural Security is installed, lines (1) and (2) may have to be changed (see *Starting the ETP Maintenance Utility* in the *Entire Transaction Propagator* documentation).

Line (4) contains the parameters for the corresponding **Replicate transactions** Entire Transaction Propagator maintenance utility screen. Parameters are entered from top to bottom, left to right. Line (5) exits from the Entire Transaction Propagator maintenance utility.

Caution: We recommend that you start replication tasks in batch mode.

The following example is for a batch file for deleting successfully replicated transactions:

```
LOGON SYSETP                                          (1)*
MENU                                                  (2)*
CLEANUP LOGFILE                                       (3)
1,1,65535,65535,,,N,00:30:00,200,1000,EXIT            (4)
EXIT                                                  (5)
FIN
```

* If Natural Security is installed, lines (1) and (2) may have to be changed (see *Starting the ETP Maintenance Utility* in the *Entire Transaction Propagator* documentation). Line (4) contains the parameters for the corresponding **Cleanup logfile** Entire Transaction Propagator maintenance utility screen. Parameters are entered from top to bottom, left to right. Line (5) exits from the Entire Transaction Propagator maintenance utility.

Caution: We recommend you to start tasks that successfully delete replicated transactions in batch mode.

If a window is displayed in batch mode, all fields are protected; the reason for this is that in most cases it is not possible to determine the number of selectable items. Therefore, the only meaningful command is `PROCESS`. The following is an example to reset the in-use flag for all replicate files:

```
MENU
RESET IN-USE * *
PROCESS
EXIT
EXIT
FIN
```

To run the above examples without problems, the following parameters in the Natural parameter module must be specified:

```
ID=','   (default setting)
IM=D
PC=OFF   (default setting)
```

Either the Natural statement `SET CONTROL '+'` or the terminal command `%=` cancel the effect of `PC=OFF`.

Installing ETP Interface for CICS

The ETP Interface for CICS (product code ETC) supports the operation of the Entire Transaction Propagator for 3GL programs running in a CICS environment. The ETP Interface for CICS is *not* needed when running Natural application programs alone with CICS.

This section describes how to install the ETP Interface for CICS.

Related Documentation:

For information on how the ETP Interface for CICS operates, refer to the *ETP Interface for CICS* in the *Entire Transaction Propagator* documentation.

Prerequisites

The following software must be installed and running before you can install the ETP Interface for CICS:

- Entire Transaction Propagator

See also ***General Prerequisites and System Support***.

Installation Medium

The **installation medium** contains the following data sets required for product installation:

Data Set Name	Contents
`ETC`*`vrs`*`.LIBR`	Source modules, macros, phases and object modules This data set contains, for example, the assembler macro `ETCPARM` for parameter module generation and an example for calling the macro `ETCPARM`.
`ETC`*`vrs`*`.LIBJ`	Sample installation jobs

Copy the data sets into your environment by performing the steps described in ***Copying Data Sets to a z/VSE Disk*** in the section *Installing Natural*.

Installation Procedure

Step 1: Create the ETCPARM Module

(Job I070, Step 5310)

Create the ETP Interface for CICS parameter module `ETCPARM` by coding the `ETCPARM` macro with appropriate parameters in an assembly file. The member `SAMPLE` contains an example which you should change according to your installation requirements. The `ETCPARM` macro call begins with the specification of one or more database IDs as positional parameters, followed by optional keyword parameters.

The following `ETCPARM` parameters are available:

dbid | ADANAME | ADMIN | AMODE31 | ASYNC | FBOPT | PUSERS | PUSERTO | SAP | STCK | TIMEOUT | TRNAME

- ***dbid* - Define Database as Entire Transaction Propagator Database**

```
ETCPARM dbid
```

Specify one or more database IDs (*dbid*) that are to be defined as Entire Transaction Propagator databases.

```
ETCPARM dbid,dbid,...
```

Multiple database IDs must be separated by commas.

```
ETCPARM *
```

To define all databases as ETP databases, specify * as the only positional parameter. This parameter must be specified. Valid database IDs are `1 - 65535`, except `255`.

```
ETCPARM (from-dbid,to-dbid)
```

When the format `(from-dbid,to-dbid)` is specified, the first database ID *from-dbid* is translated into the second *to-dbid* if the *from-dbid* is encountered in an Adabas control block. The first database ID *from-dbid* in this configuration can be zero (0).

Multiple pairs of the format `(from-dbid,to-dbid)` must be separated by commas. This format can be mixed with the format that specifies *dbid* only. Up to 512 of these pairs and/or *dbid*s may be specified.

To limit overhead, we strongly recommend that you specify only those databases containing a master file defined in the administration file (see the `ADMIN` parameter). This enables a function similar to that provided by the Natural `NTDB` macro (see the *Parameter Reference* documentation).

- **ADANAME - Specify Routine to Handle Adabas Calls**

  Syntax:

  ```
  ADANAME=routine-name
  ```

  routine-name is the name of the routine (default is `ADALNC`) to which control is passed for handling Adabas calls.

- **ADMIN - Specify Database ID and File Number of Administration File**

  Syntax:

  ```
  ADMIN=(dbid,fnr[,psw[,cipher]])
  ```

  Specify the database ID (*dbid*) and file number (*fnr*) of the administration file.

  If required, the administration file's password (*psw*) and/or cipher code (*cipher*) must also be specified.

- **AMODE31 - Enable/Disable 31-Bit Addressing Mode**

  Syntax:

  ```
  AMODE31=value
  ```

  Possible Values:

#AMODE31=NO	Setting AMODE31=NO prevents the ETP Interface for CICS from switching to 31-bit addressing mode even if such addressing is possible.
AMODE31=YES	This setting (default) allows the switch to 31-bit mode, if possible, and permits acquiring storage above the 16-MB storage line.

Specify NO only if any of the following are true:

- An application passes 24-bit addresses with the high-order (leftmost) byte is neither X'00' nor X'80'.
- An ADALNC module is being used which has been assembled and linked in 24-bit mode.

Note: We recommend that you reassemble and relink in 31-bit mode.

For operating-system-specific information, see the Adabas *Installation* documentation.

Generally, COBOL compilers generate correct 24- and 31-bit addresses.

- **ASYNCH - Specify Routine to Start Async Task**

Syntax:

```
ASYNCH=asynch-name
```

asynch-name is the name of a routine that starts an asynchronous task. The routine is addressed by a V-type address constant generated in the macro expansion and must follow standard linking conventions.

If ASYNCH is not specified and a master file definition requires the starting of an asynchronous task following and end-of-transaction (ET), an error occurs.

- **FBOPT - Provide Format Buffer Optimization**

Syntax:

```
FBOPT=(ALL,min-fbl,num-entry[,timeout])
```

```
FBOPT=(SAP,min-fbl,num-entry[,timeout])
```

```
FBOPT=NO
```

Provide format buffer optimization for large buffers if GFIDs are used extensively and Adabas has been tuned so that almost no overwrites occur. Optimization is performed only if the database calls are routed through the ETP Interface for CICS, that is, the database is defined as an ETP Interface for CICS database in the ETCPARM macro invocation (see *dbid - Define Database as Entire Transaction Propagator Database*).

Possible Values:

Value	Explanation
SAP	If the SAP R/2 application system is installed and optimization is wanted for SAP calls only, specify SAP as the first subparameter.
ALL	For optimization of calls, specify ALL.
NO	If you do not use the SAP R/2 application system, specify FBOPT=NO (the default value).
min-fbl	If either SAP or ALL are specified, the *min-fbl* value specifies the minimum format buffer length and can be any value in the range of 16-1024.
num-entry	If either SAP or ALL is specified, the *num-entry* value specifies the number of 16-byte entries in the table holding global format IDs, and can range from 64 to 32767. If this value is specified too low, optimization becomes ineffective.
timeout	If either SAP or ALL is specified, the *timeout* value specifies the time period after which the table entries defined by the *min-fbl* and *num-entry* that have not been accessed, are deleted. The value is measured in minutes and defaults to the value of the TIMEOUT parameter specified below.

- **PUSERS - Specify Number of Users for Parallel Adabas Call Execution**

  Syntax:

  ```
  PUSERS=value
  ```

  PUSERS specifies the number of users that can execute Adabas calls in parallel. For each user, a slot of 256 bytes of storage is allocated.

  value can range from 100 to 99999. The default is 100.

  A slot is only used for the time required to process an Adabas call (and all calls that may possibly be issued by the ETP Interface for CICS). The slot is then marked as being free to be reused by another user. Therefore, the default value of 100 should be sufficient for most installations. If no free slot is available, the task is terminated with the abend code ETCB. In such a case, the value of this parameter should be increased in steps of 100. PUSERS limits the number of tasks executing in ETCNUC in parallel; it does not limit the number of Adabas or CICS tasks that can be handled by ETCNUC.

- **PUSERTO - Specify Timeout Before Releasing PUSER Slot**

  Syntax:

  ```
  PUSERTO=value
  ```

  PUSERTO specifies the time, in minutes, after which a PUSERS slot is marked as being free if it has not already been released (for example, as a result of a user abend).

  value must be greater than the maximum of the values specified for the ADARUN timeout parameters TT and MXTT. Valid values are 1 to 570. The default is 30.

Note: The PUSERTO value is specified in minutes, whereas the Adabas time limits are specified in units of 1.048576 seconds.

- **SAP - Specify Adabas Support for SAP R/2 Application**

Syntax:

```
SAP=value
```

Possible Values:

SAP=YES	Specify YES if Adabas support is required for the SAP R/2 application system.
SAP=NO	If you do not use the SAP R/2 application system, use SAP=NO (the default).

Note: SAP R/2 versions that use the direct call interface without using the CICS TWA are not supported. In addition, SAP R/2 users must request a Zap from SAP support that disables SAP R/2 format buffer optimization.

- **STCK - Optional Routine for Substituting Direct STCK Instruction**

Syntax:

```
STCK=routine-name
```

routine-name specifies the name of the optional routine that substitutes direct Store Clock (STCK) instructions. The routine is addressed by a V-type address constant generated in the macro expansion, and must follow standard linking conventions.

- **TIMEOUT - Set Time Before Releasing User Work Storage**

Syntax:

```
TIMEOUT=value
```

If a user does not perform any ETP Interface for CICS actions during the number of minutes specified by TIMEOUT, the ETP Interface for CICS releases the user's work storage.

value must be greater than the maximum of the values specified for the ADARUN timeout parameters TT and MXTT. After any of the Adabas time limits (TT, TNAA, TNAE, TNAX) has expired, Adabas issues an implicit Back out Transaction (BT) for all open transactions (TT limit) and deletes the related User Queue elements. The default value is 30.

Note: The TIMEOUT value is specified in minutes, whereas the Adabas time limits are specified in units of 1.048576 seconds.

- **TRNAME - Specify Names of CICS Transactions for Further Processing**

  Syntax:

```
TRNAME=trans-name
```

```
TRNAME=(trans-name,...)
```

  trans-name denotes the name(s) of one or more CICS transactions for which further ETP Interface for CICS actions should be performed. To reduce overhead, transactions that are not specified by TRNAME are not processed further. By default, all transactions are enabled for processing with the ETP Interface for CICS.

- **Example of the ETCPARM Macro**

  The following is an example of the ETCPARM macro:

```
ETCPARM 2,(0,30),1470,ADMIN=(4,397),TIMEOUT=60,TRNAME=MYTRANS
```

  which would set the following definitions:

  - Databases 2 and 1470 are defined as ETP Interface for CICS databases.
  - If the database ID in an Adabas control block is zero, it is translated to 30 (the actual database 30 is *not* defined as an ETP Interface for CICS database, but the master files defined in the administration file must be defined with a database ID of 30, not 0).
  - File 397 on database 4 is defined as the administration file.
  - User storage for a user that remains inactive for 60 minutes is released.
  - The routine for handling Adabas calls is named ADALNC (the default).
  - The only transaction allowed to process further is MYTRANS.

Step 2: Assemble and Link the ETCPARM Module

(Job I070, Step 5311)

Assemble and link the ETCPARM module (the name of the resulting module must be ETCPARM).

Step 3: Link the Routine for the ETP Interface for CICS

(Job I080, Step 5310)

The ETP Interface for CICS can be linked with any combination of `RMODE` and `AMODE` values. You must not link the ETP Interface for CICS as a reentrant module.

Step 4: Define the CICS PPT Entry

- To identify the ETP Interface for CICS to CICS, specify the following entry in the CICS PPT for the object module that results from the linkage step:

  ```
  DFHPPT TYPE=ENTRY,PROGRAM=ada-name,RES=YES,PGMLANG=ASSEMBLER
  ```

  ada-name is the name of the object module after completing the linkage step (see the operating-system-specific installation information, above). It is not necessary to link the resulting module *ada-name* to either your 3GL applications, to Natural or to the `ADALNC` module.

  The parameter value `EXECKEY(USER)` must be set in the CICS program definition.