

Using the Administration Subsystem

This section describes how to use the Administration subsystem supplied with Natural Construct. Use the Administration main menu to access the functions available in the Administration subsystem.

This section covers the following topics:

- Access the Administration Main Menu
- Create and Maintain Natural Construct Models
- Multilingual Support for Natural Construct
- Access the Administration Main Menu in Translation Mode
- Access and Use the Sample Exit Subprograms

Access the Administration Main Menu

▶ To access the Administration main menu:

1. Enter "menu" at the Natural prompt.

The Administration main menu is displayed. For example:

```

CSDMAIN                      N a t u r a l   C o n s t r u c t          CSDMNM0
Aug 17                        Administration Main Menu          1 of 1

                               Functions
                               -----
M  Maintain Models
F  Code Frame Menu
S  Maintain Subprograms
R  Maintain Control Record
C  Compare Menu
D  Drivers Menu

                               ?  Help
                               .  Return
                               -----

Function ..... _

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help retn quit                                     main

```

Note:

For a description of the Help and Return functions, see Help and Return Codes on Menus.

2. Enter the corresponding one-character function code in Function.

The functions available through the Administration main menu are:

Code	Function	Description
M	Maintain Models	Displays the Maintain Models panel, where you can maintain the components that define a model for the Natural Construct generation process. For information, see Maintain Models Function.
F	Code Frame Menu	Displays the Code Frame menu. Using the functions available through this menu, you can maintain the code frames used by the generation models. For information, see Code Frame Menu Function.
S	Maintain Subprograms	Displays the Maintain Subprograms panel, where you can maintain the modify specification subprograms used by the generation models. For information, see Maintain Subprograms Function.
R	Maintain Control Record	Displays the Maintain Control Record panel, where you can maintain the default values for the Natural Construct control record (PF-keys, dynamic attribute characters, help indicator, etc.). For information, see Maintain Control Record Function.
C	Compare Menu	Displays the Compare menu. Using the functions available through this menu, you can compare code frames used by the models. For information, see Compare Menu Function.
D	Drivers Menu	Displays the Drivers menu. Using the driver programs available through this menu, you can access many of the utility subprograms supplied with Natural Construct. (The source code for these subprograms is not supplied.) For information, see Drivers Menu Function.

Create and Maintain Natural Construct Models

This section describes how to use the Administration main menu to define custom models and maintain the models Natural Construct uses to generate programs. The following topics are covered:

- Maintain Models Function
- Code Frame Menu Function
- Maintain Subprograms Function
- Maintain Control Record Function
- Compare Menu Function
- Drivers Menu Function

Maintain Models Function

 **To create or maintain a model:**

1. Enter "M" in Function on the Administration main menu.

The Maintain Models panel is displayed. For example:

```

CSDFM                      N a t u r a l   C o n s t r u c t                      CSDFM0
Aug 17                      M a i n t a i n   M o d e l s                      1 of 1

Action ..... __ A,B,C,D,M,N,P,R
Model ..... BROWSE_____
Description ..... *0200.1_____
                               BROWSE Program
PDA name ..... CUSCPDA_      Status window ..... Y
Programming mode ..... S_      Comment start indicator .. **_
Type ..... P Program      Comment end indicator .... __

Code frame(s) ..... CSCA?__  CSCB?__  CSCC?__  _____
Modify server specificatn CUSCMA__ CUSCMB__ CUSCMC__ CUSCMG__ _____

Modify client specificatn CUSCMA__ CUSCMB__ CUSCMC__ _____

Clear specification ..... CUSCC__      Post-generation ..... CUSCPS__
Read specification ..... CUSCR__      Save specification ..... CUSCS__
Pre-generation ..... CUSCPR__      Document specification ... CUS-D__

Command .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help retn quit frame                                main
    
```

2. Enter the action code in Action.

For example, if you are creating a new model, enter A (Add); if you are changing the settings for an existing model, enter M (Modify).

Note:

For a description of the actions available, press PF1 (help) when the cursor is in the Action field.

3. Use the following fields to define or modify the model settings:

Field	Description
Model	Name of the model you are creating or maintaining.
Description	<p>Brief description of the model or the SYSERR number that supplies the description. When a module is generated using the specified model, this description is displayed as the first heading on the panel.</p> <p>Because this description is part of the model user interface, you can use SYSERR numbers from the CSTLDA library to support dynamic translation. Within SYSERR, you can also specify substitution variables (instead of hardcoding the message). For example, SYSERR number *0200.1 corresponds to the English text, :1:Program. If you specify *0200.1 in this field for the Browse model, Natural Construct replaces :1: with the model name and the first panel heading becomes Browse Program. (The actual heading is displayed below this field.)</p> <p>For more information about dynamic translation, see Maintenance.</p>
PDA name	<p>Name of the parameter data area (PDA) for the model. This PDA is passed to the model subprograms to capture model specifications.</p> <p>For more information, see Step 1: Define the Scope of the Model.</p>

Field	Description
Status window	<p>Code that indicates whether the Status window is displayed when a module is generated.</p> <p>If the code is Y or T, you can press PF5 (optns) while generating the module to display the Status window, which contains information about the generation progress, save, and/or stow functions. You can also decide how the Status window is displayed. The following example uses symbols:</p> <pre data-bbox="539 520 852 632" style="border: 1px solid black; padding: 5px;"> <-- PREGEN CUMNGPR --> FRAME CUMN9 --> FRAME CU--B9 </pre> <p>The following example uses text:</p> <pre data-bbox="539 730 1140 842" style="border: 1px solid black; padding: 5px;"> Ending Pre-generation Subprogram CUMNGPR Starting Code Frame CUMN9 Starting Code Frame CU--B9 </pre> <ul style="list-style-type: none"> ● To display symbols, enter "Y". ● To display text, enter "T". ● If you do not want the window displayed, enter "N". <p>Note: If this field is blank, it defaults to N.</p>
Programming mode	<p>Mode for the resulting code. Valid codes are S (structured), SD (structured data), or R (reporting) mode. All supplied models use structured mode.</p>
Comment start indicator	<p>Set of characters that indicate the beginning of a comment line for the generated module. As required for Natural modules, the default value is **. You can change this value for other supported programming languages.</p>

Field	Description
Type	<p>Code for the type of module generated by this model. Valid module types are:</p> <ul style="list-style-type: none"> ● P (program) ● E (external; non-Natural) ● * (super model modules) ● N (subprogram) ● S (subroutine) ● H (helpoutine) ● M (map) ● L (local data area) ● A (parameter data area) ● G (global data area) ● J (JCL statements; mainframe) ● . (statement code block; .g) ● T (text) ● C (copycode) ● blank (determined when a module is generated using this model; model subprograms must assign the CU—PDA.#PDA-OBJECT-TYPE parameter)
Comment end indicator	<p>Set of special characters that indicate the end of a comment. For some programming languages, this set of characters is required to generate modules. For PL1, for example, the indicator is */.</p>

Field	Description
Code frame(s)	<p>Names of the code frames used to create the specified model (for information, see Naming Conventions for Code Frames). The code frames are listed in the sequence they are used during generation. You can specify a maximum of five code frame names for each model; you can only use existing code frames.</p> <p>In addition:</p> <ul style="list-style-type: none"> ● You can select a code frame and access the Code Frame editor from this panel. For information, see Select a Code Frame for Editing. ● You can use nested code frames. For information, see Nested Code Frames. <p>Note: Code frames that are used to generate maps and data areas can only have subprogram and comment lines.</p>
Modify server specificatn	Names of the subprograms executed when the Modify function is invoked by the Natural Construct nucleus for server platform generation. The subprograms are listed in execution sequence. To change the order of execution, change the order of these subprograms. You can specify a maximum of 10 subprograms.
Modify client specificatn	Names of the subprograms executed when the Modify function is invoked by the nucleus for client platform generation. The subprograms are listed in the sequence they are executed. To change the order of execution, change the order of these subprograms. You can specify a maximum of 10 subprograms.
Clear specification	Name of the subprogram executed when the Clear function is invoked by the nucleus. The Clear function is automatically invoked prior to the Read function or when a new model name is specified and the parameter data area (PDA) is different. It is typically used to set default values for the model.
Post-generation	Name of the subprogram executed when the Post-generation function is invoked by the nucleus. This subprogram applies post-generation changes to the generated program. It is typically used to perform model specification substitutions; it is not supported for models that cannot be regenerated.
Read specification	Name of the subprogram executed when the Read function is invoked by the nucleus. It is typically used to retrieve the specifications from a previously-generated module. It is not supported for models that cannot be regenerated.

Field	Description
Save specification	Name of the subprogram executed when the Save function is invoked by the nucleus (not supported for models that cannot be regenerated). This subprogram is executed immediately after the pre-generation subprogram is executed. It writes the generation specifications so the generated program can be read using the Read function. If a user marks the Save Specification Only option, this subprogram can be invoked even if generation cannot be completed due to specification errors.
Pre-generation	Name of the subprogram executed when the Pre-generation function is invoked by the nucleus. This subprogram sets up internal variables before the generation process begins. It is typically used to set PDAC-variables for code frame manipulation or to generate a module for simple models.
Document specification	Name of the subprogram executed when the Document function is invoked by the nucleus. This subprogram documents generated modules in Predict as they are saved or stowed.

Select a Code Frame for Editing

You can use the Maintain Models panel to select a code frame for editing.

To select a code frame for editing:

1. Move the cursor over the code frame you want to edit.
2. Press PF4 (frame).

The specified code frame is displayed in the Code Frame editor.

Note:

For more information about modifying the supplied code frames, see Step 5: Create Code Frame(s) and Define the Model.

Naming Conventions for Code Frames

The following example shows the Maintain Models panel for the Browse model:


```

CSDFM                      N a t u r a l   C o n s t r u c t          CSDFM0
Aug 17                      M a i n t a i n   M o d e l s              1 of 1

Action ..... ___ A,B,C,D,M,N,P,R
Model ..... BROWSE_____
Description ..... *0200.1_____
                        BROWSE Program
PDA name ..... CUSCPDA_      Status window ..... Y
Programming mode ..... S_      Comment start indicator .. **_
Type ..... P Program      Comment end indicator .... ___

Code frame(s) ..... CSCA?___  CSCB?___  CSCC?___  _____  _____
Modify server specificatn CUSCMA___ CUSCMB___ CUSCMC___ CUSCMG___ _____
                        _____  _____  _____  _____
Modify client specificatn CUSCMA___ CUSCMB___ CUSCMC___ _____
                        _____  _____  _____  _____

Clear specification ..... CUSCC___      Post-generation ..... CUSCPS___
Read specification ..... CUSCR___      Save specification ..... CUSCS___
Pre-generation ..... CUSCPR___      Document specification ... CUS-D___

Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit  frame                                  main
    
```

Notice that the code frame names listed in the Code frame(s) field end with a question mark (?). The question mark indicates a hierarchy in which the code frame with the lowest number at the end of its name is used.

All code frames supplied with Natural Construct end with an 8 (used for code frame fixes supplied between releases) or 9 (used for original code frames supplied with Natural Construct). To define a custom code frame for your model, copy the supplied code frame, change the 8 or 9 to a lower number (from 1 to 7), and modify the code frame as desired. The next time Natural Construct calls that code frame, the one with the lowest number is used.

For example, you can copy the CSCA9 code frame, change the name to CSCA7, and edit it as desired. The next time Natural Construct calls CSCA?, CSCA7 is used.

The naming conventions for code frames are:

- The first character in a code frame name is always C.
- The second and third characters are reserved for the two-character model identifiers, such as MN for Menu or dash (—) for generic code frames used by multiple models.
- The fourth character is a single letter from A-Z indicating a position within a series of code frames.
- The fifth, sixth, and seventh characters are optional. They indicate specific functions that are typically performed by nested code frames, such as wildcard support.
- The last character must be a number from 1-9, with 9 reserved for the Natural Construct-supplied code frames and 8 reserved for any future updates.

Note:

The last character refers to the last position in the code frame name, which may or may not be the eighth physical position.

Use Nested Code Frames

When code frames are referenced in code (nested code frames), their names also end with the question mark character. For example, the CSLBA9 code frame for the Browse-Select model contains the nested code frame CS-BA?:

```
Code Frame ..... CSLBA9                               SIZE 17120
Description ..... Browse-Select* model main body      FREE 82673
>                                                    > + ABS X X-Y _ S 214 L 1
Top...+...1...+...2...+...3...+...4...+...5...+...6...+...7.. T C
PROG.
REPEAT /* Repeat loop to allow escape of program from within subroutine.
*
***** Start of Main Program Logic *****
*
  RESET #FIRST-&UQ-FOUND #REDISPLAY-SCREEN #MATCH-FOUND
CS-BA?                                                    F
NOT PROCESS-SELECTION-COLUMN AND PROCESS-SELECTED-RECORD      1
/*                                                            "
/* reposition to selected field if cursor selection           "
IF #CURS-LINE > #FIRST-ACTION-LINE                            "
  IF #SEL-TBL.#&UQ(#CURS-LINE) NE #NULL-&UQ                   "
    ASSIGN #FORWARD = FALSE                                    "
    ASSIGN #MATCH-FOUND = FALSE                               "
    ASSIGN #START.#KY = #SEL-TBL.#KY(#CURS-LINE)              "
    ASSIGN #START.#&UQ = #SEL-TBL.#&UQ(#CURS-LINE)            "
  END-IF                                                       "
END-IF                                                         "
...+...1...+...2...+...3...+...4...+...5...+...6...+...7.. T
```

Code Frame Menu Function

Use this function to access the Code Frame menu.

To access the Code Frame menu:

1. Enter "F" in Function on the Administration main menu.

The Code Frame menu is displayed. For example:

```

CSMMMAIN          N a t u r a l   C o n s t r u c t          CSMMNM0
Aug 17              Code Frame Menu                          1 of 1

                Functions
                -----
                E  Edit Code Frame
                S  Save Code Frame
                L  List Code Frames
                P  Purge Code Frame
                C  Clear Edit Buffer
                H  Print Saved Code Frame

                ?  Help
                .  Return
                -----

Function ..... _
Code Frame ..... _____
Description ..... _____

Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit                                     main

```

2. Enter the one-character function code in Function.

The functions available through this menu are:

- Edit Code Frame
- Save a Code Frame
- List Code Frames for Selection
- Purge a Code Frame
- Clear Edit Buffer
- Print Saved Code Frame

Note:

For a description of the Help and Return functions, see Help and Return Codes on Menus.

Edit Code Frame

Use this function to:

- Create a New Code Frame
- Modify an Existing Code Frame

Create a New Code Frame

 **To create a new code frame:**

1. Enter "E" in Function on the Code Frame menu.

The Code Frame editor is displayed. For example:

```

Code Frame .....                               SIZE
Description .....                             FREE 56825
>                                               > + ABS X X-Y _ S      L
  ....+....1....+....2....+....3....+....4....+....5....+....6....+....7.. T C

  ....+....1....+....2....+....3....+....4....+....5....+....6....+....7.. T

```

2. Type the code frame name in Code Frame.
3. Type a brief description of the code frame in Description.
4. Use the editor to create the code frame.

The Code Frame editor supports all edit commands except the RUN, CHECK, TEST, STOW, and SAVE command. For more information about the Code Frame editor, see Using the Code Frame Editor.

5. Enter "." (period) at the > prompt to return to the Code Frame menu.

For information on saving the code frame, see Save Code Frame.

Modify an Existing Code Frame

To modify an existing code frame:

1. Type "E" in Function on the Code Frame menu.
2. Type the code frame name in Code Frame.
3. Optionally, type a brief description of the code frame in Description.
4. Press Enter.

The specified code frame is displayed in the Code Frame editor.

5. Modify the code frame.

The Code Frame editor supports all edit commands except the RUN, CHECK, TEST, STOW, and SAVE command. For more information about the Code Frame editor, see Using the Code Frame Editor.

- Enter "." (period) at the > prompt to return to the Code Frame menu.

For information on saving the code frame, see Save Code Frame.

Note:

For more information about modifying the supplied code frames, see Step 5: Create Code Frame(s) and Define the Model.

Save a Code Frame

Use this function to save the code frame that is currently in the edit buffer to the Code Frame file.

▶ To save the code frame:

- Enter "S" in Function on the Code Frame menu.

If the specified code frame name already exists, Code Frame exists. Press Enter to confirm replace is displayed. You can either change the name or press Enter to update the existing code frame.

List Code Frames for Selection

Use this function to display a list of available code frames for selection.

▶ To list the available code frames for selection:

- Enter "L" in Function on the Code Frame menu.

The Select Frames window is displayed. For example:

Frame	Description	User	Date	Time
C--BAN9	Standard banner	SAG	Sep 30,01	09:55
CBAA9	Batch define data area	SAG	Sep 30,01	09:55
CBAB9	Batch initial setup	SAG	Sep 30,01	09:55
CBAC9	Batch main body	SAG	Sep 30,01	09:55
CBOA9	Object Browse Subp define data area	SAG	Sep 30,01	09:55
CBOB9	Object Browse Subp main body	SAG	Sep 30,01	09:55
CBRA9	Object Browse Static main body	SAG	Sep 30,01	09:55
CCNA9	Callnat main body	SAG	Sep 30,01	09:55
CDRA9	Driver main body	SAG	Sep 30,01	09:55
CETA9	Extendable Input main body	SAG	Sep 30,01	09:55
CFMA9	Maint define data area	SAG	Sep 30,01	09:55

Frame _____ Detail _ Scan for ... _____
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF1
 help retrn bkwrđ frwrđ
 Position cursor or enter screen value to select

This window displays the following information:

- Each code frame name in alphabetical order
 - Brief description of the corresponding code frame
 - User ID for the user who last saved the corresponding code frame
 - Date the corresponding code frame was last saved
 - Time the corresponding code frame was last saved
2. Type the name of the code frame in Frame.

Note:

If you enter the name of a code frame that is not currently displayed, the list is repositioned.

Optionally, you can mark Detail and type a value to scan for in Scan for. Detail lines are displayed for code frames containing the scanned value only.


3. Press Enter.

Purge a Code Frame

Use this function to permanently remove a code frame from the Code Frame file.

Note:

You cannot purge a code frame if it is currently used in a model.

 **To purge a code frame:**

1. Type "P" in Function on the Code Frame menu.
2. Type the name of the code frame in Code Frame.
3. Optionally, type a brief description of the code frame in Description.
4. Press Enter.

A confirmation window is displayed to confirm the purge.

Clear Edit Buffer

Use this function to clear the current values from the Code Frame editor.

 **To clear the edit buffer:**

- Enter "P" in Function on the Code Frame menu.

Print Saved Code Frame

Use this function to print a hardcopy of the specifications for a code frame that has been saved.

Note:

To use this function, you must have access to Com-plete, CMS, TSO, or CICS with Natural/AF or Com-Pose. For more information, see Frame Hardcopy Utility.

▶ To print a hardcopy of a saved code frame:

1. Type "H" in Function on the Code Frame menu.
2. Type the name of the code frame in Code Frame.
3. Optionally, type a brief description of the code frame in Description.
4. Press Enter.

Maintain Subprograms Function

Use this function to maintain the modify specification subprograms used by the generation models.

▶ To maintain the modify specification subprograms for a model:

1. Enter "S" in Function on the Administration main menu.

The Maintain Subprograms panel is displayed. For example:

```

CSDFSP                N a t u r a l   C o n s t r u c t                CSDFSP0
Aug 17                M a i n t a i n   S u b p r o g r a m s                1 of 1

Action ..... _ A,B,C,D,M,N,P,R
Subprogram ..... _____
Description ..... _____

PF-keys Used
Backward - Forward ..... _
Test ..... _

Assign to #PDA-PF-AVAILABLE1 . _____
Assign to #PDA-PF-AVAILABLE2 . _____
Assign to #PDA-PF-AVAILABLE3 . _____

Optional Window Settings
Window height ..... _
Window width ..... _

Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help retn quit                                     main

```

Use this panel to maintain the PF-key and window settings for the model subprograms. The Natural Construct nucleus uses these settings to determine the window size and PF-key functions for the model maintenance panels and sample subprograms.

**Warning:**

You cannot change these settings for model subprograms shipped with Natural Construct; you can only change the settings for model subprograms you create.

2. Type an action code in Action.

For a description of the available actions, press PF1 (help) when the cursor is in the field.

3. Type the name of the subprogram in Subprogram.
4. Press Enter.

The PF-key and window settings for the model are displayed.

Maintain Control Record Function

Use this function to maintain the default PF-key numbers and names, special characters, and dynamic attribute settings for Natural Construct.

Note:

These settings are for Natural Construct only, not for Natural Construct-generated programs.

**To maintain the control record:**

1. Enter "R" in Function on the Administration main menu.

The Maintain Control Record panel is displayed. For example:

```

CSCTRL                      N a t u r a l   C o n s t r u c t                      CSCTRL0
Aug 17                      M a i n t a i n   C o n t r o l   R e c o r d                      1 of 1

PF-key Assignments                      Dynamic Attributes
Main ..... PF 12 NAMED *0031.5__ main          Intensify ..... <
Return ..... PF 2_ NAMED *0031.2__ retrn        Blue ..... _
Quit ..... PF 3_ NAMED *0031.3__ quit           Green ..... _
Test ..... PF 4_ NAMED *0031.4__ test           White ..... _
Backward ..... PF 7_ NAMED *0032.2__ bkwrd       Pink ..... _
Forward ..... PF 8_ NAMED *0032.1__ frwrd        Red ..... _
Move left ..... PF 10 NAMED *0032.3__ left        Turquoise ..... _
Move right ..... PF 11 NAMED *0032.4__ right       Yellow ..... _
Help ..... PF 1_ NAMED *0031.1__ help            Special Hardware
User exit ..... PF 11 NAMED *0032.5__ userX       Blinking ..... _
Help indicator ..... *0033.1__ *                 Italic ..... _
Underscore character ..... *0033.2__ ----        Underline ..... _
Of indicator (eg., 1 of 2) ... *0033.3__ of       Reverse video ..... _
Disable indicator ..... *0033.4__ -
Scroll indicator ..... *0033.5__ >>             Default return ..... >
Position indicator(s) ..... *0034/4__ 1 2 3 4 5 6 7 8 9

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit                                  main

```


2. Use the fields on this panel to specify settings for the control record.

The fields on this panel are:

Column Heading	Field	Description
PF-key Assignments	PF <i>n</i>	<p>PF-key numbers for the corresponding functions. For each function (Main, Return, Quit, etc.), specify the number of the PF-key that performs the function. These functions are:</p> <ul style="list-style-type: none"> ● Main (invokes main menu) ● Return (displays previous panel) ● Quit (terminates current session) ● Test (invokes the Test function) ● Backward (scrolls backward/up through data) ● Forward (scrolls forward/down through data) ● Move left (scrolls to panel on the left of current panel) ● Move right (scrolls to panel on the right of current panel) ● Help (invokes help for current panel) ● User exit (invokes the User Exit editor) <p>Note: Only PF-keys 1 through 12 are defined. PF-keys 13 to 24 are equivalent to PF-keys 1 to 12, respectively.</p>
	NAMED	<p>PF-key names for the corresponding functions or the SYSERR numbers that supply the names. The current names are displayed on the right (main, retrn, quit, etc.).</p> <p>Because PF-key settings are part of the user interface, you can specify a SYSERR number from the CSTLDA library as the PF-key name. For example, SYSERR number *0031.5 corresponds to the English text, "main". If you specify *0031.5 in one of the NAMED fields, the corresponding PF-key name is "main".</p>

Column Heading	Field	Description
	Help indicator	Character used to indicate that help is available for a panel field (the default is *) or the SYSERR number that supplies the character. The indicator is placed in a separate prompt to the right of the input field.
	Underscore character	One- to 4-character set used to create the underscore line for panel text (the default is ----) or the SYSERR number that supplies the character set. The specified set is repeated until all spaces are filled (80, by default). For example, if “----” is specified, the underscore line is displayed as: ----- Or if “++” is specified, the underscore line is: ++ ++ ++ ++ ++ ++ ++ ++ ++ ++
	Of	indicator Character(s) used to indicate the current panel and the number of additional panels (the default is “of” as in “1 of 2”) or the SYSERR number that supplies the character(s).
	Disable indicator	Character used to indicate that an option is unavailable on a panel (the default is -) or the SYSERR number that supplies the character.
	Scroll indicator	Character(s) used to indicate that scrolling is available for a field on a panel (the default is >>) or the SYSERR number that supplies the character(s).
	Position indicator(s)	Characters used to indicate a position in a series of positions (the defaults are 1 to 10) or the SYSERR number that supplies the characters. If you are not using SYSERR, change the default characters by typing the new characters on the lines below this field.
Dynamic Attributes		Default dynamic attributes. You can specify up to four attributes, one of which must be the return to normal display attribute (see the description for the Default return field). The attributes are:
	Intensify	Character used to intensify text.
	Blue	Blue display for color terminals.
	Green	Green display for color terminals.
	White	White display for color terminals.

Column Heading	Field	Description
	Pink	Pink display for color terminals.
	Red	Red display for color terminals.
	Turquoise	Turquoise display for color terminals.
	Yellow	Yellow display for color terminals.
Special Hardware		Options available for terminals with special hardware. Note: Due to hardware restrictions, you may not be able to use all the options listed. For more information, refer to <i>DY Session Parameter</i> in the <i>Natural Parameter Reference</i> documentation. The special hardware options are:
	Blinking	Support for blinking.
	Italic	Support for italic.
	Underline	Support for underline.
	Reverse video	Support for reverse video.
Default return		Character used to return to normal (default) display; the default is >. A character must be specified in this field.

Note:

For more information on using SYSERR, see Using SYSERR for Multilingual Support.

Compare Menu Function

Use this function to access the Compare menu.

 **To access the Compare menu:**

1. Enter "C" in Function on the Administration main menu.

The Compare menu is displayed. For example:

```

CSDCMMF          N a t u r a l   C o n s t r u c t          CSDCMMF0
Aug 08                               Compare Menu          1 of 1

                Functions
                -----
M  Compare Models
F  Compare Frames

                ?  Help
                .  Return
                -----

Function ..... _

Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit                                     main

```

2. Enter the one-character function code in Function.

The functions available through this menu are:

- Compare Models
- Compare Frames

Note:

For a description of the Help and Return functions, see Help and Return Codes on Menus.

Compare Models

Use this function to:

- Compare a Model in Different Files
- Compare Two Models in the Same File
- Compare a Range of Models in Different Files

 **To access the Compare Models function:**

- Enter "M" in Function on the Compare menu.

The Compare Models panel is displayed. For example:

```

CSDCMP                N a t u r a l   C o n s t r u c t                CSDCMP10
Apr 02                Compare Models                1 of 1
                        Old                                New
Model ..... _____
Database ... _____
File ..... _____
Version .... _____
Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help retrn quit                                     main

```

Note:

The Old and New designation does not limit the comparison to old and new versions of the same model.

Compare a Model in Different Files

Use this function to compare the components of a model in different files. You can compare the same model or different models. In the following example, the same model is compared.

 **To compare the same model in different files:**

1. Type the name of the model in Old Model and New Model on the Compare Models panel.
2. Type the database identification (DBID) number for the Natural Construct system file for the first model in Old Database.
3. Type the DBID for the second model in New Database.
4. Type the Natural Construct file number for the first model in Old File.
5. Type the Natural Construct file number for the second model in New File.
6. Type the Natural Construct version number for the first model in Old Version.
7. Type the Natural Construct version number for the second model in New Version.

For example:

```

CSDCMP                N a t u r a l   C o n s t r u c t                CSDCMP10
Apr 02                Compare Models                1 of 1
                        Old                                New
Model ..... BROWSE_____ BROWSE_____
Database ... 18_____ 18_____
File ..... 116_____ 120_____
Version .... 4.5.2      5.2.1
Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help retrn quit                                     main

```

8. Press Enter.

The Show Model Differences window is displayed, showing the differences between the two models. For example:

CSDCMPD		Natural Construct	
Aug 08		Show Model Differences	
Old	4.5.2	New	5.2.1
BROWSE		BROWSE	
Description	*0200.1		*0200.1
Save subpr	CUSCGST		CUSCS
Pre-generate	CUSCGPR		CUSCPR
Post-generate	CUSCGPS		CUSCPS
Document	CUSCDOC1		CUS-D
Modify 1			CUSCMA
Modify 2			CUSCMB
Modify 3			CUSCMC
Frame	1 CUBANNER		CSCA?
Frame	2 CUSCDA		CSCB?
Frame	3 CUSCC1		CSCC?
Frame	4 CUSCC2		
Frame	5 CUSCC3		

Compare Two Models in the Same File

Use this function to compare the components of two models in the same file.

▶ To compare two models in the same file:

1. Type the name of the first model in Old Model on the Compare Models panel.
2. Type the name of the second model in New Model.
3. Type the database identification (DBID) number for the Natural Construct system file for the models in Old Database.
4. Type the Natural Construct file number for the models in Old File.
5. Type the Natural Construct version number for the models in Old Version.

For example:

CSDCMP		Natural Construct		CSDCMP10
Apr 02		Compare Models		1 of 1
Old		New		
Model	BROWSE_____	BROWSE-SELECT_____		
Database ...	18__	_____		
File	121__	_____		
Version	5.2.1			
Command	_____			
Enter-PF1---	PF2---	PF3---	PF4---	PF5---
PF6---	PF7---	PF8---	PF9---	PF10--
PF11--	PF12---			main
help	retrn	quit		

6. Press Enter.

The Show Model Differences window is displayed, showing the differences between the two models. For example:

CSDCMPD		Natural Construct	
Aug 08		Show Model Differences	
Old	5.2.1	New	5.2.1
	BROWSE		BROWSE-SELECT

Clear subpr	CUSCC		CUSLC
Pre-generate	CUSCPR		CUSLPR
Post-generate	CUSCPS		CUSLPS
Modify Host 2	CUSCMB		CUSLMB
Modify Host 4	CUSCMG		CUSLMD
Modify Host 5			CUSCMG
Modify 4			CUSLMF
Frame	1 CSCA?		CSLA?
Frame	2 CSCB?		CSLB?
Frame	3 CSCC?		CSLC?
Date	Jul 31, 2007		Jul 31, 2007
Time	10:09.510		10:09.510
User	SAG		SAG

Compare a Range of Models in Different Files

Use this function to compare the components for a range of models in different files. You can compare the same range of models or a different range. In the following example, the same range is compared.

To compare a range of models in different files:

1. Type the starting value for the range in Old Model on the Compare Models panel.

The starting value can be either the name of a model or the first few characters in the name. You can also limit the range by entering the wildcard character (*) with the model name. For example, if you enter `Browse*`, all the Browse models are compared. For information about using wildcard characters, see *Wildcard Selection, Natural Construct Generation*.

2. Type the database identification (DBID) number for the first range of models in Old Database.
3. Type the DBID for the second range in New Database.
4. Type the Natural Construct file number for the first range of models in Old File.
5. Type the Natural Construct file number for the second range in New File.
6. Type the Natural Construct version number for the first range of models in Old Version.
7. Type the Natural Construct version number for the second range in New Version field.
8. Press Enter.

The Show Model Differences window is displayed, showing the differences between the two ranges of models. For a description of this window, see *Compare a Model in Different Files*.

Compare Frames

Use this function to:

- Compare Two Code Frames in Different Files
- Compare All Frames For Two Models
- Compare a Range of Frames in Different Files

The models containing the code frames can reside in different system files. You can also compare all code frames and nested code frames for a model. The code frames can be different code frames in the same file, the same code frames in different files, or different code frames in different files. Results are presented code frame by code frame.

For information on comparing code frames in batch mode, see Comparison Utilities.

▶ To access the Compare Frames panel:

- Enter "F" in Function on the Compare menu.

The Compare Frames panel is displayed. For example:

CSDCMP Aug 08	N a t u r a l C o n s t r u c t Compare Frames	CSDCMP20 1 of 1
	Old	New
Model	_____	_____
Frame	_____	_____
Database ...	___	___
File	___	___
Version		
Command	_____	
Enter-PF1---	PF2---	PF3---
help	retrn	quit
		main

Note:

The Old and New designation does not limit the comparison to old and new versions of the same model or code frame.

Compare Two Code Frames in Different Files

Use this function to compare two code frames in different files. You can compare the same code frame or different code frames. In the following example, the same code frame is compared.

▶ To compare the same code frame in different files:

1. Type the name of the code frame in Old Frame and New Frame on the Compare Frames panel.
2. Type the database identification (DBID) number for the Natural Construct system file for the first frame in Old Database.
3. Type the DBID for the second frame in New Database.
4. Type the Natural Construct file number for the first frame in Old File.
5. Type the Natural Construct file number for the second frame in New File.
6. Type the Natural Construct version number for the first frame in Old Version.
7. Type the Natural Construct version number for the second frame in New Version.

For example:

```

CSDCMP                      N a t u r a l   C o n s t r u c t          CSDCMP20
Aug 08                      Compare Frames                          1 of 1

                                Old                                New
Model ..... _____
Frame ..... CUBADA9_      CBAA9____
Database ... 18_          18_
File ..... 116           121
Version .... 4.5.2       5.2.1

Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help retn quit                                     main

```

8. Press Enter.

The Summary Report window is displayed, showing the differences between the two frames. For example:

```

CSDCMPFD                      Natural Construct                      CSDCMP
                               Summary Report

Old version .... 4.5.2          New version .... 5.2.1
   Frame .... CUBADA9           Frame .... CBAA9

Old   New   Matched   Deleted   Inserted   Comments
-----
 284   292    284       0         8   Frames do not match
Press ENTR to continue or any PF-key to retrn

```

The Summary Report window displays the following information:

- Version numbers
- Name of each code frame
- Number of lines of code for each code frame
- Number of lines that match
- Number of lines removed from the first code frame
- Number of lines added to the second code frame
- Whether the code frames match (in this example, they do not match)

9. Press Enter.

The Compare Frames window is displayed, showing a line-by-line comparison. For example:

```

Oct 07                      Natural Construct                      04:15 PM
                               Compare Frames                      PAGE: 1

Old version .... 4.5.2          New version .... 5.2.1

                               CUBADA9/CBAA9                      T C
-----
+ C--BAN?                      F
= DEFINE DATA
= GDA-SPECIFIED                      1
= _____ 33 more equal lines _____
= ET-SPECIFIED                      2
= 01 #HOLD-COUNT(P3)                  "
+ 01 #WRITE-LINE(A30)
= *
= Secondary file 1 key for ADABAS, VSAM, DB2 *
= _____ 161 more equal lines _____
= 01 #INPUT1                          "
= KEY-IS-REDEFINED OR KEY-IS-COMPOUND 3
+ 02 #INPUT1-FIELDS(&KEY-NAT-FORMAT)   "
+ 02 REDEFINE #INPUT1-FIELDS          "
= CUBAGRED REDEFINE-INPUT-KEY        N "
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
frwrdr      retrn      top    hcoppy      frwrdr

```

The lines in the code frames that match are marked with an equal sign (=). Lines that are in the first code frame, but not in the second, are marked with a minus sign (-). Lines that are in the second code frame, but not in the first, are marked with a plus sign (+).

- To scroll forward (down) through the information, press Enter or PF8 (frwrd).
- To return to the first line, press PF5 (top).
- To return to the Compare Frames panel, press PF2 (retrn).
- To print a hardcopy of the Code Frame Compare Utility panel, press PF6 (hcopy).

For more information on printing a hardcopy of a code frame, see Print Saved Code Frame.

Compare All Frames For Two Models

Use this function to compare all the code frames used by two models.

 **To compare all the code frames used by two models:**

1. Type the name of the first model in Old Model on the Compare Frames panel.
2. Type the name of the second model in New Model.
3. Type the database identification (DBID) number for the Natural Construct system file for the first model in Old Database.
4. Type the DBID for the second model in New Database.
5. Type the Natural Construct file number for the first model in Old File.
6. Type the Natural Construct file number for the second model in New File.
7. Type the Natural Construct version number for the first model in Old Version.
8. Type the Natural Construct version number for the second model in New Version.

For example:

```

CSDCMP                N a t u r a l   C o n s t r u c t                CSDCMP20
Aug 08                Compare Frames                1 of 1

Model ..... Old           New
      BROWSE_____ BROWSE-SELECT_____
Frame ..... _____
Database ... 18_
File ..... 116
Version .... 4.5.2      5.2.1

Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit                                main

```

9. Press Enter.

The Summary Report window is displayed, showing the differences between the two models.

10. Press Enter.

The Compare Frames window is displayed, showing a line-by-line comparison. For a description of the Summary Report and Compare Frames window, see *Compare Two Code Frames in Different Files*.

Compare a Range of Frames in Different Files

Use this function to compare the components for a range of frames in different files. You can compare the same range of frames or a different range. In the following example, the same range is compared.

To compare a range of frames in different files:

1. Type the starting value for the range in Old Frame on the Compare Frames panel.

The starting value can be either the name of a code frame or the first few characters in the name. You can also limit the range by entering the wildcard character (*) with the code frame name. For example, if you enter CFM*, all code frames that begin with CFM are compared. For more information on using wildcards, see *Wildcard Selection, Natural Construct Generation*.

2. Type the Database identification (DBID) number for the first range of frames in New Database.
3. Type the DBID for the second range in Old Database.
4. Type the Natural Construct file number for the first range of frames in New File.

5. Type the Natural Construct file number for the second range in Old File.

For example:

```

CSDCMP                N a t u r a l   C o n s t r u c t                CSDCMP20
Aug 08                Compare Frames                                1 of 1

                                Old                                New
Model ..... _____
Frame ..... CG_____
Database ... 18_          18_
File ..... 116           121
Version ....

Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help retn quit                                           main

```

6. Press Enter.

The Select Frames window is displayed. For example:

```

CSDCMPF                Natural Construct                CSDCMP0
Oct 07                Select Frames                                1 of 1

      Frame                Old                                New
-----
_ CGMA9                DATE: 01-10-03 09:46 DATE: 01-09-27 15:03
_ CGOA9                DATE: 01-09-30 09:55 DATE: 01-09-27 15:03
_ CGPA9                DATE: 01-09-30 09:55 DATE: 01-09-27 15:03
_ CGRA9                DATE: 01-09-30 09:55 DATE: 01-09-27 15:03
_ CGSA9                DATE: 01-09-30 09:55 DATE: 01-09-27 15:03
_ CHDA9                DATE: 01-09-30 09:55 DATE: 01-09-27 15:03
_ CMDA9                Does not Exist       DATE: 01-09-27 15:03
_ CMNA9                DATE: 01-09-30 09:55 DATE: 01-09-27 15:03
_ CN-BAN9             DATE: 01-09-30 09:55 DATE: 01-09-27 15:03
_ CNDA9                DATE: 01-09-30 09:55 DATE: 01-09-27 15:03
_ CNOA9                DATE: 01-09-30 09:55 DATE: 01-09-27 15:03
_ COBA9                DATE: 01-10-07 11:12 DATE: 01-09-27 15:03
Code frame name .... CFM_____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF
      help retn                                           bkwrdr frwr
Position cursor or enter screen value to select

```

Use this window to select frames and display the comparison information.

7. Type "C" in the input field for any code frame.
8. Press Enter.

The Summary Report window is displayed, showing the differences between the two ranges of frames.

9. Press Enter.

The Compare Frames window is displayed, showing a line-by-line comparison. For a description of the Summary Report and Compare Frames window, see *Compare Two Code Frames in Different Files*.

Drivers Menu Function

Use this function to access the Drivers menu, which provides access to various utility subprograms supplied with Natural Construct.

To access the Drivers menu:

1. Enter "D" in Function on the Administration main menu.

The Drivers Menu panel is displayed. For example:

```

CTEMENU                N a t u r a l   C o n s t r u c t                CTEMNM0
Oct 31                                Drivers Menu                        1 of 1

                                Functions
                                -----
                                P  Predict-Related Drivers Menu
                                N  Natural-Related Drivers Menu
                                M  Miscellaneous Drivers Menu

                                ?  Help
                                .  Return
                                -----
Function .....  ___
Command .....  _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit                                                    lang

```

The drivers used to access the utilities are grouped according to what kind of subprogram they invoke. For a description of each menu function and the subprogram it invokes, refer to the applicable *Drivers Menu Option* section in *External Objects*.

2. Enter the one-character function code in Function.

Note:

For a description of the Help and Return functions, see Help and Return Codes on Menus.

Multilingual Support for Natural Construct

You can install Natural Construct in static (single) or dynamic (multiple) language mode. If dynamic language mode is installed, you can change your *Language value at runtime and display text in another supported language. You can also use the Natural SYSERR utility to add translations for the supplied text or change the supplied text to suit your organization's standards.

- For information on installing Natural Construct in dynamic language mode, see the installation documentation.
- For information on installing Natural Construct in static language mode, see Static (One-Language) Mode.
- For more information on using SYSERR, see Using SYSERR for Multilingual Support.

Libraries Supplying Multilingual Text

In dynamic language mode, all text displayed by Natural Construct is supplied by the Natural SYSERR utility from the following libraries:


- CSTLDA (all panel and window text)
- CSTMSG (all message text)

Natural Construct checks the value of the *Language variable to determine which language to display and retrieves the text for that language from the appropriate library.

Note:

For information on the SYSERR utility, refer to the Natural utilities documentation.

Display Text in Another Language

 **To change the text displayed on panels to another supported language:**

1. Press PF12 (lang) on the Administration main menu.

The Language Preference window is displayed. For example:

CSULPS		Natural Construct	CSULPS0
Aug 08		Language Preference	1 of 1
Number	Languages		
1	English		
2	Deutsch (German)		
3	Francais (French)		
4	Espagnol (Spanish)		
5	Italiano (Italian)		
6	Dutch		
7	Turkish		
8	Danish		
9	Norwegian		
10	Albanian		
Number ... _			
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---			
help retrn		bkwrđ frwrđ	
Position cursor or enter screen value to select			

2. Select the desired language.

The main menu is displayed in the selected language.

English (*Language 1) is the default language for Natural Construct. Although other languages are listed in the Language Preference window, you must add the translations for those languages in SYSERR.

If you do not provide translated text for a selected language, Natural Construct determines which language to display based on a user-defined hierarchy of language numbers (defined in the DEFAULT-LANGUAGE field in the CNAMSG local data area for the CNUMSG subprogram). For more information, see CNUMSG Subprogram.

Maintain Panel and Message Text

To define the text for another language, you must first change the *Language value in the Language Preference window. For information, see Display Text in Another Language.

To add text for another language or modify the supplied text:

- Use the SYSERR utility to add translations or modify the supplied text for all Natural Construct screens. Using the SYSERR utility is the quickest way to translate text on all panels.

Or:

- Use the Administration subsystem in translation mode to dynamically add translations or modify the supplied text. Typically, you would use translation mode to fine tune translations that were added using the SYSERR utility. This allows you to view the translation in the context of the entire panel. For information about translation mode, see Access the Administration Main Menu in Translation Mode.

Access the Administration Main Menu in Translation Mode

To help maintain the text for Natural Construct panels, windows, and messages, the Administration subsystem is also available in translation mode. Translation mode allows you to change the text supplied in the Natural SYSERR utility without leaving Natural Construct. You can change the text displayed on the Administration main menu, as well as on panels and help or selection windows for each function available through the Administration main menu.

You can also change the text displayed on the Generation and Help Text subsystem screens. For information, see *Translate Text for the Generation Subsystem* and *Translate Text for the Help Text Subsystem*.

The current value of the *Language variable determines whether you can maintain text for the current language or for another language.

▶ To Invoke in translation mode:

- Enter "menut" at the Natural prompt.

The Administration main menu is displayed. For example:

```

CSDMAIN                      N a t u r a l   C o n s t r u c t          CSDMNM0
Aug 08                        Administration Main Menu              1 of 1

                               Functions
                               -----
M  Maintain Models
F  Code Frame Menu
S  Maintain Subprograms
R  Maintain Control Record
C  Compare Menu
D  Drivers Menu
H  Help Text Main Menu
G  Generation Main Menu
?  Help
.  Return
                               -----

Function ..... _

Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit                                          lang

```

Use this panel to access the Natural Construct Administration functions in translation mode. Notice that functions are also available to access the Help Text and Generation main menus in translation mode.

Note:

Although the panels look the same in translation mode, they do not perform the same functions. For example, edit checks are not performed on input data. We recommend that you do not use translation mode for maintenance functions, such as defining a new model; use translation mode for translation functions, such as editing text in the current language or creating multilingual specification panels and messages.

This section covers the following topics:

- Use Translation Mode

Use Translation Mode

Translation mode uses the same series of panels and windows used throughout Natural Construct. All translatable text is cursor sensitive. When you select the text and press Enter, the Translate Short Message window is displayed. You can identify translatable text by the difference in color or intensification.

Note:

If you use Entire Connection to access Natural Construct, you can display the Translate Short Message window by double-clicking on translatable text.

You can translate two types of text:

- Screen text (text displayed on panels and in windows), which is stored in the CSTLDA library in SYSERR
- Message text, which is stored in the CSTMSG library in SYSERR

Each Natural Construct panel or window is associated with a local data area (LDA) that initializes the screen prompt variables. In translation mode, these variables are initialized to a SYSERR number and the actual text values are retrieved at runtime (based on the current value of the Natural *Language system variable).

You can use SYSERR numbers for some or all screen prompts. If you specify text as an initial value, Natural Construct displays the text as entered and the prompt cannot be dynamically translated.

When you use a SYSERR number instead of text, Natural Construct retrieves the corresponding text from the CSTLDA library (for prompts) or the CSTMSG library (for messages) in SYSERR. All changes to the values stored in SYSERR are automatically applied to the panels and messages the next time they are invoked.

Note:

For more information on substitution variables, refer to *REINPUT Statement, Natural Statements* documentation.

This section describes how to perform the following tasks:

- Translate Text for the Generation Subsystem
- Translate Text for the Help Text Subsystem
- Edit Text in the Current Language
- Translate Text to Another Language
- Use Substitution Variables

Translate Text for the Generation Subsystem

 **To translate text for the Generation subsystem:**

1. Type "G" in Function on the Administration main menu in translation mode.
2. Press Enter.

The Generation main menu is displayed in translation mode.

3. Translate the text as desired.

Translate Text for the Help Text Subsystem

▶ To translate text for the Help Text subsystem:

1. Type "H" in Function on the Administration main menu in translation mode.
2. Press Enter.

The Help Text main menu is displayed in translation mode.

3. Translate the text as desired.

Edit Text in the Current Language

Using translation mode, you can dynamically edit the text displayed on Natural Construct panels in the current language — without invoking the Natural map or code editor. For example, you can change the field prompt values to match your organization's conventions.

▶ To edit text in the current language:

1. Invoke in translation mode.
2. Access the panel you want to translate.
3. Move the cursor to the prompt text you want to change (not a blank input line).
4. Press Enter.

The Translate Short Message window is displayed. For example:

```

CSUTLATE                Natural Construct
Aug 08                  Translate Short Message                1 of 1

Language Short Message ( CSTLDA1116 )
-----  .....1.....2.....3.....4.....5.....6.....+
English  Action/Subprogram                                     /+26

```

This window provides quick access to the SYSERR numbers and text. Any changes made to the text in this window are automatically applied in SYSERR. The “/+26” value in this window indicates there are up to 26 characters available for each text segment that is to be translated. For more information on using the Translate Short Message window, see Context Translation.

Note:

Take care when changing the text for SYSERR numbers that are used on other panels.

5. Edit the SYSERR text as desired.
6. Press Enter.

The panel for which you are translating text is displayed, showing the edited text.

Translate Text to Another Language

Use translation mode to add translations for prompt text on panels and windows. For example, you can create specification panels in French (*Language 3).

▶ To translate text to another language:

1. Invoke in translation mode.
2. Press PF12 (lang).

The Language Preference window is displayed. For a description of this window, see Display Text in Another Language.

3. Move the cursor to the line containing the language for which you want to translate text.
4. Press Enter.

The Administration main menu is displayed.

5. Display the panel you want to translate.

For this example, the Maintain Models panel is translated to French.

6. Move the cursor over the prompt text you want to change (not a blank input line).
7. Press Enter.

The Translate Short Message window is displayed. For example:

```

CSUTLATE                Natural Construct
Oct 07                  Translate Short Message                1 of 1

Language Short Message ( CSTLDA1116 )
-----  ....+....1....+....2....+....3....+....4....+....5....+....6....+

English Action/Subprogram                /+30
Francais

```

8. Type the French equivalent under the English text.

The “/+30” value in this window indicates that you can use up to 30 characters for each text segment that is to be translated.

9. Press Enter.

The panel for which you are translating text is displayed, showing the translated text.

10. Repeat steps 6 through 9 until all text is translated.

You can translate text on any Natural Construct panel or window by invoking that panel or window and performing the translation procedure.

Note:

To display the Generation and Help Text subsystem screens, see Translate Text for the Generation Subsystem and Translate Text for the Help Text Subsystem.

Use Substitution Variables

Within SYSERR, you can provide text in different languages for each SYSERR number. For even greater reusability, you can use a substitution variable (such as :1:) with the text. Typically, the :n: variables are used in messages and the prompt is substituted for the :n: value. The actual text displayed depends on the value of the *Language variable for the user who accessed the panel.

Note:

For more information on substitution variables, refer to *REINPUT Statement, Natural Statements* documentation.

Access and Use the Sample Exit Subprograms

Natural Construct supplies several sample exit subprograms you can use to:

- Implement security
- Restrict access to various Natural Construct modules (models, code frames, model subprograms, help text members)
- Define model aliases for use in the Generation subsystem
- Provide user-defined defaults

Tip:

Always keep a backup copy of your modified sample exit subprograms.

The Natural Construct installation tape contains the sample exit subprograms. The subprograms are initially loaded into the SYSCSTX library, which is created during installation.

 **To modify a sample exit subprogram:**

1. Use the SYSMAN utility to copy the subprogram to the SYSCST library.
2. Modify the subprogram as desired.
3. Use SYSMAN to copy the object code to the library indicated in Supplied Sample Exit Subprograms.

This section covers the following topics:

- Supplied Sample Exit Subprograms
- Define Default Specifications

Supplied Sample Exit Subprograms

The following table lists each sample exit subprogram, the library in which Natural Construct will search for the subprogram, and the function supported by the subprogram. When a user selects a module and action, Natural Construct checks the library indicated below and invokes the applicable subprogram. The supplied subprograms are:

Subprogram	Library	Function
CSXAUEXT	SYSLIBS	Support for model alias names.
CSXCNAME	SYSLIBS	Security for the Generation main menu (before the post-generation subprogram is invoked).
CSXDEFLT	SYSLIBS	User-defined default values for generation models.
CSXDUEXT	SYSCST	Security for the Administration main menu.
CSXFUEXT	SYSCST	Security for the Code Frame menu.
CSXHUEXT	SYSLIBS	Security for the Help Text main menu.
CSXMUEXT	SYSCST	Security for the Maintain Model function.
CSXPSCHG	SYSLIBS	Security for the Generation main menu (after all substitution values are generated into the program).
CSXSECX	SYSLIBS	Support for customized security routines.
CSXTRANS	SYSLIBS	Support for special processing before an END or BACKOUT TRANSACTION statement is issued. Uses the same parameters as CSXSECX, with the addition of a timestamp parameter.
CSXSUEXT	SYSCST	Security for the Maintain Subprograms function.

Define Default Specifications

Natural Construct reads the default specifications for a model into the editor whenever the clear subprogram is invoked for a model. This occurs when the:

- Clear Specifications and Editor function is invoked and a model name is specified
- Modify Specifications function is invoked for a new model

To set default values for the model parameters, edit the clear subprogram for the model.

This section covers the following topics:

- Determine the Name of the Clear Subprogram
- Set the Default Specification Values
- Use CSXDEFLT Overrides
- Assign Your Own Defaults
- Use Predict Keywords

Determine the Name of the Clear Subprogram

▶ **To determine the name of the clear subprogram for the model:**

1. Logon to the SYSCST library.
2. Enter the following on the command line:

Menu ,M

The Maintain Models panel is displayed. For example:

```

CSDFM                      ***** Natural Construct *****                      CSDFM0
Aug 18                      Maintain Models                                     1 of 1

Action ..... ___ A,B,C,D,M,N,P,R
Model ..... _____
Based on model ..... _____
Description ..... _____

PDA name ..... _____      Status window ..... _
Programming mode ..... _      Comment indicators ..... _ \ ___
Type ..... _                  Programming Language ..... *
Code frame(s) ..... _____
Modify server ..... _____

Modify client ..... _____

Clear ..... _____      Post-generation ..... _____
Read ..... _____      Save ..... _____
Pre-generation ..... _____      Document ..... _____
Validate ..... _____      Stream ..... _____
Command ..... _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help retn quit frame                                     main
    
```

3. Enter "B" in Action.

The Select Models window is displayed.

4. Select the model name.

The information for that model is displayed. For example:

```

CSDFM                      ***** Natural Construct *****                      CSDFM0
Aug 18                      Maintain Models                                      1 of 1

Action ..... ___ A,B,C,D,M,N,P,R
Model ..... OBJECT-BROWSE-DIALOG_____
Based on model ..... _____
Description ..... *0201.1_____
                        OBJECT-BROWSE-DIALOG Subprogram
PDA name ..... CUBDPDA_      Status window ..... N
Programming mode ..... S_    Comment indicators ..... **_ \ ___
Type ..... N Subprog.      Programming Language ..... NATURAL_ *
Code frame(s) ..... CBDA?___ CBDB?___ _____
Modify server ..... CUBDMA_  CUBDMB_  _____
                        _____
Modify client ..... WCNBDMA_ WCNBDMB_ _____
                        _____
Clear ..... CUBDC_          Post-generation ..... CUBDPS_
Read ..... CUBDR_          Save ..... CUBDS_
Pre-generation ..... CUBDPR_ Document ..... CUBDD_
Validate ..... CUBDVAL_    Stream ..... CUBDT_
Command ..... _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help retrn quit frame                                      main
Model OBJECT-BROWSE-DIALOG displayed successfully

```

In this example, the clear subprogram is called CUBDC and the PDA name is CUBDPDA.

Set the Default Specification Values

▶ To set the default specification values for a model:

1. Log onto the SYSCST library.
2. Edit the clear subprogram for the model.

For example, the default values in the CUBDC subprogram for the Object-Browse-Dialog model are:

```

IF #PDAX-DESCRIPTION(1) = ' ' THEN
    #PDAX-DESCRIPTION(1) :=
        'This dialog is used for the object browse ...'
END-IF

```

3. Compile CUBDC.
4. Use the SYSMAIN utility to copy the object code for the clear subprogram to the SYSLIBS library.

The new defaults will now be used.

Use CSXDEFLT Overrides

If there is a default specification value that affects several models, you can set this value in the supplied CSXDEFLT subprogram. This subprogram provides default values for model parameters that can be overridden on the specification panels, as well as internal model parameters that are not displayed on the panels.

Tip:

Natural Construct has identified the most common parameters that fit this category. To see what they are, invoke CSUGETDF from the SYSCST library.

To change the default values of these parameters, edit CSXDEFLT in the SYSCSTX library. For example, to change DATE-EDIT-MASK (by default, LLL' 'ZD','YY) to 08 Aug11, change CSXDEFLT as follows:

```
VALUE 'DATE-EDIT-MASK'
      CSADEFLT.PARM-VALUE := 'YY','LLL' 'ZD'
CSADEFLT.PARM-VALUE := 'LLL' 'ZD','YY'
```

To use the new default values, CSXDEFLT must exist in the SYSLIBS library and the clear subprogram must call this subprogram. For an example of calling CSXDEFLT, refer to the CUFMC clear subprogram in the SYSCST library. For example:

```
INCLUDE CCDEFLTA ' 'DATE-EDIT-MASK' ' '#PDA-DATE-EDIT-MASK'
```

Notice that there are three modules in CUFMC that are being used to query the defaults: CCDEFLT, CCDEFLTA, and CCDEFLTL.

The supplied INCLUDE code members retrieve the default parameter values by issuing a CALLNAT to the CSUDEFLT sample exit subprogram. Prior to returning the defaults, CSUDEFLT checks to see whether the values have been overridden by the user-defined CSXDEFLT subprogram. If so, the overridden values are returned to the model.

Normally, the model's clear subprogram requests the default values and the returned values are copied to the model parameter data area (PDA). This way, the overhead of retrieving the defaults is only incurred when the user switches to another model or issues a Clear request.

To simplify the interface to CSUDEFLT, Natural Construct supplies three parameterized copycode members. Which copycode member you choose depends on the format of the field you are providing defaults for. The copycode members are:

Copycode Member	Description
CCDEFLTA	Provides default values for alphanumeric fields.
CCDEFLTL	Provides default values for logical fields.
CCDEFLT	Provides default values for numeric fields.

Each copycode member accepts two parameters. The format of the second parameter determines which of the copycode members to use:

- The first parameter identifies the default value; this value is passed to CSXDEFLT as the CSADEFLT.PARM-NAME variable. The exact name must appear in the DECIDE statement for CSXDEFLT.
- The second parameter defines the variable to which the default value is assigned (this is typically a variable in the model PDA). The variable is assigned the value returned in CSADEFLT.PARM-VALUE.

Example of retrieving an alphanumeric default value:

```
/*
/* Assign default date edit mask to (alphanumeric) model PDA variable
INCLUDE CCDEFLTA '''DATE-EDIT-MASK''' 'CUMNPDA.#PDA-DATE-EDIT-MASK'
```

For a list of parameters that can be modified by CSXDEFLT, refer to the CSUGETDF program. CSUGETDF also indicates which parameters are currently being overridden by CSXDEFLT. The CSXDEFLT source code contains a description of the parameters.

Example of increasing the size of the left or right prompt on panels:

You can use the CSXDEFLT sample exit subprogram to increase the size of the #RIGHT-PROMPT or #LEFT-PROMPT variable in generated browse, maintenance, or batch programs. For example:

```
VALUE 'RIGHT-PROMPT-LENGTH'
  CSADEFLT.PARM-VALUE := '9'
```

If you increase the prompt length to more than 9 characters, you must also change the size of two variables in the CSUMORE generation utility subprogram in the SYSCST library. Typically, the #PROMPT value should be two characters bigger than the biggest prompt size and the #LITERAL value should be the same size as #PROMPT. For more information, see CSUMORE Subprogram.

Note:

If you change the prompt length in CSXDEFLT, you must also change the #RIGHT-PROMPT and/or #LEFT-PROMPT variable on existing maps and then regenerate the modules.

Modify the CSXDEFLT Subprogram**► To modify CSXDEFLT:**

1. Logon to the SYSCSTX library.
During installation, the CSXDEFLT subprogram is installed in the SYSCSTX library.
2. Edit and save the CSXDEFLT subprogram.
3. Use the Natural SYSMAN utility to copy CSXDEFLT to the SYSCST library.
4. Catalog CSXDEFLT in the SYSCST library.
5. Use SYSMAN to copy the CSXDEFLT object code to the SYSLIBS library.

Tip:

For testing purposes, modify CSXDEFLT in the SYSCST library and invoke CSTG, instead of NCSTG, to see the affects of your change.

Modify the DEFAULT Keyword

You can modify the DEFAULT keyword by changing the value of the DEFAULT-SPECIFICATION-KEYWORD parameter in the CSXDEFLT subprogram.

Use *ISN as a Unique Primary Key for Maintenance

For information, see Use *ISN as the Unique Primary Key for Maintenance.

Assign Your Own Defaults

You can define default values at the corporate level. For example, you can use the export data function to default information such as the export work file number and the delimiter character. To implement the defaulting mechanism, refer to the following code example. The example illustrates how a work file number and column delimiter values are defaulted.

Example of assigning corporate defaults in the clear subprogram:

```

** We want to default two internal variables: #WORKFILE-NR and
** #COLUMN-DELIMITER
DEFINE DATA
  LOCAL USING CSADEFLT          /* Must include user default
                                /* interface LDA

  LOCAL
  01 #WORKFILE-NR(N2) INIT<5>    /* Assign fallback default "5"
  01 #COLUMN-DELIMITER(A1) INIT<','> /* Assign fallback default ","
  01 #PERFORMANCE(L) INIT<FALSE> /* Assign fallback default
                                /* "FALSE"

END-DEFINE
** Assign corporate default overrides if available
INCLUDE CCDEFLT1N '''WORKFILE-NUMBER-PC-DOWN''' #WORKFILE-NR
INCLUDE CCDEFLT1A '''WORKFILE-DELIMITER-CHAR''' #COLUMN-DELIMITER
INCLUDE CCDEFLT1L '''PERFORMANCE''' #PERFORMANCE
** Note that there are 3 separate INCLUDE members: one for numeric
** defaults (CCDEFLT1N), one for alphanumeric defaults (CCDEFLT1A), and
** one for logical defaults (CCDEFLT1L)
** Continue normal processing and the initial values may have been
** overridden by a corporate-supplied defaulting routine.

```

Notes:

1. To apply the changes corporation-wide, you must add the initial variable name and its initial value in the CSXDEFLT sample exit subprogram.
2. The internal defaulting mechanism may be affected when you use this defaulting mechanism to initialize the specification panel default keyword. Use the same keyword for both mechanisms. The specification panel default keyword overrides the internal default keyword.

After adding your own parameters, modify CSUDEFLT (so the CSUGETDF subprogram can add the new parameters to the #PARM-LIST) and then set the #MAX-DEFAULTS setting (for example, if you add one parameter, add one to the #MAX-DEFAULTS value).

You can also override changes the programmer has made and insist on certain values by including statements that assign values to the model PDA in the post-generation subprogram for the model, instead of the clear subprogram. Alternatively, you can hard code a search and replace option. For example, you can create your own copy of CCSETKEY and call it MYSETKEY. To do this, add the line STACK TOP DATA FORMATTED 'CCSETKEY' 'MYSETKEY' in the post-generation subprogram. All instances of CCSETKEY in the code will be replaced by MYSETKEY.

Use Predict Keywords

You can use Predict keywords to define default values for some model input parameters (for example, primary key fields, logical hold fields, and object descriptions). If default values have been specified in Predict, Natural Construct fills in the default values when the model is accessed. This reduces the number of specifications developers must provide when using the model.

This section covers the following topics:

- Define a Default Primary Key
- Define a Default Logical Hold Field
- Define a Default Object Description

Define a Default Primary Key

You can define a default value for a primary key by specifying a descriptor name in the Sequence field for the file in Predict. Natural Construct observes the following priorities when defaulting a primary key name for a file:

1. If the value of the default Sequence field for the file is unique and a valid descriptor, Natural Construct uses this value as the primary key.
2. If the value of the default Sequence field is not unique, Natural Construct reads through the file and uses a unique descriptor field value as the primary key.
3. If the file does not have a unique descriptor field, but has only one descriptor field, Natural Construct assumes the field value is unique and uses it as the primary key.

Define a Default Logical Hold Field

You can define a default value for the logical hold field by attaching a keyword called “HOLD-FIELD” to the field in Predict.

Note:

You may have to first define the HOLD-FIELD keyword in Predict using Keyword Maintenance.

Natural Construct observes the following priorities when defaulting a hold field name for a file:

1. If the HOLD-FIELD keyword is attached to a field that meets the format criteria for a hold field, Natural Construct uses this field as the logical hold field.
2. If a field name contains any of the following strings, it is used as the logical hold field:
 - HOLDFIELD
 - HOLD-FIELD
 - HOLD_FIELD
 - TIMESTAMP

- TIME-STAMP
 - TIME_STAMP
 - LOGCOUNTER
 - LOG-COUNTER
 - LOG_COUNTER
3. If a field meets the format criteria for a hold field, Natural Construct uses this field as the logical hold field.

Define a Default Object Description

You can define a default value for the object description by specifying the default value in the Literal Name field for the file in Predict. Natural Construct uses this value as the object description when the file is referenced in messages. If the value is “Customer”, for example, messages are displayed as “Customer not found” or “Customer displayed”.