

# Map Editor

The Natural map editor is used to create maps (screen layouts).

A map can be stored in the Natural system file, from where it can be invoked by a Natural program using an INPUT USING MAP statement (for input maps) or a WRITE USING MAP statement (for output maps).

This section covers the following topics:

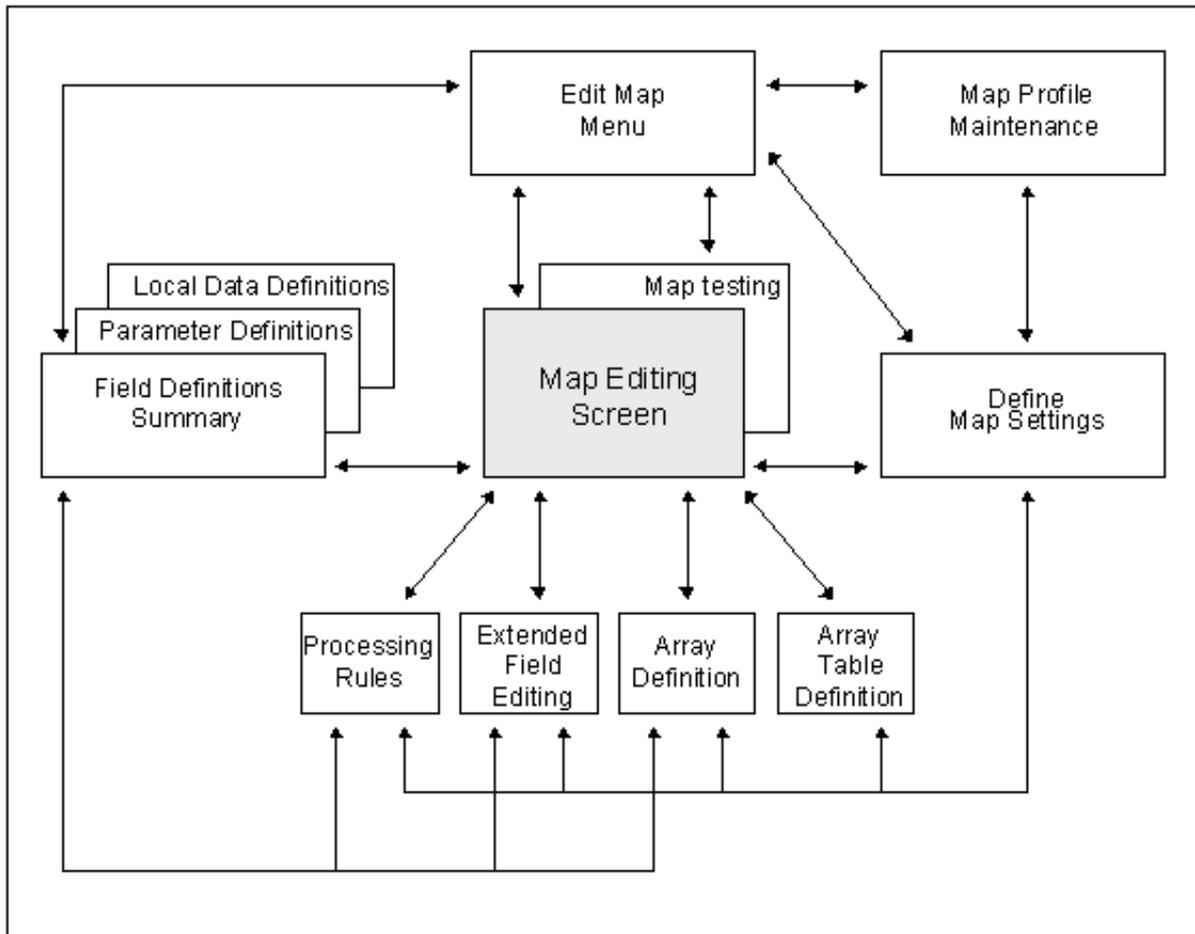
- Components of the Map Editor
- Summary of Map Creation
- Invoking the Map Editor
- Initializing a Map
- Editing a Map
- Defining Map Fields
- Extended Field Editing
- Post Assignment Function
- Array and Table Definition
- Processing Rules

See also:

- [Tutorial - Using the Map Editor](#)
-

# Components of the Map Editor

The following figure provides an overview of the various components of the map editor and also shows the possible ways to get from one component to another:



# Summary of Map Creation

Map creation involves four major steps:

## Step 1

Definition of the map profile (that is, the field delimiters, format settings, context settings and filler characters to be used) by simply selecting the desired settings from a menu.

## Step 2

Definition of the map. A map definition can be created before or after the data views that define the fields it contains. These two ways of creating a map definition are:

- First create a prototype map definition, next create the corresponding data views, then integrate the map into the application.  
Fields can be defined directly on the screen. Each field is assigned a default name. Subsequently, when the corresponding data views have been created, the actual field definitions can be assigned to the map fields (post assignment).
- Create a map definition using existing data views.  
If data views already exist, the map fields can be created by using the field definitions contained in the data views. In this case, all characteristics of a field defined in the data views are included when the field is positioned on the screen.

## Step 3

Definition of the fields to be used in the map. A full set of map editing facilities is provided which permit simple and efficient map field definition:

- Full-screen or split-screen editing. In split-screen mode, the upper half of the screen is used for the display of user views or data definitions and the lower half for map definition. Map fields can be defined directly on the screen or can be selected from a user view or data definition.
- Screen positioning commands.
- Line commands, which are used to define tables and manipulate lines.
- Field commands, which are used to define arrays and manipulate fields.
- Editor facilities, which are used to edit processing (validation) rules.

## Step 4

Storing the map definition. Once created, the map definition can be saved and/or cataloged in the Natural system file. Once saved, a map definition can be read and modified during a subsequent map editor session. Once cataloged, a map definition can be invoked from a Natural program.

### Note:

The map editor uses the Auto Save Numbers function of the program and data area editors.

# Invoking the Map Editor

You invoke the map editor with the system command:

## EDIT MAP

If there is already a map in the source area, the map definition is displayed.

If the source area is empty, the Edit Map menu, which is the main menu of the map editor, is displayed:

```

16:49:52          ***** NATURAL MAP EDITOR *****          2001-01-17
User SAG              - Edit Map -                          Library SYSTEM

          Code      Function
          ----      -
          D          Field and Variable Definitions
          E          Edit Map
          I          Initialize new Map
          H          Initialize a new Help Map
          M          Maintenance of Profiles & Devices
          S          Save Map
          T          Test Map
          W          Stow Map
          ?          Help
          .          Exit

          Code .. I      Name .. _____      Profile .. SYSPROF_

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Help          Exit Test Edit
    
```

The following entries appear on the Edit Map menu:

Entry	Explanation
User	The Natural user ID of the current user.
Library	The Natural library ID currently in effect.
Code	The code of the function to be executed (see below).
Name	<p>The source member which contains the map or help map.</p> <p>For multi-lingual maps, one digit of the source name should be reserved for the language code. For example:</p> <p style="padding-left: 40px;">USERMAP1 (language code is 1)</p> <p>The map above is called from the program by:</p> <p style="padding-left: 40px;">INPUT USING MAP 'USERMAP&amp;'</p> <p>where "&amp;" is replaced with the content of the system variable *LANGUAGE at execution time.</p>
Profile	<p>The session profile currently in effect.</p> <p>The profile name is set to the current library ID. If this profile ID is not available, it is set to the current user ID. If this profile ID is not available, the profile name is set to "SYSPROF".</p>

## Overview of Functions

The following functions appear on the Edit Map menu:

### Field and Variable Definitions

"Field Definitions" displays the following information for each map field:

- Field Name (name of the field)
- Field Mode (type of field), where:
  - D** means Data Area Field,
  - S** means System Variable,
  - U** means User-Defined Field,
  - V** means View Field,
  - blank* means Undefined Field
- Field Format (data type and field length)
- Field is an Array (A) or not ("blank")
- Number of attached Processing Rules
- Line and Column position.

This function is equivalent to line command "..E\*" entered in the first map line.

The following commands are available within the Field Definitions subfunction:

Command	Description
A	Define array
D	Delete field
E	Edit map field
Prr	Edit processing rule
- -	Top
.	Exit

"Variable Definitions" displays all non-map field parameters and all local variables used in the map.

- Name
- Format
- The **Parameter Definitions** function is invoked by pressing PF9 on the Field Definitions screen; new parameters can be added and existing parameters can be modified.
- The **Local Data Definitions** function is invoked by pressing PF10 on the Field Definitions screen; new local variables can be added and existing variables can be modified. Local variables can be used to pass values from one processing rule to another.

The following commands are available within the two Variable Definitions subfunctions:

Command	Description
A	Define array
D	Delete variable
- -	Top
.	Exit

**Note:**

Command "D" does not delete a parameter if this parameter is still applied to any map field as a control variable, start value or help parameter.

## Edit Map

Invokes the map editing screen to modify an existing map or help map definition.

The map editor starts an edit session in split-screen mode, where the upper half of the screen is used for user view definitions and the lower half for map definition. If the map being edited is a help map definition, full-screen mode is in effect.

## Initialize a New Map

This function can be executed only if no object with the same name is stored in the Natural system file.

## Initialize a New Help Map

This function should be used to create a help map, since it offers you the most flexibility when entering and editing text (leading blanks must be entered). It also provides additional checks to ensure that a valid help map is created.

The function can be executed only if no source and no object with the same name is present in the Natural system file.

A help map is stored as a map and can be referenced with the parameter "HE" in the map definition.

When initializing or editing a help map, you can specify in the map settings where the help map is to appear on the screen at execution time.

## **Maintenance of Profiles & Devices**

This function allows you to add, modify or delete session, map and device profiles.

A session profile is used to assign default map settings to be used when a map or a help map is initialized.

A map profile defines the map settings to be in effect during map definition and execution.

A device profile defines the standard characteristics and settings for a device. This profile can be used to ensure compatibility between the map definition and the device to be used.

See also the section Context and setting Device Check.

## **Save Map**

The map definition is stored in source form in the Natural system file.

## **Test Map**

The current map definition is tested to ensure that it can be executed successfully. This includes testing of all processing rules and help facilities.

When testing a map, any additionally created numeric map parameters are initialized with the value 1.

## **Stow Map**

Catalog (and save) a map definition. The map definition is cataloged and also stored in source form in the Natural system file.

## Initializing a Map

This section describes the process of defining the map settings (profile) for a map or help map definition. When you select the function "Initialize New Map" or "Initialize a New Help Map", the first screen to be invoked is the Define Map Settings screen:

```

09:36:47                Define Map Settings for MAP                2001-01-17

Delimiters                Format                Context
-----
Cls Att CD Del          Page Size ..... 23          Device Check .... _____
T   D          BLANK   Line Size ..... 79          WRITE Statement
T   I          ?      Column Shift ... 1 (0/1)      INPUT Statement  X
A   D          _      Layout ..... _____
A   I          )      dynamic ..... N (Y/N)      Help _____
O   D          +      Zero Print ..... N (Y/N)      as field default N (Y/N)
O   I          (      Case Default ... UC (UC/LC)
M   D          &      Manual Skip .... N (Y/N)      Automatic Rule Rank 1
M   I          :      Decimal Char ... .          Profile Name .... SYSPROF
                               Standard Keys .. N (Y/N)
                               Justification .. L (L/R)
                               Print Mode ..... _
                               Control Var .... _____

                               Filler Characters
                               -----
                               Optional, Partial .... _
                               Required, Partial .... _
                               Optional, Complete ... _
                               Required, Complete ... _

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           Exit                               Let
    
```

The Define Map Settings screen comprises the sections:

- Delimiters
- Format
- Context
- Filler Characters

## Delimiters

Delimiters are used as a prefix to a field or a text constant to indicate class, attribute and color to be assigned to the field or text constant.

Valid classes are:

Class	Description
A	Input field
M	Output field which is modifiable
O	Output field which is not modifiable
T	Text constant

Valid attributes are:

Attribute	Description
B	Blinking
C	Cursive/italic
D	Default (non-intensified, non-blinking, etc.)
I	Intensified
N	Non-display
U	Underlined
V	Reversed video
Y	Dynamic (attributes to be assigned dynamically by a program)

Valid colors are:

Abbreviation	Color
BL	Blue
GR	Green
NE	Neutral
PI	Pink
RE	Red
TU	Turquoise
YE	Yellow

Any special character can be defined as a delimiter character - except the control character for terminal commands, the control character for map commands and the decimal notation character.

The *default* delimiter characters and their corresponding class and attribute settings are shown in the following table:

Class (Cls)	Attribute (Attr)	Delimiter (Del)
T (text constant)	D (default)	blank
T (text constant)	I (intensified)	?
A (input only field)	D (default)	_
A (input only field)	I (intensified)	)
O (output only field)	D (default)	+
O (output only field)	I (intensified)	(
M (modifiable field)	D (default)	&
M (modifiable field)	I (intensified)	:

These defaults can be changed by the Natural administrator by creating a session profile SYSPROF. They can be changed by the user by either creating own session profiles or changing map settings during the initialization of the map. This is done by simply entering the desired delimiter value in place of the default assignment.

See the section Defining Map Fields for examples of delimiter usage.

## Format

The following map format settings can be used:

Entry	Explanation
Page Size	The number of map lines to be edited (1 - 250); if Standard Keys is set to "Y", the number of lines is restricted to 3 - 250.  For a map which is output with a WRITE statement, you specify the number of lines of the logical page output with the WRITE statement, not the map size. Thus, the map can be output several times on one page.
Line Size	The number of map columns to be edited (5 - 249).
Column Shift	Column shift (0 or 1) to be applied to the map. This feature can be used to address all 80 columns on a 80-column screen (Column Shift = 1, Line Size = 80). Positional commands (PF10, PF11) must be used to edit all map positions.
Layout	The name of a map source definition which contains a predefined layout.
dynamic	Y Specifies the layout to be dynamic. The dynamically used layout does not become a fixed part of the map at compilation time, but is executed at runtime. Thus, subsequent modifications of a layout map become effective for all maps using that layout map. If the layout map includes user-defined variables, you have to define these parameters in the map using the layout map. Input fields and modifiable fields in the layout map are not open at runtime. Parameters can be added by pressing PF9 within the Field and Variable Definitions function.  N Specifies the layout to be static. The static layout is copied into the source area when a map is initialized. Filler characters are not transferred; "N" is the default setting.
Zero Print	Y Displays a field value of all zeros as one zero only.  N Displays a zero value as blanks; "N" is the default setting.  This value is copied into the field definition when a new field is created and can be modified for individual fields using the extended field editing function.

Entry	Explanation
Case Default	<p>UC Indicates that all input entered for fields at map execution time is to be converted to upper case, that is, the session parameter AD=T is used as a field default.</p> <p>LC Indicates that no lower to upper case conversion is to be performed, that is, the session parameter AD=W is used as a field default. To make the value LC effective, you have to specify the value ON for the Natural profile parameter LC.</p> <p>This value is copied into the field definition when a new field is created and can be modified for individual fields using the extended field editing function.</p>
Manual Skip	<p>Y Does <b>not</b> automatically move the cursor to the next field in the map at execution time even if the current field is completely filled.</p> <p>N Moves the cursor automatically to the next field in the map at execution time when the current field is completely filled; "N" is the default setting.</p>
Decimal Char	The character to be used as the decimal notation character. This character can only be changed with the GLOBALS command.
Standard Keys	<p>Y Leaves the last two lines of the map empty so that function-key specifications can be entered at execution time.</p> <p>N Causes all lines to be used for the map; "N" is the default setting.</p>
Justification	<p>The type of field justification to be used for numeric and alphanumeric fields taken from a user view or data definition:</p> <p>L left justified</p> <p>R right justified</p> <p>This value is copied into the field definition when a new field is created.</p>
Print Mode	<p>The default print mode for variables:</p> <p>C Indicates that an alternative character set is to be used (special character table as defined by the Natural administrator).</p> <p>D Indicates that double byte character mode is to be used.</p> <p>I Indicates inverse print direction.</p> <p>N Indicates standard print direction.</p> <p>This value is copied into the field definition when a new field is created.</p>
Control Var	<p>The name of a control variable, the content of which determines the attribute characteristics of fields and texts that have the attribute definition AD=Y or (Y). The control variable referenced in the map must be defined in the program using that map.</p> <p>Removing a control variable from the format map settings implies that the control variable is removed from the map, too, unless it is associated to any other map field.</p>

## Context

The following map context settings can be used:

Entry	Explanation
Device Check	If a device name is entered in this field, the map settings are checked for compatibility with the device profile of the specified device. If a setting is not compatible, a warning message is issued (see also the section Maintenance of Profiles & Devices).
WRITE Statement	Marking this field with a non-blank value produces a WRITE statement at the end of the map definition process. The resulting map can then be invoked from a Natural program using a WRITE USING MAP statement. Empty lines at the end of the map are automatically deleted so that the map can be output several times on one page.
INPUT Statement	Marking this field with a non-blank value causes the result of the map definition process to be an INPUT statement. The resulting map can then be invoked from a Natural program using an INPUT USING MAP statement.
Help	The name of a helproutine which is invoked at execution time when the help function is invoked for this map (global help for map). For a detailed explanation of the syntax, refer to the specification of the parameter "HE".
as field default	<p>Y Specifies that the helproutine for the map is to apply as default to each individual field on the map, which means that the name of each field is passed individually to the helproutine.</p> <p>N Specifies that the name of the map is passed to the helproutine; "N" is the default setting.</p> <p><b>Note:</b> If you define the map settings for a help map, on the Define Map Settings for HELPMAP screen, the "Help" and "as field defaults" fields are replaced by the "Position Line Col" field described below.</p>
Position Line Col	<p>The position where the help map is to appear on the screen at execution time.</p> <p>This field only appears if you define the map settings for a help map. It replaces the "Help" and "as field defaults" fields on the Define Map Settings for HELPMAP screen.</p>
Automatic Rule Rank	The rank (priority) assigned to automatic Predict rules when they are linked to the map during field definition. Default is 1.
Profile Name	<p>The name of the profile which was active at map initialization time.</p> <p>If "ENFORCED" is displayed, the following map settings are protected:</p> <ul style="list-style-type: none"> <li>● all map delimiters</li> <li>● static and dynamic layout</li> <li>● device check</li> <li>● WRITE and INPUT statements</li> <li>● all filler characters</li> <li>● automatic rule rank</li> <li>● positioning of help maps</li> </ul> <p>The name of the profile active at the time the map is created is stored within the map. When the map is edited later and another profile is active, a warning is produced but editing is allowed.</p>

## Filler Characters

Filler characters can be assigned to indicate whether information for a field is mandatory and whether the field must be completely filled:

<b>Field Type</b>	<b>Explanation</b>
Optional, Partial	Input not mandatory, field need not be completely filled.
Required, Partial	Input mandatory, field need not be completely filled (AD=E).
Optional, Complete	Input not mandatory; if filled, field must be completely filled (AD=G).
Required, Complete	Input mandatory, field must be completely filled (AD=EG).

Filler characters can also be defined for individual fields using the extended field editing function. For definition of field types, see also the session parameter AD as described in the Natural Parameter Reference documentation.

## Editing a Map

The map editor begins an edit session always in split-screen mode, which means that the upper half of the screen is used for user-view definitions and the lower half for map definition:

```

Ob _
.
.
.
.
.
.
001  --010---+-----+-----+---030---+-----+-----+---050---+-----+-----+---070---+-

Ob D CLS ATR DEL          CLS ATR DEL
.   T  D   Blnk         T  I  ?
.   A  D   _           A  I  )
.   A  N   1           M  D  &
.   M  I   :           O  D  +
.   O  I   (
.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Mset Exit Test Edit --  -   +   Full <   >   Let
    
```

PF9 can be used to switch between full-screen and split-screen mode.

If the rightmost of the view definition windows does not contain a view, the current delimiter settings are displayed in this window instead.

Entering a period (.) in the first position of the leftmost view window returns you to the Edit Map menu.

The following commands and functions are available for editing a map:

- Commands and Function Keys for Positioning
- Line Commands
- Field Commands

## Commands and Function Keys for Positioning

The commands and PF keys listed below can be used for map positioning on the screen; you enter the commands at the beginning of a map line:

Key	Command	Function
PF1		Invoke map editor help facility.
PF2		Display/modify the current map settings.
PF3	.Q	Terminate map editing and return to Edit Map menu.
PF4		Test the map definition (without Predict rules).
PF5		Invoke extended field editing for field at which the cursor is currently positioned.
PF6	.- -	Move to top of map.
PF7	.-	Move upwards half a window page.
	.- <i>nnn</i>	Move upwards <i>nnn</i> lines.
PF8	.+	Move downwards half a window page.
	.+ <i>nnn</i>	Move downwards <i>nnn</i> lines.
	++	Move to bottom of map.
PF9	./	Switch between split-screen and full-screen mode.
PF10	.<	Move to the left half a window page.
	.< <i>nnn</i>	Move to the left <i>nnn</i> columns.
	.<<	Move to the left border of the map.
PF11	.>	Move to the right half a window page.
	.> <i>nnn</i>	Move to the right <i>nnn</i> columns.
	.>>	Move to the right border of the map.
PF12		Ignore changes made on screen subsequent to last use of ENTER.
	.*	Move top left corner to cursor position.

## Line Commands

Line commands must be entered in the form "*..line-command*" where "*..*" represents two occurrences of the control character in effect for the map definition.

It is recommended that you enter a blank at the end of each line command. This prevents the editor from attempting to interpret any information existing on the line as part of the line command.

The following line commands are available:

Command	Function
<code>..A</code>	Array table definition.
<code>..An</code>	Array table definition with <i>n</i> occurrences.  This command can be used to create a table with <i>n</i> occurrences vertically for all fields specified in the current line.
<code>..C</code>	Center a single line (that is, the line in which the command was entered).  Two " <code>..C</code> " commands entered on the same screen center the first line and adjust the rest of the selected lines.
<code>..Cn</code>	Center line and move the <i>n-1</i> lines below it accordingly.
<code>..C*</code>	Center line and move all lines below it accordingly.
<code>..D</code>	Delete a single line (that is, the line in which the command was entered).  Two " <code>..D</code> " commands entered on the same screen delete the block of lines delimited by these commands.
<code>..Dn</code>	Delete line and the <i>n-1</i> lines below it.
<code>..D*</code>	Delete line and all lines below it.  If the delete operation affects array elements the array is deleted in total.
<code>..E</code>	Invoke the extended field editing function for all fields contained in the line.  Two " <code>..E</code> " commands entered on the same screen display all fields within the range of lines delimited by these commands for possible extended field editing.
<code>..En</code>	Invoke extended field editing for the line and the <i>n-1</i> lines below it.
<code>..E*</code>	Invoke extended field editing for the line and all lines below it.  The " <code>..E</code> " commands display a screen with the name and format of the requested fields. The field names shown can be modified. The CMD column can be used to select the desired function: extended field editing, array definition and processing rule editing.
<code>..Fc</code>	Fill the empty spaces of a line with the character <i>c</i> .
<code>..I</code>	Insert a single line.  The last empty line on the screen is deleted in order to allow for the line insertion.
<code>..In</code>	Insert <i>n</i> lines below the line in which the command was entered.
<code>..I*</code>	Insert as many lines as possible below the command line.

Command	Function
..J	Join the line in which the command was entered with the line below it.  Two "..J" commands entered on the same screen joins the range of lines delimited by the commands.
..Jn	Join the line in which the command was entered with the $n-1$ lines below it.
..J*	Join the line with all lines below it.  If a join operation results in a line being too long, the lower line is split at the rightmost possible position and the left part is then joined with the previous line. The right part of the split line is then shifted to the left to align it with the line in which the command was entered.
..M	Move the line in which the command was entered below the cursor line.  If two "..M" commands are entered on the same screen, the block of lines delimited by the commands is moved below the line marked with the cursor.
..Mn	Move the line and the $n-1$ lines below it below the line marked with the cursor.
..M*	Move the line in which the command is entered and all lines below it to the line below the line marked with the cursor.  This command is only practical if the line marked with the cursor is above the line in which the command is entered.
..P	Invoke PF-key processing rule editing.  PF-key processing rules are special processing rules to define activities assigned to program sensitive function keys.
..Pn	Invoke PF-key processing rule editing for rank level $n$ .
..Q	Terminate map editing and return to the Edit Map menu.
..R	Repeat once all text constants on the line in which the command was entered. The cursor position is used to indicate the target line.  If two "..R" commands are entered on the same screen, the text constants within the block of lines delimited by the commands are repeated.
..Rn	Repeat all text constants on this and the $n-1$ following lines. If the cursor is located below the command line, the same text is repeated $n$ times.
..S	Split line at cursor position.  If two "..S" commands are entered on the same screen, the block of lines delimited by the commands are split.
..Sn	Split the line where the command is entered and the $n-1$ lines below it at the cursor position.

## Field Commands

Field commands must be entered in the form "*field-command*" where "." represents the control character in effect for the map definition. Each command must begin in the first position of a map field or text constant.

A field command can be applied to a range of fields or constants. A range can be specified in any of the following ways:

- Two or more of the same field commands can be used on the same screen. The column range (horizontal range) and the line range (vertical range) are determined by the positions of the commands. (The section Tutorial - Using the Map Editor provides examples which illustrate this.)
- A repetition factor  $n$  can be used. It can be enclosed within parentheses. The command is applied to the designated field and also to the fields in the  $n-1$  lines below it. A repetition factor of "\*" causes repetition until the bottom of the map is reached.

It is recommended that you enter a blank at the end of each field command. This prevents the editor from attempting to interpret part of the field as part of the field command.

The following field commands are available:

Command	Function
.A	<p>Define an Array. This command can be applied to a single field only and not to a range of fields.</p> <p>The array definition is specified on the screen provided. The resulting array is positioned with its left upper corner at the position where this command was entered.</p> <p>An array can be redefined by applying the ".A" command to one of its elements.</p>
.A> $n$	<p>Supply a repetition factor <math>n</math> with the ".A" command for the purpose of defining a one dimensional array (no spacing, no offsets) without having to use a separate screen.</p>
.C	<p>Center a field or range of fields between adjoining fields.</p> <p>To center a single field, enter ".C" in the field to be centered.</p> <p>To center a range of fields, enter ".C" in the first and last field to be centered, or enter ".C" in the first field and position the cursor to the last field to be centered.</p> <p>In the event that an adjoining field or fields are not present, the column boundaries in effect for the map definition are used instead.</p>
.D	<p>Delete a field or range of fields.</p> <p>To delete a single field, enter ".D" in the field to be deleted.</p> <p>To delete a range of fields, enter ".D" in the first and last field to be deleted. The field range to be deleted may extend beyond a single line. If an array element is deleted, the entire array is deleted.</p>
.E	<p>Invoke extended field editing for a field. This command can be applied only to a single field and not to a range of fields.</p> <p>Extended field editing can also be invoked by positioning the cursor to the selected field and pressing PF5.</p>
.J	<p>Join fields located on consecutive lines.</p> <p>The left boundary of the join operation corresponds to where the ".J" command is entered and the right one corresponds to the cursor position.</p>

Command	Function
.M	<p>Move a field or range of fields.</p> <p>To move a single field, enter ".M" in the field to be moved and place the cursor at the target position.</p> <p>To move a range of fields, enter ".M" in the first and last field to be moved and place the cursor at the target position.</p>
.P[ <i>n</i> ]	<p>Edit processing rules for a field.</p> <p>Supply a parameter <i>n</i> with the ".P" command to indicate the priority (rank) of the processing rule to be edited. If necessary, the value specified for <i>n</i> can be included in parentheses "()".</p>
.R	<p>Repeat (copy) a field or range of fields.</p> <p>To copy a single field, enter ".R" in the field to be copied and place the cursor at the target position.</p> <p>To copy a range of fields, enter ".R" in the first and last field to be copied and place the cursor at the target position.</p> <p>Repetition is always done downwards and from left to right. Fields generated by this command are assigned a dummy name. A valid name for each such field must be defined by using the post assignment function or the extended field editing function.</p> <p><b>Note:</b> Arrays cannot be copied.</p>
.S	<p>Split (move) a line or a line range.</p> <p>Enter ".S" in the field at which splitting is to begin and place the cursor at the target position. The line is divided at the position where the ".S" command was entered. The right portion is then moved to the cursor position.</p>
.T	<p>Truncate (delete) a field or range of fields from a line.</p> <p>Enter ".T" in the field at which truncation is to begin. If this function is used to truncate (delete) an array element, the entire array is deleted.</p>

## Defining Map Fields

The fields which are to comprise a map definition can be specified in any of the following ways:

- Defining Fields Directly on the Screen
- Selecting Fields from a User View or Data Definition
- Using System Variables in a Map Definition

### Defining Fields Directly on the Screen

The fields which are to comprise the map definition are specified by entering a delimiter character followed by the number of positions to be allocated for the field. The following characters can be used:

Character	Meaning
9	Numeric position
0	Numeric right justified
.	Decimal notation (numeric field only)
S	Sign position (numeric field only)
HH	Hexadecimal (binary) (must be entered in groups of two)
X	Alphanumeric position

A repetition factor can also be specified in the form (*n*), for example, "X(5)" is equivalent to "XXXXX".

The following are examples of field definitions (the delimiter character can be changed as desired).

:999	3 positions, numeric
:000	3 positions, numeric right justified
:99.9	3 positions numeric with decimal point
:S9(6)	6 positions, signed numeric
:HHHH	4 positions, hexadecimal
:X	1 position, alphanumeric
:X(7)	7 positions, alphanumeric

Fields entered as shown above are assigned a dummy field name by the map editor. Each field must be assigned a name prior to map execution by using either the extended field editing or post assignment function. Other field formats can be specified using extended field editing.

## Selecting Fields from a User View or Data Definition

A field can be selected from a user view or a data definition. The user view or data definition must first be specified next to the entry "Ob:" (object) on the screen (a second user view can also be specified on the right side of the screen).

To select a user view or data definition, first specify the object class and then the object name. Valid object classes are:

Class	Description
A	Parameter Data Area
C	Predict Conceptual Files (only if Predict is installed)
G	Global Data Area
H	Helproutine
L	Local Data Area
M	Map
N	Subprogram
P	Program
S	Subroutine
V	View

Programs, subroutines, subprograms and helproutines can only be used if they contain a DEFINE DATA statement. Data areas should only be used if they are STOWed.

Once a user view has been selected, it can be positioned forwards or backwards on the screen using positioning commands (+,-,++,--, +n,-n).

To include a user view field in the map definition, enter a delimiter character followed by the number (left-side view) or letter (right-side view) of the desired field. A group or items preceded by a period cannot be selected:

**:3 (field 3 of the left-side view is selected)**

**:C (field C of the right-side view is selected)**

Once all user view fields have been selected, press ENTER to show the fields on the map definition. If a selected field contains an edit mask, this is denoted by the notation "M".

The user view field name is used as the map field name for fields selected from a user view.

## Using System Variables in a Map Definition

Natural system variables can also be specified in a map definition. The Natural Programming Reference documentation contains a complete description of Natural system variables.

A system variable must be preceded by an output delimiter:

**(\*TIME**  
**(\*DATE**  
**(\*APPLIC-ID**

## Extended Field Editing

Extended field editing is used to define field attributes.

Extended field editing is invoked by entering the line command ".E" or the field command ".E"; the following screen is displayed:

```

Fld START-NAME                                     Fmt A8
-----
AD= MIT'_'_____  ZP=          SG=          HE= _____  Rls 0
AL= _____      CD=  ___      CV= _____  Mod User
PM=  ___  DF=          DY= _____
EM= _____

001  --010---+-----+-----+---030---+-----+-----+---050---+-----+-----+---070---+-----

      Please enter starting name .EXXXXXXXX

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  <---  --->  --  -  +  <  >  Let

```

It is possible to invoke extended field editing for the next or previous field in the map by pressing PF4 or PF5 respectively, or to invoke extended field editing for any field in the map by moving the cursor onto the desired field and pressing ENTER.

The sample screen above contains the following entries:

Entry	Explanation
Fld	<p>The field or array name.</p> <p>Field name assignment is related to the method with which the field was originally defined.</p> <p>If the field was taken from a user view or data definition, it is assigned the same name as the field in the user view or data definition.</p> <p>If the field was specified as a Natural system variable, it is assigned the name of the specified variable.</p>
Arr	<p>If the field is neither of the above, it is assigned a dummy name. You must assign a name to such a field prior to map execution. The name of a field can be changed. However, a prefix cannot be used for a field which did not have a prefix assigned previously. To obtain a prefixed field name, select the field from a user view or data definition.</p> <p><b>Note:</b> Duplicate field names are only allowed for fields defined as "output only fields".</p> <p>See the section Defining Map Fields for additional information.</p>
Fmt	<p>The format and length of the field.</p> <p>These can be changed by overwriting the current entry.</p>
AL/FL/NL	<p>The length to be used when displaying the field.</p>
Rls	<p>The number of processing rules currently defined for the field.</p>
ZP	<p>Zero printing.</p> <p>OFF indicates that zero values for the field are <b>not</b> to be printed.</p> <p>ON indicates that zero values are to be printed.</p> <p>ZP appears on the screen only if the field is numeric.</p>
SG	<p>Sign position for numeric fields.</p> <p>OFF indicates that <b>no</b> sign position is to be allocated (default).</p> <p>ON indicates that a sign position is to be allocated.</p> <p>SG appears on the screen only if the field is numeric.</p>
PM	<p>Print Mode.</p> <p>C indicates that an alternative character set is to be used (as defined by the Natural administrator).</p> <p>D indicates that double-byte character set is to be used.</p> <p>I indicates inverse print direction.</p> <p>N indicates that it is not possible to print a hardcopy of the field content.</p>

Entry	Explanation
DF	<p>Date format (applies only to date fields):</p> <p>Determines the length of a date when converted to alphanumeric representation without an edit mask being specified:</p> <ul style="list-style-type: none"> <li>S 8-byte representation with 2-digit year component and delimiters (<i>yy-mm-dd</i>).</li> <li>I 8-byte representation with 4-digit year component and no delimiters (<i>yyyymmdd</i>).</li> <li>L 10-byte representation with 4-digit year component and delimiters (<i>yyyy-mm-dd</i>).</li> </ul> <p>For further information, see the session parameter DF as described in the Natural Parameter Reference documentation.</p>
DY	<p>Dynamic string attributes.</p> <p>The dynamic string parameter is used to define certain characters contained in the text string of an alphanumeric variable to control the attribute setting. See also the session parameter DY as described in the Natural Parameter Reference documentation.</p>
HE	<p>The name of a helproutine to be assigned to the field.</p> <p>For the syntax of the HE parameter, see below.</p> <p>For a detailed explanation of the operands used in the HE option, see the session parameter HE as described in the Natural Parameter Reference documentation.</p> <p><i>Operand1</i> can be the name of a helproutine specified in single quotes (') or a variable name.</p> <ul style="list-style-type: none"> <li>● If a field with the name specified as <i>operand1</i> in the HE option exists as a field of a map, the parameter references this field.</li> <li>● If no field with that name exists in the map, the parameter is defined as A8 (default format assumed) in the map.</li> </ul> <p>The format/length of <i>operand2</i> is defined in the following way:</p> <ul style="list-style-type: none"> <li>● If the parameter specified as <i>operand2</i> in the HE option is defined as a field of a map, the parameter references this field.</li> <li>● If no field with that name exists, the parameter is defined as N7 (default format assumed) in the map.</li> </ul> <p>Removing a parameter from the HE option implies that the parameter is also removed from the map, unless it is a map field or it is associated with any other map field as a help parameter or "Starting from" value.</p> <p>Non-map field parameters can be edited in the Field and Variable Definitions screen using PF9.</p> <p>Entering "HE=+" opens a window, which provides sufficient space for specifying multiple parameters to be passed to a helproutine.</p>
AD	<p>Field attributes.</p> <p>For source optimization reasons, the default values "D", "H", "F" and "W" are accepted but not retained (see also session parameter AD as described in the Natural Parameter Reference documentation).</p>
CD	<p>Color attributes.</p>

Entry	Explanation
CV	<p>Control variable for dynamic field attributes.</p> <p>The name of a variable which contains the attributes to be used for this field. This variable must be defined with format C in the program.</p> <p>The control variable also contains a MODIFIED data tag, which indicates whether the field has been modified following map execution.</p> <p>A single control variable can be applied to several map fields, in which case the MODIFIED data tag is set if any of the fields referencing the control variable has been modified.</p> <p>The control variable can be expanded up to three dimensions, for example, CONTR(*), CONTR(*,*), CONTR(*,*,*), depending on the rank of the corresponding array.</p> <p><b>Note:</b> Removing a control variable from a field implies that the control variable is removed from the map, too, unless it is associated to any other map field.</p>
EM	Edit mask to be used for the field.
MODE	<p>Mode indicates how the field was created:</p> <p>DATA The field was created by selecting a field from a DEFINE DATA definition.</p> <p>SYS The field is a system variable.</p> <p>UNDEF The field was created directly on the screen and has a dummy name.</p> <p>USER The name of the field was created by extended field editing.</p> <p>VIEW The field was created by selecting a field from a view (file).</p>

### HE Parameter Syntax:

$$HE = operand1 \left[ , \left\{ \begin{matrix} operand2 \\ = \end{matrix} \right\} [, operand2] \dots 19 \right]$$

Operand	Possible Structure		Possible Formats										Referencing Permitted	Dynamic Definition			
Operand1	C	S														no	no
Operand2	C	S														no	no

## Post Assignment Function

A field which has been previously defined (in layout) directly on the screen can be assigned the field name and field attributes of a user view field or a DEFINE DATA definition.

**Note:**

Duplicate field names are only allowed for fields defined as "output-only fields".

A map field which has been created using a DDM field definition can be redefined using the field definition from a view defined in a data area.

Post assignment can be done by entering the user view field number (or letter) as shown in the view window directly behind the delimiter of the field.

This function can only be used if the formats of the layout agrees with the field definition. N and P are considered to be identical numeric.

This function cannot be used for view arrays if one or more dimensions of that array are smaller than the dimensions of the array in the layout.

If a length conflict occurs, an AL/FL/NL attribute is generated to map the field definition to the layout definition with truncation or expansion. Data are truncated when AL/FL/NL is specified.

# Array and Table Definition

The array definition function is used to define the occurrences and layout of an array.

Array definition is initiated by the field command ".A" or by issuing the line command ".E" and then marking the desired field with the function code "A".

The table definition function is used to define the occurrences and layout of more than one array at the same time. The arrays must begin in the same map line.

Table definition is invoked by the line command "..A".

Below you will find information on:

- Array Definition
- Table Definition

## Array Definition

The upper portion of the following screen is displayed for the purpose of array definition:

```

Name #001                                     Upper Bnds 1___ 1___ 1___
-----
Dimensions          Occurrences  Starting from  Spacing
0 . Index vertical   1___          _____  0 Lines
0 . Index horizontal 1___          _____  1 Columns
0 . Index (h/v) V    1___          _____  0 Cls/Ls

001  --010---+-----+-----030---+-----+-----050---+-----+-----070---+-----

                Please enter starting name .AXXXXXXXXXXXXXXXXXXXXX

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Mset  Exit          --   -   +           <   >       Let
    
```

You can specify the following:

Entry	Explanation
Upper Bnds	<p>Indicates the upper bounds of the array; that is, the highest occurrence in (from left to right) the first, second and third dimension.</p> <p>If a field defined in a program is used to define the map array, the upper bounds of that field (user-defined variable or database field), as defined in the program, are used; these cannot be overwritten on the array definition screen.</p> <p>If the map array is derived from a user view array or a data definition, the dimensions of the map array must not exceed the dimensions shown in this field.</p> <p>If the map array is not derived from a user view array or a data definition, the dimensions of the map array must not exceed the dimensions as defined in the Natural program.</p>
Dimensions	<p>An array can have up to three dimensions. The order in which the dimensions of the array are mapped to the map layout is determined by the values entered to the left of the Index operands.</p>
Occurrences	<p>The number of occurrences to be defined for a dimension.</p>
Starting From	<p>The starting index value for a dimension. A numeric value can be used, or a variable name can be used to indicate that the actual value is supplied in the Natural program which invokes the map definition.</p> <p>If the variable is not defined otherwise as a field in the map, it is assumed to be of format/length N7. If so, it can be edited using PF9 in the Field and Parameter Definition screen.</p> <p><b>Note:</b> Removing a "Starting from" value from an array implies that the variable is removed from the map, too, unless it is a map field or it is associated to any other map field as a "Starting from" value or help parameter. To edit "Starting from" values press PF9 in the Field and Variable Definitions screen.</p>
Spacing	<p>The number of blank lines (for vertical dimensions) or blank columns (for horizontal dimensions) to be inserted between each dimension occurrence.</p>

## Examples of Array Definitions

### Example 1:

A one-dimensional array consisting of 10 vertical occurrences with 2 blank lines to be inserted between each occurrence.

Name #001		Upper Bnds 10	1	1
-----				
Dimensions	Occurrences	Starting from	Spacing	
1 . Index vertical	10	_____	2	Lines
0 . Index horizontal	1	_____	1	Columns
0 . Index (h/v) V	1	_____	0	Cls/Ls

### Example 2:

Same as example 1 except that the array is to be horizontal.

Name #001		Upper Bnds 10	1	1
-----				
Dimensions	Occurrences	Starting from	Spacing	
0 . Index vertical	1	_____	0	Lines
1 . Index horizontal	10	_____	1	Columns
0 . Index (h/v) V	1	_____	0	Cls/Ls

### Example 3:

A two-dimensional array. The first dimension consists of 10 vertical occurrences with 1 blank line between each occurrence. The second dimension consists of 5 horizontal occurrences with 2 blank columns between each occurrence.

Name #001		Upper Bnds 10	5	1
-----				
Dimensions	Occurrences	Starting from	Spacing	
1 . Index vertical	10	_____	1	Lines
2 . Index horizontal	5	_____	2	Columns
0 . Index (h/v) V	1	_____	0	Cls/Ls

**Example 4:**

Same as example 3 except that the order of the dimensions is reversed.

Name #001		Upper Bnds	5____	10____	1____
-----					
Dimensions		Occurrences	Starting from	Spacing	
2 . Index vertical		10_	_____	1	Lines
1 . Index horizontal		5_	_____	2	Columns
0 . Index (h/v) V		1_	_____	0	Cls/Ls

**Example 5:**

A three-dimensional array. The first dimension consists of 3 vertical occurrences with 1 blank line between each occurrence. The second dimension consists of 5 horizontal occurrences with 2 blank columns between each occurrence. The third dimension consists of 2 occurrences, expanded vertically within each occurrence of the first dimension.

Name #001		Upper Bnds	3____	5____	2____
-----					
Dimensions		Occurrences	Starting from	Spacing	
1 . Index vertical		3_	_____	1	Lines
2 . Index horizontal		5_	_____	2	Columns
3 . Index (h/v) V		2_	_____	0	Cls/Ls

**Example 6:**

An example using "Starting from". The first dimension consists of 10 vertical occurrences starting from index I. 'I' is defined in the map editor with format/length N7 by default. The second dimension consists of 5 horizontal occurrences starting from the index 3.

Name #001		Upper Bnds	10____	5____	1____
-----					
Dimensions		Occurrences	Starting from	Spacing	
1 . Index vertical		10_	I_____	1	Lines
2 . Index horizontal		5_	3_____	2	Columns
0 . Index (h/v) V		1_	_____	0	Cls/Ls

**Example 7:**

An example of making a two-dimensional display from a one-dimensional array. The array consists of 40 elements. It is displayed in two columns with 20 lines each. This is achieved by specifying 0 as the horizontal index.

Name #001		Upper Bnds	40____	1____	1____
-----					
Dimensions		Occurrences	Starting from	Spacing	
1 . Index vertical		20_	_____	0	Lines
0 . Index horizontal		2_	_____	10	Columns
0 . Index (h/v) V		1_	_____	0	Cls/Ls

## Table Definition

A table of one or more arrays which all begin in the same map line is defined with the "..A" line command. When you enter the "..A" line command, the following screen is invoked:

```

16:52:39          ***** NATURAL MAP EDITOR *****          2001-01-17
                   - Array Table Definition -

Main  Index:  Vert. Occur.  1      Starting from _____ Spacing 0  Lines
Second Index:  Direction(H/V) V      _____              0  Cls/Ls
Third  Index:  Direction(H/V) V      _____              0  Cls/Ls
-----
Name of Variable          Col  Dimension Size      Order  2.Ind  3.Ind
(truncated)              Pos  Ind1  Ind2  Ind3  M S T  Occ.  Occ.
-----
#001                      2   1    1    1
#002                      25  1    1    1

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit          --   -   +                               Let

```

### Note:

Applying the "..A" command to arrays which were not defined by an "..A" command but by an ".A" command, may result in a modification or even a destruction of these arrays.

The sample screen above contains the following entries:

<b>Entry</b>	<b>Explanation</b>
Main Index	The number of vertical occurrences, the starting position and the number of lines to be skipped between each dimension occurrence.
Second Index	<p>The direction (horizontal or vertical), the starting position and the number of lines/columns to be skipped between each dimension occurrence.</p> <p>The second dimension only applies if one of the arrays has more than one dimension. In this case the second dimension can be displayed either horizontally (in which case there must be enough space in the line for all selected occurrences) or vertically (in which case there must be enough lines on the map to display main dimension times second dimension occurrences, including line spacing).</p>
Third Index	<p>The direction (horizontal or vertical), the starting position and the number of lines/columns to be skipped between each dimension occurrence.</p> <p>The third dimension only applies if one of the arrays has more than two dimensions. In this case the third dimension can be displayed either horizontally (in which case there must be enough space in the line for all selected occurrences) or vertically (in which case there must be enough lines on the map to display main dimension times second dimension times third dimension occurrences, including line spacing).</p>
Name of Variable	All names of field arrays contained in the table are displayed.
Col Pos	The column position in which the field is located. This is displayed for informational purposes only.
Dimension Size	The size of the array as defined in a user view or data definition, or as in a Natural program. If the map array is derived from a user view array or data definition, the dimensions of the map array must not exceed the dimensions shown in this field. If the map array is not derived from a user view array, the dimensions of the map array must not exceed the dimensions as defined in the Natural program.
Order	The order in which the dimensions are to be defined. M, S and T correspond to Main, Second and Third.
2nd Ind Occ.	The number of occurrences to be defined for the second index.
3rd Ind Occ.	The number of occurrences to be defined for the third index.





## Processing Rules

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

### Field-Related Processing Rules

Three types of processing rules can be defined:

- Inline processing rules
- Free Predict rules
- Automatic Predict 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.

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

To edit a free processing 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 created by selecting it from a view or a data definition, and if the field is a database field, 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 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.

If Predict rules are modified subsequently by the Predict administrator, or new automatic rules are linked to a database field, or automatic rules are removed, it is sufficient to recatalog the map.

**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 substituted with the fully qualified name of the field using the rule; this does not apply if the rule is used by individual array elements.

**Example:**

```
IF & = ' ' REINPUT 'ENTER NAME' MARK *&
```

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 "&" 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
- Free Predict rules

Function-key-related processing rules can be used to assign activities to program sensitive function keys during map processing. For function 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

### How to Select 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 function-key-related processing rules defined for the 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.

While working in the processing-rule editor, processing rules can be edited by entering the following commands in the editor command line:

<b>Command</b>	<b>Function</b>
<u>A</u> DD[( <i>n</i> )]	This command adds <i>n</i> empty lines in source code.  See the more detailed description of the ADD command in the section Editor Commands.
<u>C</u> HANGE ' <i>string1</i> ' <i>string2</i> '	This command 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> HECK	This command checks the rule.
<u>C</u> LEAR	This command clears the edit area (including the line markers "X" and "Y").
DX, DY, DX-Y	This command deletes the X-marked line; or the Y-marked line; or the block of lines delimited by "X" and "Y".
EX, EY, EX-Y	This command 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	This command terminates the rule editing function and return to map editing.
.	
P	This command positions forward to the next rule defined for the field.
P*	This command selects a rule from the selection menu.
P <i>rr</i>	This command selects the rule with rank <i>rr</i> .
P= <i>rr</i>	This command changes the rank of a processing rule to rank <i>rr</i> .
<u>P</u> OINT	This command positions the line in which the line command ".N" was entered to the top of the current screen.
<u>R</u> ESET	This command 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>	This command saves a rule as copycode with the name <i>name</i> .
<u>S</u> CAN [ <i>'scan-value'</i> ]	This command scans for data in the source area. Entering SCAN without any parameter displays the SCAN menu. Entering SCAN ' <i>scan-value</i> ' results in a scan for <i>scan-value</i> .
<u>S</u> CAN = [+ -]	This command 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 the section Editor Commands.
<u>S</u> HIFT [- + <i>nn</i> ]	This command 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 - -	This command shifts each source line delimited by the X and Y markers to the leftmost position. Comment lines are not shifted.
<u>S</u> HIFT ++	This command shifts each source line delimited by the X and Y markers to the rightmost position (maximum 99 positions). Comment lines are not shifted.

<b>Command</b>	<b>Function</b>
<u>S</u> PLIT [ <u>E</u> ND]	This command switches between split-screen mode and full-screen mode (see also the section Split-Screen Commands).
<u>T</u> EST	This command tests a map.
<u>U</u> NLINK	This command unlinks an inline or Predict free rule from the field.

**Note:**

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

## Editor Commands for Positioning

Editor commands for positioning are also entered in the command line of the rule editor. The following commands are available:

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

In split screen mode, all positioning commands must be preceded by an "S" (for Split Screen). See further information in the section Split Screen Commands.

## Line Commands

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

Command	Function
.C( <i>nnnn</i> )	Copies the line in which the command was entered.
.CX( <i>nnnn</i> ) .CY( <i>nnnn</i> )	Copies the X-marked or the Y-marked line. See also the commands ".X" and ".Y" in the following section.
.CX-Y( <i>nnnn</i> )	Copies the block of lines delimited by the X and Y markers.  If the direction indicator is "+", the copied lines are placed after the line in which the command was entered. If the direction indicator is "-", the copied lines are placed before the line in which the command was entered.
.D( <i>nnnn</i> )	Delete line or lines. The default is 1 line.
.I( <i>n</i> )	Inserts <i>n</i> empty lines. With the next ENTER, lines that are left blank are eliminated again.
.I( <i>obj,ssss,nnnn</i> )	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.  See the more detailed description of the .I line commands in the section Line Commands.
.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	Undoes all modifications that have been made to the line since the last time ENTER was pressed.
.MX .MY	Moves the X-marked or the Y-marked line. See also the commands ".X" and ".Y" below.
.MX-Y	Moves the block of lines delimited by the X and Y markers.  If the direction indicator is set to "+", the moved lines are placed after the line in which the command was entered. If the direction indicator is set to "-", the moved lines are placed before the line in which the command was entered.
.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	Inserts <i>n</i> empty lines.  With the next ENTER, lines that are left blank are eliminated again.
.X	Marks a line, or the beginning of a block of lines, to be processed.
.Y	Marks a line, or the end of a block of lines, to be processed.  <b>Note:</b> 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".