CST-Modify and CST-Modify-332 Models

This section describes the CST-Modify and CST-Modify-332 models, which are used to create the modify (maintenance) subprograms for a model.

- CST-Modify generates specification panels that support dynamic translation.
- CST-Modify-332 generates specification panels that do not support dynamic translation; it is supplied for those who want to continue using maintenance subprograms that were generated using previous versions of Natural Construct.

This section covers the following topics:

Introduction

After defining the model PDA and creating the clear, read, and save subprograms; maintenance maps; and translation LDAs, you must create one or more maintenance subprograms to collect user-supplied specification parameters (#PDAX variables), perform validation checks, and set the condition codes and #PDA variables (optional).

Maintenance subprograms are executed in the same order as they appear on the Maintain Models panel. Usually, there is one maintenance subprogram for every left/right (horizontal) maintenance panel. Data edits should only be applied if the developer presses Enter or PF11 (right). Either the maintenance subprogram or the maintenance map can validate the parameters.

You should only trap PF-keys that perform specialized functions related to the panel. If you want the PF-key settings to be dependent on the default settings specified on the control record, the subprogram should not contain hardcoded PF-keys (check the PF-key values using the variables specified in CU—PDA).

The CST-Modify and CST-Modify-332 models are described in the following sections. We recommend using the CST-Modify model to create new maintenance subprograms.

Note:
A maintenance subprogram can test the value of CU—PDA.#PDA-PHASE to identify the phase during which it was invoked (G for generation, M for modification, L for translation, U for sample user exits, etc.).

Example of a Maintenance Subprogram

The following example shows the first 40 lines of the CUMNMA maintenance subprogram:

```
0010 **SAG GENERATOR: CST-MODIFY                        VERSION: 4.4.1
0020 **SAG TITLE: Menu Model Modify Subp
0030 **SAG SYSTEM: NATURAL-CONSTRUCT
0040 **SAG DATA-AREA: CUMNPDA
```

```
0050 **SAG MAP: CU--MA0
0060 **SAG DESCS(1): This subprogram is used as modify panel 1
0070 **SAG DESCS(2): 1 of 2
0080 **SAG HEADER2: *0311.1,+/54
0090 **SAG TRANSLATION-LDA(1): CU--MAL
0100 **SAG DYNAMIC-TRANSLATION: X
0110 *****************************************************************
0120 * Program  : CUMNMA
0130 * System   : NATURAL-CONSTRUCT
0140 * Title    : Menu Model Modify Subp
0150 * Generated: May 03,02 at 05:33 PM by REGEN41
0160 * Function : This subprogram is used as modify panel 1
0170 *            1 of 2
0180 *
0190 *
0200 * History
0210 *****************************************************************
0220 DEFINE DATA
0230   PARAMETER USING CUMNPDA       /* Model specific data
0240   PARAMETER USING CU--PDA       /* Standard model parameters
0250   PARAMETER USING CSASTD        /* Standard message passing area
0260   LOCAL USING CNAMSG            /* Message retrieval passing area
0270   LOCAL USING CSLRCODE          /* Message return codes
0280   LOCAL USING CSAMARK           /* Field mark information
0290   LOCAL USING CSLPHASE          /* Valid generation phases
0300   LOCAL USING CSLSTD            /* Local message passing area
0310   LOCAL USING CSACURS      /* Used by CSUCURS to translate prompts
0320   LOCAL USING CU--MAL      /* Translation LDA
0330   LOCAL
0340     01 #PROGRAM (A8)
0350     01 LOCAL-TRANSLATION
0360       02 TEXT
0370         03 #HEADER2 (A54)
0380             INIT<'*0311.1,+/54'>
0390       02 REDEFINE TEXT
0400         03 TRANSLATION-TEXT
....
```

For an example of a maintenance subprogram subpanel generated by the CST-Modify model, refer to CUMNMBA in SYSCST.

CST-Modify Model

The CST-Modify model generates maintenance subprograms that support dynamic translation and multiple languages. To implement dynamic translation, you must also create a maintenance map and one or more translation local data areas (LDAs) for each maintenance subprogram.

The CST-Modify model generates either a main maintenance subprogram panel (defined on the Maintain Models panel) or a maintenance subprogram subpanel (invoked from the main maintenance subprogram panel using a PF-key). To reduce the amount of information on a panel, we recommend grouping similar parameters, such as windowing information, and moving that information to a subpanel.

If desired, you can use a subroutine to display a subpanel. Subroutines typically control processes that do not require a panel or subpanel to be displayed. For example, a subroutine can enable backward or forward scrolling or test a function that does not require mandatory edits for generation. Both subprograms and subroutines are invoked by PF-keys from the main maintenance subprogram panel.

All maintenance subprograms require a VALIDATE-INPUT subroutine to process mandatory edits. At generation time, the edits for the maintenance subprogram subpanel are processed first, then the edits for the main maintenance subprogram panel are processed. Therefore, any subroutine edits should also be included in the VALIDATE-INPUT subroutine.

Tip:
To avoid confusion about the order of execution of the panel and subpanel subroutines, place edit checks in programs rather than in subroutines.

The CST-Modify model also allows you to override the headers and PF-keys defined on the Subprogram record.

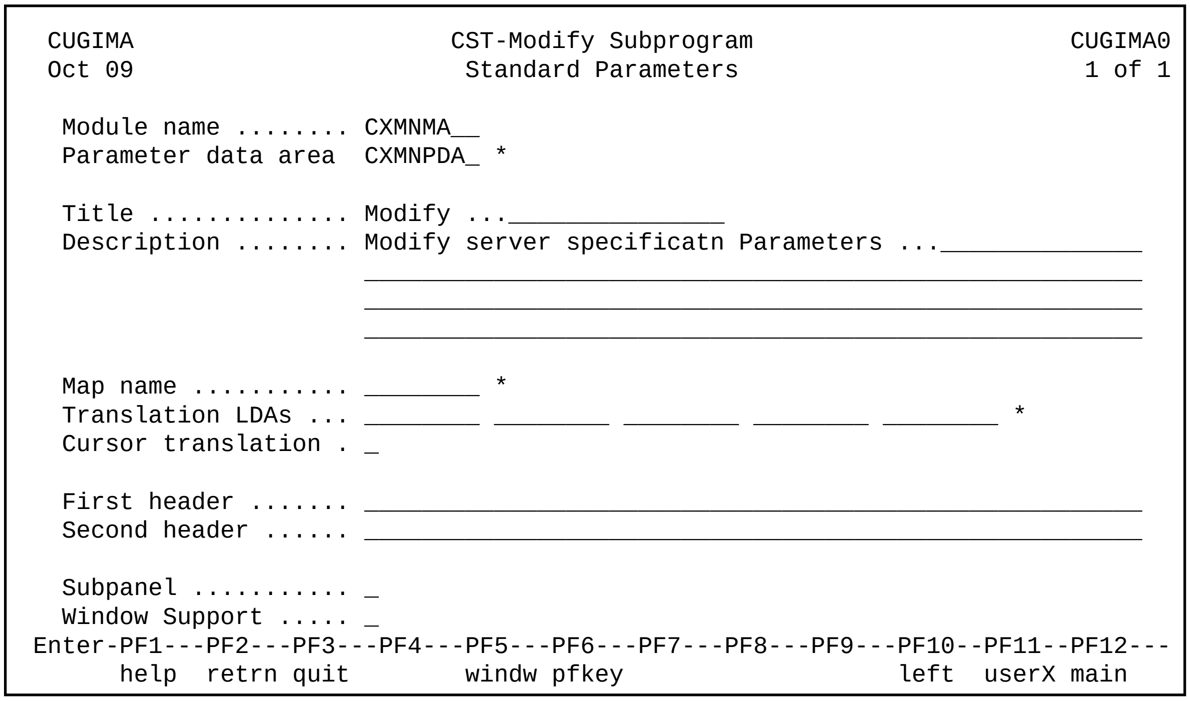
This section covers the following topics:

- Parameters for the CST-Modify Model
- User Exits for the CST-Modify Model

Parameters for the CST-Modify Model

Use the CST-Modify model to generate a maintenance subprogram that supports dynamic translation. This model has one specification panel, Standard Parameters.

Standard Parameters Panel



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

   Module name ........ CXMNMA__
   Parameter data area  CXMNPDA_ *

   Title .............. Modify ...______________
   Description ........ Modify server specificatn Parameters ...______________
                        ______________________________________________________
                        ______________________________________________________
                        ______________________________________________________

   Map name ........... ________ *
   Translation LDAs ... ________ ________ ________ ________ ________ *
   Cursor translation . _

   First header ....... ______________________________________________________
   Second header ...... ______________________________________________________

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

Use this panel to define standard parameters, such as the map and translation LDAs used with the maintenance subprogram and whether cursor translation is supported on the generated panel or subpanel. You can also use this panel to override the first and second headings or specify subpanel and window support.

Using PF-keys on this panel, you can change the default window settings (PF5 windw) or override the PF-key settings (PF6 pfkey).

The input fields on the Standard Parameters panel are:

Field	**Description**
Module name	Name specified on the Generation main menu. The name of the maintenance subprogram must be alphanumeric and no more than eight characters in length. Use the following naming conventions: ● Panel: CX*xx*M*y* where *xx* uniquely identifies your model and *y* is a letter from A–J that identifies the maintenance panel (A for the first maintenance panel, B for the second, etc.) ● Subpanel: CX*xx*M*yz* where *xx* uniquely identifies your model, *y* is a letter from A–J that identifies the maintenance panel (A for the first maintenance panel, B for the second, etc.), and *z* is a letter from A–J that identifies the subpanel.
Parameter data area	Name of the parameter data area (PDA) for your model. Natural Construct determines the PDA name based on the Module name specified on the Generation main menu. For example, if you enter "CXMNMA", Natural Construct assumes the PDA name is CXMNPDA. Use the following naming convention: `CX`*`xx`*`PDA` where *xx* uniquely identifies your model.
Title	Title for the generated subprogram. The title identifies the subprogram for the List Generated Modules function on the Generation main menu and is used internally for program documentation.
Description	Brief description of the subprogram. The description is inserted in the banner at the beginning of the subprogram and is used internally for program documentation.

Field	Description

Map name	Name of the map used for the maintenance subprogram. Natural Construct determines the name of the map based on the Module name specified on the Generation main menu. For example, if you enter CXMNMA as the subprogram name, Natural Construct assumes the map name is CXMNMA0.
	
	The specified map must exist in the current library and the map name should correspond to the maintenance subprogram name, with the addition of a zero. The zero indicates that the map has no hard-coded text and is used for dynamic translation. For example:
	
	`Program            Map`
	`CXMNMA      CXMNMA0`
	`CXMNMB       CXMNMB0`

Translation LDAs	Names of the translation local data areas (LDAs) for the maintenance subprogram. You can specify the names of up to five translation LDAs. The specified translation LDAs must exist. The LDA name should correspond to the maintenance subprogram name, with the addition of an "L". For example:
	
	`Program          Translation LDA`
	`CXMNMA      CXMNMAL`
	`CXMNMB       CXMNMBL`

Cursor translation	Indicates whether users can modify the text on this panel while in translation mode. To support cursor translation, mark this field.
First header	First heading displayed on the generated subprogram panel or the SYSERR number(s) that supplies the heading. By default, this header is automatically populated with the description specified on the model record. To override this default, specify the new header in this field. To specify the positioning of the heading, use special syntax after the text or SYSERR numbers. By default, the header is displayed at the left margin. To center *First Heading* across 50 bytes for example, type: `First Heading,+/50` The text before `,+/` indicates the heading displayed. The number after`,+/` indicates the number of bytes within which the heading is centered. For information about SYSERR message numbers, see Use SYSERR References or refer to the SYSERR utility in the Natural Utilities documentation. **Note:** Data substitution within SYSERR references is not supported in this context.

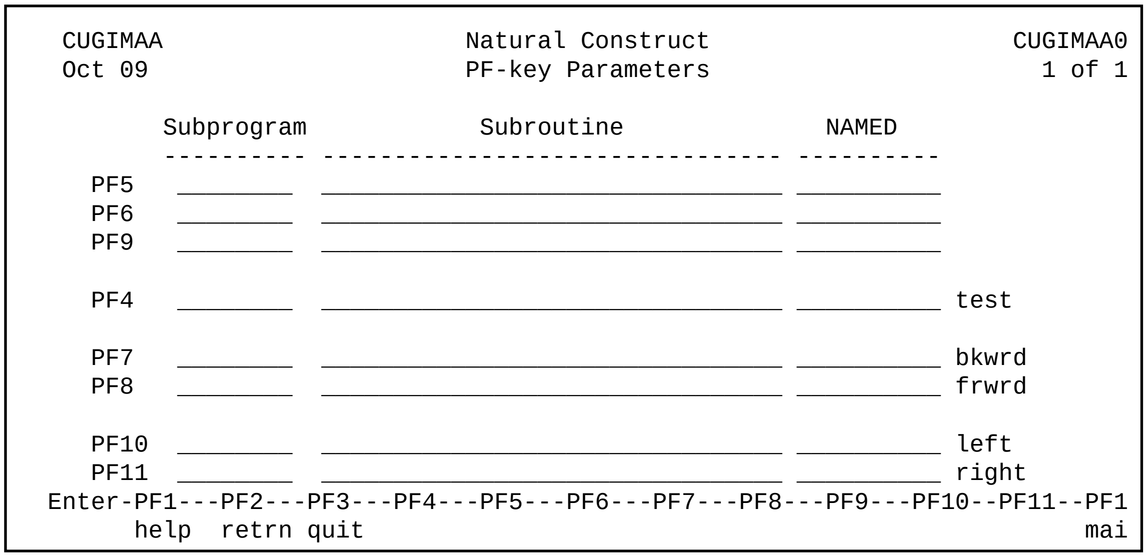
Field	Description
Second header	Second heading displayed on the generated panel or the SYSERR number(s) that supplies the heading. By default, this header is populated with the description specified on the subprogram record, if it exists. Unlike the model record, which populates the first header field, the subprogram record only exists if you create it. To supply a second header (if no subprogram record exists) or to override the default, specify a new header in this field. **Note:** We recommend using this field to define the second heading, instead of the description on the Maintain Subprograms panel. The Natural Construct nucleus does not reference the Subprogram record for supplied models, so the description used to populate the second header will not exist unless you create it. To specify the heading position, use special syntax after the text or SYSERR number. By default, the header is displayed at the left margin. To center *Second Heading* across 50 bytes for example, type: `Second Heading,+/50` The text before `,+/` indicates the heading displayed. The number after`,+/` indicates the number of bytes within which the heading is centered. For information about SYSERR message numbers, see Use SYSERR References or refer to the SYSERR utility in the Natural Utilities documentation.
Subpanel	Indicates whether the generated subprogram is created as a subpanel that is invoked from a main panel (such as a help selection window). To create the subprogram as a subpanel, mark this field. By default, the Natural Construct nucleus controls the help, retrn, quit, left, right, and main PF-keys (defined on the control record) for a main panel, and the help, retrn, quit, and main PF-keys for a subpanel. To define the processing for additional keys (the left and right keys, for example) on a subpanel, press PF6 (pfkey) on the Standard Parameters panel. For more information, see Define Non-Standard PF-Keys.
Window support	Indicates whether the generated subprogram is displayed in a window. To display the generated subprogram in a window, mark this field. By default, the PF-keys and messages are displayed within the generated window, and a frame (border) is displayed around the generated window. To change the default window settings, press PF5 (windw) on the Standard Parameters panel. For more information, see Change the Default Window Settings.

Define Non-Standard PF-Keys

To define the processing for non-standard PF-keys:

1. Press PF6 (pfkey) on the Standard Parameters panel.

   The PF-Key Parameters window is displayed. For example:



```
  CUGIMAA                      Natural Construct                    CUGIMAA0
  Oct 09                       PF-key Parameters                      1 of 1

         Subprogram             Subroutine             NAMED
         ---------- -------------------------------- ----------
    PF5   ________  ________________________________ __________
    PF6   ________  ________________________________ __________
    PF9   ________  ________________________________ __________

    PF4   ________  ________________________________ __________ test

    PF7   ________  ________________________________ __________ bkwrd
    PF8   ________  ________________________________ __________ frwrd

    PF10  ________  ________________________________ __________ left
    PF11  ________  ________________________________ __________ right
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF1
       help  retrn quit                                                  mai
```

   By default, the Natural Construct nucleus controls the help, retrn, quit, left, right, and main PF-keys for a main panel (defined on the control record), and the help, retrn, quit, and main PF-keys for a subpanel. Using this window, you can override the nucleus-controlled PF-keys displayed on a subpanel.

   Note:
   The left and right PF-keys are available only if the maintenance subprogram is a subpanel.

2. Define the processing and name for the non-standard PF-key.

   Note:
   You can also change the processing and/or name for a non-standard PF-key currently defined in the window.

   Use the following input fields to define the non-standard PF-key:

Field	Description
Subprogram	Name of the subprogram executed when the corresponding PF-key is pressed. This subprogram is invoked during generation to process the VALIDATE-INPUT subroutine.
Subroutine	Name of the subroutine executed when the corresponding PF-key is pressed.
NAMED	Name of the PF-key (either text or a valid SYSERR message number). If this field is blank, the default key names are used. For information about SYSERR message numbers, see Use SYSERR References or refer to the SYSERR utility in the Natural Utilities documentation.

3. Press Enter.

User Exits for the CST-Modify Model

```
CSGSAMPL                    CST-Modify Subprogram                       CSGSM0
Oct 09                            User Exits                            1 of 1

               User Exits             Exists    Sample   Required Conditional
    -------------------------------- -------- ---------- -------- ------------
 _  CHANGE-HISTORY                            Subprogram
 _  PARAMETER-DATA
 _  LOCAL-DATA
 _  START-OF-PROGRAM
 _  BEFORE-CHECK-ERROR                         Example
 _  BEFORE-STANDARD-KEY-CHECK                  Example
 _  ADDITIONAL-TRANSLATIONS
 _  ADDITIONAL-INITIALIZATIONS                 Example
 _  BEFORE-INPUT
 _  INPUT-SCREEN                               Example                 X
 _  AFTER-INPUT
 _  BEFORE-INVOKE-SUBPANELS                                            X
 _  AFTER-INVOKE-SUBPANELS                                             X
 _  BEFORE-REINPUT-MESSAGE
 _  VALIDATE-DATA                             Subprogram
 _  MISCELLANEOUS-SUBROUTINES                  Example
 _  END-OF-PROGRAM                             Example
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
frwrd help  retrn quit                    bkwrd frwrd
```

For information about these user exits, see Supplied User Exits. For information about using the User Exit editor, see *User Exit Editor*, *Natural Construct Generation*.

CST-Modify-332 Model

Use the CST-Modify-332 model to generate a maintenance subprogram that does not support dynamic translation. This model is provided for those who want to continue using maintenance subprograms that were generated under previous versions of Natural Construct.

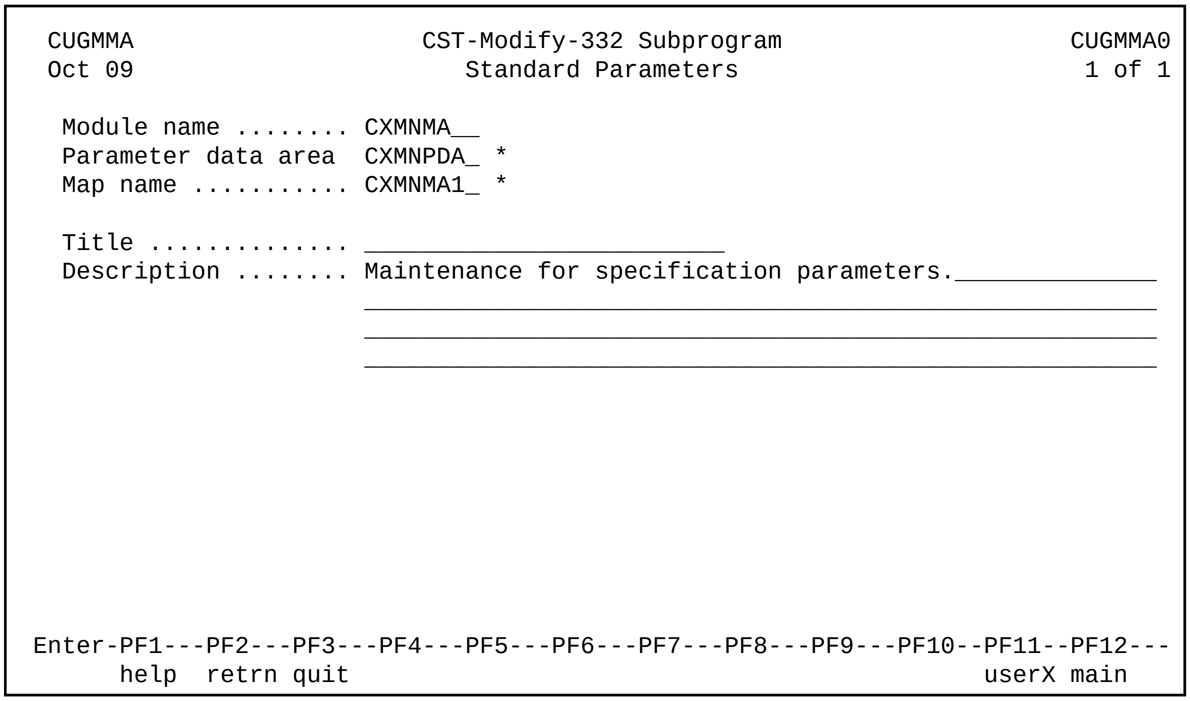
This section covers the following topics:

- Parameters for the CST-Modify-332 Model
- User Exits for the CST-Modify-332 Model

Parameters for the CST-Modify-332 Model

Use the CST-Modify-332 model to generate the maintenance subprogram. This model has one specification panel, Standard Parameters.

Standard Parameters Panel



```
 CUGMMA                    CST-Modify-332 Subprogram                   CUGMMA0
 Oct 09                       Standard Parameters                       1 of 1

  Module name ........ CXMNMA__
  Parameter data area  CXMNPDA_ *
  Map name ........... CXMNMA1_ *

  Title .............. _________________________
  Description ........ Maintenance for specification parameters.______________
                       _______________________________________________________
                       _______________________________________________________
                       _______________________________________________________









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

The input fields on the Standard Parameters panel are:

Field	**Description**
Module name	Name specified on the Generation main menu. The name of the maintenance subprogram must be alphanumeric and no more than eight characters in length. Use the following naming convention: `CX`*`xx`*`M`*`y`* where *xx* uniquely identifies your model and *y* is a letter from A–J that identifies the maintenance panel (A for the first maintenance panel, B for the second, etc.).
Parameter data area	Name of the parameter data area (PDA) for your model. Natural Construct determines the PDA name based on the Module name specified on the Generation main menu. For example, if you enter "CXMNMA", Natural Construct assumes the PDA name is CXMNPDA. Use the following naming convention: `CX`*`xx`*`PDA` where *xx* uniquely identifies your model.

Map name	Name of the map used for the maintenance subprogram. Natural Construct determines the name of the map based on the Module name specified on the Generation main menu. For example, if you enter CXMNMA as the subprogram name, Natural Construct assumes the map name is CXMNMA1 (for English). The map must exist in the current library, and the map name should correspond to the maintenance subprogram name, with the addition of the language code. For example:
	
	`Program            Map`
	`CXMNMA      CXMNMA1`

Title	Title for the generated subprogram. The title identifies the subprogram for the List Generated Modules function on the Generation main menu and is used internally for program documentation.
Description	Brief description of the subprogram. The description is inserted in the banner at the beginning of the subprogram and is used internally for program documentation.

User Exits for the CST-Modify-332 Model

```
CSGSAMPL                  CST-Modify-332 Subprogram                 CSGSM0
Oct 09                            User Exits                        1 of 1

               User Exits             Exists    Sample   Required Conditional
    -------------------------------- -------- ---------- -------- ------------
 _  CHANGE-HISTORY                            Subprogram
 _  LOCAL-DATA
 _  START-OF-PROGRAM
 _  AFTER-INPUT                                Example
 _  PROCESS-SPECIAL-KEYS                      Subprogram              X
 _  VALIDATE-DATA                             Subprogram
```

For information about these user exits, see Supplied User Exits. For information about using the User Exit editor, see *User Exit Editor, Natural Construct Generation*.