

MAINT MODEL

This document describes the Maint model, which generates a program that maintains a file using a unique key and, optionally, a related secondary file. The Maint model generates the code necessary to scroll through the MU/PE fields of a primary file or the records of a secondary file.

The following topics are covered:

- **Introduction**, page 2
- **Parameters for the Maint Model**, page 4
- **User Exits for the Maint Model**, page 16

Introduction

Natural Construct provides two alternatives to generate a maintenance process:

- 1 Use the Object models: Object-Maint-Dialog, Object-Maint-Dialog-Subp, and Object-Maint-Subp (which also generates the object PDA and restricted PDA). These models provide all the functionality needed for application development at the production level. The separation of dialog and data makes future changes to the maintenance process easier to implement. For information about the Object models, see:
 - **Object-Maint-Dialog Model**
 - **Object-Maint-Dialog-Subp Model**
 - **Object-Maint-Subp Model**
- 2 Use the Maint model. This model creates fast prototypes or simple maintenance processes that are temporary. The process is faster to implement because you need only create one or two Natural objects: the maintenance program generated by the Maint model and, optionally, a map. Conversely, because dialog and data are combined, this maintenance process is not as easy to modify as the process generated by the Object models.

The differences between the two types of models include:

| Object Models | Maint Model |
|--|--|
| Maintain up to four levels of files: primary, secondary, tertiary, and quaternary. | Maintains one or two levels of files: primary and secondary. |
| Support multiple scrolling regions. | Supports one scrolling region. |
| Support a link between scrolling regions on multiple panels. | Does not support a link between scrolling regions on multiple panels. |
| Provide automatic cursor repositioning after an error. | Does not provide automatic cursor repositioning after an error. (This functionality is available in user exits.) |

The following example shows a maintenance program generated using the Maint model:

```
NCTFRPRD          ***** TABLE SUBSYSTEM *****          NCTFRPRD1
Oct 30             - PRODUCT MAINTENANCE -                    10:45 AM

*Action (A,B,C,D,M,N,P,R): _ Product ID: 256733

Description   : OATS AND BARLEY CEREAL_____
Reorder Point: 22_____
Unit Cost    : 20.00_____

Manufacturer's Address  Street   : 45 CRESENT STREET____
                        City      : EDMONTON_____
                        Province  : ALBERTA_____
                        Postal Code: P9E 8E4

Direct Command: _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
confm help retrn quit      flip  pref                                main
Next PRODUCT ID displayed
```

Sample Output from the Maint Model

Note: Notice that PF5 (flip) and PF6 (pref) are available. For a description of these special function keys, see *Natural Construct Generation*.

By default, a maintenance program generated using the Maint model prompts users to press the Enter key to confirm a Purge action. If you specify a confirmation key other than Enter, the program will force confirmation of Add, Modify, and Purge actions. For a description of how to change the confirmation key, see *Natural Construct Generation*.

Parameters for the Maint Model

The following sections describe the parameters and specification panels for the Maint model. After specifying the required parameters on one panel, click Next to proceed to the following panel. To generate the module, click Finish on the last specification panel.

Standard Parameters

The screenshot shows a dialog box titled "MAINT - Page 1 of 3". On the left side, there is a logo for "NATURAL Construct" and the word "Model". At the bottom left, there is a logo for "SOFTWARE AG". The main area of the dialog is titled "Standard Parameters" and contains several input fields and checkboxes. The fields are: "Module:" (empty), "System:" (SYSPL451), "Global data area:" (CDGDA), "With block:" (empty), "Title:" (Maintain...), "Description:" (This program is used to maintain the....), "First heading:" (empty), and "Second heading:" (empty). Below these fields are three checkboxes: "Command", "Message numbers", and "Password", all of which are unchecked. At the bottom of the dialog, there are five buttons: "Help", "Cancel", "Back", "Next", and "Finish".

Standard Parameters for the Maint Model

Use this panel to specify standard parameters for your maintenance program. These parameters are similar for all models. For a description of the standard parameters, see **General Model Specifications**.

Additional Parameters

Additional Parameters for the Maint Model

The fields on this panel are:

| Field | Description |
|---------------|---|
| Predict view | Name of the Predict view. A file definition for the specified file must exist in Predict. |
| Natural (DDM) | Name of the primary file (the DDM). All fields in the primary file must be in the specified DDM. (Since this field defaults to the primary file name, only enter a DDM name if it is different from the primary file name.) |

| Field | Description (continued) |
|--------------------|---|
| Primary key | <p data-bbox="630 338 1372 535">Name of the key in Predict for the primary file. This key becomes the primary key for accessing the Predict view for the maintenance program. It may be a descriptor, superdescriptor, or subdescriptor. If the specified key does not exist in the corresponding file, a message is displayed indicating that the key field was not found.</p> <p data-bbox="528 556 1372 682">Note: Maintenance programs generated by Natural Construct are only intended for use with views having a unique primary key. However, the primary key does not need to be defined with the UQ option.</p> <p data-bbox="528 703 1372 829">Note: You can use a combination of fields to define the primary key for a table. To do this, add a superdescriptor in Predict that defines the combination. Specify this superdescriptor as the primary key name.</p> <p data-bbox="528 850 1372 955">Note: You do not need to mark the primary key as a descriptor, as long as the field is a valid UDF redefinition of a key field from the master segment.</p> |
| Record description | <p data-bbox="630 976 1372 1075">Description used in messages. By default, “Record” is displayed and messages are displayed in the form: “Record not found” or “Record displayed.”</p> |
| Log file name | <p data-bbox="630 1096 1372 1260">Name of the view used to perform update logging. All fields in the primary file (with matching fields in the log file) are written to the log records. Other fields, such as LOG-DATE and LOG-USER, must also appear in the log view. For more information, see <i>Natural Construct Generation</i>.</p> |
| Natural (DDM) | <p data-bbox="630 1281 1372 1413">Name of the DDM (data definition module) used for logging. If the log file DDM is different from the update log file, enter the name of the DDM used for logging. The update log file must be a view of the log file DDM.</p> |

| Field | Description (continued) |
|-------------------|--|
| Input using map | <p>Name of an external map for the input panel. To use an external map, enter a map name in this field.</p> <p>To help you create maintenance maps, Natural Construct supplies the CDLAYFM1 and CDLAYFM2 layout maps. If the generated program does not maintain MU/PE fields or a secondary file, use CDLAYFM1. If the generated program does maintain MU/PE fields or a secondary file, use CDLAYFM2. This layout map contains all the variables needed to control scrolling. The #LINE-CV variable on the CDLAYFM2 map is only applicable to a secondary file.</p> <p>If you do not specify a map name, Natural Construct generates an INPUT statement and designs the panel with all prompts and fields aligned on the left. MU/PE fields are not included. The fields are in the same order as the fields in the Predict field definition.</p> <p>Note: If you are generating a program that maintains both a primary and secondary file, use an external map or a primary file containing MU/PE fields (for information, see Secondary File Parameters, page 12).</p> <p>Note: For a list of variables you can use on a map, see Variables You Can Use on a Map, page 9.</p> |
| Minimum key value | <p>Minimum key for the range of records. To limit the range of records the program can access, enter a starting (minimum) value.</p> |
| Maximum key value | <p>Maximum key for the range of records. To limit the range of records the program can access, enter an ending (maximum) value. The minimum and maximum values form a “window” in which the program looks. Users cannot access keys outside this range.</p> <p>Note: If the key is defined as numeric, the minimum and maximum key values must be numeric and the specified maximum key must be greater than or equal to the specified minimum key. If the key values are variable, you can assign them in the START-OF-PROGRAM user exit.</p> |
| Single prompt | <p>If this field is marked, a single prompt is displayed for all fields (for example, Date: ____ _ __). This option applies if the key is a superdescriptor or redefined in Predict.</p> |

| Field | Description (continued) |
|----------------------|--|
| Multiple prompts | If this field is marked, one prompt is displayed for each field (for example, Year: ____ Month: __ Day: __). This option applies if the key is a superdescriptor or redefined in Predict. For more information, see <i>Natural Construct Generation</i> . |
| | Note: If the key is redefined and you want to have prompting for some redefined fields — but no prompting for other fields — you can use a map to display the prompts the way you want them to appear. |
| Mark cursor field | Name of the field marked by default when an error occurs. To avoid ambiguity, fully qualify the field name with a structure name. |
| Action Parameters | Action parameters. By default, all actions except Recall and Former are marked. (Recall is not a default action because values on the secondary file are not recalled.) To select an action, mark the corresponding field. At least one action must be selected. |
| #ACTION field length | Length (1, 2, or 3) of the field used for #ACTION names. For example, to use “DI” for Display, type “2” in this field. |
| Add | Adds records to the file. |
| Clear | Clears specified record values from the panel. |
| Modify | Modifies specified records. |
| Purge | Purges specified records from the file. |
| Browse | Browses records in the file. To include the Browse action, specify a subprogram name in this field. (Use the Browse-Subp or Browse-Select-Subp model to generate the browse subprogram.) |
| Display | Displays specified records. |
| Next | Displays the next record in the file. |
| Recall | Recalls the values for the last record cleared from the panel following a Display, Modify, or Purge action. |
| Former | Displays the contents of the record having the next lower primary key value from the current key value. If no lower value exists, the Start of Data reached message is displayed. |

| Field | Description (continued) |
|---------------------|---|
| | Note: To add user-defined actions, define the SELECT-ADDITIONAL-ACTIONS and ADDITIONAL-ACTIONS-PROCESSING user exits. For more information, see <i>Natural Construct Generation</i> . |
| Push-button support | If this field is marked, actions are presented as cursor-sensitive push buttons. Users can press the Tab key to move from action to action. For more information about implementing actions as push buttons, see the description of the #KD-LINES(*) variable in the following section. |

Variables You Can Use on a Map

You can use the following variables on a map:

| Variable | Format | Definition | Description |
|---------------|--------|-----------------------|--|
| #PROGRAM | A8 | Output | Contains the name of the program that invoked the map. |
| #HEADER1 | A60 | Output | Contains the first heading for the maintenance program. |
| #HEADER2 | A58 | Output | Contains the second heading for the maintenance program. |
| #LEFT-PROMPT | A9 | Output | Indicates the number of panels to the left of the current panel. If the current panel is the leftmost panel, this variable contains the date. |
| #RIGHT-PROMPT | A9 | Output | Indicates the number of panels to the right of the current panel. If the current panel is the rightmost panel, this variable contains the time. |
| #HPARM | A65 | Output/ Nondisplay | Contains the key to the passive help file for the current program (system name concatenated with program name). You can place this variable on the map and pass it to the CD-HELPR help routine. |
| #VAL-ACT | A18 | Output | Contains the list of available actions. |

| Variable | Format | Definition | Description (continued) |
|--|---------------|-------------------|---|
| #VAL-ACT2 | A79 | Output | Contains a second list of available actions (to accommodate long lists). |
| #ACTION | A1 | Modifiable | Contains the action applied to the current record. |
| #DIRECT-COMMAND | A60 | Modifiable | Indicates support for direct command processing in the generated module. |
| #KD-LINE1/ #KD-LINE2/ #KD-LINES(*) | A79 | Output | <p>Contain the names of actions available to the user (Add, Modify, Purge, etc.) or alternate PF-key display formats. Place these variables (supplied in CDDIALDA) either at the top of the map or at the bottom of the map above the standard PF-key lines.</p> <p>To use the push button feature, place these variables on the map(s) with the #KD-LINES-CV control variable and the '00V(NP'02 dynamic attribute. To display the push buttons in red, use the '00VRE(NP'02 dynamic attribute.</p> <p>Note: The CDLAYFM3 layout map supports push buttons.</p> |
| #BKWRD-LAB1 #BKWRD-LAB2 #BKWRD-LAB3 #BKWRD-LAB4 #FRWRD-LAB1 #FRWRD-LAB2 #FRWRD-LAB3 #FRWRD-LAB4 | A2 | Output | <p>Control backward/forward scrolling (supplied in CDKEYLDA). Place these variables on the map(s) next to the fields where you want to enable push buttons for backward and forward scrolling.</p> <p>Note: Include reverse video display among the push button attributes.</p> |
| #LIN | P3 | Output | Contains a single dimension array to display the current index value beside a scroll region (sequential numbers starting from 1). The occurrences of this array match the highest upper bounds value specified for any scroll region. |

| Variable | Format | Definition | Description (continued) |
|-----------------|---------------|-------------------|---|
| #TOP-LINE | P3 | Modifiable | Indicates the index of the first displayed occurrence of the fields in the scroll region. You can change the value of this field to reposition the scroll region to the specified occurrence. |
| #PANEL | N7 | | Determines which panel is displayed when PF10 (left) or PF11 (right) is pressed. For programs containing more than one panel, subtract 1 from #PANEL for PF10 and add 1 to #PANEL for PF11. |
| #LINE-CV(*) | C | | Control variable used for secondary file fields. |

Indexing Fields on a Map

The #LINE variable indexes fields that are scrolled using PF7 and PF8 (the standard forward and backward keys). The upper bounds value for MU/PE fields should match the MAX-OCCURS value for the field as defined in Predict.

Secondary file fields are added to the map as an array. The upper bounds value for these fields should match the value specified in the Scrollable records field on the Secondary File Parameters panel.

The occurrence value specified for all scrolling fields should match the value in the Screen records field on the Secondary File Parameters panel.

If a map is created in the Map editor, you can use the V command to select fields from the database views of the file. If you do so, you must temporarily alter the name of the secondary file fields so you can define them as arrays.

All secondary file fields must be defined with the #LINE-CV(*) control variable. This variable allows the program to only update modified secondary file records.

Secondary File Parameters

Secondary File Parameters for the Maint Model

If the program maintains two files, periodic groups (PEs), multiple-valued fields (MUs), or uses more than one panel of input data, use this panel to specify the secondary parameters. The fields on this panel are:

| Field | Description |
|-------------------|--|
| Horizontal panels | Number of panels required to specify all data in the view. By default, "1" is displayed. (The view may involve either one or two files.) The #PANEL value ranges from 1 to the number specified in this field. This field is used in conjunction with the Input using map field. |
| Note: | If you specify more than one panel, PF10 (left) and PF11 (right) are activated in the generated program to allow the user to move from panel to panel. |

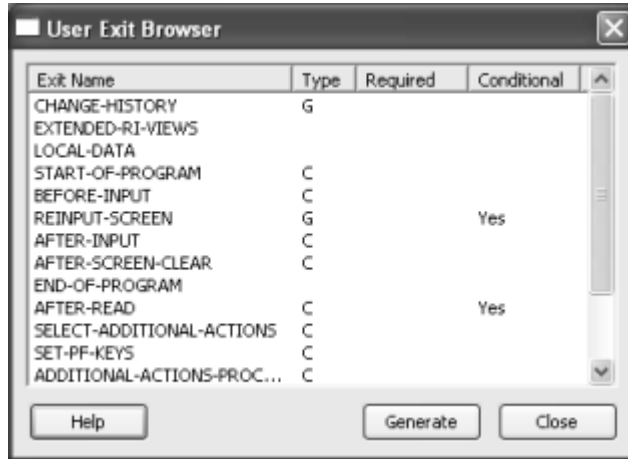
| Field | Description (continued) |
|-------------------------|---|
| Scrollable records | If you specify a secondary file, enter the maximum number of secondary file records that can be read or saved. If scrolling MU/PE fields in the primary file is supported, this value represents the highest value that may be scrolled. (This value does not affect the number of MU/PE occurrences obtained; the MAX-OCCURS value in Predict determines this number.) |
| Scroll lines per screen | Number of MU/PE elements or secondary file records displayed on the panel at one time. |
| Input using map | <p>If you specified a map in the Input using map field on the Additional Parameters panel, the map name is displayed. The field is repeated on this panel in case you want to change the map name.</p> <p>When you specify multiple panels, Natural Construct scans the specified map names for the asterisk (*) character and replaces it with the current panel number. For example, if you specify a TSTMAP* map name and “2” in the Total maintenance panels field, TSTMAP1 and TSTMAP2 are used. The left/right scrolling mechanism (generated into the program) allows windowing between these two maps, starting at TSTMAP1.</p> |
| | <p>Note: To specify any option on the Secondary File Parameters panel, specify a map name.</p> |
| Secondary view | Name of the Predict view that is coupled with the primary file for the maintenance program. A file definition for the file must exist in Predict. |
| Natural (DDM) | Name of the DDM (data definition module) for the secondary file. All fields in the secondary file must be in this DDM. If you do not specify a secondary file DDM, this field defaults to the value in the Secondary view name field. |
| Secondary key | Key in the secondary file that is related to the key in the primary file. The key can be a descriptor, superdescriptor, or subdescriptor. |
| Related keys must match | If this field is marked, the secondary file key must be identical to the primary file key. |

| Field | Description (continued) |
|-----------------------|---|
| Primary key as prefix | <p data-bbox="630 338 1382 432">If this field is marked, the key of the primary file is a prefix of the secondary file key (secondary file records are always displayed in the secondary key order).</p> <p data-bbox="630 453 1382 737">If the primary file key is a prefix of the secondary file key, specify how the relationship between the two files is established by using either the Line Number as Suffix options or the Redefine or Super as Suffix options. The biggest difference between the two options is that for the Line Number as Suffix options, you do not need to enter the suffix value for the secondary key. Because the suffix value is a line number, the suffix is determined during each update session within the generated program.</p> <p data-bbox="630 758 1382 821">For all four options, the secondary file key can be either a descriptor or a superdescriptor.</p> <p data-bbox="528 842 1382 1031">Note: The secondary key suffix may consist of one or more distinct fields that determine the sort sequence of the secondary file records. If this is the case, define a superdescriptor in Predict containing the fields that relate the secondary file to the primary file, followed by the fields that determine the sort order of the secondary records.</p> |
| Natural format | <p data-bbox="630 1062 1382 1209">Natural format and length of the line number (N4, I2, for example). The generated program assumes the secondary file key is made up of the primary file key value, plus a line number. The value of the suffix can be displayed on the panel, but cannot be modified.</p> |
| Remove empty lines | <p data-bbox="630 1251 1382 1335">If this field is marked, the suffix components of the secondary file keys are renumbered (starting at 1) after an occurrence of the view is saved.</p> |
| Save empty lines | <p data-bbox="630 1367 1382 1451">If this field is marked, the suffix components of the secondary file keys are not renumbered after an occurrence of the view is saved.</p> |

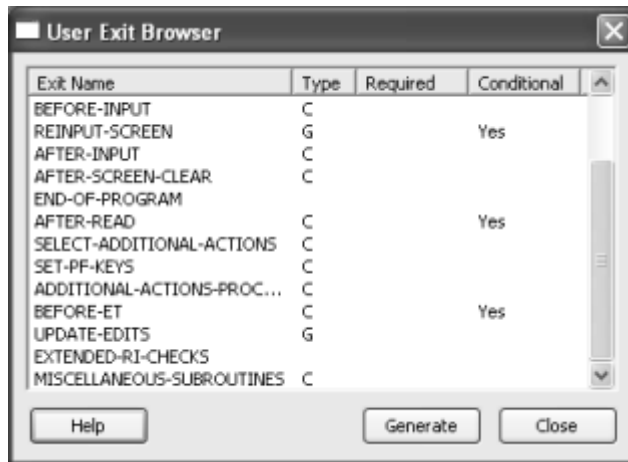
| Field | Description (continued) |
|------------------|---|
| Force uniqueness | <p>If the secondary key suffix is more than a line number, mark this field to use the redefinition of the secondary key field in Predict to determine the suffix.</p> <p>The secondary key suffixes for each secondary file record should be modifiable when displayed on the map. The generated program also ensures a unique secondary file key. If the secondary key is a superdescriptor, place the trailing fields (beyond the primary key length) on the map.</p> |
| Allow duplicates | <p>If this field is marked, the generated program does not check the secondary key for duplicates.</p> |

User Exits for the Maint Model

The following examples show the User Exit Browser for the Maint model:



User Exit Browser — Part 1



User Exit Browser — Part 2

For information about selecting and defining user exits, see **User Exits for the Generation Models**.