Natural Construct

Installation Guide for Mainframes

Manual Order Number: CST451-010IBM

This document applies to Natural Construct Version 4.5 and to all subsequent releases.

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

Readers' comments are welcomed. Comments may be addressed to the Documentation Department at the address on the back cover or to the following e-mail address:

Documentation@softwareag.com

TABLE OF CONTENTS

PREFACE

This preface explains the structure of the *Natural Construct Installation Guide for Mainframes*, as well as the document conventions used in the guide.

Note: Before installing Natural Construct, refer to the release notes for the latest information about the installation procedure, system requirements, and new features.

The following topics are covered:

Structure of this Guide

This guide describes the installation of Natural Construct V4.5 on mainframe platforms. It is intended for administrators who are responsible for installing Natural Construct.

This guide covers the following topics:

Chapter	Title	Description
1	**Before You Install**, page 11	This chapter describes the installation tape for Natural Construct. It includes a description of the installation tape, information about System Maintenance Aid (SMA), and sample JCL you can use to copy the datasets from tape to disk.
2	**Installing Natural Construct**, page 27	This chapter describes how to install Natural Construct V4.5 as a new or current user, how to install a catalog and runtime environment, and how to install a runtime-only environment. It also describes how to install the processor source, define libraries to Natural Security, activate Natural Construct, install in uppercase, and install in one language only (static mode).
3	**Verifying the Installation**, page 65	This chapter describes several procedures you can use to verify the installation of Natural Construct.
4	**Maintaining Natural Construct**, page 69	This chapter describes the requirements to maintain and use Natural Construct. It includes information about accessing the data file and model maintenance facilities, as well as the user exit subprograms and the generation and runtime environments. It also lists the modules supplied with Natural Construct and provides a summary of the installation datasets.

Document Conventions

Within this guide, the following conventions apply:

Term	Description
Enter	Type a value in a field and press the Enter key.
Field	In general, any area on a screen where users can type information, select a value from a pop-up window, or indicate a preference by marking a box or circle.
Invoke	Activate or execute a program or menu.
Mark	Type a non-blank character in an input field (for example, an X) to select the field.
Panel	A full screen of information displayed by a program, etc.
Select	One of the following actions: • Move the cursor to a value and press the Enter key. • Scroll through a selection box and highlight a value. • Double-click on a value. • Type the name of a value in a key field and press the Enter key.
Specify	One of the following actions: • Type a value in a field. • Select a value from a selection window.
Window	A partial screen of information that overlays the current panel.

BEFORE YOU INSTALL

This chapter contains information you should read before installing Natural Construct. It lists the prerequisites and operating environments, describes the installation tape, and provides sample JCL you can use to copy the datasets from tape to disk. It includes a section on using System Maintenance Aid (SMA) in a mainframe environment.

The following topics are covered:

Prerequisites

Natural Construct V4.5.1 is compatible with the following Software AG products and versions:

- Natural V3.1.6
- Construct Spectrum V4.5
- Predict V4.2 and V4.3

Note: Predict V4.3 is compatible with Natural Construct V4.4 and V4.5.

- Adabas V7.1 or higher

Note: To read files in descending sequence, Natural must compile with Adabas V6.2 or higher files. Natural V3.1.6 defaults to Adabas V7.

The Natural Construct installation tape includes sample JCL for the installation (see the CST*nnn*.JOBS dataset).

This guide assumes you are installing Natural Construct for the first time or upgrading from Natural Construct V4.4.1. If you are upgrading from any other version, read the release notes and installation guides for all versions between your current version and V4.4.1 and perform the required procedures.

As some components of Natural Construct require access to the Software AG editor, ensure that the editor is fully installed and operational. To use the online job submission features, you must also ensure that the NATRJE module is fully installed and operational. For information about the editor or NATRJE module, see your Natural Installation and Operations documentation.

Operating Environments

This section describes the operating environments for Natural Construct. The information in this section includes:

- List of the operating systems on which you can run Natural Construct
- List of the databases that can be accessed by Natural Construct-generated modules
- List of the Natural systems that can be used with Natural Construct
- Information about security for Natural Construct

Operating Systems

Natural Construct V4.5 requires Predict 4.2 or higher. It functions in any environment that supports Natural V3.1.6 or higher, including the following operating systems:

- OS/390
- VSE/ESA
- VM/CMS
- BS2000/OSD
- MSP
- MSP/EX
- AS/400 WANG

TP Monitors

Like Natural, Natural Construct runs under the following teleprocessing (TP) monitors:

- Com-plete
- IMS/DC
- CICS
- TSO

Data Access

Natural Construct generates applications that access the following data structures:

- Adabas
- DB2
- SQL/DS
- VSAM
- DL/1

You can also create your own models to access other data structures.

Natural Software Systems

Natural Construct operates in conjunction with any Natural software system, including:

- Natural Security
- Natural ISPF
- Predict V4.2 and V4.3
- Entire Connection
- Natural Process
- Construct Spectrum V4.5

Security

Access to Natural Construct is protected by Natural Security facilities. Natural Construct-generated modules operate within, and are controlled by, the Natural Security system.

Dataset-Naming Conventions

Natural Construct installation tapes use a dataset-naming convention that identifies the product, version number, release number, system maintenance level, and dataset type. For example, the Natural Construct installation tape includes the following dataset:

`CST`*`nnn`*`.INPL`

This dataset name consists of the following components:

Component	Description
CST	Product code
n	Version number
n	Release number
n	System maintenance level
INPL	Dataset type

Description of the Installation Tape

The Natural Construct installation tape contains the datasets listed below. The *Report of Tape Creation* lists the volume serial number, density, media type, dataset names, and dataset sequence numbers.

The following sections describe the datasets for all users, as well as the datasets for VSE/ESA users and those for OS/390 and BS2000/OSD users.

All Users

The installation tape contains the following datasets for all users, where *nnn* indicates the version number, release number, and system maintenance (SM) level of Natural Construct. The common datasets for all operating systems are:

Dataset Name	Description
CST*nnn*.INPL	INPL dataset containing all the modules required to generate, catalog, and run Natural Construct-generated modules.
CST*nnn*.IUPD	IUPD dataset containing updates and fixes to the main INPL dataset. It is only included if there are updates or fixes.
Note:	If this dataset is included in the installation package, it should be loaded after the CST*nnn*.INPL dataset is loaded.
CST*nnn*.INPC	INPL dataset containing the modules required to catalog and run generated modules. This dataset is a subset of CST*nnn*.INPL.
CST*nnn*.INPE	INPL dataset containing the modules required to run (but not catalog) generated modules. This dataset is a subset of CST*nnn*.INPC.
CST*nnn*.SYSF	Dataset used to unload the Natural Construct SYSF system file. Use this data file as input to the Adabas load utility (ADALOD).
Note:	This file only contains header information and some test data.
CST*nnn*.SYSH	Dataset containing help text. This dataset is a subset of CST*nnn*.SYSF.
CST*nnn*.SYSM	Dataset containing program models. This dataset is a subset of CST*nnn*.SYSF.
CST*nnn*.SYSR	Dataset containing code frames. This dataset is a subset of CST*nnn*.SYSF.

Dataset Name	Description (continued)
CST*nnn*.ERRN	SYSERR dataset containing messages used by Natural Construct-generated applications.
CST*nnn*.DA4	Dataset containing Predict views used in the demo application.
CST*nnn*.VINP	INPL dataset containing the Natural Construct demo programs to access VSAM data files.
CST*nnn*.VD4	Dataset containing Predict definitions for the VSAM demo application.
CST*nnn*.DINP	INPL dataset containing the Natural Construct demo programs to access DB2 data files.
CST*nnn*.DD4	Dataset containing Predict definitions for the DB2 demo application.
CST*nnn*.CP42	Dataset containing all Predict interface modules for users running Natural Construct V4.5 with Predict V4.2.
CST*nnn*.C441	Dataset containing the code frame listings used to compare the code frames in Natural Construct V4.5 with those in V4.4.
CST*nnn*.NCPD	Dataset containing the source for the command processor used in the demo application.

OS/390 and BS2000/OSD Users

For OS/390 and BS2000/OSD users, the installation tape also contains the following dataset:

Dataset Name	Description
CST*nnn*.JOBS	Dataset containing sample OS JCL for all functions required to install Natural Construct.

VSE/ESA Users

For VSE/ESA users, the installation tape also contains the following dataset:

Dataset Name	Description
CST*nnn*.LIBJ	Dataset containing sample JCL to install Natural Construct.

Installation Jobs

To install Software AG's products, use the installation jobs. You can either create these jobs manually using JCL or use System Maintenance Aid (SMA).

If you are not using SMA, the JCL for installing is included in **Copying the Datasets to Disk**, page 21. You must adapt this sample job to your requirements.

Using System Maintenance Aid

Note: System Maintenance Aid is not available for the CMS operating system.

If you are using SMA, perform the following steps before generating jobs:

1. Load the SMA table data as described in the System Maintenance Aid documentation (if you have not already done so).
2. Set CST45*n* in the list of available products for your environment to TO BE INSTALLED.
3. If you are performing a first-time installation, adapt the FCST and FCST-DBID parameters in the FILNUM parameter group to the required values.

   To use SMA, set the following options in your SMA environment (specified on the Modify Environment panel). Verify all options before generating jobs, as they may be set from a previous install.

SMA Name	Description
CST-FIRST-INSTALL	To load a new Natural Construct system file, set this option to "Y". To migrate your existing system file to the current version, set this option to "N". The new help text, code frames, and models will be loaded.
CST-FULL-ENV	To establish an environment in which you can generate, catalog, and run generated modules, set this option to "Y". This will load the CST*nnn*.INPL dataset.
Note:	The CST*nnn*.INPL dataset contains the Predict modules and will automatically install theses modules in your environment.
CST-CAT-ENV	To establish an environment in which you can catalog and run (but not generate) generated modules, set this option to "Y". This will load the CST*nnn*.INPC dataset (subset of CST*nnn*.INPL).

SMA Name	Description (continued)
CST-EXE-ENV	To establish an environment in which you can run (but not generate or catalog) generated modules, set this option to "Y". This will load the CST*nnn*.INPE dataset (subset of CST*nnn*.INPC).
CST-LOAD-PRD	To load the Predict definitions for the demo system and system file, set this option to "Y".
CST-LOAD-DB2	To install the demo system to access DB2 tables, set this option to "Y". (You must have Natural for DB2 installed.)
CST-LOAD-VSAM	To install the demo system to access VSAM files, set this option to "Y". (You must have Natural for VSAM installed.)
FCST	File number of the Natural Construct system file.
ESIZE-NAT (BS2000/OSD only)	ESIZE for batch environments (minimum is 220).
ESIZE-NRT (BS2000/OSD only)	ESIZE for TIAM environments (minimum is 220).
ESIZE-NUT (BS2000/OSD only)	ESIZE for UTM environments (minimum is 220).
FCST-DBID (OS/390 only)	Database ID of the Natural Construct system file.
ESIZE-BATCH (OS/390 only)	ESIZE for batch environments (minimum is 120).
ESIZE-ONLINE (OS/390 only)	ESIZE for online environments (minimum is 220).
ESIZE (VSE/ESA only)	ESIZE for VSE/ESA environments (minimum is 220).
DATSIZE-BATCH	DATSIZE for batch environments (minimum is 180).
DATSIZE-ONLINE	DATSIZE for online environments (minimum is 180).
SYNERR	SYNERR option, which must be set to "ON".

Copying the Datasets to Disk

You may want to copy the datasets on the installation tape to disk. The disk storage required for each dataset is given in the *Report of Tape Creation*, which accompanies the installation tape.

The procedure to copy the datasets to disk differs, depending on whether you are using System Maintenance Aid (SMA). If you are using SMA, refer to the System Maintenance Aid (SMA) documentation (included on the current edition of the Natural documentation CD).

If you are not using SMA, see the following sections.

OS/390

Note: If the datasets for more than one product are on the tape, the COPY.JOB dataset contains the JCL to unload the datasets for all products. You must then perform the individual installation procedure for each component.

Step 1: Copy the COPY.JOB Dataset from Tape to Disk

The COPY.JOB dataset (label 2) contains the JCL to unload all other datasets from the installation tape to disk. To unload COPY.JOB, use the following sample JCL:

```
//SAGTAPE JOB SAG,CLASS=1,MSGCLASS=X
//* --------------------------------
//COPY EXEC PGM=IEBGENER
//SYSUT1 DD DSN=COPY.JOB,
// DISP=(OLD,PASS),
// UNIT=(CASS,,DEFER),
// VOL=(,RETAIN,SER=<Tnnnnn>),
// LABEL=(2,SL)
//SYSUT2 DD DSN=<hilev>.COPY.JOB,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,VOL=SER=<vvvvvv>,
// SPACE=(TRK,(1,1),RLSE),
// DCB=*.SYSUT1
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//
```

where:

`<hilev>`	is a valid high level qualifier
`<Tnnnnn>`	is the tape number
`<vvvvvv>`	is the desired volser

Step 2: Modify COPY.JOB to Conform to Your Naming Conventions

Before submitting this job, set the following parameters:

- Set HILEV to a valid high-level qualifier.
- Set LOCATION to a storage location.
- Set EXPDT to a valid expiration date.

Step 3: Submit the Job

To unload all other datasets from the tape to your disk, submit COPY.JOB.

BS2000/OSD

To copy the tape to disk, perform the following steps.

Step 1: Copy the SRVnnn.LIB Library from Tape to Disk

Note: This step is not required if you have already copied the SRVnnn.LIB library from another Software AG tape. For more information, see the #READ-ME file in this library.

The SRVnnn.LIB library is stored on the tape as the SRVnnn.LIBS sequential file containing LMS commands. The current version nnn can be obtained from the *Report of Tape Creation*.

To convert the SRVnnn.LIBS sequential file into an LMS-library, execute the following commands:

```
/IMPORT-FILE  SUPPORT=*TAPE(FILE-NAME=SRVnnn.LIBS, -
/   VOLUME=<volser>, DEV-TYPE=<tape-device>)
/ADD-FILE-LINK  LINK-NAME=EDTSAM, FILE-NAME=SRVnnn.LIBS, -
/   SUPPORT=*TAPE(FILE-SEQ=3), ACC-METH=*BY-CAT, -
/   BUF-LEN=*BY-CAT, REC-FORM=*BY-CAT, REC-SIZE=*BY-CAT
/START-EDT
@READ  '/'
@SYSTEM 'REMOVE-FILE-LINK  EDTSAM'
@SYSTEM 'EXPORT-FILE  FILE-NAME=SRVnnn.LIBS'
@WRITE  'SRVnnn.LIBS'
@HALT
/ASS-SYSDTA  SRVnnn.LIBS
/MOD-JOB-SW  ON=1
/START-PROG  $LMS
/MOD-JOB-SW  OFF=1
/ASS-SYSDTA  *PRIMARY
```

where:

`<tape-device>`	is the device-type of the tape (for example, TAPE-C4)
`<volser>`	is the VOLSER of the tape (see the *Report of Tape Creation*)

Step 2: Copy the COPY.PROC Procedure from Tape to Disk

To copy the COPY.PROC procedure to disk, call the P.COPYTAPE procedure in the SRVnnn.LIB library:

```
/CALL-PROCEDURE  (SRVnnn.LIB,P.COPYTAPE), -
/   (VSNT=<volser>, DEVT=<tape-device>)
```

Note: If you use a TAPE-C4 device, you can omit the DEVT parameter.

Step 3: Copy all Product Files from Tape to Disk

To copy all Software AG product files from tape to disk, enter the COPY.PROC procedure:

```
/ENTER-PROCEDURE  COPY.PROC, DEVT=<tape-device>
```

Note: If you use a TAPE-C4 device, you can omit the DEVT parameter. The result from this procedure is written to the 'L.REPORT.SRV' file.

VSE/ESA

Note: If the datasets for more than one product are on the tape, the COPYTAPE.JOB member contains the JCL to unload the datasets for all delivered products (except datasets you can install directly from tape, such as INPL objects). You must then perform the individual installation procedure for each component.

Step 1: Copy the COPYTAPE.JOB Dataset from Tape to Disk

The COPYTAPE.JOB dataset (file 5) contains the JCL to unload all other datasets from tape to disk. To unload COPYTAPE.JOB, use the following sample JCL:

```
* $$ JOB JNM=LIBRCAT,CLASS=0,                                          +
* $$ DISP=D,LDEST=(*,UID),SYSID=1
* $$ LST CLASS=A,DISP=D
// JOB LIBRCAT
* ****************************************
*     CATALOG COPYTAPE.JOB TO LIBRARY
* ****************************************
// ASSGN SYS004,NNN                                <------  tape address
// MTC REW,SYS004
// MTC FSF,SYS004,4
ASSGN SYSIPT,SYS004
// TLBL IJSYSIN,'COPYTAPE.JOB'
// EXEC LIBR,PARM='MSHP; ACC S=lib.sublib'         <------- for catalog
/*
// MTC REW,SYS004
ASSGN SYSIPT,FEC
/*
/&
* $$ EOJ
```

where:

`NNN`	is the tape address
`lib.sublib`	is the library and sublibrary of the catalog

Step 2: Modify COPYTAPE.JOB to Conform to Your Naming Conventions

Before submitting this job, modify COPYTAPE.JOB to conform to your local naming conventions and set the disk space parameters.

Step 3: Submit COPYTAPE.JOB

To unload all other datasets from the tape to your disk, submit COPYTAPE.JOB.

VM/CMS

To copy the tape to disk, perform the following steps.

Step 1: Position the Tape for the TAPE LOAD Command

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

If the sequence number of CST*nnn*.TAPE, as shown in the *Report of Tape Creation*, is n, you must position over 3n - 2 tape marks (i.e., FSF 1 for the first dataset, FSF 4 for the second, etc.).

Step 2: Access the Disk

Access the disk that is to contain the installation files as disk A. The disk size must be at least 1500 4-KB blocks (for example, 10 cylinders on 3380-type disks or 12000 blocks FB-512).

Step 3: Attach a Tape Drive to Your Virtual Machine

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

Step 4: Position the Tape for the CMS Command

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

Step 5: Load Natural Construct under CMS

Load Natural Construct under CMS installation material by issuing the CMS command:

```
TAPE LOAD * * A
```

Step 6: Keep the Tape Drive Attached to Your Virtual Machine

Keep the tape drive attached to your virtual machine, as the tape is required during the installation procedure.

INSTALLING NATURAL CONSTRUCT

This chapter describes how to install Natural Construct V4.5 as a new or current user, how to install a catalog and runtime environment, and how to install a runtime-only environment. It also describes how to install the processor source, define libraries to Natural Security, and activate Natural Construct. It includes the procedures to install in uppercase and install in one language only (static mode).

Note: To use the Construct Windows interface, Construct Spectrum must be installed. For information, see *Construct Spectrum and SDK Installation Guide for Mainframes*.

The following topics are covered:

Installing Natural Construct Over an Existing Version

This section describes how to upgrade to Natural Construct V4.5 from V4.4. It is intended for users who want to generate, catalog, and run the generated modules. The CST*nnn*.INPL dataset installs a complete administration, development, generation, and execution environment.

- To establish an environment to catalog and run generated applications, install Natural Construct using the CST*nnn*.INPC dataset (subset of CST*nnn*.INPL).
- To establish an environment to run (but not catalog) generated applications, install Natural Construct using the CST*nnn*.INPE dataset (subset of CST*nnn*.INPC).

For information on installing the CST*nnn*.INPC or CST*nnn*.INPE dataset, see **Installing Catalog and Runtime Environments**, page 55.

Version 4.3 Users

Before upgrading, make the following changes to the Natural Construct system file if you currently have Natural Construct V4.3 installed:

1. Extend the ED field from 7 to 8 bytes.
2. Extend the EG field from 1 to 8 bytes.
3. Extend the EH field from 1 to 8 bytes.
4. Extend the FE field from 1 to 8 bytes.

After saving the changes, refer to **Upgrading Natural Construct**, page 29.

Version 4.4 Users

Before upgrading, make the following changes to the Natural Construct system file if you currently have Natural Construct V4.4 installed:

1. Release the S8 descriptor.
2. Release the S9 descriptor.

After saving the changes, refer to **Upgrading Natural Construct**, page 29.

Upgrading Natural Construct

➢ To upgrade to Natural Construct V4.5 from V4.3 or V4.4, perform the following steps:

- **Step 1: Create Backup of SYSCST**, page 30
- **Step 2: Define Natural Construct NATPARM**, page 30
- **Step 3: Link Natural Nucleus (batch) with New NATPARM**, page 31
- **Step 4: Load Updated Modules and New Error Messages**, page 32
- **Step 5: (Optional) Install DB2 or VSAM Demo Application**, page 34
- **Step 6: Load Updates and Fixes for Main INPL Dataset**, page 35
- **Step 7: Link Natural Nucleus (online) with New NATPARM**, page 36
- **Step 8: (Optional) Load Command Processor and Demo Application Definitions**, page 36
- **Step 9: Copy Natural Utility Subprograms**, page 38
- **Step 10: Load Updated Help Text**, page 39
- **Step 11: Load Updated Models**, page 39
- **Step 12: Load Updated Code Frames**, page 40
- **Step 13: Set Natural Construct System File Version**, page 40
- **Step 14: (Optional) Install in Uppercase Only**, page 41
- **Step 15: (Optional) Translate Error Messages**, page 41
- **Step 16: (Optional) Install in Static (One-Language) Mode**, page 41

The remainder of this section describes these steps.

Note: The CST*nnn*.JOBS dataset contains sample JCL.

Step 1: Create Backup of SYSCST

- Job I051, Step 1300

  Before upgrading, use the SYSMAIN utility to move all modules in the SYSCST library to a temporary library. This ensures that no obsolete modules remain in your SYSCST library, and any modules you created or modified using other versions of Natural Construct are not overwritten. You can use this backup copy to restore your custom modules.

  Because of the large number of modules in the SYSCST library, we recommend that you create a new application, called CSTTEMP, in Natural Security and move these modules in batch mode. Ensure that CSTTEMP is defined to Natural Security before running the following job:

```
LOGON SYSMAIN
MENU
M,ALL,*,FM,SYSCST,DBID,xxx,FNR,yyy,TO,CSTTEMP,DBID,aaa,%
FNR,bbb,REP
FIN
```

  where:

- `xxx` indicates the DBID value for your FNAT system file
- `yyy` indicates the FNR value for your FNAT system file
- `CSTTEMP` indicates the name of the temporary library
- `aaa` indicates the DBID for your FUSER system file
- `bbb` indicates the FNR for your FUSER system file

  Ensure that the IM=D parameter is set in your NATPARM.

Step 2: Define Natural Construct NATPARM

Natural Construct uses a special data file that must be identified to the Natural transaction used for Natural Construct.

Note: When using Natural Construct, ensure that BPSFI=OFF. If BPSFI=OFF, Search Sequence 1 is used (alternating search in buffer pool and database for each library). If BPSFI=ON, Search Sequence 2 is used (search in buffer pool first for all libraries and then search database).

To identify this file, use one of the following methods:

- Use the LFILE parameter to dynamically identify the file

  or

- Code the NTFILE macro in the Natural parameter module to make the file part of the Natural nucleus

If you identify the Natural Construct data file using the LFILE dynamic parameter method, use the LFILE=(227,DBID,FNR) dynamic parameter when invoking Natural. The DBID and FNR values indicate the database ID and file number for the Natural Construct data file.

If you identify the Natural Construct data file using the NTFILE macro in the Natural parameter module, the following example describes its use:

```
ACMPARM TITLE 'Natural Parameters for Natural Construct'
         NTPRM FNAT=(10,5),              System libraries, errors
               FUSER=(10,6),              User programs
               FDIC=(10,7),              Predict data, DDMs
               FSEC=(10,8),              Natural Security file
               LS=132,
               PS=60,
               WORK=7                     Recommended for download
               ESIZE=220,                 Minimum recommended ESIZE
               DATSIZE=180,              Minimum recommended DATSIZE
               SSIZE=70                  Minimum recommended SSIZE
               SYNERR=ON,
         NTFILE DBID=nnn,FNR=nnn,ID=227  Natural Construct data file
         END
```

The DBID value is the database number used when loading the Natural Construct data file; the FNR value is the file number used when loading the file. The ID value must be 227.

The NTFILE macro allows you to load the Natural Construct data file into any file in any database. It translates the logical file number used in the Natural Construct modules (227) into the database ID and file number for the data file.

To assemble and link the parameter module, use the same procedures as you used to assemble and link the standard Natural V3 parameter module.

Step 3: Link Natural Nucleus (batch) with New NATPARM

Using the Natural link job (I060) that linked the standard Natural nucleus, create an executable Natural nucleus. Include the NATPARM parameter module assembled and linked in Step 2.

Note: Some components of Natural Construct require access to the Software AG editor. Ensure that the editor is installed and operational.

Step 4: Load Updated Modules and New Error Messages

To load the updated modules and messages:

A Load the updated modules for the demo, SYSTEM, SYSCST, SYSCST00, SYSCSTX, and SYSLIBS libraries.

B Load the updated Predict interface modules.

C Load the new error messages.

To load the updated modules, use the Natural INPL utility to load the applicable dataset and set work file 1 to the dataset name. For example:

```
INPL
B
```

These procedures are described in the following sections.

4A: Load Updated Natural Construct Modules

- Job I061, Step 1300; CST*nnn*.INPL dataset

  Load the modules from the CST*nnn*.INPL dataset into the appropriate system files. This dataset contains modules that update modules in the demo, SYSTEM, SYSCST, SYSCST00, SYSCSTX, and SYSLIBS libraries. Ensure that the FNAT, FUSER, and FDIC parameter values correctly identify the desired system files.

  The INPL utility loads all the data areas, copycode, and external modules used by the generated applications into the FNAT SYSTEM library. Ensure that you load the modules and the DDMs.

4B: Load Updated Predict Interface Modules

- Job I061, Step 1304; CST*nnn*.CP42

  Load the modules from the CST*nnn*.CP42 dataset into the appropriate system files. This dataset contains the updated Predict V4.2 interface modules.

4C: Load New Error Messages

- Job I061, Step 1305; CST*nnn*.ERRN dataset

  The CST*nnn*.ERRN dataset contains the new error messages for Natural Construct-generated applications. This dataset was created using the SYSERR utility, ERRULDUS. All messages are loaded into SYSERR.

  Note: For information about SYSERR, see the Natural documentation.

  – Messages 8000 to 8200 are loaded into the SYSTEM and SYSCST libraries
  – Messages 8300 to 8500 are loaded into the CSTAPPL library
  – Messages 1 to 9999 (error message text) are loaded into the CSTMSG library
  – Messages 1 to 9999 (screen prompt text) are loaded into the CSTLDA library
  – Messages 1 to 9999 (text for Actions) are loaded into the CSTACT library
  – Messages 1 to 9999 (text for PF-keys) are loaded into the CSTPFK library

  To load these messages into your FUSER system file, invoke Natural in batch mode with work file 2 assigned to the CST*nnn*.ERRN dataset. Use the following statements to load the messages into SYSERR:

```
LOGON SYSTEM
ERRLODUS
FIN
```

  Note: If the SYSTEM library is not defined as the highest-level steplib under Natural Security, use the SYSMAIN utility to copy the messages from the SYSTEM library to your alternate steplib.

Step 5: (Optional) Install DB2 or VSAM Demo Application

If you are installing in an environment that accesses DB2 or VSAM, you can install the corresponding demo application.

Note: Use the Predict definitions to generate the required physical files. For information, see the Predict documentation.

Install the DB2 Demo Application

- Job I061, Step 1308; CST*nnn*.DINP

  The CST*nnn*.DINP dataset contains the demo application to access DB2 tables. The DB2 demo application accesses the following files:

  – NCSTDB2-CUSTOMER
  – NCSTDB2-ORDER_DISTRIBUTION
  – NCSTDB2-ORDER_HEADER
  – NCSTDB2-ORDER_INSTRUCTIONS
  – NCSTDB2-ORDER_LINES
  – NCSTDB2-PRODUCT
  – NCSTDB2-WAREHOUSE

➢ To load the demo application for DB2:

1 Use the Natural INPL utility and set work file 1 to CST*nnn*.DINP.

2 Enter the following:

```
INPL
B
```

➢ To copy the data used by the DB2 demo application:

1 Log onto the NCSTDEM2 library.

2 Catalog and run the CSGMIGA2 program.

Install the VSAM Demo Application

- Job I061, Step 1307; CST*nnn*.VINP

  The CST*nnn*.VINP dataset contains the demo application to access VSAM data files. The VSAM demo application accesses the following files:

  – NCSTVSAM-CUSTOMER
  – NCSTVSAM-ORDER-DISTRIBUTION
  – NCSTVSAM-ORDER-HEADER
  – NCSTVSAM-ORDER-LINES
  – NCSTVSAM-PRODUCT
  – NCSTVSAM-WAREHOUSE

➢ To load the demo application for VSAM:

1 Use the Natural INPL utility and set work file 1 to CST*nnn*.VINP.

2 Enter the following:

```
INPL
B
```

➢ To copy the data used by the VSAM demo application:

1 Log onto the NCSTDEMV library.

2 Catalog and run the CSGMIGAV program.

Step 6: Load Updates and Fixes for Main INPL Dataset

- Job I061, Step 1309; CST*nnn*.IUPD dataset

  This dataset contains updates and fixes to the main INPL dataset (CST*nnn*.INPL); it is only included in the installation package if there are updates or fixes to the INPL dataset.

➢ To load the updates and fixes:

1 Invoke the Natural INPL utility.

2 Set work file 1 to the applicable dataset name (CST*nnn*.IUPD)

3 Enter the following:

```
INPL
B
```

Step 7: Link Natural Nucleus (online) with New NATPARM

Using the Natural link job (I080) that linked the standard Natural nucleus, create an executable Natural nucleus. Include the NATPARM parameter module assembled and linked in **Step 2: Define Natural Construct NATPARM**, page 30.

Note: As some components of Natural Construct require access to the Software AG editor, ensure that the editor is installed and operational.

Note: To use the online job submission features of Natural Construct, ensure that the NATRJE module is installed and operational.

Step 8: (Optional) Load Command Processor and Demo Application Definitions

Optionally, you can load the command processor and updated Predict file definitions for the demo application.

➢ To load the command processor and updated file definitions:

A Load the command processor source.

B Load the updated Predict file definitions for your environment (Adabas, DB2, or VSAM).

8A: Load Command Processor

- Job I200, Step 1300; CST*nnn*.NCPD dataset

  The CST*nnn*.NCPD dataset contains the source for the NCPDEMO command processor defined for the Natural Construct demo application.

➢ To load the command processor source into the FUSER system file:

1 Invoke the SYSNCP utility in batch mode with work file 3 assigned to the CST*nnn*.NCPD dataset. If the Startup transaction is “Menu”, ensure that batch execution on the Natural Security library definition for SYSNCP is set to “N”.

2 If Natural Security is installed, define the demo library to Natural Security and assign the functional security levels for the NCPDEMO command processor on the Additional Options panel. (Be sure to set the KEYWORD DEFAULT to “Public”.)

For information on verifying the installation of the command processor, see **(Optional) Test the Installation of the Command Processor**, page 68.

8B: Load Updated Predict File Definitions

➢ To load the Natural Construct file layout into Predict:

1 Invoke Natural in batch mode with work file 1 assigned to the CST*nnn*.*XXx* dataset (where *XX* corresponds to the two-letter code for your environment and *x* corresponds to your Predict version).

2 Specify an FDIC parameter to correspond to the Predict dictionary where you want the Natural Construct file layout to reside.

3 Specify an FNAT parameter to correspond to the Natural system file where Predict is installed.

4 Use the Predict definitions to generate the physical files for DB2 and/or VSAM. For information, see the Generate function in the Predict documentation.

The following input loads the Natural Construct file layouts into Predict:

```
LOGON SYSDICBE
MENU
SET ALF-TYPE=1
LOAD OBJECTTYPE ALL,REPLACE=YES,ADA=N
.
FIN
```

For more information, see the Predict documentation.

Load Definitions for Adabas Demo Application

- Job I200, Step 1301; CST*nnn*.DA4 dataset

  The CST*nnn*.DA4 dataset contains the unloaded Predict view definitions for the Natural Construct help file and the Adabas demo application.

Load Definitions for VSAM Demo Application

- Job I200, Step 1302; CST*nnn*.VD4 dataset

  The CST*nnn*.VD4 dataset contains the unloaded Predict view definitions for the Natural Construct help file and the VSAM demo application.

Load Definitions for DB2 Demo Application

- Job I200, Step 1303; CST*nnn*.DD4 dataset

  The CST*nnn*.DD4 dataset contains the unloaded Predict view definitions for the Natural Construct help file and the DB2 demo application.

Note: The datasets were created using the V4 Migration utility.

Step 9: Copy Natural Utility Subprograms

- Job I500, Step 1301

  Some generated modules use utility subprograms installed with Natural. Use the CVUSRCOP program to copy the following subprograms:

Subprogram	From Library	To Library
MAPBOXCA	SYSMAP	SYSLIBS
MAPBOXCT	SYSMAP	SYSLIBS
MAPBOXCV	SYSMAP	SYSLIBS
MAPBOXEX	SYSMAP	SYSLIBS
MAPBOXIN	SYSMAP	SYSLIBS
MAPBOXW1	SYSMAP	SYSLIBS
USR0120N	SYSEXT	SYSLIBS
USR0120N	SYSEXT	SYSTEM
USR0320N	SYSEXT	SYSCST
USR0320N	SYSEXT	SYSLIBS
USR0320N	SYSEXT	SYSTEM
USR0622N	SYSEXT	SYSLIBS
USR1002N	SYSEXT	SYSLIBS
USR1002N	SYSEXT	SYSTEM
USR1005N	SYSEXT	SYSLIBS
USR1005N	SYSEXT	SYSTEM
USR1009N	SYSEXT	SYSLIBS
USR1025N	SYSEXT	SYSLIBS
USR1031N	SYSEXT	SYSLIBS
USR1038N	SYSEXT	SYSLIBS
USR1051N	SYSEXT	SYSLIBS
USR1051N	SYSEXT	SYSTEM

Subprogram	From Library	To Library (continued)
USR1057N	SYSEXT	SYSLIBS
USR3013N	SYSEXT	SYSLIBS

Tip: If the CVUSRCOP utility returns a NAT4890 or NAT4891 error, modify the Natural Security definitions for the affected libraries and adjust the Utilities flag to "N". Run CVUSRCOP again to complete the copy procedure.

Step 10: Load Updated Help Text

- Job I500, Step 1302; CST*nnn*.SYSH dataset

  Load the new help text into the Natural Construct system file. Use the CSHLOAD batch utility, described in **Utilities**, page 71, *Natural Construct Help Text*, to load the help text.

➢ To load the updated help text:

1 Assign the CST*nnn*.SYSH dataset to work file 1.

2 Log onto the SYSCST library.

3 Specify the following input for the CSHLOAD utility:

```
*,,,,Y
```

Step 11: Load Updated Models

- Job I500, Step 1303; CST*nnn*.SYSM dataset

  Load the updated models into the Natural Construct system file. The CSMLOAD batch utility, described in **Multiple Model Import Utility**, page 750, *Natural Construct Generation*, loads the models.

➢ To load the updated models:

1 Assign the CST*nnn*.SYSM dataset to work file 1.

2 Log onto the SYSCST library.

3 Specify the following input for the CSMLOAD utility:

```
*,Y
```

Step 12: Load Updated Code Frames

- Job I500, Step 1304; CST*nnn*.SYSR dataset

  Load the updated code frames into the Natural Construct system file. The CSFLOAD batch utility (in the SYSCST library), described in **Multiple Code Frame Import Utility**, page 395, *Natural Construct Administration and Modeling*, loads the code frames.

➢ To load the updated code frames:

1 Assign the CST*nnn*.SYSR dataset to work file 1.

2 Log onto the SYSCST library.

3 Specify the following input for the CSFLOAD utility:

```
*,Y
```

4 After loading the updated code frames, remove all code frames supplied as fixes to older versions of Natural Construct.

To remove old code frames, list existing code frames ("MENU F L" in the SYSCST library) and scan for suffixes of "8" or less (fixes supplied by Software AG are identified by the suffix "8"). For example, COPA9 is an original module that was installed during the installation procedure, while COPA8 was delivered later as a fix.

Warning:
If you used 8 as the suffix in the name of a modified code frame, replace the 8 with 7 or less and then purge all code frames with this suffix.

Use the Compare Code Frame utility to review any suffixes lower than "8". Compare the old version against the new code frames to determine which modifications have been made and may no longer be valid. If you made more extensive model changes than those supplied on this release tape, you may want to apply the changes manually.

The installation tape contains the CST*nnn*.C441 dataset, which compares the new code frames with those in Natural Construct V4.4.

Once copied to disk, you can view these datasets online or route them to a printer.

Step 13: Set Natural Construct System File Version

- Job I500, Step 1305

  After new versions of the models, code frames, and help text are installed, set the version number on the Control record. This allows Natural Construct to determine inconsistencies between the supplied models and their subprograms.

  To assign the new version number to the Natural Construct system file, run the CVSETVER program in the SYSCST library.

Step 14: (Optional) Install in Uppercase Only

If you are installing in an environment that does not support lowercase Latin characters, run the CVUPPERC utility in the SYSCST library to translate the components into uppercase. As this is a resource-intensive process, you should run this utility in batch mode. Ensure that the batch job defines the correct Natural Construct logical file number (227) and FUSER system file.

The CVUPPERC utility performs the following functions:

- Converts all data in the Natural Construct system file to uppercase (CVLO2HIA).
- Converts all English messages to uppercase (CVLO2HIM).
- Converts all supplied source to uppercase (CVLO2HIS).
- Performs a CATALL in the SYSCST library.

When this process is completed, copy all object modules beginning with CU and CG to the SYSLIBS library in the FNAT system file. You can use the following input:

```
LOGON SYSTEM
SYSMAIN
MENU C,C,CU*,FM,SYSCST,DBID,xxx,FNR,yyy,TO,SYSLIBS,DBID,xxx,%
FNR,yyy,REP
SYSMAIN
MENU C,C,CG*,FM,SYSCST,DBID,xxx,FNR,yyy,TO,SYSLIBS,DBID,xxx,%
FNR,yyy,REP
FIN
```

Ensure that the IM=D parameter is set in your NATPARM and then copy all modules beginning with CD and CC to the SYSTEM library (or alternate steplib) in your FNAT system file.

Step 15: (Optional) Translate Error Messages

Natural Construct-generated applications use external error messages that are defined in SYSERR in the SYSTEM and SYSCST libraries (message numbers 8000 to 8200). If you want to generate applications in a language for which these messages have not been translated, you can translate the messages using the Translate function in SYSERR. Natural Construct-generated applications also use text defined in the CSTAPPL library in SYSERR.

The Generation subsystem uses messages defined in the CSTMSG and CSTLDA libraries in SYSERR. You can translate this text to another language.

Step 16: (Optional) Install in Static (One-Language) Mode

By default, Natural Construct is installed in dynamic mode (multilingual version that allows users to display Natural Construct in any available language). If desired, you can install in one language only (static mode). For details, see **Installing in Static (One-Language) Mode**, page 59.

Installing Natural Construct for the First Time

This section describes how to install Natural Construct for the first time. It is intended for users who want to generate, catalog, and run generated modules. The CST*nnn*.INPL dataset installs a complete administration, development, generation, and execution environment.

- To establish an environment to catalog and run generated applications, install Natural Construct using the CST*nnn*.INPC dataset (subset of CST*nnn*.INPL).
- To establish an environment to run (but not catalog) generated applications, install Natural Construct using the CST*nnn*.INPE dataset (subset of CST*nnn*.INPC).

  For information on installing the CST*nnn*.INPC or CST*nnn*.INPE dataset, see **Installing Catalog and Runtime Environments**, page 55.

➢ To install Natural Construct for the first time, perform the following steps:

- ❑ **Step 1: Establish Natural Construct Data File**, page 43
- ❑ **Step 2: Define the Natural Construct NATPARM**, page 43
- ❑ **Step 3: Link Natural Nucleus (batch) with New NATPARM**, page 44
- ❑ **Step 4: Load Modules and Error Messages**, page 45
- ❑ **Step 5: (Conditional) Add Natural Security Definitions**, page 47
- ❑ **Step 6: (Optional) Install DB2 or VSAM Demo Application**, page 49
- ❑ **Step 7: Load Dataset Containing Updates and Fixes**, page 50
- ❑ **Step 8: Link Natural Nucleus (online) with New NATPARM**, page 51
- ❑ **Step 9: (Optional) Load Command Processor and Demo Application Definitions**, page 51
- ❑ **Step 10: Copy Natural Utility Subprograms**, page 52
- ❑ **Step 11: (Optional) Install in Uppercase Only**, page 54
- ❑ **Step 12: (Optional) Translate Error Messages**, page 54
- ❑ **Step 13: (Optional) Install in Static (One-Language) Mode**, page 54

The remainder of this section describes these steps.

Note: The CST*nnn*.JOBS dataset contains sample JCL.

Step 1: Establish Natural Construct Data File

- Job I050, Step 1300; CST*nnn*.SYSF dataset

  The CST*nnn*.SYSF dataset is an unloaded Adabas V6 file and is input to the Adabas ADALOD utility, which loads the Natural Construct data file. The DBID and FNR values you use to load the file must be the same as those specified in the NTFILE macro of the NATPARM module.

  The space required by the Natural Construct system file depends on the number of models and help text members defined within Natural Construct. We recommend setting the following minimum sizes for the ADALOD utility:

- MAXISN=20000
- DSSIZE=5
- NISIZE=300B
- UISIZE=50B

Step 2: Define the Natural Construct NATPARM

Natural Construct uses a special data file that must be identified to the Natural transaction used for Natural Construct.

Note: When using Natural Construct, ensure that BPSFI=OFF. If BPSFI=OFF, Search Sequence 1 is used (alternating search in buffer pool and database for each library). If BPSFI=ON, Search Sequence 2 is used (search in buffer pool first for all libraries and then search database).

To identify this file, use one of the following methods:

- Use the LFILE parameter to dynamically identify the file

  or

- Code the NTFILE macro in the Natural parameter module to make the file part of the Natural nucleus

  If you identify the Natural Construct data file using the LFILE dynamic parameter method, use the LFILE=(227,DBID,FNR) dynamic parameter when invoking Natural. The DBID and FNR values indicate the database ID and file number for the Natural Construct data file.

If you identify the Natural Construct data file using the NTFILE macro in the Natural parameter module, the following example describes its use:

```
ACMPARM TITLE 'Natural Parameters for Natural Construct'
         NTPRM FNAT=(10,5),              System libraries, errors
               FUSER=(10,6),              User programs
               FDIC=(10,7),              Predict data, DDMs
               FSEC=(10,8),              Natural Security file
               LS=132,
               PS=60,
               WORK=7                     Recommended for download
               ESIZE=220,                 Minimum recommended ESIZE
               DATSIZE=180,              Minimum recommended DATSIZE
               SSIZE=70                  Minimum recommended SSIZE
               SYNERR=ON,
         NTFILE DBID=nnn,FNR=mmm,ID=227  Natural Construct data file
         END
```

The DBID value is the database number used when loading the Natural Construct data file; the FNR value is the file number used when loading the Natural Construct data file. The ID value must be 227.

The NTFILE macro allows you to load the Natural Construct data file into any file in any database. It translates the logical file number used in the Natural Construct modules (227) into the database ID and file number for the data file.

To assemble and link the parameter module, use the same procedures as you used to assemble and link the standard Natural V3 parameter module.

Step 3: Link Natural Nucleus (batch) with New NATPARM

Using the Natural link job (I060) that linked the standard Natural nucleus, create an executable Natural nucleus. Include the NATPARM parameter module assembled and linked in Step 2.

Some components of Natural Construct require access to the Software AG editor. Ensure that the editor is installed and operational.

Step 4: Load Modules and Error Messages

To load the required modules and error messages:

A Load the modules for the demo, SYSTEM, SYSCST, SYSCST00, SYSCSTX, and SYSLIBS libraries.

B Load the Predict interface modules.

C Load the error messages.

To load the modules, use the Natural INPL utility to load the applicable dataset and set work file 1 to the dataset name. For example:

```
INPL
B
```

These procedures are described in the following sections.

4A: Load Natural Construct Modules

- Job I061, Step 1300; CST*nnn*.INPL dataset

  Load the modules from the CST*nnn*.INPL dataset into the appropriate system files. This dataset contains modules for the demo, SYSTEM, SYSCST, SYSCST00, SYSCSTX, and SYSLIBS libraries. Ensure that the FNAT, FUSER, and FDIC parameter values correctly identify the desired system files.

  The INPL utility loads all the data areas, copycode, and external modules used by the generated applications into the FNAT SYSTEM library. Ensure that you load the modules and the DDMs.

4B: Load Predict Interface Modules

- Job I061, Step 1304; CST*nnn*.CP42

  Load the modules from the CST*nnn*.CP42 dataset into the appropriate system files. This dataset contains the Predict V4.2 interface modules.

4C: Load Error Messages

- Job I061, Step 1305; CST*nnn*.ERRN dataset

  The CST*nnn*.ERRN dataset contains the error messages for Natural Construct-generated applications. This dataset was created using the SYSERR utility, ERRULDUS. All messages are loaded into SYSERR.

  Note: For information SYSERR, see the Natural documentation.

  - Messages 8000 to 8200 are loaded into the SYSTEM and SYSCST libraries
  - Messages 8300 to 8500 are loaded into the CSTAPPL library
  - Messages 1 to 9999 (error message text) are loaded into the CSTMSG library
  - Messages 1 to 9999 (screen prompt text) are loaded into the CSTLDA library
  - Messages 1 to 9999 (text for Actions) are loaded into the CSTACT library
  - Messages 1 to 9999 (text for PF-keys) are loaded into the CSTPFK library

  To load these messages into your FUSER system file, invoke Natural in batch mode with work file 2 assigned to the CST*nnn*.ERRN dataset. Use the following statements to load the messages into SYSERR:

```
LOGON SYSTEM
ERRLODUS
FIN
```

  Note: If the SYSTEM library is not defined as the highest-level steplib under Natural Security, use the SYSMAIN utility to copy the messages from the SYSTEM library to your alternate steplib.

Step 5: (Conditional) Add Natural Security Definitions

If you have Natural Security installed, use it to add definitions for the SYSCST and SYSCSTX libraries. SYSCST contains the generator maintenance facilities for Natural Construct; SYSCSTX contains sample user exit routines.

You can restrict access to these libraries to only those users who maintain the definitions that control the generation process. Users who generate modules or maintain help text using Natural Construct do not require access to the libraries.

Note: For a list of the libraries you must define to Natural Security, see **Defining Libraries to Natural Security**, page 58.

Adabas Demo Library

For Adabas, the NCSTDEMO library contains examples of how to use the supplied models. It should be accessible to all developers. The modules in this library access the following DDMs:

- NCST-ORDER-HEADER
- NCST-ORDER-LINES
- NCST-ORDER-DISTRIBUTION
- NCST-CUSTOMER
- NCST-PRODUCT
- NCST-WAREHOUSE
- NCST-INS-POLICY

These files should be publicly available.

To allow programs that access the Natural Construct help file to be recataloged, define the following DDMs to Natural Security:

- NCST-HELP
- NCST-HELP-LINES
- NCST-PROFILE

DB2 Demo Library

For DB2, the NCSTDEM2 library contains examples of how to use the supplied models. It should be accessible to all developers. The modules in this library access the following DDMs:

- NCSTDB2-CUSTOMER
- NCSTDB2-ORDER_DISTRIBUTION
- NCSTDB2-ORDER_HEADER
- NCSTDB2-ORDER_INSTRUCTIONS
- NCSTDB2-ORDER_LINES
- NCSTDB2-PRODUCT
- NCSTDB2-WAREHOUSE

VSAM Demo Library

For VSAM, the NCSTDEMV library contains examples of how to use the supplied models. It should be accessible to all developers. The modules in this library access the following DDMs:

- NCSTVSAM-CUSTOMER
- NCSTVSAM-ORDER-DISTRIBUTION
- NCSTVSAM-ORDER-HEADER
- NCSTVSAM-ORDER-LINES
- NCSTVSAM-PRODUCT
- NCSTVSAM-WAREHOUSE

Step 6: (Optional) Install DB2 or VSAM Demo Application

If you are installing in an environment that accesses DB2 or VSAM, you can install the corresponding demo application. To copy the file definitions into Predict, use the Natural INPL utility to load the applicable dataset.

Note: Use the Predict definitions to generate the required physical files. For information, see the Predict documentation.

Install DB2 Demo Application

- Job I061, Step 1308; CST*nnn*.DINP

  The CST*nnn*.DINP dataset contains the demo application to access DB2 tables. The DB2 demo application accesses the following files:

  – NCSTDB2-CUSTOMER
  – NCSTDB2-ORDER_DISTRIBUTION
  – NCSTDB2-ORDER_HEADER
  – NCSTDB2-ORDER_INSTRUCTIONS
  – NCSTDB2-ORDER_LINES
  – NCSTDB2-PRODUCT
  – NCSTDB2-WAREHOUSE

➢ To load the demo application for DB2:

1 Use the Natural INPL utility and set work file 1 to CST*nnn*.DINP.

2 Enter the following:

```
INPL
B
```

➢ To copy the data used by the DB2 demo application:

1 Log onto the NCSTDEM2 library.

2 Catalog and run the CSGMIGA2 program.

Install VSAM Demo Application

- Job I061, Step 1307; CST*nnn*.VINP

  The CST*nnn*.VINP dataset contains the demo application to access VSAM data files. The VSAM demo application accesses the following files:

  – NCSTVSAM-CUSTOMER
  – NCSTVSAM-ORDER-DISTRIBUTION
  – NCSTVSAM-ORDER-HEADER
  – NCSTVSAM-ORDER-LINES
  – NCSTVSAM-PRODUCT
  – NCSTVSAM-WAREHOUSE

➢ To load the demo application for VSAM:

1 Use the Natural INPL utility and set work file 1 to CST*nnn*.VINP.

2 Enter the following:

```
INPL
B
```

➢ To copy the data used by the VSAM demo application:

1 Log onto the NCSTDEMV library.

2 Catalog and run the CSGMIGAV program.

Step 7: Load Dataset Containing Updates and Fixes

- Job I061, Step 1309; CST*nnn*.IUPD dataset

  This dataset contains updates and fixes to the main INPL dataset (CST*nnn*.INPL); it is only included in the installation package if there are updates or fixes to the INPL dataset.

➢ To load the updates and fixes to the main INPL dataset:

1 Invoke the Natural INPL utility.

2 Set work file 1 to the applicable dataset name (CST*nnn*.IUPD)

3 Enter the following:

```
INPL
B
```

Step 8: Link Natural Nucleus (online) with New NATPARM

Using the Natural job (I080) that linked the standard Natural nucleus, create an executable Natural nucleus. Include the NATPARM parameter module assembled and linked in **Step 2: Define the Natural Construct NATPARM**, page 43.

To use the online job submission features of Natural Construct, ensure that the NATRJE module is installed and operational.

Step 9: (Optional) Load Command Processor and Demo Application Definitions

Optionally, you can load the command processor and Predict file definitions for the demo application. To load the command processor and file definitions:

A Load the command processor source.

B Load the Predict file definitions for your environment (Adabas, DB2, or VSAM).

9A: Load Command Processor

- Job I200, Step 1300; CST*nnn*.NCPD dataset

  The CST*nnn*.NCPD dataset contains the source for the NCPDEMO command processor defined for the Natural Construct demo application.

➢ To load the command processor source into the FUSER system file:

1 Invoke the SYSNCP utility in batch mode with work file 3 assigned to the CST*nnn*.NCPD dataset. If the Startup transaction is “Menu”, ensure that batch execution on the Natural Security library definition for SYSNCP is set to “N”.

2 If Natural Security is installed, define the demo library to Natural Security and assign the functional security levels for the NCPDEMO command processor on the Additional Options panel. (Be sure to set the KEYWORD DEFAULT to “Public”.)

9B: Load Predict File Definitions

➢ To load the Natural Construct file layout into Predict:

1 Invoke Natural in batch mode with work file 1 assigned to the CST*nnn.XXx* dataset (where *XX* corresponds to the two-letter code for your environment and *x* corresponds to your Predict version).

2 Specify an FDIC parameter to correspond to the Predict dictionary where you want the Natural Construct file layout to reside.

3 Specify an FNAT parameter to correspond to the Natural system file where Predict is installed.

4 Use the Predict definitions to generate the physical files for DB2 and/or VSAM. For information, see the Generate function in the Predict documentation.

The following input loads the Natural Construct file layouts into Predict:

```
LOGON SYSDICBE
MENU
SET ALF-TYPE=1
LOAD OBJECTTYPE ALL,REPLACE=YES,ADA=N
.
FIN
```

For more information, see the Predict documentation.

Load Definitions for Adabas Demo Application

- Job I200, Step 1301; CST*nnn*.DA4 dataset

  The CST*nnn*.DA4 dataset contains the unloaded Predict view definitions for the Natural Construct help file and the Adabas demo application.

Load Definitions for VSAM Demo Application

- Job I200, Step 1302; CST*nnn*.VD4 dataset

  The CST*nnn*.VD4 dataset contains the unloaded Predict view definitions for the Natural Construct help file and the VSAM demo application.

Load Definitions for DB2 Demo Application

- Job I200, Step 1303; CST*nnn*.DD4 dataset

  The CST*nnn*.DD4 dataset contains the unloaded Predict view definitions for the Natural Construct help file and the DB2 demo application.

Note: The datasets were created using the V4 Migration utility.

Step 10: Copy Natural Utility Subprograms

- Job I500, Step 1301

  Some generated modules use utility subprograms installed with Natural. Use the CVUSRCOP program in the SYSCST library to copy the following subprograms:

Subprogram	From Library	To Library
MAPBOXCA	SYSMAP	SYSLIBS
MAPBOXCT	SYSMAP	SYSLIBS
MAPBOXCV	SYSMAP	SYSLIBS

Subprogram	From Library	To Library (continued)
MAPBOXEX	SYSMAP	SYSLIBS
MAPBOXIN	SYSMAP	SYSLIBS
MAPBOXW1	SYSMAP	SYSLIBS
USR0120N	SYSEXT	SYSLIBS
USR0120N	SYSEXT	SYSTEM
USR0320N	SYSEXT	SYSCST
USR0320N	SYSEXT	SYSLIBS
USR0320N	SYSEXT	SYSTEM
USR0622N	SYSEXT	SYSLIBS
USR1002N	SYSEXT	SYSLIBS
USR1002N	SYSEXT	SYSTEM
USR1005N	SYSEXT	SYSLIBS
USR1005N	SYSEXT	SYSTEM
USR1009N	SYSEXT	SYSLIBS
USR1025N	SYSEXT	SYSLIBS
USR1031N	SYSEXT	SYSLIBS
USR1038N	SYSEXT	SYSLIBS
USR1051N	SYSEXT	SYSLIBS
USR1051N	SYSEXT	SYSTEM
USR1057N	SYSEXT	SYSLIBS
USR3013N	SYSEXT	SYSLIBS

Tip: If the CVUSRCOP utility returns a NAT4890 or NAT4891 error, modify the Natural Security definitions for the affected libraries and adjust the Utilities flag to “N”. Run CVUSRCOP again to complete the copy procedure.

Step 11: (Optional) Install in Uppercase Only

If you are installing in an environment that does not support lowercase Latin characters, you can run the CVUPPERC utility to translate the product into uppercase. To run this utility, you must be in the SYSCST library. As this is a resource-intensive process, we recommend that you run the utility in batch mode. Ensure that the batch job defines the correct Natural Construct logical file number (227) and FUSER system file.

The CVUPPERC utility performs the following functions:

- Converts all data in the Natural Construct system file to uppercase (CVLO2HIA).
- Converts all English messages used by Natural Construct to uppercase (CVLO2HIM).
- Converts all supplied source to uppercase (CVLO2HIS).
- Performs a CATALL in the SYSCST library.

When this process is completed:

1 Copy all object modules beginning with CU and CG to the SYSLIBS library in the FNAT system file.
You can use the following batch input:

```
LOGON SYSMAIN
MENU
C,C,CU*,FM,SYSCST,DBID,xxx,FNR,yyy,TO,SYSLIBS,DBID,xxx,%
FNR,yyy,REP
MENU
C,C,CG*,FM,SYSCST,DBID,xxx,FNR,yyy,TO,SYSLIBS,DBID,xxx,%
FNR,yyy,REP
FIN
```

2 Copy all modules beginning with CD and CC to the SYSTEM library.

Step 12: (Optional) Translate Error Messages

Generated applications use external error messages defined in SYSERR in the SYSTEM and SYSCST libraries (message numbers 8000 to 8200). If you want to generate applications in a language for which these messages have not been translated, you can translate the messages using the Translate function in SYSERR. Generated applications also use text defined in the CSTAPPL library in SYSERR.

The Generation subsystem uses messages defined in the CSTMSG and CSTLDA libraries in SYSERR. You can translate this text to another language.

Step 13: (Optional) Install in Static (One-Language) Mode

By default, Natural Construct is installed in dynamic mode (a multilingual version that allows users to display Natural Construct in any available language). If desired, you can install in one language only (static mode). For details, see **Installing in Static (One-Language) Mode**, page 59.

Installing Catalog and Runtime Environments

This section describes how to set up environments in which you can catalog and run, or run (but not catalog), generated modules without installing Natural Construct in its entirety. In a catalog and runtime environment, you can catalog as well as run generated modules. In a runtime environment, you can run generated modules, but you cannot catalog.

➢ To install catalog and runtime or runtime-only environments, perform the following steps:

- ❑ **Step 1: Establish Natural Construct Data File**, page 55
- ❑ **Step 2: Load Updated Natural Construct Modules**, page 56
- ❑ **Step 3: Load Error Messages**, page 57
- ❑ **Step 4: Copy Natural Utility Subprograms**, page 57

The remainder of this section describes these steps.

Note: The CST*nnn*.JOBS dataset contains sample JCL.

Step 1: Establish Natural Construct Data File

- Job I050, Step 1300; CST*nnn*.SYSF dataset

If your applications use help text in Natural Construct's help facility, you may have to provide access to the Natural Construct system file to establish a production environment.

Note: If you are an existing user and have application help text in your production environment, use the CSHUNLD utility to unload your help text before loading this dataset; use the CSHLOAD utility to reload your help text after loading this dataset.

For information about loading the help text data file, see **Step 10: Load Updated Help Text**, page 39.

Step 2: Load Updated Natural Construct Modules

You can load the updated Natural Construct modules for a catalog and runtime environment or a runtime-only environment.

For a Catalog and Runtime Environment

- Job I061, Step 1301; CST*nnn*.INPC dataset

  Using the Natural INPL utility, load the modules from the CST*nnn*.INPC dataset into the appropriate system files. This dataset is a subset of the CST*nnn*.INPL dataset and contains the modules required for a catalog and runtime environment.

  The INPL utility loads all the data areas, copycode, and external modules used by the generated applications into the FNAT SYSTEM library. Ensure that you load both the modules and the DDMs. For example:

```
INPL
B
```

For a Runtime-Only Environment

- Job I061, Step 1302; CST*nnn*.INPE dataset

  Using the Natural INPL utility, load the modules from the CST*nnn*.INPE dataset into the appropriate system files. This dataset is a subset of the CST*nnn*.INPC dataset and contains the modules required for a runtime-only environment. If you want to run generated applications and do not want to install Natural Construct in its entirety (using the CST*nnn*.INPL dataset), load this dataset into the system files for runtime environments.

  The INPL utility loads all external modules used by generated applications into the FNAT SYSTEM library. Ensure that you load the modules and the DDMs. For example:

```
INPL
B
```

Step 3: Load Error Messages

- Job I061, Step 1305; CST*nnn*.ERRN dataset

  To establish a production environment for Natural Construct, you may also have to provide access to the error messages supplied in SYSERR. The CST*nnn*.ERRN dataset contains the error messages. For information about loading the error message file, see **4C: Load New Error Messages**, page 33.

Step 4: Copy Natural Utility Subprograms

- Job I500, Step 1310

  Copy the following USR routines from SYSEXT to the SYSTEM library in the FNAT file:

  – USR0120N
  – USR0320N
  – USR1002N
  – USR1005N
  – USR1051N

Defining Libraries to Natural Security

If you are running under Natural Security, define the following Natural Construct libraries:

Library	Description
SYSCST	Natural Construct administration library.
SYSCSTX	Source for user exits.
CSTAPPL	Application for Natural Construct message text in SYSERR.
NCSTDEMO	Adabas demo application.
NCSTDEM2	DB2 demo application.
NCSTDEMV	VSAM demo application.

Activating Natural Construct

After installing Natural Construct, enter “ncstg” at the Natural Next prompt to activate the program. The Generation main menu is displayed if the installation was successful.

Installing in Static (One-Language) Mode

By default, Natural Construct is installed in dynamic (multilingual) mode, which allows users to display Natural Construct in any available language. If you intend to run the system in one language only, you can install Natural Construct in static (one-language) mode.

Note: Installing in static mode does not limit your ability to generate multilingual applications; static mode applies to the interface only.

There are two options for installing in static mode. You can take either or both options.

- Using the Static Install option, you can create performance local data areas (LDAs) for the supplied models (modules that begin with “CU” or “CG”). This option eliminates dynamic lookups to the SYSERR file every time the model specification panels are invoked.
- Using the Create Performance LDAs and Create Performance Subps functions, you can create performance LDAs and the corresponding subprograms for the Natural Construct nucleus. This option creates one-language versions of the 10 most commonly used modules for which source is not supplied.

Note: If you are installing a static version in any language except English, review all messages in the CSTLDA library in SYSERR to ensure they are translated into the desired language.

These options are described in the following sections.

Installing Natural Construct in Static Mode

➢ To install Natural Construct in static mode:

1 Log onto the SYSCST library.

2 Enter “NCSTI” (Natural Construct Install) at the Natural prompt.
The Natural Construct Installation main menu is displayed:

```
NCSTI           ***** N A T U R A L   C O N S T R U C T *****
Mar 27                    - Installation Main Menu -                  9:52 AM

               Code Function
               ---- ----------------------------------------
                S   Static Install (one language)
                L   Create Performance LDAs
                I   Create Performance Subps
                ?   Help
                .   Terminate
               ---- ----------------------------------------
          Code: _




Direct command...: _____________________________________________________________
Enter--PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12-
       help  retrn quit        flip                                    main
```

Natural Construct Installation Main Menu

3 Enter “S” in the Code field.
The following window is displayed:

```
INSTALL         ***** N A T U R A L   C O N S T R U C T *****
Mar 27                - Static Install (one language) -             9:57 AM

Enter the language in which you would like Natural Construct
installed (Any PF-key to quit): 1_
```

Static Install Window

4 Enter the number for the language in which you want to install Natural Construct (for example, “2” for German, “3” for French).
Natural Construct recreates all the LDAs for the model specification panels and replaces the SYSERR references to field prompts with the text for the language specified. The following window is displayed:

```
INSTALL         ***** N A T U R A L   C O N S T R U C T *****
Mar 27                - Static Install (one language) -             10:37 AM

All data areas have been populated with text appropriate to
language 1  .
In order to complete this process, please recompile all
Natural modules in the SYSCST library beginning with CU, CG
and copy the object code for these modules to SYSLIBS.
```

Window Displayed when Process is Finished

Note: Set the runtime parameter for Natural to “20”.

5 Perform a CATALL on modules beginning with “CU” or “CG” in the SYSCST library. You need only select the subprogram and local data area (LDA) modules. In addition, mark the Catalog ALL Source-programs option to catalog all source modules.

6 You may want to do this step in batch mode, because many modules are affected. You can use the following input:

```
LOGON SYSCST
CATALL CU*,,X,,,X,,,,X,,,,,
CATALL CG*,,X,,,X,,,,X,,,,,
FIN
```

7 Copy the object code from these modules into the SYSLIBS library.
If you prefer to do this in batch mode, the SYSMAIN input commands are supplied below. Ensure that the parameter IM=D is set in your NATPARM.

Use the following batch input:

```
LOGON SYSTEM
SYSMAIN
MENU C,C,CU*,TYPE,N,FM,SYSCST,DBID,xxx,FNR,yyy,TO,SYSLIBS,DBID,xxx,%
FNR,yyy,REP
SYSMAIN
MENU C,C,CG*,TYPE,N,FM,SYSCST,DBID,xxx,FNR,yyy,TO,SYSLIBS,DBID,xxx,%
FNR,yyy,REP
FIN
```

Creating Performance LDAs and Subprograms

Regardless of whether you choose the Static Install function or not, this option will enhance performance by creating some subprograms that eliminate calls to SYSERR to build many of the frequently used screens (such as the Generation main menu). Because these programs are not supplied in source form, use the Create Performance LDAs function to create LDAs containing the text appropriate to the desired language and then use the Create Performance Subps function to create the performance subprograms. You can repeat these two steps as many times as desired, depending on how many languages you want to make available.

Note: Natural Construct supplies the performance subprograms for English. If you are running Natural Construct in a language for which these subprograms have not been created, the English subprograms will be invoked.

➢ To create performance LDAs and subprograms for the Natural Construct nucleus:

1 Copy the contents (source and object) of the SYSCST00 library into the SYSCST*nn* library (where *nn* is the language code for the language you want to support, such as "1" for English, "2" for German, "3" for French).

2 Log onto the SYSCST*nn* library.

3 Enter "NCSTI" (Natural Construct Install) at the Natural prompt.
The Natural Construct Installation main menu is displayed.

Note: When running NCSTI to create these LDAs and subprograms, the DC and ID characters must be set to the default (DC=. and ID=,).

4 Enter "L" in the Code field.
The following window is displayed:

```
INSTALL2        ***** N A T U R A L   C O N S T R U C T *****
Mar 27                  - Create Performance LDAs -               10:18 AM

NOTE: You must be in library SYSCSTnn (where nn represents
      the language number) in order to execute this function.
      This step may be repeated for as many languages
      as desired.

You are currently in library: SYSCST01
About to create performance LDAs for language: 1
Press ENTER to continue - any PF-key to stop.
```

Create Performance LDAs Window

5 Press Enter.
The following confirmation window is displayed:

```
INSTALL2        ***** N A T U R A L   C O N S T R U C T *****
Mar 27                  - Create Performance LDAs -               10:21 AM

All data areas have been populated with text appropriate to
language 1   .  Please CATALL this library ( SYSCST01 )
before creating the Performance Subprograms.
```

Confirmation Window

Note: You must be in the SYSCSTnn library corresponding to the language for which you are creating the LDAs. This allows multiple languages to be supported, since the LDAs are created in different libraries.

6 Press Enter.

7 Perform a CATALL on this library, ensuring that all 10 LDAs are cataloged successfully.

8 Log onto the SYSCST library.

9 Enter “NCSTI” at the Natural prompt.
The Natural Construct Installation main menu is displayed.

10 Enter “I” in the Code field.
The Create Performance Subps window is displayed:

```
CSTTRANS        ***** N A T U R A L   C O N S T R U C T *****
Mar 27                  - Create Performance Subps -              10:24 AM

NOTE:  This function must be executed from library SYSCST

Enter the language number for which you would like
performance subprograms generated: __
(Press any PF-key to stop)
```

Create Performance Subps Window

11 Enter the number of the language for which you have created performance LDAs. Natural Construct creates object-only performance subprograms for the specified language.

12 Copy the performance subprograms from the SYSCST library to SYSLIBS.
These modules begin with “CZ” and end with the *Language value for the language in which you are installing (for example, “CZHOBJ2” for German).

13 Log onto the SYSCSTX library and edit the CSXDEFLT subprogram as follows:

- Set the PERFORMANCE default to TRUE (must be in uppercase). For example:

```
**SAG DEFINE EXIT GENERATE-CODE
*
* Your code to implement defaulting for your CST models.
DECIDE ON FIRST VALUE CSADEFLT.PARM-NAME
  VALUE 'PERFORMANCE'
    ASSIGN CSADEFLT.PARM-VALUE = 'TRUE'
  NONE
    IGNORE
END-DECIDE
**SAG END-EXIT
```

- Save the CSXDEFLT subprogram in the SYSCSTX library.
- Use the Natural SYSMAIN utility to copy CSXDEFLT to the SYSCST library.
- Catalog CSXDEFLT in the SYSCST library.
- Use the SYSMAIN utility to copy the CSXDEFLT object code to SYSLIBS.

VERIFYING THE INSTALLATION

After installing Natural Construct, there are several procedures you can perform to ensure that the program has been installed correctly. This chapter describes these procedures.

The following topics are covered:

Regenerate and Recatalog the Demo System

➢ To regenerate and recatalog the demo system:

1 Log onto the demo system as follows:
- "NCSTDEMO" for Adabas
- "NCSTDEM2" for DB2
- "NCSTDEMV" for VSAM

2 Enter "NCSTBGEN" at the Next prompt.
The Multiple Generation panel is displayed:

```
CSBINOL                Natural Construct                    CSBM0
Jan 02               Multiple Generation                  1 of 1

Module
 ________   ________   ________   ________   ________
 ________   ________   ________   ________   ________

Model
 ____________________________   ____________________________
 ____________________________   ____________________________
 ____________________________   ____________________________
 ____________________________   ____________________________
 ____________________________   ____________________________

Catalog regenerated modules .. _
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--P
      help  retrn
```

Multiple Generation Panel

3 Type "*" in the first Module field.

4 Type "x" in the Catalog regenerated modules field.

5 Press Enter.
The utility regenerates all modules and produces a report listing the names of the modules it generated.

Note: If the modules cannot be regenerated or recataloged, check the Natural parameters. If a NAT0082 error occurs, one or more modules are not loaded. Check the INPL jobs to determine which modules are missing.

Invoke the Demo System

➢ To invoke the demo system:

1 Enter “menu” at the Next prompt.
The Order Entry main menu should be displayed.

2 Select a function from the menu.
Ensure that the function is available and working.

Invoke the Help Text Subsystem

➢ To invoke the Help Text subsystem:

1 Enter “ncsth” at the Next prompt.
The Help Text main menu should be displayed.

2 Enter “L” in the Function field to list the available help members.
Ensure that the Natural Construct help members are loaded.

Invoke the Administration and Modeling Subsystem

➢ To invoke the Administration and Modeling subsystem:

1 Logon to the SYSCST library.

2 Enter “menu” at the Next prompt.
The Administration main menu should be displayed.

3 Enter “M” in the Function field.
The Maintain Models panel is displayed.

4 Enter “B” in the Action field.
The Select Model window is displayed. Ensure that the supplied models are loaded.

(Optional) Test the Installation of the Command Processor

➢ To test whether the installation of the command processor was successful:

1 Log onto the demo library.

2 Enter “ncstg” at the Next prompt.

3 Read “Menu” into the edit buffer.

4 Modify the specifications on the first panel to refer to the NCPDEMO command processor.

5 Regenerate and test “Menu” as follows:

- Enter “Add order” on the Menu command line.
  The order maintenance panel should be displayed.
- Enter “Add customer” on the Order panel command line.
  The customer maintenance panel should be displayed.

MAINTAINING NATURAL CONSTRUCT

This chapter describes the operational requirements to maintain and use Natural Construct. It includes information about accessing the Natural Construct data file and model maintenance facilities, as well as the supplied user exit subprograms and the generation and runtime environments. It also lists the modules supplied with Natural Construct and provides a summary of the installation datasets.

The following topics are covered:

Accessing the Natural Construct Data File

To run Natural Construct-generated modules, you must have access to the Natural Construct data file. The supplied models, as well as help text used by Natural Construct and Natural Construct-generated modules, are stored in the Natural Construct data file. Therefore, every Natural transaction that uses Natural Construct facilities must either:

- Be linked to a NATPARM module that includes the NTFILE definition for the Natural Construct data file (for more information, see **Step 1: Establish Natural Construct Data File**, page 43).

  or

- Be invoked using the LFILE dynamic parameter to specify the Natural Construct data file, for example:

  ```
  LFILE=(227,DBID,FNR)
  ```

  where 227 identifies Natural Construct and DBID and FNR identify the database ID and file number of the Natural Construct data file.

➢ To create an environment that does not require the NTFILE macro or LFILE dynamic parameter:

1 Recatalog the NCST-HELP, NCST-HELP-LINES, and NCST-PROFILE DDMs.

2 Recatalog the following subprograms:

- CD-HELP
- CD-HELPR
- CD-HELPL
- CD-HPRED

These subprograms can then use the Natural Construct data file without requiring the NTFILE macro or LFILE dynamic parameter.

Accessing the Model Maintenance Facilities

Access the Natural Construct models, code frames, and Control record maintenance facilities from the SYSCST library in the FNAT system file. To use these facilities, the Natural transaction must use a Natural FNAT system file in which Natural Construct is loaded. You can restrict access to the maintenance facilities using Natural Security.

The Natural Construct definition and model maintenance facilities do not require Predict.

User Exit Subprograms to Implement Security

Natural Construct supplies user exit subprograms for the Administration and Help Text subsystems. These subprograms allow you to implement security or restrict access to various objects (models, code frames, model subprograms, help text members, etc.).

If the user exit subprograms are in a library within the Administration or Help Text subsystem, Natural Construct invokes the applicable subprogram to enforce security when an object and action are requested. Depending on the value specified, the subprogram grants or denies access.

The following table lists the supplied user exit subprograms, the panels they secure, and the libraries they are invoked from:

Subprogram	Panel	Library
CSXDUEXT	Administration Main Menu	SYSCST
CSXMUEXT	Maintain Models	SYSCST
CSXFUEXT	Code Frame Menu	SYSCST
CSXSUEXT	Maintain Subprograms	SYSCST
CSXHUEXT	Help Text Main Menu	SYSLIBS

Note: The CST*nnn*.INPL dataset contains samples of the user exit subprograms. The sample subprograms are initially loaded into the SYSCSTX library. If you want to activate a subprogram, modify the subprogram and copy it to the library indicated above. Always keep a backup copy of any modified user exit subprogram.

Natural Construct Generation Environment

This section describes the generation environment for Natural Construct. It contains information about the applicable FNAT and FUSER system files, how to control access to the generation environment, the modules required in the environment, and any Predict and steplib considerations.

FNAT System File

The modules you require to use the Generation and Help Text subsystems are stored in the SYSLIBS protected library in the FNAT system file. To use these facilities, the Natural transaction must use the FNAT system file into which Natural Construct is loaded.

Controlling Access to the Generation Environment

You can control access to the Generation and Help Text maintenance facilities using Natural Security. When you disallow access to NCSTG and NCSTBGEN, users cannot access the generation facilities. (The NCSTBGEN module invokes the Natural Construct batch regeneration facility.) When you disallow access to NCSTH, users cannot access the help text facilities.

Required Modules

The supplied code frames and models require certain modules when a Natural Construct-generated module is cataloged, stowed, or run. These modules are necessary because the generated modules contain INCLUDE statements and references to external global, local, and parameter data areas. When Natural Construct is installed, modules that support the INCLUDE statements and external data definitions are loaded into the FNAT SYSTEM library.

Using Predict

To use the generation facilities, Predict must be available in the generation environment and the Natural transaction must use Predict V4.2.

Natural Construct Runtime Environment

This section describes the runtime environment for Natural Construct. It contains information about required modules and steplib considerations.

Required Modules

At runtime, Natural Construct-generated modules must have access to several supplied modules in the current library or a steplib. Generated modules use the Natural subprograms contained in the supplied modules to provide command processing, password checking, and context-sensitive help. These supplied modules must reside in either:

- the current library at the time the generated module is invoked or run

  or

- a steplib

Using Steplibs

Natural Construct installs the CD* and CC* runtime modules in the FNAT SYSTEM library. If you place modified versions in your FUSER SYSTEM library, these modules can be overridden. The CD* modules include the external data areas, the command-processing and password-checking programs, and the context-sensitive helproutines and subprograms. The CC* modules contain shared copy code members.

Modules Supplied with Natural Construct

During installation, modules from the installation tape datasets must be loaded into the target library. You can find the modules in more than one dataset and take the modules from any dataset; you do not have to use all of the datasets.

When you run the NATLOAD utility, the modules are automatically loaded into the target library unless you specify “ALL LIBS” or “STEPLIB” (under Natural Security).

Note: You may have to copy the modules into these libraries in a separate procedure.

The following table lists the prefixes that identify different types of modules supplied with Natural Construct and whether source code is provided for the modules:

Prefix	Source	Type of Module
CC*	Yes	Copycode members used by generated modules. You can modify these members as desired.
CD*	Yes	Default objects (helproutines, subprograms, programs, map layouts, and data areas) used by some generated modules. You can modify these objects as desired.
CG*	Yes	Statement model subprograms. These subprograms collect specification parameters and generate code for the .g models. You can modify these subprograms as desired.
CJ*	Some	Natural objects related to JCL generation.
CNH*	No	Helproutines for user-defined models; these helproutines provide active help for Natural system file information.
CNU*	No	Utility subprograms for user-defined models; they access information in the Natural system files.
CNA*	Yes	Data areas for CNU* subprograms.
CPH*	No	Helproutines for user-defined models; these helproutines provide active help for Predict file information.
CPU*	No	Utility subprograms for user-defined models; these subprograms access Predict file information.
CPA*	Yes	Data areas for CPU* subprograms.

Prefix	Source	Type of Module (continued)
CSU*	No	Utility subprograms for user-defined models; these subprograms provide specialized generation and data manipulation functions.
CSA*	Yes	Data areas for CSU* subprograms.
CS*	No	Natural Construct system programs.
CTE*	Yes	Driver programs.
CU*	Yes	Model subprograms; these subprograms collect specification parameters and generate program portions. You can modify the subprograms as desired.
CV*	Some	Conversion utility modules. You may require these modules when you convert to a new version of Natural Construct.
NCST*	No	Startup modules and error routines. These modules must reside in the SYSTEM library (or steplib).

Note: When you modify CU modules, you must maintain or reapply the modifications with each new release of Natural Construct. You should either rename the modified CU modules (by changing the "CU" prefix to a "CX" prefix, for example) or modify them in a higher-level steplib.

Summary of Installation Datasets

This section describes the datasets required for the following environments:

- Generation and maintenance
- Catalog and runtime
- Runtime-only
- Natural Construct/Construct Spectrum

Generation and Maintenance Environment

The following table lists the datasets containing modules required to generate and maintain models and help text. It includes the names of the files or libraries in which the dataset contents are stored and the contents of the datasets:

Dataset	Target File/Library	Dataset Contents
CST*nnn*.INPL	FNAT/SYSCST	Code frame and model maintenance modules
CST*nnn*.INPL	FNAT/SYSLIBS	Generation and help text modules
CST*nnn*.IUPD	FNAT/SYSCST	Updates and fixes to the main INPL dataset (only included if there are updates or fixes)
CST*nnn*.CP42	FNAT/SYSCST	CPA*, CPU*, and CPH* modules for Predict V4.2
CST*nnn*.DA4	Predict system	Predict data
CST*nnn*.ERRN	FUSER application messages	Error messages

Catalog and Runtime Environment

The following table lists the datasets containing modules required to catalog and run generated modules. It includes the names of the files or libraries in which the dataset contents are stored and the contents of the datasets:

Dataset	Target File/Library	Dataset Contents
CST*nnn*. INPL or CST*nnn*.INPC	FNAT/SYSTEM	CC* and CD* modules and data areas
CST*nnn*.SYSF	NTFILE, LFILE, DBID, and FNR	Natural Construct data file
CST*nnn*.ERRN	FUSER application messages	Error messages

Runtime-Only Environment

The following table lists the datasets containing modules required to run (but not catalog) generated modules. It includes the names of the files or libraries in which the dataset contents are stored and the contents of the datasets:

Dataset	Target File/Library	Dataset Contents
CST*nnn*. INPL or CST*nnn*.INPE	FNAT/SYSTEM	CD* programs, helproutines, subprograms, and maps
CST*nnn*.SYSF	NTFILE, LFILE, DBID, and FNR	Natural Construct data file
CST*nnn*.ERRN	FUSER application messages	Error messages

Natural Construct/Construct Spectrum Environment

The following table lists the datasets containing modules required to create a Natural Construct/Construct Spectrum environment. It includes the names of the files or libraries in which the dataset contents are stored and the contents of the datasets:

Dataset	Target File/Library	Dataset Contents
CST*nnn*.ERRN	FUSER application messages	Error messages

SMA Tables

The following table lists each dataset and the corresponding job and step names:

Dataset Name	Job Name	Step Name
CST*nnn*.CP42	I061	1304
CST*nnn*.DA4	I200	1301
CST*nnn*.DD4	I200	1303
CST*nnn*.DINP	I061	1308
CST*nnn*.ERRN	I061 I061	1305 1376
CST*nnn*.INPC	I061	1301
CST*nnn*.INPE	I061	1302
CST*nnn*.INPL	I061	1300
CST*nnn*.IUPD	I061	1309
CST*nnn*.NCPD	I200	1300
CST*nnn*.SYSF	I050	1300
CST*nnn*.SYSH	I500	1302
CST*nnn*.SYSM	I500	1303
CST*nnn*.SYSR	I500	1304
CST*nnn*.VD4	I200	1302
CST*nnn*.VINP	I061	1307