

# Map Editor

- Summary of the Map Creation Process
- Map Fields
- Invoking the Map Editor
  - Creating a New Map
  - Editing an Existing Map
- Map Editor Menu
- Creating a Text Constant
- Creating a User-Defined Variable
  - Using Natural System Variables in a Map
- Modifying a User-Defined Variable - Field Editing
  - Rule Editing - Processing Rules
  - Array Editing
  - AD - Attribute Definition
- Selecting Fields from a DDM or User View
- Defining Fields for a Parameter or Local Data Definition
  - Parameter Definitions
  - Local Data Definitions
- Map Profile
  - Map Profile Settings
  - Filler Characters
- Post Assignment
- Field-Sensitive Processing
  - Advantages of Field-Sensitive Processing
  - Defining a Map as Field-Sensitive

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

A map can be stored in a Natural library, 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).

## Summary of the Map Creation Process

There are four major steps involved in the creation of a map:

1. Definition of the map profile (that is, the format settings and filler characters to be used). A menu is provided from which you select the desired items.
2. Definition of the map. A map can be created in two different ways:
  - First create a prototype map, 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.
  - Create a map 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.
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 field definition.
4. The saving and/or cataloging of the map definition. Once defined, the map can be saved in source form and/or object form in a Natural library. Once saved, a map definition can be read and modified during a subsequent map editor session. Once stowed, a map definition can be invoked from a Natural program.

## Map Fields

A map consists of fields. A field can be either a text constant or a data variable. The fields which are to comprise a map definition can be specified in any of the following ways:

- The field can be defined directly on the screen as a text constant, or a user-defined variable.
- The field can be selected from a user view or data definition for any existing Natural object type in the library.
- A Natural system variable can be used.

Data variables can be either system variables or data copied from a view or data area. These variables can also be user-defined or copied from other source objects.

The following sections describe each of these methods in detail.

## Invoking the Map Editor

You invoke the Natural map editor to create a new map or edit an existing one as follows:

### Creating a New Map

On the Natural Main Menu, select "Direct" and press ENTER to invoke the Direct Command window.

**Note:**

You can also invoke the Direct Command window by selecting the <DIRECT COMMAND> from the library selection list that is displayed when you select "Library" from the Natural Main Menu.

In the Direct Command window, enter the following command:

**EDIT MAP or in short form: E M**

Press ENTER. The Map Editor Menu will be displayed:

```

.....
                                NATURAL MAP EDITOR (Esc to select field)      .
Create   Modify   Erase   Drag   Info OFF  Lines   Ops. Map  Quit   .
.....

```

### Editing an Existing Map

On the Natural Main Menu, select "Direct" and press ENTER to invoke the Direct Command window.

In this window, enter the following command:

**EDIT *map-name* or in short form: E *map-name***

If you do not remember the name of the map you wish to edit, select "Library" on the Natural Main Menu and press ENTER. A list of all available libraries will be displayed.

From the list, select the desired library with the cursor, and press ENTER. A list of all objects in this library will be displayed.

**Note:**

File designations are listed under the column "Pgm. Type".

With the cursor keys, scroll through the list until the desired map appears on the list. Then mark it with an "E" (Edit) in the column before the object names and press ENTER. The map editor will then be invoked for the selected map.

**Note:**

The F2 key invokes a window listing valid functions (for example, C=Check, D=Read, E=Edit) for this specific object for the item Ops Map.

Regardless of the map invoked, you are prompted at the bottom of the screen to use the cursor keys to select a field and to press ENTER. The selected field is highlighted.

**Note:**

The rightmost position of the bottom line displays the column and row number of the highlighted field.

Press ESC and the Map Editor Menu is invoked.

**Note:**

To change the programming mode (structured/reporting) for an existing map, enter the following commands in the Direct Command window:

READ map-name

GLOBALS SM=ON/OFF

SAVE map-name.

## Map Editor Menu

The Map Editor Menu, as shown above, is the main menu of the map editor. The following items can be selected from this menu:

- Create
- Modify
- Erase
- Drag
- Info ON/OFF
- Lines
- Ops. Map
- Quit

### Create

If you press ENTER, a list containing the following items is displayed:

- **A** - Parameter Data Area
- **G** - Global Data Area
- **H** - Helproutine
- **L** - Local Data Area
- **M** - Map
- **N** - Subprogram
- **P** - Program
- **S** - Subroutine
- **T** - Text Constant
- **U** - User Defined
- **V** - View Defined
- **1** - Parameter Defined
- **2** - Local Defined

These items are used to select fields/variables from an object for an object of this class. The object class can also be invoked by entering "C" and the object class abbreviation (key-sensitive). For more detailed information, see the section Selecting Fields from a DDM or User View.

### Modify

Modify enables you to modify a selected field. The selected field is the current field and it is highlighted.

A window displaying field attributes (extended field editing) is displayed in which the contents of these attributes can be modified.

### Erase

Erase enables you to delete the current field. You are then prompted "Delete field (Y/N)?"

All responses are key-sensitive. Caution is recommended!

If the current field is an array field, the entire array (not just the field) will be deleted.

## Drag

Drag enables you to move the current field to any unoccupied position on the screen. The selected field can be moved without restriction, using the cursor.

Once the field is positioned and ENTER is pressed, this field position takes effect.

## Info ON/OFF

Info ON/OFF is used to switch the display of the field information window ON and OFF (toggle switch). OFF is the default value.

The ENTER key or "I" (key-sensitive) is used to switch between ON and OFF.

## Lines

Lines invokes a selection list from which you can select the following line-specific functions:

Insert After
Erase Line
Copy After
Duplicate Line
Move After
Split Line
Join Line

These functions can be selected to perform operations on an entire line (not a single field) in a map. All operations are performed on the current line.

### Note:

These functions are self-explanatory and prompts appear at the bottom of the screen for each item selected.

## Ops. Map

Ops. Map (map operations) invokes the following selection list:

C Check Map
E Edit Map
K Key Rules
L List Map
P Prof. Map
R Read Map
S Save Map
T Test Map
W Stow Map

The list contains the following items:

<b>Check Map</b>	Causes syntax checking and generation of source code.
<b>Edit Map</b>	Invokes the map editing screen for modification of an existing map definition. The map editor will start a new edit session.
<b>Key Rules</b>	Invokes editing of function-key-related processing rules.
<b>List Map</b>	Generates source code and lists it.
<b>Prof. Map</b>	Invokes a map profile window, which is described under Map Profile.
<b>Read Map</b>	Invokes the map editing screen to read an existing map definition.
<b>Save Map</b>	Performs a source code generation check and then saves the map. The map definition is saved in source form in the Natural library.
<b>Test Map</b>	The current map definition is tested to ensure that it can be executed successfully. This includes testing of all processing rules and help facilities.
<b>Stow Map</b>	Performs a source code generation check, as well as a save and catalog of a map definition. The map definition is cataloged and also saved in source form in the current Natural library.

## Quit

Quit terminates the map editor session.

If you have edited the map (not saved), selected "Quit" and pressed ENTER, the following prompt appears:

```
Modifications have not been saved, quit anyway Y/N ?
```

### Note:

Replies are key-sensitive; caution is recommended.

If the editor session is terminated, the Natural Main Menu is redisplayed.

## Creating a Text Constant

Select "Text Constant" from the Create list and press ENTER.

Depending on the map type invoked, a screen appears.

At the bottom of the screen, you are always prompted to position the cursor and enter text.

Position the cursor to the start of an empty field.

**Note:**

You cannot overwrite existing fields.

Enter the text. The first character entered causes the line to be highlighted. Highlighting indicates the maximum space available for text entry. Characters can be entered or deleted until you press ENTER.

**Note:**

The ESC key cancels text entry.

Press PF2 to select an attribute and color to be used for the text entered. Use the UP/DOWN ARROW keys to scroll through and select one of the available attributes/colors or simply use the corresponding abbreviation (for example, B=Blinking or RE=Red) and press ENTER. Using the LEFT/RIGHT ARROW keys, you can toggle between attribute and color definition.

Text entry is now complete.

**Note:**

If you want to create and/or define a data variable after having defined the text, see the section Using Natural System Variables in a Map.

## Creating a User-Defined Variable

Select "User Defined" from the field list and press ENTER.

The following window appears:

```

Extended Field Editing
Field :
Format: A Len:          AL:          PM:          ZP: N  SG: N
Rules : 0 Rule Editing: N Array:      Array Editing: N Mode:
AD:          CD:          CV:          DY: N  HE: N
EM:

```

A message appears at the bottom of the screen, prompting you to:

```

Position cursor and press Enter or format char.

```

Position the cursor to the start of a field position and press ENTER.

A line containing the maximum available length is displayed as well as a selection list of the valid data variable types.

### Note:

If the data type is known, the type character can be entered directly. The display of available data types is thus avoided.

Select the data variable and press ENTER. The field attributes window appears in the bottom portion of the screen.

The format is now the data type entered and a default name (for example, "#1") is assigned to the field.

There are two length fields displayed on this screen:

1. ... AL (Alphanumeric length) The display length.
2. Len The internal length of the data type.

The lengths of user-defined variables are defined by performing the following steps:

1. Enter the length of the first field (for example, Alphan. Len... ).
2. When the field length definition is complete, press ENTER.
3. You are then prompted to enter a field name for the variable. Having selected the appropriate field name, press ENTER.

The definition function for the first field is now complete.

The cursor moves automatically to the second length field (Len... ). Change the length field definition or use the TAB key. Continue this process to define all other pertinent information to be used for the field being defined.

For further information on these fields, see [Modifying a User-Defined Variable - Field Editing](#).

When this definition is complete, press ENTER.

## Using Natural System Variables in a Map

Natural system variables can also be specified in a map definition.

A Natural system variable can be selected with Create user-defined field.

Select "\*" from the field list and select the system variable from the list provided.

The format of the specified system variable is inserted in the field definition form.

## Modifying a User-Defined Variable - Field Editing

The map editor is used to define a field with all its attributes.

Select the item Create or Modify on the field editing selection menu and press ENTER. The following window displaying field attributes for the current field appears:

```

Extended Field Editing
Field :
Format: A Len:          AL:          PM:          ZP: N  SG: N
Rules : 0 Rule Editing: N Array:      Array Editing: N Mode:
AD:          CD:          CV:          DY: N  HE: N
EM:
    
```

With the field attributes window, any selected field can be modified. A selected field is the current field, which is highlighted.

Entry	Explanation
<b>Field</b>	<p>The field name. 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> <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.</p> <p>The name of a field can be changed. However, a prefix must not 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. You are prompted to enter a name. If modifications have been made you must press ENTER to continue. Otherwise, you can move through the field attributes using the TAB key.</p> <p><b>Note:</b> Duplicate field names are only allowed for fields defined as "output only fields".</p>
<b>Format</b>	The format and length of the field. These can be changed by overwriting the current entry.
<b>Len</b>	The internal program length of the variable.
<b>AL or NL or FL or DF</b>	<p>The length to be used when displaying the field. What is displayed in this field depends on the entered format.</p> <p>AL - (alphanumeric length) for formats alphanumeric, logical, time and system;                      NL - (numeric length) for formats binary, integer, numeric, packed numeric;                      FL - for format floating point;                      DF - for format date.</p>
<b>PM</b>	<p>Print Mode:</p> <p><b>C</b> - indicates that an alternative character set is to be used (special character table as defined by you or the designated Natural administrator).</p> <p><b>I</b> - indicates inverse print direction.</p> <p><b>CI</b> - indicates normal print direction</p> <p>For Numeric fields only:</p> <p><b>ZP</b> - N indicates that zero values for the field are not to be printed. Y - indicates that zero values are to be printed.</p> <p><b>SG</b> - N indicates that no sign position is to be allocated. Y - indicates that a sign position is to be allocated.</p>
<b>Rules</b>	The number of processing rules currently defined for the field.

<b>Entry</b>	<b>Explanation</b>
<b>Rule Editing</b>	Editing of processing rules; you are prompted: "Inline Processing Rule Editing (Y, <PF2>=EDIT)?". <b>Note:</b> The source code used to define a processing rule is entered/edited in the same way as with the Natural program editor.
<b>Array</b>	Indicates whether the field is an array or not ("blank").
<b>Array Editing</b>	Editing of arrays; you are prompted: "Array Editing (Y, <PF2>=EDIT)?".
<b>Mode</b>	Indicates how the field was created: <b>DATA</b> - Field was created by selecting a field from a DEFINE DATA definition. <b>SYS</b> - Field is a system variable. <b>UNDEF</b> - Field was created directly on the screen and has a dummy name. <b>USER</b> - The name of the field was created by extended field editing. <b>VIEW</b> - Field was selected from a view.
<b>AD</b>	Field attributes; you are prompted: "Attribute definitions (<PF2>=EDIT)". An Attribute definition window, containing the following items is displayed: - Representation - Alignment - I/O Characteristics - Mandatory Characters - Length Characteristics - Upper/Lower Case - Filler Character This function incorporates a toggle feature. If the attribute definition character displayed is correct, use the ESC key and the character is not changed. To change the attribute characteristics of a field, select the desired attribute and press ENTER; the modification is inserted in the AD definition. Press ESC to exit this function. Each item selected invokes an attribute window. These windows are self-explanatory.
<b>CD</b>	Colour attributes; you are prompted: "Color definitions (<PF2>=EDIT <CSR-UP/DN>=Select (Esc=Cancel Enter=OK)); i.e. press PF2 to edit the color definition, use the cursor to select a color and confirm with ENTER.
<b>CV</b>	Control variable for dynamic field attributes; you are prompted: "(<PF2>=Edit Rank if Array) ". 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. The control variable also contains a MODIFIED data tag, which indicates whether the field has been modified following map execution. 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.
<b>DY</b>	Dynamic string attributes; you are prompted: "(Y,<PF2>=Edit)". This parameter is used to define certain characters contained in the text string of an alphanumeric variable to control the attribute setting. See the session parameter DY.

Entry	Explanation
<b>HE</b>	<p>The name of a helproutine to be assigned to the field; you are prompted: "(Y, &lt;PF2&gt;=Edit)". A window is displayed which provides sufficient space to specify multiple parameters.</p> <p>This option consists of two operands:</p> <p>One which specifies the name of the helproutine to be invoked. It can be specified as a text constant or as a user-defined variable which contains the name of the helproutine;</p> <p>One which consists of up to 20 parameters which can be passed to the help routine. The parameters can be specified as constants or as user-defined variables which contain the values of the parameters. If an "=" is specified as a parameter, the name of the field as defined in the map definition is passed to the helproutine. In the case of a help routine which is assigned to a map, "=" denotes the name of the map. As no explicit DEFINE DATA PARAMETER statement can be specified in the map editor, the format/length of the second operand is defined in the following way:</p> <ul style="list-style-type: none"> <li>- If a parameter specified as second operand in the HE option is defined as a field of a map, the parameter will reference this field.</li> <li>- If no field with that name exists, the parameter field must be defined as N7 (default format assumed) in the program that uses the map.</li> </ul> <p><b>Note:</b> For a detailed explanation of the operands used in the HE option, see the session parameter HE.</p>
<b>EM</b>	Edit mask to be used for the field.

Three of the above items are of special interest:

- Rule Editing
- Array Editing
- AD (Attribute Definition)

## Rule Editing - Processing Rules

### 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.

**Note:**

Field-related inline processing rules can also be executed on a field-by-field basis; for further information, see the section Field-Sensitive Processing.

Free Predict rules have a name assigned and are stored in the Predict Dictionary. You cannot modify an existing free Predict rule (this can only be done in Predict); however, you can read a free rule into the editor, modify it, and store it under a different name to create a new free rule.

Free Predict rules can also be defined by accessing Predict on a remote OpenVMS, UNIX or mainframe server. To be able to do so, you must have set up your NATPARM parameter file accordingly and established a corresponding link by using Natural RPC. For information on how to assign dictionary servers and how to use Natural RPC, refer to your Installing and Setting Up Natural on UNIX or Installing and Setting Up Natural on OpenVMS.

Inline rules can become free Predict rules (and vice versa) if you assign/remove the rule name.

Predict automatic rules apply to database fields and are defined by the Predict administrator. If 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 profile settings (default 1).

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

An ampersand "&" within the source code of a processing rule is dynamically substituted with the fully qualified name of the field using the rule.

#### Example:

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

### 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 Ranks

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

### Processing-Rule Editing

To edit field-related processing rules, select "Rule Editing" on the Extended Field Editing window. To edit function-key-related processing rules, select "Key Rules" in the "Ops. Map" window.

The following window containing the options Rules and Fields appears:

```

..Current Field: PERSONNEL.STREET.....
.
.                R U L E   E D I T I N G (Esc = Quit)                .
. Rules                                Fields                                .
.....

                                Hobby:  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Address
-
State      :  XX
Zip        :  99999
City       :  XXXXXXXXXXXXXXXXXXXX
Street/Number: XXXXXXXXXXXXXXXXXXXX  99999

Create or modify a rule for this field
    
```

Select one of these options with the cursor and press ENTER.

If you select Fields, a list of variables used in the current map appears (for information purposes only).

If you select Rules, a selection list of existing rules for the current field appears.

On each list, the Predict rules are identified by their names, the inline rules by their first three source code lines.

There are the following ways to define processing rules for a field or key:

- Create a new processing rule
  - define a new rule
  - modify an existing rule and save it under a new name
- Assign an existing rule and

- edit or
- move or
- copy or
- unlink the rule.

### **To define a new processing rule**

1. Select Create.

An empty rule editor is displayed.

2. Enter the rule. (Use source code in the same way as in the Natural program editor.)

3. If you want the rule to be a free Predict rule, name it. If you want the rule to be an inline rule, do not name it before saving it.

4. Save the rule, see the section Commands for Processing Rule Editing.

Important: Once you have saved the rule, you can only modify it in Predict.

### **To modify an existing processing rule**

1. In the Rule Editor header enter the name of a Predict rule in the field Rule and press ENTER.

The rule is displayed in the rule editor.

2. Modify the rule. (Use source code in the same way as in the Natural program editor.)

3. Rename the rule and save it, see section Commands for Processing Rule Editing.

Important: Once you have saved the rule, you can only modify it in Predict.

### **To assign an existing rule**

1. Select an existing rule from the list.

2. Press ENTER.

A window with the following options is displayed: edit, move, copy, unlink.

- Edit  
Select edit to modify the rule. The name and contents of the rule (if it is a free rule) are displayed in the editor. See Step 2 of the section to modify an existing processing rule.
- Move  
Select move to modify the rule's rank. When you press ENTER a list is displayed from which you can select the new rank.
- Copy  
Select copy to copy the rule but assign it a new rank. When you press ENTER a list is displayed from which you can select the new rank.
- Unlink  
Select unlink to remove the rule from the field.
- Quit with ESC.

### **Note:**

If rules are written referencing a database statement, a label should be used, not a line reference number.

After the desired field processing rule has been entered, issue the command "P=*nn*" (where *nn* is the processing rule rank). This command saves the rule automatically.

### Commands for Processing Rule Editing

In the processing rule editor, processing rules can be selected for editing by using the following commands in the editor command line:

Cmd.	Function
<b>P<math>nn</math></b>	Select rule with rank $nn$ .
<b>P*</b>	Select rule from selection list.
<b>P</b>	Advance to next rule defined for the field.
<b>P=<math>nn</math></b>	Assign rule on current rank to rank $nn$ and save automatically.
<b>U</b>	Unlink (Delete)
<b>.</b>	End processing rule editing and save the rule.

### Array Editing

To invoke array editing, use the TAB key to select the item "Array Editing?" on the field attributes window. Replace the character "N" with "Y" (key sensitive) or press PF2 and the Array Definition window appears:

•Array Definition•••••			
Name #1		Upper Bnds	1____ 1____ 1____
-----			
Dimensions	Occurrences	Starting from	Spacing
1 . Index vertical	1__	_____	0 Lines
0 . Index horizontal	1__	_____	1 Columns
0 . Index (H/V) V	1__	_____	0 Cls/Ls
•••••			

This window displays the fields as defined on the map. If changes are not required, press TAB to proceed to the next field; if you want to return to a previous field, press SHIFT+TAB.

The Array Definition window contains the following entries:

Entry	Explanation
<b>Upper Bnds</b>	Indicates the upper bounds of the array; that is, the highest occurrence in (from left to right) the first, second and third dimension. 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. 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. 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.
<b>Dimensions</b>	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; the abbreviations used are: H=Horizontal and V=Vertical.
<b>Occurrences</b>	The number of occurrences to be defined for a dimension.
<b>Starting From</b>	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.
<b>Spacing</b>	The number of blank lines (for vertical dimensions) or blank columns (for horizontal dimensions) to be inserted between each dimension occurrence.

Enter the desired information and press ENTER. You are returned to the next item ("AD") listed on the field attributes window.

### 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.

```

•Array Definition.....
Name #1                               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.

```

•Array Definition.....
Name #1                               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.

•Array Definition•			
Name #1		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.

•Array Definition•			
Name #1		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.

•Array Definition•			
Name #1		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.

•Array Definition•			
Name #1	Upper Bnds 10___ 5___ 1___		
-----			
Dimensions	Occurrences	Starting from	Spacing
1 . Index vertical	10_	1_____	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.

•Array Definition•			
Name #1	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
.....			

**AD - Attribute Definition**

Attribute definition editing is performed as follows. Use the TAB key to move to the AD item in the field attribute window.

Press PF2 to invoke the following Attribute Definition window:

Children	: 99	Years-Educ:	99
Family Status:	XXXXXXXXXX	Years-Comp:	99
Sex	: X	Vacation-D:	99
		Sick-Days :	99
		Hobby:	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Address			
-			
State	: XX	•Attribute Definition•	
Zip	: 99999	•Representation	.
City	: XXXXXXXXXXXX	•Alignment	.
Street/Number:	XXXXXXXXXXXX	•I/O Characteristics	.
		•Mandatory Characters	.
		•Length Characteristics	.
		•Upper/Lower Case	.
		•Filler Character	.
•Extended Field Editing•			
•Field=	PERSONNEL.STREET		.
•Format=	A Len= 20 Alphan. Len= 20 PM=		.
•Rules:	0 Rule Editing? N Array:	Array Editing? N	Mode= Data .
•AD=	FHWOil CD= CV=	DY= -> HE= ->	.
•EM=			.
.....			

Select the desired item with the cursor keys.

Press ENTER and an additional selection window for each item selected appears.

**Note:**

This function has a toggle feature. If the attribute definition character displayed is correct, use the ESC key and this character is not changed. The ESC key terminates AD editing.

## Selecting Fields from a DDM or User View

New fields can be created by selecting existing variable definitions from a DDM or a user view. A field can be selected from any available DDM.

To select a DDM/user view, first select the object class. Valid object classes are:

<b>A</b>	Parameter Data Area
<b>G</b>	Global Data Area
<b>H</b>	Helproutine
<b>L</b>	Local Data Area
<b>M</b>	Map
<b>N</b>	Subprogram
<b>P</b>	Program
<b>S</b>	Subroutine
<b>V</b>	View

Programs, subroutines, subprograms and helproutines can only be used if they contain a DEFINE DATA statement.

For demonstration purposes, the object "view" will be used. The leftmost item on the Field Editing selection menu displays a list. Either select the item "View Defined" from this list and press ENTER or enter "C" and then the abbreviation "V".

The following library list appears on the screen:

```

.....
.
      NATURAL MAP EDITOR (Esc to select field)
.
·Create  Modify  Erase   Drag    Info OFF  Lines   Ops. Map  Quit  ·
.....
N· A Parameter Data Area ·XXXX·ACTIO      · : XXXXXXXXXXXXXXXXXXXX
I· G Global Data Area   ·      ·ACTION      · : 999999
F· H Help Routine       ·XXXX·AEH-BEDIENSTETER ·ion: 999999
C· L Local Data Area    ·      ·AEH-HDAT     ·duc: 99
F· M Map                 ·      ·BED          ·omp: 99
S· N Subprogram         ·      ·EMPLOYEES    ·n-D: 99
· P Program             ·      ·EMPLOYEES-FILE ·ys : 99
· S Subroutine          ·      ·FUNC         ·
· T Text Constant      ·      ·FUNCTION     ·XXXXXXXXXXXXXXXXXXXX
A· U User Defined       ·      ·GEN_CODE     ·
-· V View Defined       ·      ·HILFSDAT    ·
S· 1 Parm Defined      ·      ·MAP          ·
Z· 2 Local Defined     ·      ·OBJ          ·
C.....·OBJECTTYPE     ·
Street/Number: XXXXXXX·PERSONNEL     ·
                ·PERSONNEL-FILE ·
                ·.....LIB= SYSTEM·

Take variable definition from view
    
```

Select the desired view (for example, PERSONNEL) and press ENTER. The selected view is displayed in a window:

```

*** Personnel Data Detail Display Function ***

Person Data                                Employment Data
-
Name      : XXXXXXXXXXXXXXXXXXXXXXXX      Job       : XXXXXXXXXXXXXXXXXXXXXXXX
Initial   : X                               Salary    : 999999
First-Name : XXXXXXXXXXXXXXXXXXXXXXXX      Commission: 999999
Children  : 99                               Years-Educ: 99
Family Status: XXXXXXXXXXXXX              Years-Comp: 99
Sex       : X                               Vacation-D: 99
                                                Sick-Days : 99

                                                Hobby:   XXXXXXXXXXXXXXXXXXXXXXXX

Address
-
..PERSONEL.....
. 1 AA PERSONEL-NUMBER          N 8.0 D .
. 1 AA PERSONAL-NUMMER         N 8.0 D .
. 1 AA NUMERO-PERSONNEL        N 8.0 D .
.G 1 G1 PERSON                 .
. 2 BA NAME                     A 20 N D .
.....

HD=PERSONNEL/NUMBER
    
```

This window displays, for the highlighted field, three additional information lines at the bottom of the screen.

These lines display the following information for the current field:

- Edit mask
- Header
- Comments

The field name consists of the view name concatenated with the field name by a period, for example:

**personal.personnel-number**

Select the fields which are to be included in your map and press ENTER.

The library window, listing the selected field in the "Field= " line appears and you are prompted to: "Position cursor and press Enter".

Position the cursor to the start of a field position and press ENTER.

Continue modifying fields as described under Creating a User-Defined Variable.

Repeat this procedure until all views required for your map have been completed.

**Note:**

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

## Defining Fields for a Parameter or Local Data Definition

New parameters or local variables can be created or existing ones can be modified by selecting the item "Parm Defined" or "Local Defined" from the field editing list and pressing ENTER.

### Parameter Definitions

If you select the "Parm Defined" item, the following window appears:

```

..PARAMETER.....
.<CREATE>.....
. TEST1          L      001:003      .
. TEST2          L      001:003,001:004 .
. TEST3          L      001:003,001:004,001:005 .
.....
    
```

With this function, new parameters can be added and existing parameters can be modified.

If you want to modify an existing parameter (for example, TEST3), select it with your cursor and press ENTER. A window pops up prompting you to define the desired action: you can Edit the selected parameter, Delete it, or Cancel the function.

If you define the action Edit, the following window appears, in which you can edit the selected parameter:

```

..PARAMETER.....
.Name.....: TEST3
.Format...: L
.
.Dimension: 3
.
. Lower Bnds : Upper Bnds
.1. Index.: 1 3
.2. Index.: 1 4
.3. Index.: 1 5
.....
    
```

If you want to create a new parameter, select the <CREATE> option. The same window appears, but this time it is empty so that you can make the specifications for the parameter to be created.

When you leave the above window, a further window pops up prompting you whether you want to either save your modifications/specifications or cancel the function.

### Local Data Definitions

If you select the "Local Defined" item, the following window appears:

```

..LOCAL.....
.<CREATE>.....
. TESTA          L      001:003      .
. TESTB          L      001:003,001:004 .
. TESTC          L      001:003,001:004,001:005 .
.....
    
```

With this function, 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.

If you want to modify an existing local variable (for example, TESTC), select it with your cursor and press ENTER. A window pops up prompting you to define the desired action: you can Edit the selected variable, Delete it, or Cancel the function.

If you define the action Edit, the following window appears, in which you can edit the selected local variable:

```

..LOCAL.....
•Name.....: TESTC      .
•Format...: L          .
.                      .
•Dimension: 3          .
.      Lower Bnds  :  Upper Bnds  .
•1. Index.:          1           3  .
•2. Index.:          1           4  .
•3. Index.:          1           5  .
.....

```

If you want to create a new local variable, select the <CREATE> option. The same window appears, but this time it is empty so that you can make the specifications for the variable to be created.

When you leave the above window, a further window pops up prompting you whether you want to either save your modifications/specifications or cancel the function.

# Map Profile

This section describes the process of defining the map settings (profile) for a map.

The Map Profile is invoked by selecting "Prof. Map" from the "Ops. Map" selection list. The following screen is displayed:

```

.....
.
.          NATURAL MAP EDITOR (Esc to select field)          .
.....
·Cr·Map Settings.....
  ·Format          Context          .
  -----          -----          .
Add·Page Size .....: 24          .
- ·Line Size .....: 79          WRITE Statement ...: N          .
Sta·Layout .....:          .
Zip·dynamic .....: N          Help Routine .....:          .
Cit·Zero Print .....: N          Help Parameter .....:          .
Str·Upper Case .....: Y          as field default ..: N          .
  ·Documentation Skip ....: N          Help Text .....: N          .
  ·Decimal Char ...: .          Position Line ....: 0          Column : 0          .
  ·Standard Keys ..: N          AutoRuleRank .....: 1          .
  ·Right Justify ..: Y          .          .          .
  ·Print Mode .....:          Filler Characters          .
  .          -----          .
  ·Control Var ....:          Optional, Partial ..: _          .
  .          Required, Partial ..: _          .
  ·Field Sensitive: N          Optional, Complete ..: _          .
          Required, Complete ..: _          .
.....
Modify map profile
    
```

This map profile screen comprises Map Profile Settings and Filler Characters.

## Map Profile Settings

The following map profile settings can be used:

Entry	Explanation
<b>Page Size</b>	The number of map lines to be edited (1 - 250); if "Std Keys" (see below) is set to "Y", the number of lines is restricted to 3 - 250. For a map which is output with a WRITE statement (see the entry "WRITE Statement" in the Context column of the Map Settings), you specify the number of lines of the logical page output with the WRITE statement, not the map size. Thus, the map may be output several times on one page.
<b>Line Size</b>	The number of map columns to be edited (5 - 249).
<b>Layout dynamic</b>	The name of a map source definition which contains a predefined layout. <b>Y</b> - 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 F9 within the Field and Variable Definitions function. <b>N</b> - 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.

Entry	Explanation
<b>Zero Print</b>	<p><b>Y</b> - displays a field value of all zeros as one zero only.  <b>N</b> - displays a zero value as blanks.  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>
<b>Upper Case</b>	<p><b>Y</b> - indicates that all input entered for fields at map execution time is to be converted to upper case.  <b>N</b> - indicates that no lower to upper case conversion is to be performed.  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>
<b>Documentation Skip</b>	<p><b>Y</b> - Does <i>not</i> automatically move the cursor to the next field in the map at execution time even if the current field is completely filled.  <b>N</b> - 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>
<b>Decimal Char</b>	<p>The character to be used as the decimal notation character. This character can only be changed with the GLOBALS command.</p>
<b>Standard Keys</b>	<p><b>Y</b> - leaves the last two lines of the map empty so that function-key specifications can be entered at execution time.  <b>N</b> - causes all lines to be used for the map.</p>
<b>Right Justify</b>	<p>The type of field justification to be used for numeric and alphanumeric fields taken from a user view or the data definition:  <b>Y</b> - right justified  <b>N</b> - left justified</p>
<b>Print Mode</b>	<p>The default print mode for variables:  <b>C</b> - indicates that an alternative character set is to be used (special character table).  <b>I</b> - indicates inverse print direction.  <b>IC</b> - indicates standard print direction. This value is copied into the field definition when a new field is created.</p>
<b>Control Var</b>	<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 maximum length of a control variable is limited to 8 characters. The control variable referenced in the map must be defined in the program using that map.</p>
<b>Field Sensitive</b>	<p><b>Y</b> - specifies that processing rules attached to map fields are executed on a field-by-field basis; that is, immediately after you have left a given field.  <b>N</b> - specifies that no field sensitivity is to be defined for the map, which means that the map is not to be processed until you have entered all necessary values in the map fields and pressed ENTER (or any F key).</p>
<b>WRITE Statement</b>	<p><b>Y</b> - Marking this field with <b>Y</b> produces a WRITE statement at the end of the map definition process. The resulting map may then be invoked from a Natural program using a WRITE USING FORM statement. Empty lines at the end of the map are automatically deleted so that the map can be output several times on one page.  <b>N</b> - Marking this field with <b>N</b> will cause the result of the map definition process to be an INPUT statement. The resulting map may then be invoked from a Natural program using an INPUT statement.</p>
<b>Help Routine</b>	<p>The name of a helproutine which is invoked at runtime when the help function is invoked for this map (global help for map). For detailed explanation of the syntax, see the parameter "Help" in the section Modifying a User-Defined Variable - Field Editing.</p>

Entry	Explanation
<b>Help Parameter</b>	The help parameter which is invoked at execution time when the help function is invoked. <b>Note:</b> A maximum of 20 help parameters are possible. If you enter more, they are simply ignored.
<b>as field default</b>	<b>Y</b> - specifies that the help routine 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 help routine. <b>N</b> - specifies that the name of the map is passed to the help routine.
<b>Help Text</b>	<b>Y</b> - specifies that this map is actually help text; default = "N".
<b>Position Line Column</b>	The position where the help map is to appear on the screen at execution time.
<b>AutoRuleRank</b>	The rank (priority) assigned to automatic Predict rules when they are linked to the map during field definition. Default is 1.

## Filler Characters

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

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

Filler characters may also be defined for individual fields using the extended field editing function. For definition of field types, see also the session parameter AD.

## Post Assignment

A field which has been previously defined (in layout) directly on the screen may 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 may be redefined using the field definition from a view defined in a data area. This, however, is only possible if it is the same database field.

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

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

## Field-Sensitive Processing

In the map profile, you can specify whether a map is to be processed after you have entered all necessary values in the map fields and pressed ENTER, or whether the processing rules attached to the map fields are to be handled on a field-by-field basis.

When handled on a field-by-field basis, the processing rules attached to a field are executed immediately as soon as you have filled this field entirely or you move the cursor to another position; they are executed in the same order as without field sensitivity.

### Advantages of Field-Sensitive Processing

The advantages of field-sensitive processing are:

- dynamic filling of fields based on user input,
- improved user guidance,
- security based on user input,
- rapid data entry.

**Note:**

To exploit the advantages of field-sensitive processing, you need to adapt your existing processing rules accordingly. For more information on processing rules, see Rule Editing - Processing Rules.

### Dynamic Fillings of Fields

Without field-sensitive processing, the application would wait until all the fields have been filled in and you have pressed ENTER, before checking if the data entered are valid. In the worst case, you would have wasted time and effort filling in all the fields, before being informed of the error.

However, with field-sensitive processing, as soon as you have entirely filled a field or leave the field by either pressing END or SHIFT+END or moving the cursor with one of the cursor movement keys, the data input is checked immediately, and other fields can be pre-filled automatically (via REINPUT FULL) with data based on the value entered in the previous field. Depending on this value, these data may also be the result of a database query.

The information under which condition a field was left can be retrieved from the \*PF-KEY system variable, which contains either "FULL" (if the field was entirely filled) or the name of the key most recently pressed as shown in the following table.

<b>Pressed Key</b>	<b>*PF-KEY</b>
LEFT ARROW	LEFT
RIGHT ARROW	RIGT
UP ARROW	UP
DOWN ARROW	DOWN
PAGE UP	PGUP
PAGE DOWN	PGDN
TAB	TAB
HOME	HOME
END	END
BACKTAB	BTAB
ENTER	ENTR
Field is full	FULL

### **Improved User Guidance**

With field-sensitive processing, you can be guided from field to field, depending on the values entered. The cursor can skip fields that are pre-filled and take you to the next input field.

As messages are displayed immediately after a field has been checked, you are informed more promptly and more precisely as to what to do.

### **Security Based on User Input**

Depending on the value entered in a particular field, you can be prompted, for example, for a special password to access the requested information.

### **Rapid Data Entry**

All these mechanisms described above make interaction with the application much faster and more efficient.

## **Defining a Map as Field-Sensitive**

To be able to use field-sensitive processing, you first need to define a map as being field-sensitive by setting Field Sensitive to "Y" in the Map Profile. A window appears asking you whether field sensitivity should be with or without automatic field recognition:

```

.....
.                NATURAL MAP EDITOR (Esc to select field)                .
·Cr·Map Settings·.....
  ·Format                Context                .
  -----                -----                .
Add·Page Size .....: 24                .
- ·Line Size .....: 79                WRITE.....                .
Sta·Layout .....:                . Automatic Field Recognition ? .                .
Zip·dynamic .....: N                Help ·                YES    NO                .
Cit·Zero Print .....: N                Help .....                .
Str·Upper Case .....: Y                as field default .: N                .
·Documentation Skip ...: N                Help Text .....: N                .
·Decimal Char ..: .                Position Line .....: 0    Column: 0                .
·Standard Keys .: N                AutoRuleRank .....: 1                .
·Right Justify .: Y                .                .                .
·Print Mode .....:                Filler Characters                .
.                -----                .
·Control Var ...:                Optional, Partial .: _                .
.                Required, Partial .: _                .
·Field Sensitive: Y                Optional, Complete .: _                .
.                Required, Complete .: _                .
.....
Modify map profile

```

Field sensitivity within a Natural map can be achieved in two different ways:

1. With automatic field recognition, which means that you need not code any conditions for activating the right processing rule.

**Note:**

If you upload such a map to the mainframe, the Natural mainframe map editor does not understand the syntax of the uploaded map, because of new syntax extensions.

2. Without automatic field recognition, which means that you are responsible for being informed about which field was left with the last keystroke. You can achieve this by using the system variables \*CURS-COL and \*CURS-LINE in your Natural program.

**Note:**

If you upload such a map to the mainframe, STOW the map on the mainframe, and the same source code will be generated.