Installing Natural under VM/CMS

This document describes how to install Natural under VM/CMS.

The following topics are covered:

- Prerequisites
- Preparing the VM System for Natural CMS
- Preparing the Adabas System for Natural
- Building the Natural CMS System
- Installation Tape for Natural under CMS
- Transferring a License Key File from PC to Host with FTP
- Installation Procedure for Natural under CMS
- Installation Verification for Natural under CMS

For information on how to run Natural in a CMS environment, refer to *Natural under CMS* in the Natural *Operations* documentation.

For installation-related information on Unicode and code page support, refer to *Configuration and Administration of the Unicode/Code Page Environment* in the *Unicode and Code Page Support* documentation.

Notation vrs or vr: If used in the following document, the notation *vrs* or *vr* stands for the relevant **v**ersion, **r**elease, **s**ystem maintenance level numbers. For further information on product versions, see Version in the *Glossary*.

Prerequisites

The following products must be installed:

- A supported version of the z/VM operating system must be installed. For the supported versions of the operating systems, refer to *Operating/Teleprocessing Systems Required* in the current Natural *Release Notes* for Mainframes.
- A supported version of Adabas must be installed. For the supported versions, refer to *Natural and Other Software AG Products* in the current Natural *Release Notes* for Mainframes.
- As a rule of thumb, each major Software AG product requires approximately 20 MB space in the Adabas database to store the Natural objects supplied by Software AG.

Preparing the VM System for Natural CMS

Natural is completely reentrant under CMS. It is recommended that Natural under CMS be installed as a Discontiguous Shared Segment (DCSS) to avoid unnecessary paging activity by VM.

For better performance, Natural uses a buffer pool in which Natural object programs are stored and executed. If you execute a Natural program, it is fetched from the database and stored in the buffer pool. If you subsequently invoke the same program, it will be executed directly in the buffer pool. Thus, repeated fetching from the database and accompanying IUCV overhead is avoided.

In a shared buffer pool, one Natural under CMS user can execute Natural programs that have been placed in the buffer pool by another Natural under CMS user. A shared buffer pool is implemented by installing the buffer pool in a writeable saved segment.

You are recommended to install two DCSSs: one for the Natural nucleus and one for the buffer pool. Of course, several buffer pools at different virtual addresses can be defined and several Natural nucleus DCSSs with different functionalities, for example, different static parameter settings (`NATPARM`) or different Assembler options of the CMS driver (`NATCMS`).

The buffer pool shared segment is defined as `SN` (see the corresponding step of the installation procedure) and the first Natural session automatically initializes the buffer pool. When the last Natural user in VM exits Natural, the use count of the shared segment becomes 0, the buffer pool is purged from CP storage, and the next Natural session has to initialize the buffer pool again.

To avoid repeated initialization, the use count can be artificially held at 1 by including the following statement line in the `PROFILE EXEC` of a disconnected service machine, *`segment`* being the name of the Natural buffer pool shared segment:

```
SEGMENT LOAD segment
```

In this way, the Natural buffer pool is initialized only once and immediately available to a Natural session as long as the disconnected server is running.

Preparing the Adabas System for Natural

The Natural system programs are stored in the Natural system file `FNAT`. User-written Natural programs are stored in the Natural system file `FUSER`. These system files reside in Adabas files. The corresponding step of the installation procedure (see below) describes how these Adabas files can be built in an Adabas system running under CMS.

Building the Natural CMS System

Building the Natural CMS system comprises the following steps:

- building the Natural nucleus and
- loading the Natural system file.

The Natural nucleus should be built

- as a module (for testing) and
- as a saved segment (for production use).

The Natural system file is loaded directly from tape using the installation `EXEC`, `NATINPL`; see the corresponding step of the installation procedure.

The CMS driver `NATCMS` is included in the Natural module and in the saved segment. `NATCMS ASSEMBLE` consists of one macro that generates the driver.

Description of the Macro Parameters

For a detailed description of the macro parameters, refer to the corresponding step of the installation procedure. The driver `NATCMS` and the parameter module `NATPARM` are assembled during the installation process. Optionally, the modules `NATTEXT`, `NATTXT2`, `NATTXT3`, `NATCONFG` and `NATPM` can be modified and reassembled.

The list of text files to be included in the Natural module or DCSS is contained in REXX program `NAT$LOAD EXEC` (variable `LOADLIST`). To customize your Natural system, modify this `EXEC` with `XEDIT` by changing the `LOADLIST` as required.

User ID for Performing the Installation

The user ID used for performing the installation should have a minidisk of 80 cylinders on 3390-type disks or an SFS directory of 14000 blocks. CP privilege class E is needed to issue the command `CP SAVESYS`. The size of the virtual machine allowed for the user ID must be adequate to accommodate the highest address of the saved segments that you plan to generate.

Installation Tape for Natural under CMS

The installation tape was created under z/OS; it has standard z/OS labels and headers. It contains the datasets listed in the table below. The sequence of the datasets is shown in the *Report of Tape Creation* which accompanies the installation tape.

Dataset Name	Contents
NAT*vrs*.LICS	Natural License Key File. For further information on license key file, license key file installation, product license check and product license check FAQs, see *Licensing Natural*. If a license key file is supplied as an e-mail attachment, see *Transferring the License Key File from PC to Host with FTP*
NAT*vrs*.SYSF	Empty Natural system file.
NAT*vrs*.TAPE	CMS installation material. This dataset is in TAPE DUMP format and must be loaded onto the installation minidisk.
NAT*vrs*.ERRN	Natural error messages.
NAT*vrs*.LDEL	Instructions to delete Natural system objects of Version 4.1
NAT*vrs*.INPL	Natural system objects.
NAT*vrs*.EXPL	Natural example objects.

Copying the Tape Contents to a VM/CMS Disk

1. On the Natural installation tape, a license key file is provided.

   If the sequence number of NAT*vrs*.LICS, as shown by the *Report of Tape Creation*, is *n*, you must position over 3 *n* - 2 tape marks (that is, FSF 1 for the first dataset, FSF 4 for the second, etc.).

   ```
   FILEDEF IN TAP1 (RECFM FB LRECL 80 BLKSIZE 3120
   FILEDEF OUT DISK NATLIC DATA A (RECFM F LRECL 80 BLKSIZE 80
   MOVEFILE IN OUT
   ```

2. To position the tape for the TAPE LOAD command, calculate the number of tape marks as follows:

   If the sequence number of NAT*vrs*.TAPE, as shown by the *Report of Tape Creation*, is *n*, you must position over 3 *n* - 2 tape marks (that is, FSF 1 for the first dataset, FSF 4 for the second, etc.).

3. Access the disk that is to contain the Natural installation files as disk A.

   The size of the disk must be at least 14000 4-KB blocks, for example, 80 cylinders on 3390-type disks.

4. Ask the system operator to attach a tape drive to your virtual machine at address X'181' and mount the Natural installation tape.

5. Position the tape by issuing the CMS command:

   ```
   TAPE FSF fsfs
   ```

   where *fsfs* is the number of tape marks and is calculated as described above.

6. Load the Natural under CMS installation material by issuing the CMS command:

```
TAPE LOAD * * A
```

7. Keep the tape drive attached to your virtual machine, because the tape is still needed during the installation procedure.

Transferring a License Key File from PC to Host with FTP

If a license key file is supplied as an e-mail attachment, you must transfer the attached license key file `nat`*`vr`*`.xml` from the PC to the mainframe, using native FTP commands:

> **Warning:**
> **Using utilities instead of native FTP commands for the license key file transfer may corrupt the license key and thus prevent Natural from execution later on. This applies for example to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.**

To transfer a license key file from the PC to the mainframe, perform the following steps:

1. Save the product license key file e-mail attachment on your PC hard disk.

2. Open a command prompt window. In the command prompt window, change to the directory where you saved the product license key file.

3. Start an FTP session for communication with the z/VM host:

   ```
   ftp host-name
   ```

   Where *`host-name`* is the name of the z/VM host.

4. Enter your z/VM host login ID and password; for example: `<`*`cmsmachine`*`.by.`*`user`*`>`

5. Once the FTP session has been established, specify the CMS file system (SFS) or access a minidisk for the license key file:

   ```
   ftp>cd 'SFS'
   ```

6. Switch to binary data mode (the license key file must retain its ASCII format during the transfer):

   ```
   ftp>binary
   ```

7. Specify that the dataset for the license key file must be written with `RECFM FIX 80`.

   ```
   ftp>quote site fix 80
   ```

8. Write the license as a dataset on the z/VM system. For example, if the license key file name is nat*vrs*.xml, you might enter:

   ```
   ftp>put natvr.xml
   ```

   This command will create a dataset called `NAT`*`vr`*`.LICS`.

9. Rename or copy the dataset as follows:

   ```
   NATLIC DATA A
   ```

   The license key information stored in the dataset will be in ASCII format.

10. Stop your FTP session by entering:

    ```
    ftp>quit
    ```

Installation Procedure for Natural under CMS

Step 1: Define the Natural Discontiguous Saved Segments to VM

- Consult your system programmer for the virtual address range where the Natural DCSS should be defined. The DCSS should not overlap with any other DCSS that you might be using during execution of a Natural session. In particular, if you plan to call 3-GL programs that use Language Environment services, or if you set `NATCMS` parameter `LE370=YES`, make sure that your Natural DCSS does not overlap with the DCSS named `SCEE` and `SCEEX`.

  Execute the `DEFSEG` class `E` command for the Natural nucleus and for the Natural buffer pool.

  Examples:

  ```
  DEFSEG NATvr 3000-3FFF SR
  DEFSEG NATBPvr 500-5FF SN
  SAVESEG NATBPvr
  ```

  Note:
  Segment `NATBP`*`vr`* is saved as an empty segment. Segment `NAT`*`vr`* will be saved later when the Natural nucleus has been loaded into it.

Step 2: Load the FNAT System File

- If you are installing into an existing Natural 4.1 `FNAT` file, skip this step.

  Load the empty Natural system file (dataset `NAT`*`vrs`*`.SYSF`) using the `ADALOD` utility.

  This file will contain all Natural objects supplied by Software AG. Its size depends on the number of products to be installed later. As a rule of thumb, 20 MB can be assumed for each major Software AG product.

  The following `ADALOD` parameters must not be altered:

  ```
  ISNREUSE=YES
  ```

  To avoid Natural errors NAT9988 and NAT7397 after reorganization of the `FNAT` system file using `ADAULD/ADALOD`, the parameter `USERISN` should be set to `YES`.

  The file number *`fnat`* of the `FNAT` system file can be chosen as described under Natural profile parameter `FNAT` in the Natural *Parameter Reference* documentation.

Please, also note the following when defining a system file:

On the Natural installation tape, an input file for the ADALOD utility is provided.

It is easiest to perform the ADALOD operation from disk. You are recommended to copy the file NAT*vrs*.SYSF from tape to disk and to make additional copies for loading the FUSER, FDIC and FSEC system files:

```
FILEDEF IN TAP1 SLn VOLIDnnnnnn (RECFM VB LRECL 9996 BLKSIZE 10000
FILEDEF OUT DISK NATvrSYS LODDATA A
MOVEFILE IN OUT
```

To make additional copies, issue the following commands, for example:

```
COPYFILE NATvrSYS LODDATA A NATvrUSE LODDATA A
COPYFILE NATvrSYS LODDATA A NATvrDIC LODDATA A
COPYFILE NATvrSYS LODDATA A NATvrSCR LODDATA A
```

Modify the member NAT*vr*SYS LODLIB to meet your requirements. In particular replace the question mark (?) specified for the FILE parameter with the file number chosen for your FNAT system file.

You can now invoke the ADALOD EXEC for loading the Natural system file:

```
ADALOD NATvrSYS
```

Step 3: Load the FUSER System File

- You have the following options:

  1. You can use an existing Version 4.1 FUSER file, then you can skip this step.
  2. You can use a new FUSER file for Version 4.2.
  3. You can use an existing Version 4.1 FUSER file to be shared by Versions 4.1 and 4.2.
  4. You can use an existing Version 4.1 FUSER file to be used by Version 4.2 only.

Use a New Version 4.2 FUSER System File - First-Time Installation

For the use of a new and empty FUSER system file for Natural Version 4.2, no additional system-file-related actions are necessary.

If you do *not* want to share the FUSER system file, proceed as follows:

Load the empty Natural user file contained in dataset NAT*vrs*.SYSF using the ADALOD utility.

In this file, all user-written Natural programs are stored.

The following ADALOD parameters *must not* be altered:

```
ISNREUSE=YES
```

The file number *fuser* of the FUSER system file can be chosen as described under Natural profile parameter FUSER in the Natural *Parameter Reference* documentation.

Reuse an Existing Version 4.1 FUSER System File - Migration from Natural Version 4.1

If you want to use the existing Natural Version 4.1 FUSER system file and you do not want to share the FUSER system file, skip this step.

Using a Version 4.1 FUSER File to be Shared by Natural Versions 4.1 and 4.2

If you use an existing Natural Version 4.1 FUSER system file to be shared by Natural Versions 4.1 and 4.2, you must upgrade your Natural Version 4.1 installation to Version 4.1.4.

Natural Version 4.1.4 Service Pack I003 or a subsequent Service Pack is required. Service Pack I003 and all subsequent Service Packs contain all the necessary Version 4.1 based solutions for Natural Version 4.2.

User Application Programming Interfaces USR* in Library SYSEXT

The USR* programs from the delivered library SYSEXT run in a special mode. As a result, the USR* programs do not need to set further steplibs to execute related objects for processing. This reduces the impact on the Natural buffer pool search logic and improves the performance significantly if user exits are used extensively within user written applications.

It is necessary that the user exits are cataloged with Natural Version 4.2. This implies that the user exits cannot be executed with Natural Version 4.1.

Use of USR* Programs

Usually, the access of USR* programs by an application requires that the user application programming interfaces be copied from library SYSEXT to either the application libraries on the FUSER system file or to library SYSTEM on the FUSER system file or to library SYSTEM on the FNAT system file, respectively, or any other library which is defined as steplib for the application. Library SYSEXT can also be used as steplib. Due to the fact that the delivered user application programming interfaces will always be cataloged with the latest Natural version, we recommend that the user application programming interfaces should reside on the FNAT system file. This will ensure that the right version is executed and will separate user written applications from Software AG modules.

If applications which call user application programming interfaces should run with both Natural Version 4.1 and Natural Version 4.2, it must be made sure that the user application programming interfaces delivered with the corresponding Natural version are used.

The following scenarios may be considered:

Using a Version 4.1 FUSER File for Natural Version 4.1 and 4.2

If the same FUSER system file shall be used in a Natural Version 4.1 and Version 4.2 environment in parallel the following steps are recommended:

- Remove all USR* modules you have copied from library SYSEXT into application libraries on your FUSER system file.
- In both environments, copy the used USR* modules from library SYSEXT to library SYSTEM on the corresponding FNAT system file.

  - Alternatively, the USR* modules can be moved to another system library on FNAT which then must be defined as steplib, or library SYSEXT can be used as steplib for the applications. Then automatically in both environments the right versions of the user application programming interfaces are executed.

Using a Version 4.1 FUSER File for Natural Version 4.2 only

If you want to use the existing Natural Version 4.1 FUSER system file and you do not want to share the FUSER system file, then it is still possible to replace all USR* modules you have copied from library SYSEXT into application libraries with the new USR* objects from the Version 4.2 library SYSEXT.

But the preferred way is to remove all user application programming interfaces on the FUSER system file and copy the used user application programming interfaces from library SYSEXT to library system of the FNAT system file or use a SYS library on FNAT as steplib.

Using a New FUSER File for Natural Version 4.2

If you want to port existing applications to a new FUSER system file, copy all application objects but no Software AG USR* objects to the new FUSER system file. Then proceed as described in the scenario above.

Migration

The FIND function of the Natural utility SYSMAIN can be used to search for all USR* modules stored in a specific library on the FUSER system file or across the whole system file. In addition, Predict cross reference data can be used to determine all referenced user application programming interfaces.

The file number *fuser* of the FUSER system file can be chosen as described under Natural profile parameter FUSER (in the Natural Parameter Reference documentation).

Modify the member NAT *vr*USE LODLIB to suit your requirements. In particular replace the question mark (?) specified for the FILE parameter with the file number chosen for your FUSER system file.

You can now invoke the ADALOD EXEC for loading the Natural system file:

```
ADALOD NATvrUSE
```

Step 4: Load the Scratch-Pad File

- The scratch-pad file (which is a Natural-internal system file) can be used exclusively by the new Natural version or it can be shared by different versions of Natural.

  If you do *not* want to use a scratch-pad file, skip this step.

  If you do want to use a scratch-pad file; that is, if you want to use read-only system files (profile parameter ROSY=ON), see also *Natural Scratch-Pad File* in the Natural *Operations* documentation, proceed as follows:

Load the empty scratch-pad file contained in dataset NAT*vrs*.SYSF, using the ADALOD utility as described below.

The following ADALOD parameter *must not* be altered:

```
ISNREUSE=YES
```

For the optional scratch-pad file inclusion, the following NATPARM parameters must be added or, if already present, updated with:

```
LFILE=(212,dbid,fnr)
ROSY=ON
```

Modify the member NAT*vr*SCR LODLIB to suit your requirements. In particular replace the question mark (?) specified for the FILE parameter with the file number chosen for your scratch-pad file.

You can now invoke the ADALOD EXEC for loading the Natural system file:

```
ADALOD NATvrSCR
```

Step 5: Load the FDIC System File

- Skip this step:
  - if you want to install Predict (in this case, use the corresponding installation step in the Predict *Installation* documentation), or
  - if you want to use an existing FDIC system file (an existing FDIC system file can be shared by Natural Versions 4.1 and 4.2), or
  - if you do not use your own FDIC system file.

Load the empty FDIC system file contained in dataset NAT*vrs*.SYSF using the ADALOD utility, as described below.

The following ADALOD parameters *must not* be altered:

```
ISNREUSE=YES
```

The file number *fdic* of the FDIC system file can be chosen as described under Natural profile parameter FDIC in the Natural *Parameter Reference* documentation.

Modify the member NAT*vr*DIC LODLIB to meet your requirements. In particular replace the question mark (?) specified for the FILE parameter with the file number chosen for your FDIC system file.

You can now invoke the ADALOD EXEC for loading the Natural system file:

```
ADALOD NATvrDIC
```

Step 6: Load the FSEC System File

- Skip this step,
  - if you do not use Natural Security, or
  - if you want to use an existing FSEC system file, or
  - if you do not want to use an own FSEC system file.

  If you use Natural Security, refer to *Installing Natural Security* in the Natural Installation documentation.

Step 7: Generate a TEXT file from the Software AG license key file

- The file NATLIC EXEC uses as input the NATLIC DATA A which you have already copied with MOVEFILE.

  Execute NATLIC EXEC to generate the TEXT file NATLIC TEXT A.

  This TEXT file must be included in the Natural nucleus, or loaded dynamically at Natural start-up by using the profile parameter RCA.

Step 8: Customize the Natural Parameter Module

- The file NATPARM ASSEMBLE contains sample settings of Natural profile parameters.

  Use XEDIT to set the Natural profile parameters FNAT, FUSER, FSEC and FDIC to the correct values for your installation.

  Note that these parameters are in effect for the INPL procedure in one of the following steps.

  You must define the Natural buffer pool using the parameter macro NTBPI.

  Example:

  If you defined a buffer pool shared segment named NATBP*vr*, code:

  ```
  NTBPI TYPE=NAT,NAME=NATBPvr
  NTBPI TYPE=NAT,SIZE=1024
  ```

  Natural will be instructed to load the shared segment NATBP*vr* and use it as its buffer pool. If the shared segment cannot be loaded, a buffer pool of 1024 KB is allocated from CMS free storage and the initialization error message NAT1074 is displayed.

  You can suppress the display of initialization error messages by coding:

  ```
  IMSG=OFF
  ```

  You may wish to include

  ```
  HCAM=CMS
  ```

to enable the *Hardcopy Function* (%H) as described in the Natural *Operations* documentation. See also *Hardcopy Output* in the Natural *Terminal Commands* documentation).

Set other profile parameters as required by your installation.

For a description of these parameters, refer to the Natural *Parameter Reference* documentation.

z/VM does not support CPU usage time measurement; as a result, the profile parameter MT reacts to elapsed time instead of CPU usage time. Therefore, set the profile parameter MT to MT=0.

Step 9: Customize the INCLUDE List for Linking of Natural

- The program NAT$LOAD EXEC is used in the following two steps to build Natural for CMS.

  You may wish to use XEDIT to customize the contents of variable LOADLIST, for example, to include your own routines that you defined as CSTATIC in NATPARM ASSEMBLE.

Step 10: Linking the Adabas Interface

- Before linking Natural, copy the Adabas interface from the Adabas text library.

  You can use the following CMS commands to perform this task:

  ```
  FILEDEF IN DISK ADAVvrs TXTLIB fm (MEMBER ADABAS
  FILEDEF OUT DISK ADABAS TEXT A
  MOVEFILE IN OUT
  ```

  where *ADAVvrs* stands for your current Adabas text library and *fm* for the filemode where you access this text library.

Step 11: Generate a Natural Module

- Issue the following CMS command:

  ```
  NATBLDM
  ```

  You are prompted for the name of the Natural module.

  Enter a valid CMS file name, for example, NAT*vrs*.

  NATBLDM checks NATCMS TEXT and NATPARM TEXT.

  If a TEXT file is not found or if it is older than the corresponding source file, NATBLDM asks you whether you want to assemble (reassemble) the source file. If so, you are given the opportunity to XEDIT the source first, in order to define installation-dependent parameters.

  Note that NATCMS TEXT does not exist during an initial installation and that an assembly is forced.

  Enter YES when asked whether you want to edit NATCMS.

  The source program NATCMS ASSEMBLE contains a call to the macro NTCMS which generates the Natural VM/CMS interface.

Set the parameters that are described in NATCMS ASSEMBLE as required to meet your requirements.

Parameters in Macro NTCMS

The following parameters can be specified in the macro NTCMS:

Parameter	Description
ATTNKEY=*key*	This parameter specifies the attention key that is used to interrupt a running Natural program. When the specified key is pressed, Natural error NAT1016 will be raised. Valid keys are PA1 to PA3 and PF1 to PF24. Specify NONE if no attention key is to interrupt a Natural program. If the ATTNKEY parameter is omitted, all attention keys will cause error message NAT1016 to be issued.
CEEXOPT=(*LE run-time options*)	Specifies the LE run-time options if Natural runs LE-enabled. This parameter takes effect only with LE370=YES. For a description of LE run-time options, please refer to IBM documentation *Language Environment Customization*. Do not specify the TRAP option other than TRAP=(ON,NOSPIE). The entire list of options must be enclosed in parentheses. If you plan to call subprograms that run in AMODE24, specify CEEXOPT=(ALL31=OFF,STACK=(,,BELOW)).

LE370=YES
LE370=NO
	With LE370=YES, Natural runs as an LE program. You can call external subprograms according to the IBM calling conventions.
	With LE370=NO, you can only call main programs of the Language Environment. A new LE enclave is created and terminated for each CALL to a 3GL LE-enabled program.
	LE370=NO is the default.

Parameter	Description
LOADSS=(*name1,name2,...*)	This parameter specifies the names of programs that reside in saved segments and are to be loaded dynamically during execution of the Natural CALL statement. If the name of a program differs from the name of the saved segment in which it is contained, specify the two names in parentheses. If, for example, you specify LOADSS=(ALPHA,(BETA,GAMMA)) and a Natural program executes CALL 'ALPHA', the saved segment named ALPHA is loaded and control is passed to it. When a Natural program executes CALL 'BETA', the saved segment named GAMMA is loaded and given control. Control is always passed to the start address of the saved segment. If an attempt to load a saved segment fails, normal dynamic load logic is executed, which means that a LOAD macro searches for TEXT files and for TXTLIB and LOADLIB members.
MAXSIZE=*nnn*	This parameter specifies the amount of CMS free storage (in KB) to be obtained for Natural; the default setting is 1024. A standard way of calculating the space needed is to add all buffer sizes (ESIZE, etc.) explicitly defined in your NATPARM parameter module, plus 40 KB. If you specify MAXSIZE=0, each Natural buffer is obtained individually through the CMSSTOR OBTAIN service. Storage use by Natural is then only limited by the amount of free storage available in your virtual machine. See also *Using the MAXSIZE Parameter* in the Natural *Operations* documentation.
OPID=*name*	This parameter specifies the name of the CMS machine to which the specified operator message is sent after a CALL 'CMWTO' request has been issued from a Natural program. An asterisk (*) as value causes the message to be sent to the current CMS machine; specifying OP directs the message to the VM system operator. The default value is OP.

NATBLDM also generates a load map whose file name is the same as the file name of the module, with a file type of MAP. Keep this map, because it may be helpful in locating errors.

Step 12: Generate a Natural DCSS

- Issue the following CMS command:

  ```
  NATBLDS
  ```

  You are prompted for the name of the Natural DCSS (discontiguous shared segment) that you defined in the corresponding installation step.

  Like `NATBLDM` in the corresponding step, `NATBLDS` checks `NATCMS TEXT` and `NATPARM TEXT`, and if necessary, allows you to edit and reassemble the respective source before building the saved segment.

  `NATBLDS` also generates a load map whose file name is the same as the name of the DCSS, with a file type of `MAP`. Keep this map, because it may be helpful in locating errors.

  Natural then offers to build the "bootstrap" for the DCSS which will load and then pass control to the saved segment. If you enter `YES`, Natural will ask you for the name you want to use to invoke Natural.

  You can also build the bootstrap yourself using the following CMS command:

  ```
  XEDIT NATBOOT ASSEMBLE
  ```

  Set the parameter NSS to the name of the Natural DCSS and file it under the name you want to use, for example, `NAT`*`vr`*.

  Issue the following CMS commands:

  ```
  GLOBAL MACLIB NATvrs DMSGPI
  ASSEMBLE NATvr
  LOAD NATvr (ORIGIN TRANS
  GENMOD NATvr
  ```

Step 13: Delete Natural System Objects

- This step is optional but recommended to avoid data inconsistencies.

  If you are using a Version 4.1 Natural `FNAT` system file, you can delete obsolete Version 4.1 Natural objects.

  If only base Natural Version 4.1 is installed

  Delete the Version 4.1 Natural objects by loading the `NAT`*`vrs`*`.LDEL` data set with the Natural `INPL` utility.

  (Job I061, Step 0010)

  If Natural Security Version 4.1 is installed

  Delete the Version 4.1 Natural Security objects by loading the `NSC`*`vrs`*`.LDEL` data set with the Natural `INPL` utility.

(Job I061, Step 0099)

See also *Delete Natural Security Objects* in *Installing Natural Security*.

Step 14: Load the Natural System File and Error Messages

- If the tape drive used when copying the contents of the installation tape to disk was detached from your virtual machine, ask the system operator to attach a tape drive to your virtual machine at address `X'181'` and mount the Natural installation tape.

  Before you can start the Natural `INPL`, the Adabas environment for your CMS machine must have been set up (as described in the Adabas documentation).

  Issue the following CMS command:

  ```
  NATINPL
  ```

  You are prompted for the name of the command to invoke Natural.

  Enter one of the following:

  - the name of the Natural module built in the corresponding step, or
  - the name of the Natural DCSS bootstrap created in the corresponding step.

  `NATINPL` then positions the tape and loads the Natural system file and error messages. In addition to the English error message short and long texts, error message short texts in German language (`ULANG=2`) are loaded.

  Use the `ERRUPPER` program to convert the error message texts to upper case. This program is provided by the Natural SYSERR utility described in the *Utilities* documentation.

  This is the most time-consuming step of the installation procedure. The elapsed time varies greatly with machine type and system load and can in some cases exceed an hour.

  The Natural system file definitions as coded in the Natural parameter module (which was created in a previous step) apply.

  Three reports are produced on your virtual printer.

  Check these reports to ensure that no errors have occurred.

Load the Japanese Error Messages

This step is only required if you wish to load the error message short texts in Japanese language (`ULANG=59`) or if you wish to replace the English error message long texts with their Japanese equivalents.

The Natural error message short and long texts in Japanese language are both contained in the Natural Japanese Language Pack. This is a separate product (product code NCJ) that can be loaded optionally. If you do not load the Natural error message long texts in Japanese language, English error message long texts will appear. If you do not load the Natural error message short texts in Japanese language, English error message short texts will appear.

Issue the following CMS command:

```
NCJINPL
```

You are prompted for the name of the command to invoke Natural.

Enter one of the following:

- the name of the Natural module built in the corresponding step, or
- the name of the Natural DCSS bootstrap created in the corresponding step.

You are also prompted whether you want to load error message short texts, error message long texts, or both.

`NCJINPL` positions the tape and loads the requested error message texts in Japanese language.

The Natural system file definitions as coded in the Natural parameter module (which was created in a previous step) apply.

After loading each type of error message text, a report is produced on your virtual printer.

Check these reports to ensure that no errors have occurred.

Step 15: Make Natural Modules Available to the Users

- Copy the Natural module and the DCSS bootstrap to a public disk or have your system programmer copy them to the CMS Y-disk.

Installation Verification for Natural under CMS

To verify your installation, perform the following steps:

1. Issue as CMS command the name of the Natural module created in a previous step.
2. Check that the following results are available:
   - Communication between Adabas and Natural is working.
   - The Natural system files have been loaded.
   - Batch Natural is operational.