

# Installing Natural Construct

This section describes how to install Natural Construct without Natural Business Services. The following topics are covered:

- Dataset Summary
- Modules Supplied with Natural Construct
- Install Natural Construct Over an Existing Version
- Install Natural Construct for the First Time
- Install Natural Construct Catalog and Runtime Environments
- Install in Static (One-Language) Mode
- Activate Natural Construct
- Verify the Installation of Natural Construct
- Maintain Natural Construct

**Note:**

By default, Natural Construct is installed during the installation of the Business Service repository and server components. For information, see *Installing the Business Services Repository and Server Components*.

---

## Dataset Summary

This section describes the datasets required for the following environments:

- Generation and Maintenance Environment
- Catalog and Runtime Environment
- Runtime-Only Environment
- SMA Tables

### 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
	FNAT/SYSLIBS	Generation and help text modules
	FNAT/FUSER	Messages used by Natural Construct-generated applications
CST $nnn$ .CPRD	FNAT/SYSCST	CPA*, CPU*, and C.H.* modules for Predict V4.5
CST $nnn$ .DA4	Predict system	Predict data

The INPL dataset loads the messages used by Natural Construct-generated applications into SYSERR as follows:

- 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
- Messages 1 to 9999 are loaded into the SYSBIZ library

**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.

## 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 $nnn$ .INPE or CST $nnn$ .INPC	FNAT/SYSTEM	CC* and CD* modules and data areas
CST $nnn$ .SYSF	NTLFILE, LFILE, DBID, and FNR	Natural Construct data file

## 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 $nnn$ .INPE	FNAT/SYSTEM	CD* programs, help routines, subprograms, and maps
CST $nnn$ .SYSF	NTLFILE, LFILE, DBID, and FNR	Natural Construct data file

## SMA Tables

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

Dataset Name	Job Name	Step Name
CST $nnn$ .SYSF	I050	1300
CST $nnn$ .INPL	I061	1300
CST $nnn$ .INPC	I061	1301
CST $nnn$ .INPE	I061	1302
CST $nnn$ .CPRD	I061	1306
CST $nnn$ .VINP	I061	1307
CST $nnn$ .DINP	I061	1308
CST $nnn$ .DA4	I200	1301
CST $nnn$ .VD4	I200	1302
CST $nnn$ .DD4	I200	1303
CST $nnn$ .SYSH	I500	1302
CST $nnn$ .SYSM	I500	1303
CST $nnn$ .SYSR	I500	1304
CST $nnn$ .FUPD	I500	1312

## 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.
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.

# Install Natural Construct Over an Existing Version

This section describes how to upgrade to Natural Construct V5.3 from a previous version. 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. The `CST $nnn$ .JOBS` dataset contains sample JCL.

**Note:**

For information on installing catalog and runtime environments, see *Install Natural Construct Catalog and Runtime Environments*.

This section covers the following topics:

- Installation Prerequisites
- Step 1: Create Backup of SYSCST
- Step 2: Load Updated Modules
- Step 3: Install DB2 or VSAM Demo Application (Optional)
- Step 4: Load Updated Predict File Definitions (Optional)
- Step 5: Load Updated Help Text
- Step 6: Load Updated Models
- Step 7: Load Updated Code Frames
- Step 8: Set Natural Construct System File Version
- Step 9: Copy Natural Utility Subprograms
- Step 10: Install in Upper Case Only (Optional)
- Step 11: Translate Error Messages (Optional)
- Step 12: Install in Static (One-Language) Mode (Optional)

## Installation Prerequisites

Before beginning the upgrade procedure, ensure the following prerequisites have been met (some prerequisites may have been fulfilled during earlier installations):

- All code frames supplied as fixes to previous versions of Natural Construct are deleted (code frames with an “8” suffix). For example, COPA9 is an original code frame installed with the previous version, while COPA8 was delivered later as a fix. To determine which code frames to delete, list the existing code frames in the SYSCST library ("MENU F L") and scan for suffixes of “8”.

**Important:**

If you used “8” as the suffix in the name of a customized code frame, change the “8” to a “7” or less before deleting the code frames.

- The S8 and S9 superdescriptors are released from the Natural Construct system file.
- The File Definition Table (FDT) for the Natural Construct system file contains the following fields and values:

Field Name	Format and Length	Option
ED	A8	Null
EG	A8	Null
EH	A8	Null
FE	A8	Null
BX	A253	Null

If these fields or values are not correct, use the Adabas utilities to add missing fields or modify the field values or options. For more information, refer to the Adabas utilities documentation.

**Tip:**

As the physical placement of the BX field within the FDT can have a small effect on performance, ensure that the BX field is placed after the BM field. Note that the BX field does not have to follow immediately after the BM field (for example, the placement of the field should not interfere with the fields in a PE group).

## Step 1: Create Backup of SYSCST

- Job I051, Step 1300

**Note:**

Before beginning this procedure, ensure all installation prerequisites have been met. For information, see Installation Prerequisites.

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 that 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, yy, 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: Load Updated Modules

### To load the applicable dataset:

1. Invoke the Natural INPL utility.
2. Set work file 1 to the dataset name.
3. Enter the following:

```
INPL
B
```

This section covers the following topics:

- 2A: Load Updated Natural Construct Modules
- 2B: Load Updated Predict Interface Modules

### 2A: Load Updated Natural Construct Modules

- Job I061, Step 1300; *CSTnnn.INPL* dataset

The *CSTnnn.INPL* 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.

### 2B: Load Updated Predict Interface Modules

- Job I061, Step 1306; *CSTnnn.CPRD*

The *CSTnnn.CPRD* dataset contains the updated Predict V4.5 interface modules.

## Step 3: Install DB2 or VSAM Demo Application (Optional)

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. Invoke the Natural INPL utility.
2. Set work file 1 to CST*nnn*.DINP.
3. Enter the following:

```
INPL  
B
```

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

1. Log onto the SYSCSTD2 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. Invoke the Natural INPL utility.
2. Set work file 1 to *CSTnnn.VINP*.
3. Enter the following:

```
INPL
B
```

▶ **To copy the data used by the VSAM demo application:**

1. Log onto the SYSCSTDV library.
2. Catalog and run the CSGMIGAV program.

## Step 4: Load Updated Predict File Definitions (Optional)

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

▶ **To load the Natural Construct file layout into Predict:**

1. Invoke Natural in batch mode with work file 1 assigned to the *CSTnnn.XXx* dataset, where:
  - *XX* corresponds to the two-letter code for your environment
  - *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 META ALL,REPLACE=YES
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 5: Load Updated Help Text

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

Use the CSHLOAD batch utility to load the new help text into the Natural Construct system file (for information, see *Utilities, Natural Construct 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 6: Load Updated Models

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

Use the CSMLOAD batch utility to load the models into the Natural Construct system file (for information, see *Multiple Model Import Utility, Natural Construct Generation*).

### 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 7: Load Updated Code Frames

- Job I500, Step 1312; CST $nnn$ .FUPD dataset

### Note:

This dataset is only supplied when there are updates or fixes for the CST $nnn$ .SYSR dataset.

Use the CSFLOAD batch utility to load the updated code frames into the Natural Construct system file (for information, see Multiple Code Frame Import Utility).

### To load the updated code frames:

1. Assign the CST $nnn$ .FUPD dataset to work file 1.
2. Log onto the SYSCST library.
3. Specify the following input for the CSFLOAD utility:

\* ,Y

After loading the code frames, use the Compare Code Frame utility to review any suffixes lower than “8”. These suffix numbers indicate customized code frames. Compare the old code frames with the new frames to determine which customizations have been made and may no longer be valid. If you made extensive model changes, you may want to apply the changes manually.

### Note:

You must delete all code frames supplied as fixes to previous versions of Natural Construct (code frames using an “8” suffix). For more information, see Installation Prerequisites.

The installation tape contains the CST $nnn$ .C521 dataset, which compares the new code frames with those in Natural Construct V5.2.

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

## Step 8: 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 Natural Construct 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:

1. Log onto the SYSCST library.

2. Run the CVSETVER program.

## Step 9: Copy Natural Utility Subprograms

Some generated modules use utility subprograms installed with Natural. The method used to copy these USR routines differs, based on whether you are installing Natural Business Services or installing Natural Construct alone.

### Tip:

To produce a report listing the modules that have been copied by the CVUSRCOP utility, enter "CVUSRCOP y" when invoking the utility from the client.

### Natural Business Services

- Job I200, Step 1371

To configure Natural Business Services, run the CVSPEVR conversion utility in batch mode. This utility:

1. Sets the system file version to NBS53.
2. Invokes the CVUSRCOP utility to copy USR modules from the SYSEXT library to the SYSBIZ library.

### When Using Natural Security

#### Note:

Before installing Natural Business Services in a secure environment, ensure that you have applied the latest hot fixes for Natural Security.

If you are using Natural Security, the following errors may occur while running the CVUSRCOP utility:

- 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.

- Error indicating that the CVUSRCOP utility cannot copy the USR1051N routine

This error occurs if Predict is not installed. The USR1051N routine is installed in the SYSEXT library during the Predict installation procedure.

- NAT4889 error: SYSMAIN 4889 : Library is not defined to Natural Security

If you use the Natural Security Administration option to define all Natural system libraries, ensure that the SYSLIBS library is defined as public. (By default, SYSLIBS is protected.)

### Natural Construct

- Job I500, Step 1301

If you are installing Natural Construct without Natural Business Services, run the CVUSRCOP utility to copy the following subprograms:

<b>Subprogram</b>	<b>From Library</b>	<b>To Library</b>
MAPBOXCA	SYSMAP	SYSLIBS
MAPBOXCT	SYSMAP	SYSLIBS
MAPBOXCV	SYSMAP	SYSLIBS
MAPBOXEX	SYSMAP	SYSLIBS
MAPBOXIN	SYSMAP	SYSLIBS
MAPBOXW1	SYSMAP	SYSLIBS
USR0050N	SYSEXT	SYSLIBS SYSTEM
USR0120N	SYSEXT	SYSLIBS SYSTEM
USR0320N	SYSEXT	SYSCST SYSLIBS SYSTEM
USR0360N	SYSEXT	SYSCST SYSLIBS SYSTEM
USR0622N	SYSEXT	SYSLIBS
USR1002N	SYSEXT	SYSLIBS SYSTEM
USR1005N	SYSEXT	SYSLIBS SYSTEM
USR1009N	SYSEXT	SYSLIBS
USR1025N	SYSEXT	SYSLIBS
USR1031N	SYSEXT	SYSLIBS
USR1038N	SYSEXT	SYSLIBS
USR1051N	SYSEXT	SYSLIBS SYSTEM
USR1057N	SYSEXT	SYSLIBS
USR3013N	SYSEXT	SYSLIBS

Subprogram	From Library	To Library
USR4011N	SYSEXT	SYSLIBS SYSTEM

### When Using Natural Security

If you are using Natural Security with Natural Construct, see When Using Natural Security for troubleshooting information.

## Step 10: Install in Upper Case Only (Optional)

If you are installing in an environment that does not support lower case Latin characters, run the CVUPPERC utility in the SYSCST library to translate the components into upper case. 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:

1. Converts all data in the Natural Construct system file to upper case (CVLO2HIA).
2. Converts all messages to upper case (CVLO2HIM).

#### Note:

By default, the CVUPPERC utility converts all English messages to upper case. To translate other languages to upper case, either re-run the CVUPPERC utility using a different language code or use the CVLO2HIM utility and define the language code, error message numbers, and application/library IDs for the desired language. If you use the CVLO2HIM utility, only the messages in the following Natural Construct libraries are converted: SYSTEM, CSTAPPL, CSTACT, CSTPFK, CSTMSG, CSTLDA, SYSCUA, SYSSPEC, SYSLIB, and SYSCST.

3. Converts all supplied source to upper case (CVLO2HIS).
4. Performs a CATAL 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 11: Translate Error Messages (Optional)

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 12: Install in Static (One-Language) Mode (Optional)

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 information, see Static (One-Language) Mode.

# Install 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<sub>nnn</sub>.INPL* dataset installs a complete administration, development, generation, and execution environment. The *CST<sub>nnn</sub>.JOBS* dataset contains sample JCL.

### Note:

For information on installing catalog and runtime environments, see Install Natural Construct Catalog and Runtime Environments.

This section covers the following topics:

- Step 1: Establish Natural Construct Data File
- Step 2: Load Required Modules
- Step 3: Copy Natural Utility Subprograms
- Step 4: Install DB2 or VSAM Demo Application (Optional)
- Step 5: Load Predict File Definitions (Optional)
- Step 6: Install in Upper Case Only (Optional)
- Step 7: Translate Error Messages (Optional)
- Step 8: Install in Static (One-Language) Mode (Optional)

## Step 1: Establish Natural Construct Data File

- Job I050, Step 1300; *CST<sub>nnn</sub>.SYSF* dataset

The *CSTnnn.SYSF* dataset is an unloaded Adabas V7 file and is input to the Adabas ADALOD utility, which loads the Natural Construct data file. The DBID and FNR values used to load the file must be the same as those specified in the NTLFILE 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 that you set the following minimum sizes for the ADALOD utility:

ADALOD Parameter	Minimum Size Required
MAXISN	20000
DSSIZE	5
NISIZE	300B
UISIZE	50B

## Step 2: Load Required Modules

 To load the required modules and error messages:

1. Invoke the Natural INPL utility.
2. Set work file 1 to the dataset name.
3. Enter the following:

```
INPL
B
```

This section describes the following procedures:

- 2A: Load Natural Construct Modules
- 2B: Load Predict Interface Modules

### 2A: Load Natural Construct Modules

- Job I061, Step 1300; *CSTnnn.INPL* dataset

Load the modules from the *CSTnnn.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.

### 2B: Load Predict Interface Modules

- Job I061, Step 1306; *CSTnnn.CPRD*

Load the modules from the *CSTnnn.CPRD* dataset into the appropriate system files. This dataset contains the Predict V4.5 interface modules.



## Step 3: Copy Natural Utility Subprograms

Some generated modules use utility subprograms installed with Natural. The method used to copy these USR routines differs, based on whether you are installing Natural Business Services or Natural Construct alone.

### Tip:

To produce a report listing the modules that have been copied by the CVUSRCOP utility, enter "CVUSRCOP y" when invoking the utility from the client.

### Natural Business Services

- Job I200, Step 1371

To configure Natural Business Services, run the CVSPEVR conversion utility in batch mode. This utility:

1. Sets the system file version to NBS53.
2. Invokes the CVUSRCOP utility to copy USR modules from the SYSEXT library to the SYSBIZ library.

### When Using Natural Security

#### Note:

Before installing Natural Business Services in a secure environment, ensure that you have applied the latest hot fixes for Natural Security.

If you are using Natural Security, the following errors may occur while running the CVUSRCOP utility:

- 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.

- Error indicating that the CVUSRCOP utility cannot copy the USR1051N routine

This error occurs if Predict is not installed. The USR1051N routine is installed in the SYSEXT library during the Predict installation procedure.

- NAT4889 error: SYSMAIN 4889 : Library is not defined to Natural Security

If you use the Natural Security Administration option to define all Natural system libraries, ensure that the SYSLIBS library is defined as public. (By default, SYSLIBS is protected.)

### Natural Construct

- Job I500, Step 1301

If you are installing Natural Construct without Natural Business Services, run the CVUSRCOP utility to copy the following subprograms:

<b>Subprogram</b>	<b>From Library</b>	<b>To Library</b>
MAPBOXCA	SYSMAP	SYSLIBS
MAPBOXCT	SYSMAP	SYSLIBS
MAPBOXCV	SYSMAP	SYSLIBS
MAPBOXEX	SYSMAP	SYSLIBS
MAPBOXIN	SYSMAP	SYSLIBS
MAPBOXW1	SYSMAP	SYSLIBS
USR0050N	SYSEXT	SYSLIBS SYSTEM
USR0120N	SYSEXT	SYSLIBS SYSTEM
USR0320N	SYSEXT	SYSCST SYSLIBS SYSTEM
USR0360N	SYSEXT	SYSCST SYSLIBS SYSTEM
USR0622N	SYSEXT	SYSLIBS
USR1002N	SYSEXT	SYSLIBS SYSTEM
USR1005N	SYSEXT	SYSLIBS SYSTEM
USR1009N	SYSEXT	SYSLIBS
USR1025N	SYSEXT	SYSLIBS
USR1031N	SYSEXT	SYSLIBS
USR1038N	SYSEXT	SYSLIBS
USR1051N	SYSEXT	SYSLIBS SYSTEM
USR1057N	SYSEXT	SYSLIBS
USR3013N	SYSEXT	SYSLIBS
USR4011N	SYSEXT	SYSLIBS SYSTEM

## When Using Natural Security

If you are using Natural Security with Natural Construct, see *When Using Natural Security* for troubleshooting information.

## Step 4: Install DB2 or VSAM Demo Application (Optional)

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.

This section covers the following topics:

- Install DB2 Demo Application
- Install VSAM Demo Application

### Install DB2 Demo Application

- Job I061, Step 1308; *CSTnnn.DINP*

The *CSTnnn.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. Invoke the Natural INPL utility.
2. Set work file 1 to *CSTnnn.DINP*.
3. Enter the following:

```
INPL
B
```

▶ **To copy the data used by the DB2 demo application:**

1. Log onto the SYSCSTD2 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. Invoke the Natural INPL utility.
2. Set work file 1 to CST $nnn$ .VINP.
3. Enter the following:

```
INPL  
B
```

▶ **To copy the data used by the VSAM demo application:**

1. Log onto the SYSCSTDV library.
2. Catalog and run the CSGMIGAV program.

## **Step 5: Load Predict File Definitions (Optional)**

Optionally, you can load the Predict file definitions for the demo application.

▶ **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

- $x$  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 META ALL,REPLACE=YES
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 6: Install in Upper Case Only (Optional)

If you are installing in an environment that does not support lower case Latin characters, you can run the CVUPPERC utility to translate the product into upper case. 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:

1. Converts all data in the Natural Construct system file to upper case (CVLO2HIA).
2. Converts all messages to upper case (CVLO2HIM).

**Note:**

By default, the CVUPPERC utility converts all English messages to upper case. To translate other languages to upper case, either re-run the CVUPPERC utility using a different language code or use the CVLO2HIM utility and define the language code, error message numbers, and application/library IDs for the desired language. If you use the CVLO2HIM utility, only the messages in the following Natural Construct libraries are converted: SYSTEM, CSTAPPL, CSTACT, CSTPFK, CSTMSG, CSTLDA, SYSCUA, SYSSPEC, SYSLIB, and SYSCST.

3. Converts all supplied source to upper case (CVLO2HIS).
4. 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 7: Translate Error Messages (Optional)

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 8: Install in Static (One-Language) Mode (Optional)

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 information, see Static (One-Language) Mode.

# Install Natural Construct 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.

**Note:**

The *CSTnnn.JOBS* dataset contains sample JCL.

To install catalog and runtime or runtime-only environments:

- Step 1: Establish Natural Construct Data File
- Step 2: Load Updated Natural Construct Modules
- Step 3: Copy Natural Utility Subprograms

## Step 1: Establish Natural Construct Data File

- Job I050, Step 1300; *CSTnnn.SYSF* dataset

If your applications use help text in the Natural Construct help subsystem, 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 5: Load Updated Help Text.

## 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; *CSTnnn.INPC* dataset

The *CSTnnn.INPC* dataset is a subset of the *CSTnnn.INPL* dataset and contains the modules required for a catalog and runtime environment.

 **To load the updated Natural Construct modules:**

1. Invoke the Natural INPL utility.
2. Set work file 1 to *CSTnnn.INPC*.

3. Enter the following:

```
INPL
B
```

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 a Runtime-Only Environment

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

The CST*nnn*.INPE 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.

#### To load the updated Natural Construct modules:

1. Invoke the Natural INPL utility.
2. Set work file 1 to CST*nnn*.INPE.
3. Enter the following:

```
INPL
B
```

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.

## Step 3: Copy Natural Utility Subprograms

Some generated modules use utility subprograms installed with Natural. The method used to copy these USR routines differs, based on whether you are installing Natural Business Services or Natural Construct alone.

### Tip:

To produce a report listing the modules that have been copied by the CVUSRCOP utility, enter "CVUSRCOP y" when invoking the utility from the client.

### Natural Business Services

- Job I200, Step 1371

To configure Natural Business Services, run the CVSPEVR conversion utility in batch mode. This utility:

1. Sets the system file version to NBS53.
2. Invokes the CVUSRCOP utility to copy USR modules from the SYSEXT library to the SYSBIZ library.



## When Using Natural Security

### Note:

Before installing Natural Business Services in a secure environment, ensure that you have applied the latest hot fixes for Natural Security.

If you are using Natural Security, the following errors may occur while running the CVUSRCOP utility:

- 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.

- Error indicating that the CVUSRCOP utility cannot copy the USR1051N routine

This error occurs if Predict is not installed. The USR1051N routine is installed in the SYSEXT library during the Predict installation procedure.

- NAT4889 error: SYSMAIN 4889 : Library is not defined to Natural Security

If you use the Natural Security Administration option to define all Natural system libraries, ensure that the SYSLIBS library is defined as public. (By default, SYSLIBS is protected.)

## Natural Construct

- Job I500, Step 1301

If you are installing Natural Construct without Natural Business Services, run the CVUSRCOP utility 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
USR0050N	SYSEXT	SYSLIBS SYSTEM
USR0120N	SYSEXT	SYSLIBS SYSTEM

Subprogram	From Library	To Library
USR0320N	SYSEXT	SYSCST SYSLIBS SYSTEM
USR0360N	SYSEXT	SYSCST SYSLIBS SYSTEM
USR0622N	SYSEXT	SYSLIBS
USR1002N	SYSEXT	SYSLIBS SYSTEM
USR1005N	SYSEXT	SYSLIBS SYSTEM
USR1009N	SYSEXT	SYSLIBS
USR1025N	SYSEXT	SYSLIBS
USR1031N	SYSEXT	SYSLIBS
USR1038N	SYSEXT	SYSLIBS
USR1051N	SYSEXT	SYSLIBS SYSTEM
USR1057N	SYSEXT	SYSLIBS
USR3013N	SYSEXT	SYSLIBS
USR4011N	SYSEXT	SYSLIBS SYSTEM

### When Using Natural Security

If you are using Natural Security with Natural Construct, see [When Using Natural Security](#) for troubleshooting information.

## Install 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 to improve performance and reduce database calls. For information, see [Static \(One-Language\) Mode](#).

**Note:**

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

## Activate 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.

## Verify the Installation of Natural Construct

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

The following topics are covered:

- Regenerate and Recatalog the Demo System
- Access the Demo System
- Access the Help Text Subsystem
- Access the Administration and Modeling Subsystem

### Regenerate and Recatalog the Demo System

 **To regenerate and recatalog the demo system:**

1. Log onto the demo system as follows:
  - "SYSCSTDE" for Adabas
  - "SYSCSTD2" for DB2
  - "SYSCSTDV" for VSAM
2. Enter "ncstbgen" at the Next prompt.

The Multiple Generation panel is displayed:

```

CSBINOL          ***** Natural Construct *****          CSBM0
Mar 09              Multiple Generation              1 of 1

Module
  _____  _____  _____  _____  _____
  _____  _____  _____  _____  _____

Model
  _____  _____
  _____  _____
  _____  _____
  _____  _____

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

```

3. Type "\*" in the first Module field.
4. Type "x" in Catalog regenerated modules.
5. Press Enter.

The utility regenerates all modules and produces a report listing the names of the modules it generated and information about modules that were not generated. For example:

```

Skipped          AWHK      is a PDA. Note: Browse PDAs regenerate by Obj
                 AWHP      is a PDA. Note: Browse PDAs regenerate by Obj
                 BCUSTS20 is a PDA. Note: Browse PDAs regenerate by Obj
                 BORDNS20 is a PDA. Note: Browse PDAs regenerate by Obj
                 CALC      Module CALC was not generated by Construct
                 FLIPSTR  Module FLIPSTR was not generated by Construct
                 FLIPSTRA is a PDA. Note: Browse PDAs regenerate by Obj
                 GCDA      is a PDA. Note: Browse PDAs regenerate by Obj
                 GCDN      Module GCDN was not generated by Construct
                 MODCV     Module MODCV was not generated by Construct
                 MYMAP     is a map; Construct does not re-generate maps
                 NCCFM11  is a map; Construct does not re-generate maps
                 NCCFM21  is a map; Construct does not re-generate maps
                 NCCMAI11 is a map; Construct does not re-generate maps
                 NCCMENU   is a map; Construct does not re-generate maps
                 NCGDA     Module NCGDA was not generated by Construct
                 NCLAYMN1 is a map; Construct does not re-generate maps
                 NCLAYOUT  is a map; Construct does not re-generate maps

```

**Tip:**

If the modules cannot be regenerated or recataloged, check the Natural parameters.

**Tip:**

If a NAT0082 error occurs, one or more modules are not loaded. Check the INPL jobs to determine which modules are missing.

## Access the Demo System

### ▶ To access 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.

## Access the Help Text Subsystem

### ▶ To access 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.

## Access the Administration and Modeling Subsystem

### ▶ To access 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.

## Maintain Natural Construct

This section describes the operational requirements to maintain and use Natural Construct. The following topics are covered:

- Access the Natural Construct Data File
- Access the Model Maintenance Facilities

- Implement Security Using Sample Exit Subprograms
- Maintain the Natural Construct Generation Environment
- Maintain the Natural Construct Runtime Environment

## Access 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 NTLFILE definition for the Natural Construct data file (for more information, see Step 1: Establish Natural Construct Data File).


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
- *DBID* identifies the database ID of the Natural Construct data file
- *FNR* identifies the file number of the Natural Construct data file

 **To create an environment that does not require the NTLFILE macro or LFILE dynamic parameter:**

1. Recatalog the following DDMs:
  - NCST-HELP
  - NCST-HELP-LINES
  - NCST-PROFILE
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 NTLFILE macro or LFILE dynamic parameter.

## Access 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 use Natural Security to restrict access to the maintenance facilities.

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

## Implement Security Using Sample Exit Subprograms

Natural Construct supplies sample 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 sample 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 sample exit subprograms, the panels they secure, and the libraries they are invoked from:

Subprogram	Panel Secured	Library Invoked From
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 *CSTnnn.INPL* dataset contains examples of the sample 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 sample exit subprogram.

## Maintain the 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.

The following topics are covered:

- FNAT System File
- Control Access to the Generation Environment
- Required Modules

- Predict Support

## 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.

## Control Access to the Generation Environment

You can use Natural Security to control access to the Generation and Help Text maintenance facilities.

- 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.

## Predict Support

To use the generation facilities, Predict must be available in the generation environment and the Natural transaction must use Predict V4.5 or higher.

## Maintain the 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



## Steplib Support

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 help routines and subprograms. The CC\* modules contain shared copy code members.