

Processing Rules

This section describes the types of processing rule that exist and how to edit them.

- Field-Related Processing Rules
 - Function-Key-Related Processing Rules
 - Processing Rule Editing
-

Field-Related Processing Rules

Three types of field-related processing rules can be defined:

- Inline processing rules
- Predict free rules
- Predict automatic rules

Inline processing rules are defined within a map source and do not have a name assigned. The availability of Predict is not required for inline rules.

Predict free rules have a name assigned and are stored in Predict.

To edit a free rule, enter the rule during map creation and assign a name to it. Inline rules can become Predict rules (and vice versa) by assigning/removing the rule name.

Predict automatic rules apply to database fields and are defined by the Predict administrator. When a field is selected from the data definitions in another Natural object, all automatic rules for that field are linked to the map definition. All automatic rules are concatenated and treated as a single map rule.

The rank of the automatic rules is defined in the map settings (default is 1).

Automatic rules cannot be modified using the map editor. They can, however, be assigned a different rank either by using the command `P=n` or by just overwriting the old rank.

Note:

If a field with linked Predict processing rules is renamed, the rules are lost and must be linked again.

An ampersand (&) within the source code of a processing rule is dynamically replaced by the fully-qualified name of the field for which the rule is defined. Array indexes are not affected by the replacement; you must explicitly specify the index notation after the ampersand (&) as shown in the following example.

Examples:

```
IF &      = ' ' THEN REINPUT 'ENTER NAME' MARK *&    /* For a scalar field
IF &(1) = ' ' THEN MOVE 'X' TO &(*)                /* For an array field
```

The field name notation `&.field-name` within the source code of a processing rule allows you to have DDM-specific rules that cross-check the integrity of values between database fields, without having to explicitly qualify the fields with a view name. As *field-name* you specify the name of the database field as defined in the DDM, and at compilation time, Natural dynamically qualifies the field by replacing the ampersand (&) with the corresponding view name. This allows you to use the same processing rule for specific fields, regardless of which view the fields are taken from.

Function-Key-Related Processing Rules

Two types of function-key-related processing rules can be defined:

- Inline processing rules
- Predict free rules

Function-key-related processing rules can be used to assign activities to program sensitive function keys (PF keys) during map processing. For PF keys which already have a command assigned by the program, this command is executed without any rule processing.

Example:

```
IF *PF-KEY = 'PF3'  
  ESCAPE ROUTINE  
END-IF
```

When this rule is executed, map processing is terminated without further rule processing.

Processing Rule Editing

Processing-rule editing is invoked by the field command `.P`, or by issuing the line command `..E` and then placing the function code `P` next to the field for which processing rule editing is to be performed. PF-key processing rule editing is invoked by the command `..P`.

A parameter can be used (`.Prr`) to indicate the rank (priority) of the processing rule to be defined/edited. A field can have up to 100 processing rules (rank 0 to 99). At map execution time, the processing rules are executed in ascending order by rank and screen position of the field. PF-key processing rules are always assumed to have the first screen position.

For optimum performance, the following assignments are recommended when assigning ranks to processing rules:

Rank	Processing Rule
0	Termination rule
1 - 4	Automatic rules
5 - 24	Format checking
25 - 44	Value checking for individual fields
45 - 64	Value cross-checking between fields
65 - 84	Database access
85 - 99	Special purpose

Note:

When modifying or adding a processing rule, keep in mind that you must recatalog the map(s) that reference this rule so that the changes become effective.

The section below contains information on:

- Selecting a Rule for Editing
- Direct Commands
- Editor Commands for Positioning
- Line Commands

Selecting a Rule for Editing

If you enter the field command `.P*` in a map field, you obtain a list of all processing rules defined for the field.

If you enter the line command `. .P*` in any map line, you obtain a list of all processing rules defined for the PF keys used in this map.

On each list, the Predict rules are identified by their names, the inline rules by their first three source code lines. From each list you can select a rule for editing by entering its rank.

The screen for processing rule editing (with a processing rule example) is shown below:

```

Variables used in current map                                MOD
MODTXT(A3)                                                U
FVAR(A75/1:6)                                             U
FTYP(A1/1:6)                                              U
RULEMODE(A6)                                              U
RULE-NAME(A32)                                           D
FIELDAN(A5)                                               D

Rule _____ Field FULCB3.CBCOM
> > + Rank 0      S 1   L 1   Struct Mode
ALL  ....+....10...+....20...+....30...+....40...+....50...+....60...+....70..
0010 *
0020 IF & EQ MASK('?')
0030 REINPUT USING HELP
0040 END-IF
0050 *
0060
0070
0080
0090
0100
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Mset Exit Test      --   -   +   Full Sc=      Let

```

During processing rule editing you can switch between split-screen and full-screen mode using PF9 or SPLIT/SPLIT E command. The upper half of the split screen displays the definitions of all map fields (except system variables). This display can be positioned by split-screen commands.

The source code used to define the processing rule is entered/edited in the same way as with the Natural program editor.

Direct Commands

While working in the processing-rule editor, processing rules can be edited by entering one of the following commands in the editor command line (>). In the table below, an underlined portion of a keyword represents an acceptable abbreviation. For further explanations of the syntax characters used in this table, refer to *System Command Syntax* in the *System Commands* documentation.

Command	Explanation
<u>ADD</u> [(<i>n</i>)]	Adds <i>n</i> blank lines to the source code. See the more detailed description of the ADD command in <i>Editor Commands</i> in the <i>Program Editor</i> documentation.
<u>C</u> HANGE ' <i>string1</i> ' <i>string2</i> '	Scans for the value entered as <i>string1</i> and replaces each such value found with the value entered as <i>string2</i> .
<u>C</u> CHECK	Checks the rule.
<u>C</u> LEAR	Clears the editing area (including the line markers X and Y).

Command	Explanation
DX or DY or DX-Y	Deletes the X-marked line; or the Y-marked line; or the block of lines delimited by X and Y.
EX or EY or EX-Y	Deletes source lines from the top of the source area to, but not including, the X-marked line; or from the source line following the Y-marked line to the bottom of the source area; or all source lines in the source area excluding the block of lines delimited by X and Y.
EXIT .	Terminates the rule editing function and returns to map editing.
P	Positions forward to the next rule defined for the field.
P*	Selects a rule from the selection menu.
Prr	Selects the rule with rank <i>rr</i> .
P= <i>rr</i>	Changes the rank of a processing rule to rank <i>rr</i> .
<u>P</u> POINT	Positions the line in which the line command .N was entered to the top of the current screen.
<u>R</u> ESET	Deletes the current X and/or Y line markers and any marker previously set with the line command .N.
<u>S</u> AVE <i>name</i>	Saves a rule as copycode with the name <i>name</i> .
<u>S</u> CAN [' <i>scan-value</i> ']	Scans for data in the source area. Entering SCAN without any parameter displays the SCAN/REPLACE menu. Entering SCAN ' <i>scan-value</i> ' results in a scan for <i>scan-value</i> .
<u>S</u> CAN = [+ -]	Scans for the next occurrence of the scan value. The direction of the scan operation is determined by the setting of the direction indicator. See the more detailed description of the SCAN commands in <i>Editor Commands</i> in the section <i>Program Editor</i> .
<u>S</u> HIFT [- + <i>nn</i>]	Shifts each source line delimited by the X and Y markers to the left or right. <i>nn</i> represents the number of characters the source line is to be shifted. Comment lines are not shifted.
<u>S</u> HIFT --	Shifts each source line delimited by the X and Y markers to the leftmost position. Comment lines are not shifted.

Command	Explanation
<u>S</u> HIFT ++	Shifts each source line delimited by the X and Y markers to the rightmost position (maximum 99 positions). Comment lines are not shifted.
<u>S</u> PLIT [E]	Switches between split-screen mode and full-screen mode. See also SPLIT in <i>Editor Commands for Positioning</i> .
<u>T</u> EST	Tests a map.
<u>U</u> NLINK	Unlinks an inline or Predict free rule from the field.

Note:

To select a rule from all Predict free rules, enter a question mark (?) in the rule name field of the processing rule editing screen.

Editor Commands for Positioning

Editor commands for positioning are entered in the command line (>) of the rule editor. The commands available are listed in the following table; an underlined portion of a keyword represents an acceptable abbreviation.

Command	Explanation
+P	Positions forwards one page.
+	
-P	Positions backwards one page.
-	
+H	Positions forwards half a page.
-H	Positions backwards half a page.
<u>T</u> OP	Positions to top of rule.
--	
<u>B</u> OTTOM	Positions to bottom of rule.
++	
+ <i>nnnn</i>	Positions forwards <i>nnnn</i> lines (maximum 4 digits).
- <i>nnnn</i>	Positions backwards <i>nnnn</i> lines (maximum 4 digits).
<i>nnnn</i>	Positions to line <i>nnnn</i> .
X	Positions to the line marked with X.
Y	Positions to the line marked with Y.
<u>S</u> PLIT[- -- + ++]	Positions backwards (- or --) or forwards (+ or ++) in the split screen.

Line Commands

In addition to the editor commands, the following line commands can be used when editing a processing rule:

Command	Explanation
.C (nnnn)	Copies the line in which the command was entered.
.CX (nnnn) or .CY (nnnn)	Copies the X-marked or the Y-marked line. See also the commands .X and .Y in the following section.
.CX-Y (nnnn)	Copies the block of lines delimited by the X and Y markers. If the direction indicator is a plus sign (+), the copied lines are placed after the line in which the command was entered. If the direction indicator is a minus sign (-), the copied lines are placed before the line in which the command was entered.
.D (nnnn)	Deletes line or lines. The default is 1 line.
.I (n)	Inserts <i>n</i> blank lines. With the next ENTER, lines that are left blank are eliminated again.
.I (obj, ssss, nnnn)	Inserts an object contained in the current library or in the steplib into the source. The <i>ssss</i> entry can be used to indicate the line number at which the insert operation is to begin. The <i>nnnn</i> entry can be used to indicate the number of lines to be inserted. For detailed information on the .I line commands, see <i>Line Commands</i> in the section <i>Editing a Map</i> .
.J	Joins the current line with the next line. If the resulting line length is greater than the length of the editor screen line, the line is marked with L and must then be separated again using the .S command (see below), before it can be modified.
.L	Ignores all modifications that have been made to the line since the last time ENTER was pressed.
.MX or .MY	Moves the X-marked or the Y-marked line. See also the commands .X and .Y below.

Command	Explanation
.MX-Y	<p>Moves the block of lines delimited by the X and Y markers.</p> <p>If the direction indicator is set to a plus sign (+), the moved lines are placed after the line in which the command was entered. If the direction indicator is set to a minus sign (-), the moved lines are placed before the line in which the command was entered.</p>
.P	Positions the line marked by this command to the top of the screen.
.S	Splits the line at the position marked by the cursor.
.W	<p>Inserts <i>n</i> blank lines.</p> <p>With the next ENTER, lines that are left blank are eliminated again.</p>
.X	Marks a line, or the beginning of a block of lines, to be processed.
.Y	<p>Marks a line, or the end of a block of lines, to be processed.</p> <p>Note: If both the commands .X and .Y are applied to one line, it is treated as being marked with X and with Y; the line marker actually shown to reflect this status is a Z.</p>