Supplied Administration Utilities

This section describes the utilities supplied with Natural Construct for use in the Administration subsystem. The following topics are covered:

- Introduction
- Import and Export Utilities
- Frame Hardcopy Utility
- Comparison Utilities
- Upper Case Translation Utility
- Additional Utilities

Introduction

This section describes the utilities supplied with Natural Construct for all supported platforms. To invoke a utility, enter its name at the Next prompt (Direct Command box for Unix).

Note:
When a description refers to “your print file” for mainframe users, it refers to Print File 1. When a description refers to “your print file” for Unix users, it refers to DEVICE LPT1.

Import and Export Utilities

This section explains how to transfer data across dissimilar platforms (for example, from Unix to mainframe).

Natural Construct’s import and export utilities read and write their data from and to work file 1. This is true for each of the following utilities:

Utility	Described in
CSFLOAD CSFUNLD	Multiple Code Frame Import Utility Multiple Code Frame Export Utility
CSHLOAD CSHUNLD	*Natural Construct Help Text*
CSMLOAD CSMUNLD	*Natural Construct Generation*

A work file written on one platform (such as Unix) can be read by another platform (such as mainframe) if the following conditions are met:

- The work file must be an ASCII file. For example:

Platform	**How to save as an ASCII file**
Mainframe	Define work file 1 as a PC file and activate PC Connection before running the utility. (PC Connection translates from EBCDIC to ASCII.)
Unix	Set the work file specification in your NATPARM to any extension other than “SAG”.

- When transferring the work file between platforms, select the appropriate translator. For example, the file transfer method you select to move a file from a PC to a Unix machine must translate the PC’s CR/LFs to CRs.

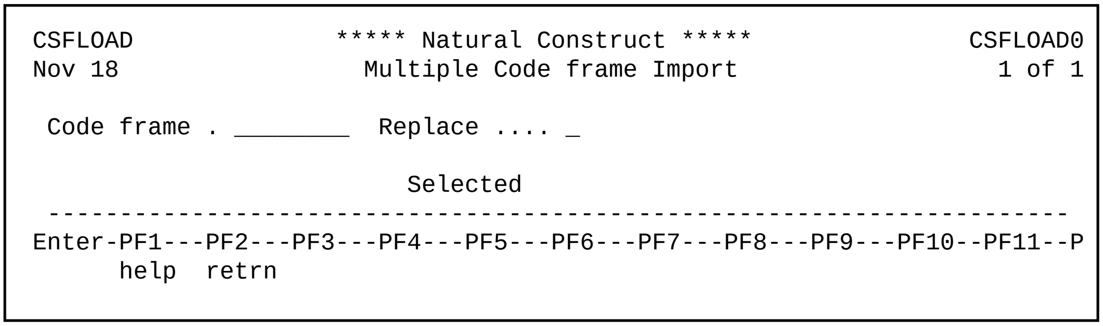
Multiple Code Frame Import Utility

The CSFLOAD frame import utility imports selected code frames from work file 1 to the code frame file. A report of the imported code frames is written to your print file.

CSFLOAD accepts up to 100 frame names and replace options in the form:

```
Code frame . ________  Replace .... _
```

The following example shows the CSFLOAD window:



```
CSFLOAD              ***** Natural Construct *****              CSFLOAD0
Nov 18                 Multiple Code frame Import                 1 of 1

 Code frame . ________  Replace .... _

                         Selected
 -----------------------------------------------------------------------
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--P
      help  retrn
```

Note:
To replace the existing code frames with code frames with the same names in work file 1, mark the Replace field. If you do not want to replace the existing code frames, leave the Replace field blank.

Examples of Input Values

Values entered	Result
Code frame: *	Imports all code frames from work file 1. If a code frame with the same name exists in the code frame file, it is not replaced.
Code frame: MENU	Imports the “MENU” code frame from work file 1. If the “MENU” code frame exists in the code frame file, it is not replaced.
Code frame: MENU Replace: X	Imports the “MENU” code frame from work file 1. If the “MENU” code frame exists in the code frame file, it is replaced.
Code frame: FM*	Imports all code frames beginning with “FM” from work file 1. If a code frame with the same name exists in the code frame file, it is not replaced.
Code frame: . (period)	Terminates the CSFLOAD utility. **Note:** When running in batch mode, the CSFLOAD utility will terminate with RC=0 if an error occurs due to problems with the internal layout structure of work file 1. To terminate the batch Natural session with RC=99, add “Y” to the end of the last frame input combination (for example: FM*,,Y).

Multiple Code Frame Export Utility

The CSFUNLD frame export utility exports selected code frames from the code frame file to work file 1. A report of the exported code frames is written to your print file.

Note:
You can export a maximum of 1000 code frames at one time.

Enter each code frame name one name at a time. As you enter the names, they are automatically displayed on the panel.

For each exported code frame, you can specify whether to export its recursive (nested) code frames — if any exist. To export recursive code frames, mark the Include recursive code frames field. If you do not want to export recursive code frames, leave the field blank.

Examples of Input Values

Values entered	Result
*	Exports all code frames to work file 1.
MENU X	Exports the “MENU” code frame including any recursive (nested) code frames to work file 1.
FM*	Exports all code frames beginning with “FM” to work file 1.

Enter a period (.) to terminate the input.

Frame Hardcopy Utility

The CSFHCOPY frame hardcopy utility allows you to print a hardcopy list of your code frames, regardless of your teleprocessing monitor. All output is routed to your print file.

Enter each code frame name one name at a time. As you enter the names, they are automatically displayed on the panel.

Examples of Input Values

Values entered	Result
*	Routes all code frames to your print file.
MENU	Routes the “MENU” code frame to your print file.
FM*	Routes all code frames beginning with “FM” to your print file.

Enter a period (.) to terminate the input.

Comparison Utilities

This section describes utilities you can use to compare two Natural source modules and to compare a range of models in different libraries.

CSGCMPS Utility

This program compares two Natural source modules. You can compare the contents of two saved modules or you can compare the contents of the module currently in the source buffer to the contents of a saved module.

Specify the library ID, module name, database ID, and file number for each module you want to compare. In addition, you can specify the following options:

- ignore comment lines
- ignore trailing comments
- ignore leading spaces
- provide summary only

When you invoke the CSGCMPS utility online, the following window is displayed:

```
                              Compare Criteria
                       Library  Object   Database File or Source Area
                       ======== ======== ======== ====    ===========
      Old version ===> CST411M_ CSGCMPS_   017    029          _
      New version ===> CST411M_ CSGCMPS_   017    029          _

      Options...
         Ignore comment lines...... _
         Ignore trailing comments.. _
         Ignore leading spaces..... _
         Summary only.............. _
      Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF1
                        quit
```

CSGCMPL Utility

This program compares a range of modules in one library to the same modules in another library.

Specify the library ID, database ID, file number, and range value for the modules you want to compare. In addition, you can specify the following options:

- summary only
- only report if different
- ignore comment lines
- ignore trailing comments
- ignore leading spaces
- only compare object types

Online

When you invoke the CSGCMPL utility online, the following window is displayed:

```
                       Source Range Compare Facility
                                   Library  Database File Dominant
                                   ======== ======== ==== ========
        Old library............... ________   __17   __29     X
        New Library............... ________   __17   __29     _
        Program range............. ________ thru ________
        Summary only.............. _
        Only report if different.. _
        Ignore comment lines...... _
        Ignore trailing comments.. _
        Ignore leading spaces..... _
        Only compare object types  __________ (ACGHLMNPST)
```

The Dominant column indicates the range of modules to be compared. Only modules that exist in the dominant library and in the other specified library are included in the compare results. Modules that only exist in the non-dominant library are not included.

The Only compare object types field limits the comparison to modules of a specified object type. Valid object types are:

Object Type	Description
A	Parameter
C	Copycode
G	Global data area
H	Helproutine
L	Local data area
M	Map
N	Subprogram
P	Parameter data area
S	Subroutine
T	Text

In Batch

Batch mode is the most efficient method of comparing many modules. The following SYSIN shows an example of using this utility in batch:

```
LOGON CST421M
CSGCMPL OLD-LIB,001,002,X,NEW-LIB,003,004, ,BEGIN,END,S,D,C,T,L,NPH
FIN
```

where:

OLD-LIB	Indicates the name of a library containing modules to be compared.
001	Indicates the database ID for OLD-LIB.
002	Indicates the system file number for OLD-LIB.
X	Indicates that the OLD-LIB is dominant (all modules in the dominant library are compared to matching modules in the other specified library).
NEW-LIB	Indicates the name of a library containing modules to be compared.
003	Indicates the database ID for NEW-LIB.
004	Indicates the system file number for NEW-LIB.
blank	If blank, indicates that OLD-LIB is dominant. If X, indicates that NEW-LIB is dominant.
BEGIN	Indicates the name of the first module in the range compared.
END	Indicates the name of the last module in the range compared.
S	Indicates a summary report (does not display detailed differences). This option displays the names of the modules and whether the module contents are the same in both libraries.
D	Indicates that only modules that are different are included on the output report. Modules that are identical in both libraries are not included.
C	Indicates that Natural comment lines (lines beginning with “*” or “/*”) are not compared.
T	Indicates that trailing comments (comments beginning with “/*”) are not compared.
L	Indicates that leading spaces are not compared (changes in alignment will not show up as differences).
NPH	Indicates the list of Natural object types compared within the specified range of modules.

Upper Case Translation Utility

If you are developing applications in a language that does not support lower case Latin characters, use the supplied CVUPPERC utility to convert the Natural Construct components to upper case. This utility converts all Natural Construct-installed SYSERR message text and source code, as well as the contents of the Natural Construct system file, to upper case.

Notes:

1. Since this conversion requires a significant amount of processing, only run this utility in a batch environment.
2. Before running this utility, ensure that the batch job defines the correct Natural Construct logical file, FUSER system file, and FNAT system file.

Use the following SYSIN to invoke the CVUPPERC utility:

```
LOGON SYSCST
CVUPPERC
FIN
```

After converting the components to upper case, this utility issues a CATALL in the SYSCST library. To reflect the changes in your production environment, manually transfer all modules from the SYSCST library to the SYSLIBS library after the modules have been cataloged.

Additional Utilities

The utilities in this section generate cross-reference information for all subprograms referenced by a code frame or model. You can use these utilities either online (recommended) or in batch mode to determine which subprograms are invoked. If subprograms are missing, the utility will write a report to the screen.

Determine Which Subprograms Are Referenced by Code Frames

To determine which subprograms are referenced by code frames, run the CVUVALF utility either online or in batch mode. The CVUVALF utility:

- Verifies that all subprograms referenced by all code frames exist in the current library.
- Generates CALLNAT statements for all subprograms used in the code frames. Online, these statements are generated into the program editor and you can view the parameter list.

Online

▶ **To invoke the CVUVALF utility online:**

1. Logon to the SYSCST library.
2. Run the CVUVALF program.

In Batch

▶ **To invoke the CVUVALF utility in batch mode:**

- Use the following SYSIN:

  ```
  LOGON SYSCST
  CVUVALF
  FIN
  ```

  In this example, the utility provides a list of all subprograms that are referenced by all code frames and do not exist in the SYSCST library.

Determine Which Subprograms Are Referenced by Models

To determine which subprograms are referenced by models, run the CVUVALM utility either online or in batch mode. The CVUVALM utility:

- Verifies that all subprograms referenced by all models exist in the current library.

- Generates CALLNAT statements for all subprograms used in the models. Online, these statements are generated into the program editor and you can view the parameter list.

Online

▶ **To invoke the CVUVALM utility online:**

1. Logon to the SYSCST library.
2. Run the CVUVALM program.

In Batch

▶ **To invoke the CVUVALM utility in batch mode:**

- Use the following SYSIN:

  ```
  LOGON SYSCST
  CVUVALM
  FIN
  ```

  In this example, the utility provides a list of all subprograms that are referenced by all models and do not exist in the SYSCST library.