

New Model Example

This section provides a step-by-step example of creating a new model using the procedure described in Build a New Model. The model, Menu, generates a program that displays several choices to a user and allows the user to select one.

Note:

For an example of a generated menu program, refer to NCMAIN in the demo library.

The procedure to build a new model is:

- Step 1: Define the Scope of the Model
 - Step 2: Create the Prototype
 - Step 3: Scrutinize the Prototype
 - Step 4: Isolate the Parameters in the Prototype
 - Step 5: Create a Code Frame and Define the Model
 - Step 6: Create the Model PDA
 - Step 7: Create Translation LDAs and Maintenance Maps
 - Step 8: Create the Model Subprograms
 - Step 9: Implement the Model
-

Step 1: Define the Scope of the Model

A program generated by the Menu model will provide a list of options and descriptions to the user for selection. The INPUT statement can be generated by Natural Construct or supplied by the developer.

Step 2: Create the Prototype

After defining the scope of the model, create a prototype to handle the most complex function and then refine the prototype to handle the simpler functions.

The following example shows the output from the NCMAIN prototype:

```

NCMAIN                      ***** ACME DEPARTMENT STORES *****          NCLAYMN1
Oct 09                      - MAIN MENU -                                     04:11 PM

      Code | Subsystem
      +---+-----+
      C   | Customer
      T   | Table Maintenance
      O   | Order
      ?   | Help
      .   | Terminate
      +---+-----+
Code:  __
Direct Command: _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      help  retrn quit          flip                               main

```

Step 3: Scrutinize the Prototype

After creating the prototype, follow the steps outlined in Step 3: Scrutinize the Prototype, to ensure that all of the assumptions are correct and the scope of the model has been addressed.

Step 4: Isolate the Parameters in the Prototype

Next, identify data that must be supplied by parameters.

Parameters for the Program Header

The parameters supplied for the program header are:

- Name of the program being generated.
- Application to which the generated program belongs.
- Date and time the program was generated.
- Title and description of the program.

Parameters for the Program Body

The parameters supplied for the program body are:

- Name of the global data area (GDA).
- Map used by the generated program.

- List of functions and their descriptions.

Step 5: Create a Code Frame and Define the Model

This section describes how to create a code frame and define the model.

Create the Code Frame

Once you have identified all data that must be supplied by parameters, you can create the code frame (CMNA?) for the model. For more information, see [Create the Code Frames](#).

Note:

For an example of the code frame for the Menu model, display the CMNA? code frame (stored in the SYSCST library) in the Code Frame editor.

 **To create the code frame:**

1. Read the prototype into the Code Frame editor and define the substitution parameters.
2. Create the user exits.

To allow developers to specify additional parameters, local data, or Natural statements, include the following user exits:

User Exit	Description
CHANGE-HISTORY	Generates comment lines indicating the date and ID of the person who created or modified the program. The developer provides a description of changes.
LOCAL-DATA	Defines additional local variables used in the generated program.
START-OF-PROGRAM	Defines code that is executed once at the beginning of the generated program — after all standard initial values are assigned. For example, this user exit code can initialize input values from globals.
BEFORE-INPUT	Defines code that is executed immediately before the INPUT statement is executed (before each input panel is displayed). For example, this user exit code can issue the SET CONTROL statements.
AFTER-INPUT	Defines code that is executed immediately after the INPUT statement is executed (after each input panel is displayed).
BEFORE-PROCESSING-MENU-CODES	Defines code that is executed before the menu code is processed.
SPECIAL-CODE- PROCESSING	Defines code that is executed when a menu code does not FETCH a program.
END-OF-PROGRAM	Contains code that is executed once before the program is terminated. For example, this user exit code can assign a termination message.
SET-PF-KEYS	Defines code that is executed before the PF-keys are set and allows non-standard PF-keys to be added to the program. (The additional PF-keys are defined in the CDKEYLDA local data area.)

3. Create the code frame conditions.

To create conditional code, insert the condition name and condition level number in the code frame. To view some examples of conditional code, display the CMNA? code frame in the Code Frame editor and refer to the following condition names:

- GDA-SPECIFIED
- DIRECT-COMMAND-PROCESSING
- MAP-USED

Define the Model

At this point, you can define the model to Natural Construct using the Maintain Models function on the Administration main menu. For more information, see Define the Model.

Model subprograms are prefixed by CUMN, where CU identifies the subprogram as a Natural Construct model subprogram and MN identifies the model (Menu).

Note:

The CU prefix is used by the models supplied with Natural Construct. When you create a new model or modify a supplied model, use a CX prefix. For this example, we use a CU prefix.

To add the Menu model to Natural Construct:

1. Invoke the Maintain Models function from the Administration main menu.
2. Specify the following parameters on the Maintain Models panel.

For example:

CSDFM	N A T U R A L C O N S T R U C T				CSDFM0
Oct 09	Maintain Models				1 of 1
Action	__ A,B,C,D,M,N,P,R				
Model	MENU_____				
Description	*0200.1_____				
	MENU Program				
PDA name	CUMNPDA_	Status window	Y		
Programming mode	S_	Comment start indicator ..	**_		
Type	P Program	Comment end indicator	__		
Code frame(s)	CMNA?_	_____	_____	_____	_____
Modify server specificatn	CUMNMA_	CUMNMB_	_____	_____	_____
Modify client specificatn	CUMNMA_	CUMNMB_	_____	_____	_____
Clear specification	CUMNC_	Post-generation	CUMNPS_		
Read specification	CUMNR_	Save specification	CUMNS_		
Pre-generation	CUMNPR_	Document specification ...	CUMND_		
Command	_____				
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---					
help retn quit frame					main

Most of the model components are listed on this panel. The components that are not listed are assigned through subprograms or code frames. For example, the CUMNMA0 and CUMNMB0 maps are invoked through the CUMNMA and CUMNMB maintenance subprograms, respectively, and the generation subprogram is assigned through the CMNA? code frame.

Step 6: Create the Model PDA

Use the CST-PDA model in the Generation subsystem to create the parameter data area (PDA) for the model (CUMNPDA).

For an example of the parameter data area for the Menu model, refer to the CUMNPDA parameter data area in the SYSCST library.

To create the model PDA:

1. Type the following parameter values on the Generation main menu:

Parameter	Value
Function	M
Module	CUMNPDA
Model	CST-PDA

2. Press Enter.

The Standard Parameters panel for the CST-PDA model is displayed.

3. Enter "Menu" in Model.

For example:

```

CUPDMA                      CST-PDA Parameter Data Area          CUPDMA1
Oct 09                      Standard Parameters                    1 of 1

Module ..... CUMNPDA_
Model ..... Menu_____ *

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

The Generation main menu is displayed.

4. Enter "G" in Function.

Natural Construct generates the PDA.

5. Enter "E" in Function.

The Natural data area editor is displayed.

- Each substitution parameter in the model code frame corresponds to a user area variable in the model PDA that has the same name and a #PDAX- or #PDA- prefix.
- Each condition variable in the model code frame corresponds to a condition variable in the model PDA that has the same name and a #PDAC- prefix.

6. Specify the type and length of each #PDAX variable.

7. Add any #PDA variables required by the model.

Step 7: Create Translation LDAs and Maintenance Maps

This section describes how to create the translation LDA and maintenance map for your model.

Create the Translation LDAs

To support dynamic translation of text and messages, you can create up to five translation local data areas (LDAs) for each maintenance map; the module that invokes the map must have a translation LDA. Translation LDAs contain the names of the fields on the map that can be translated. To assign the INIT values for these fields, use SYSERR references.

For an example of the translation LDAs for the Menu model, refer to the CU--MAL and CUMNMBL LDAs in the SYSCST library.

The following example shows a translation LDA:

```

Local      CUXXMAL   Library SYSCST                      DBID 19 FNR 26
Command                                         > +
I T L Name                                     F Leng Index/Init/EM/Name/Comment
All - -----
* * **SAG TRANSLATION LDA
* * * used by map CUXXMX0.
  1 CUTRMAL
  2 TEXT                                     /* Corresponds to syserr message
  3 #GEN-PROGRAM                           A 20 INIT<'*2000.1,.'>
  3 #TITLE                                  A 20 INIT<'*2001.1,.'>
  3 #DESCS                                  A 20 INIT<'*2001.2,.'>
  3 #DATA-AREA                             A 20 INIT<'*2097.3,.'>
  3 #LANGUAGE                              A 20 INIT<'*1309.2,.'>
R 2 TEXT
  3 TRANSLATION-TEXT
  4 TEXT-ARRAY                             A 1 (1:100)
  2 ADDITIONAL-PARMS
  3 #MESSAGE-LIBRARY                       A 8 INIT<'CSTLDA'>
  3 #LDA-NAME                              A 8 INIT<'CUXXMAL'>
  3 #TEXT-REQUIRED                         L INIT<TRUE>
  3 #LENGTH-OVERRIDE                      I 4 /* Explicit len to translate
----- S 17 L 1

```

To create your translation LDAs

1. Copy an existing translation LDA.
2. Define the fields for which you want dynamic translation.

All translation LDAs must have the format shown in the example above. For more information, see Step 7: Create the Translation LDAs and Maintenance Maps.

Create the Maintenance Maps

The model uses one or more maintenance maps to accept parameters from a user. To create the maintenance maps, use one of the following methods:

- Copy an existing maintenance map and modify it to suit your requirements.
- Create the map in the Natural Map editor.
- Create the map using the Natural Construct Map model.

For an example of the maintenance maps for the Menu model, refer to the CU--MA0 and CUMNMB0 maps in the SYSCST library.

The CU--MA0 maintenance map contains the following input fields:

Field	Description
Module	Name of the menu to be generated.
System	Name of the system (usually the library name).
Global data area	Name of the global data area (GDA) used by this menu program. Developers can display a field-level help window to select a value for this field.
With block	Name of the GDA block used by this menu program (if desired).
Title	Title for the menu program. The title identifies the program for the List Generated Modules function on the Generation main menu and is used internally for program documentation.
Description	Brief description of the menu program. The description is inserted in the banner at the beginning of the program and is used internally for program documentation.
First header	First heading displayed on the generated menu.
Second header	Second heading displayed on the generated menu.
Command	Indicates whether the menu supports a Direct Command line.
Message numbers	Indicates whether the menu uses message numbers or message text.
Password	Indicates whether the menu is password protected.

The CUMNMB0 maintenance map contains the following input fields:

Field	Description
Map layout	Name of the map layout (form) used to create the menu panel. Developers can display a field-level help window to select a value for this field.
Code	1 or 2-character code used to invoke the functions listed on the menu. Each code must have a corresponding function.
Functions	Functions listed on the menu. Each function must have a corresponding code. If desired, developers can change the word, Functions, to another value.
Program Name	Name of the program that is invoked when the corresponding function is selected. Developers can display a field-level help window to select a value for this field.
Optional Parameters	Indicates whether additional input parameters are required (user must enter a value) or optional. Developers can specify a maximum of four additional parameters (using PF5). On the menu, the parameters are displayed as column headings to the right of the Function heading and as input fields below the Code field. If additional parameters are specified, Natural Construct generates a legend ® for Required, O for Optional). The legend is aligned under the first occurrence of a Required or Optional indicator.

Step 8: Create the Model Subprograms

After creating the code frame, PDA, maintenance maps, and translation LDAs for the Menu model, you are ready to create the model subprograms. The following sections describe how to create each of the model subprograms:

- Create the Maintenance Subprograms
- Create the Pre-Generation Subprogram
- Create the Post-Generation Subprogram
- Create the Clear Subprogram
- Create the Save Subprogram
- Create the Read Subprogram
- Create the Generation Subprogram
- Create the Documentation Subprogram
- Test the Model Subprograms

Create the Maintenance Subprograms

Use the CST-Modify model in the Generation subsystem to create the maintenance subprograms (CUMNMA and CUMNMB). These subprograms invoke the CUMNMA0 and CUMNMB0 maps, respectively.

For an example of the maintenance subprograms for the Menu model, refer to the CUMNMA and CUMNMB subprograms in the SYSCST library.

▶ To create the CUMNMA maintenance subprogram:

1. Display the Standard Parameters panel for the CST-Modify model.
2. Specify the following parameters:

```

CUGIMA                      CST-Modify Subprogram                      CUGIMA0
Oct 09                      Standard Parameters                          1 of 1

Module ..... CUMNMA_
Parameter data area CUMNPDA_ *

Title ..... Menu Model Modify Subp___
Description ..... This subprogram is used as modify panel 1_____
                  1 of 2_____
                  _____
                  _____

Map name ..... CU--MA0_ *
Translation LDAs ... CU--MAL_ _____ *
Cursor translation . X

First header ..... _____
Second header ..... *0311.1,+/54_____

Subpanel ..... _
Window support ..... _
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help retrn quit          windw pfkey          left userX main
    
```

3. Generate the subprogram.

For information, see *Natural Construct Generation*.

▶ To create the CUMNMB maintenance subprogram:

1. Display the Standard Parameters panel for the CST-Modify model.
2. Specify the following parameters:

```

CUGIMA                      CST-Modify Subprogram          CUGIMA0
Oct 09                      Standard Parameters          1 of 1

Module ..... CUMNMB__
Parameter data area CUMNPDA_ *

Title ..... Menu Model Modify Subp__
Description ..... This subprogram is used as modify panel 2_____
                2 of 2_____
                _____
                _____

Map name ..... CUMNMB0_ *
Translation LDAs ... CUMNMBL_ _____ *
Cursor translation . X

First header ..... _____
Second header ..... *0310.1,+/54_____

Subpanel ..... _
Window support ..... _
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      help  retrn quit          windw pfkey          left  userX main

```

3. Generate the subprogram.

For information, see *Natural Construct Generation*.

Create the Pre-Generation Subprogram

Use the CST-Pregen model in the Generation subsystem to create the pre-generation subprogram.

For an example of the pre-generation subprogram for the Menu model, refer to the CUMNPR subprogram in the SYSCST library.

To create the CUMNPR pre-generation subprogram:

1. Display the Standard Parameters panel for the CST-Pregen model.
2. Specify the following parameters:

```

CUGPMA                      CST-Pregen Subprogram          CUG-MA0
Oct 09                      Standard Parameters             1 of 1

Module ..... CUMNPR__
Parameter data area CUMNPDA_ *

Title ..... Menu Model Pregen Subp
Description ..... Pre-generate subprogram.  ..._____
                  Set conditions and assign shared PDA variables.
                  _____
                  _____

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

```

3. Generate the subprogram.

For information, see *Natural Construct Generation*.

Create the Post-Generation Subprogram

Use the CST-Postgen model in the Generation subsystem to create the post-generation subprogram.

For an example of the post-generation subprogram for the Menu model, refer to the CUMNPS subprogram in the SYSCST library.

To create the CUMNPS post-generation subprogram:

1. Display the Standard Parameters panel for the CST-Postgen model.
2. Specify the following parameters:

CUGOMA Oct 09	CST-Postgen Subprogram Standard Parameters	CUGOMA0 1 of 1
Module	CUMNPS__	
Model	MENU_____ *	
Title	Menu Model Post-Gen Subp_	
Description	Post-generation parameters for the Menu model._____	

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

3. Generate the subprogram.

For information, see *Natural Construct Generation*.

Create the Clear Subprogram

Use the CST-Clear model in the Generation subsystem to create the clear subprogram. The Menu model requires a clear subprogram because the #PDA-USER-AREA field is redefined into non-alphanumeric variables (for example, #PDA-USER-PARM-LENGTH and #PDA-CODE-LENGTH) and the Description field on the first maintenance panel requires default text.

For an example of the clear subprogram for the Menu model, refer to the CUMNC subprogram in the SYSCST library.

To create the CUMNC clear subprogram:

1. Display the Standard Parameters panel for the CST-Clear model.
2. Specify the following parameters:

```

CUGCMA                      CST-Clear Subprogram          CUG-MA0
Oct 09                      Standard Parameters            1 of 1

Module ..... CUMNC____
Parameter data area CUMNPDA_ *

Title ..... Menu Model Clear Subp____
Description ..... Clear specification parameters and assign initial value
_____
_____
_____

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

```

3. Generate the subprogram.

For information, see *Natural Construct Generation*.

Create the Save Subprogram

Use the CST-Save model in the Generation subsystem to create the save subprogram. The save subprogram allows the model to read a previously-generated program.

For an example of the save subprogram for the Menu model, refer to the CUMNS subprogram in the SYSCST library.

To create the CUMNS save subprogram:

1. Display the Standard Parameters panel for the CST-Save model.
2. Specify the following parameters:

CUGAMA	CST-SAVE Subprogram	CUG-MA0
Oct 09	Standard Parameters	1 of 1
Module	CUMNS____	
Parameter data area	CUMNPDA_ *	
Title	Menu Model Save Subp_____	
Description	Save specification parameters for the menu model_____	

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

3. Generate the subprogram.

For information, see *Natural Construct Generation*.

Create the Read Subprogram

Use the CST-Read model in the Generation subsystem to create the read subprogram.

For an example of the read subprogram for the Menu model, refer to the CUMNR subprogram in the SYSCST library.

To create the CUMNR read subprogram:

1. Display the Standard Parameters panel for the CST-Read model.
2. Specify the following parameters:

```

CUGRMA                      CST-Read Subprogram          CUG-MA0
Oct 09                      Standard Parameters          1 of 1

Module ..... CUMNR_____
Parameter data area CUMNPDA_ *

Title ..... Menu Model Read Subp_____
Description ..... Read parameter specifications _____
_____
_____
_____

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

```

3. Generate the subprogram.

For information, see *Natural Construct Generation*.

Create the Generation Subprogram

Use the CST-Frame model in the Generation subsystem to create the generation subprogram.

For an example of the generation subprogram for the Menu model, refer to the CUMNGGL subprogram in the SYSCST library.

To create the CUMNGGL generation subprogram:

1. Display the Standard Parameters panel for the CST-Frame model.
2. Specify the following parameters:


```

CUGFMA                      CST-Frame Subprogram          CUG-MA0
Oct 09                      Standard Parameters             1 of 1

Module ..... CUMNGGL_
Parameter data area CUMNPDA_ *

Title ..... Menu Model Frame Subp____
Description ..... Generation parameter variables (if length and format
                  are specified)_____
                  _____
                  _____

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

```

3. Generate the subprogram.

For information, see *Natural Construct Generation*.

Create the Documentation Subprogram

Use the CST-Document model in the Generation subsystem to create the documentation subprogram.

Note:

For an example of the documentation subprogram for the Menu model, refer to the CUMND subprogram in the SYSCST library.

To create the CUMND documentation subprogram:

1. Display the Standard Parameters panel for the CST-Document model.
2. Specify the following parameters:

```

CUGDMA                      CST-Document Subprogram          CUGDMA0
Oct 09                      Standard Parameters                1 of 2

Module ..... CUMND____
Model ..... Menu_____ *
Maps ..... CU--MAO_ CUMNMBO_ _____ *
                                     _____ *
Translation LDAs ... CU--MAL_ CUMNMBL_ _____ *
                                     _____ *

Title ..... Menu Model Document Subp_
Description ..... Writes Predict documentation for the Menu model____
                                     _____
                                     _____

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

3. Press PF11 (right).

The Additional Parameters panel is displayed.

4. Specify the following parameters:

```

CUGDMB                      CST-Document Subprogram          CUGDMB0
Oct 09                      Additional Parameters        2 of 2

Help Text ..... Type ..... 0
                                     Major .... Model_____
                                     Minor .... Menu_____

Description
1 _____
2 _____
3 _____
4 _____
5 _____
6 _____
7 _____
8 _____
9 _____
10 _____

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

5. Generate the subprogram.

For information, see *Natural Construct Generation*.

Test the Model Subprograms

Natural Construct supplies a utility to help test the model subprograms.

To invoke the model subprogram test utility:

1. Log onto the SYSCST library.
2. Enter "CSUTEST" at the Next prompt (Direct Command box for Unix).

The Single Module Test Program panel is displayed. For information about this panel, see Test the Model Subprograms.

Step 9: Implement the Model

After creating and testing the code frames and model components (data areas, model subprograms, maps, etc.), copy all components to the SYSLIBS library to implement the model.

To implement the model:

1. Invoke the SYSMAIN utility from the Next prompt.
2. Copy all the model components to the SYSLIBS library.

Your new model is now ready for use in the Generation subsystem.