

Map Editor

The Natural map editor is used to create a Natural object of type map. A map is a screen layout that can be referenced in a Natural object such as a program by using either an `INPUT USING MAP` statement (for input maps) or a `WRITE USING MAP` statement (for output maps).

A map contains text fields and data fields. Text fields are literal strings and data fields are variables. Data fields can be either user-defined variables or Natural system variables.

Once a map has been created, it can be stored as a source object and a cataloged object in a library in a Natural system file.

Note:

Using Natural Studio in a Windows environment, the map editor supports fields with Unicode format and Unicode strings. However, when reading the source of a Unicode map into the editing area of a map editor in a local UNIX, OpenVMS or mainframe environment, all Unicode strings will be removed from the source.

The *Map Editor* documentation covers the following topics:

- Creating a Map
 - Invoking the Map Editor
 - Map Editor Menu
 - Creating a Text Constant
 - Creating a User-Defined Variable
 - Modifying a User-Defined Variable - Field Editing
 - Selecting Data Definitions
 - Defining Fields for a Parameter or Local Data Definition
 - Map Profile
 - Post Assignment
 - Field-Sensitive Processing
-

Creating a Map

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

1. Definition of the map profile (that is, the field delimiters, format settings, context 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 defined in two different ways:

- First define a prototype map, next make the corresponding data definitions in the object that references the map, then integrate the map into the application.

Fields can be defined directly on the map editing screen. Each field is assigned a default name. Subsequently, when the corresponding data definitions have been made in the respective object, these data definitions can be assigned to the map fields (post assignment).

- Define a map using existing data definitions.

If data definitions already exist in an object that references the map, the map fields can be created by using the data definitions contained in this object. In this case, all characteristics of the data definitions are copied into the map.

3. Definition of the fields to be used in the map. Map fields can be created by either typing the field definitions directly in the map editing screen or by selecting data definitions from another Natural object (see also *Selecting Data Definitions*).

4. The saving and/or cataloging of the map definition. Once a map has been defined as described in the previous steps, it can be saved as a source object and/or cataloged object in the current library and Natural system file. Once saved as a source object, the map can be read and modified during a subsequent map editor session. Once saved as a cataloged object, a map can be invoked from a Natural program.

Invoking the Map Editor

You invoke the Natural map editor to create a new map or edit an existing one as described in the following section.

- Creating a New Map
- Editing an Existing Map

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 menu items are described in the following section.

- 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 (DDM) 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 Data Definitions*.

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
V Reverse 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:

Item	Description
Check Map	Causes syntax checking and generation of source code.
Edit Map	Invokes the map editing screen for modification of an existing map definition. The map editor will start a new edit session.
Reverse Map	This feature only applies if the default code page supports languages that are written from right-to-left (RTL) such as Hebrew. Reverses the screen direction of a map from left-to-right (this is the default) to right-to-left, or vice versa. This is only a visual change; the physical field position defined in the map source is retained. The screen direction remains reversed until you select "Reverse Map" again.
Key Rules	Invokes editing of function-key-related processing rules.
List Map	Generates source code and lists it.
Prof. Map	Invokes a map profile window, which is described under <i>Map Profile</i> .
Read Map	Invokes the map editing screen to read an existing map definition.
Save Map	Performs a source code generation check and then saves the map. The map definition is saved in source form in the Natural library.
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.
Stow Map	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-ARROW and DOWN-ARROW keys to scroll through and select one of the available attributes/colors or use the corresponding abbreviation (for example, B=Blinking or RE=Red) and press ENTER. Using the LEFT-ARROW and 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.

An Extended Field Editing window similar to the example below 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 selection list appears, with all valid Natural data formats such as "A" for data type alphanumeric.

Note:

If you know the Natural data format, you can enter it directly in the field position where you placed the cursor. The display of available data formats is thus avoided.

Select the Natural data format required and press ENTER.

The Natural data format is now entered in the Extended Field Editing window 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 field.

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 Natural data 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. An Extended Field Editing window similar to the example below appears, which displays all attributes for the current field:

```

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 Extended Field Editing window, any selected field can be modified. A selected field is the current field, which is highlighted.

The field attributes contained in the Extended Field Editing window are explained in the following table:

Field Attribute	Explanation
Field	<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 data definition in another Natural object, it is assigned the same name as the field definition in this object.</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 the data definition in another Natural object. 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>Note: Duplicate field names are only allowed for fields defined as "output only fields".</p>
Format	The Natural data format of the field. These can be changed by overwriting the current entry.
Len	The internal program length of the variable.
AL or NL or FL or DF	<p>The length to be used when displaying the field. The field label depends on the Natural data format entered:</p> <p>AL for formats A (alphanumeric), L (logical) and T (time).</p> <p>NL for formats B (binary), I (integer), N (numeric) and P (packed numeric).</p> <p>FL for format F (floating point).</p> <p>DF for format D (date).</p>

Field Attribute	Explanation
PM	<p>Print Mode.</p> <p>This input field corresponds to the session parameter PM. For detailed information on using this field and valid input values, see <i>PM - Print Mode</i> in the <i>Parameter Reference</i>.</p>
ZP	<p>Zero printing. You can only enter a value in ZP if the field is numeric or a time system variable.</p> <p>This input field corresponds to the session parameter ZP. For detailed information on using this field and valid input values, see <i>ZP - Zero Printing</i> in the <i>Parameter Reference</i>.</p>
SG	<p>Sign position. You can only enter a value in SG if the field is numeric or a time system variable.</p> <p>This input field corresponds to the session parameter SG. For detailed information on using this field and valid input values, see <i>SG - Sign Position</i> in the <i>Parameter Reference</i>.</p>
Rules	<p>The number of processing rules currently defined for the field.</p>
Rule Editing	<p>Editing of processing rules; you are prompted:</p> <p>"Inline Processing Rule Editing (Y, <PF2>=EDIT)?".</p> <p>Note: The source code used to define a processing rule is entered/edited in the same way as with the Natural program editor.</p>
Array	<p>Indicates whether the field is an array or not ("blank").</p>
Array Editing	<p>Editing of arrays; you are prompted:</p> <p>"Array Editing (Y, <PF2>=EDIT)?".</p>
Mode	<p>Indicates how the field was created:</p> <p>DATA - Field was copied from the data definition in another Natural object (except DDM).</p> <p>SYS - Field is a system variable.</p> <p>UNDEF - 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 - Field was copied from the field definition in a DDM.</p>

Field Attribute	Explanation
AD	<p>Field attributes.</p> <p>This field corresponds to the session parameter AD. For detailed information on using this field and valid input values, see the relevant sections (referenced below) in <i>AD - Attribute Definition</i> in the <i>Parameter Reference</i> documentation.</p> <p>You are prompted:</p> <p>"Attribute definitions (<PF2>=EDIT)".</p> <p>An Attribute definition window, containing the following items is displayed:</p> <ul style="list-style-type: none"> ● Representation <i>See Field Representation.</i> ● Alignment <i>See Field Alignment.</i> ● I/O Characteristics <i>See Field Input/Output Characteristics.</i> ● Mandatory Characters <i>See Mandatory Input.</i> ● Length Characteristics <i>See Length of Input Value.</i> ● Upper/Lower Case <i>See Field Upper/Lower Case Characteristics.</i> ● Filler Character <i>See Filler Character.</i> <p>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.</p>
CD	<p>Colour attributes; you are prompted:</p> <p>"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.</p>

Field Attribute	Explanation
CV	<p>Control variable for dynamic field attributes; you are prompted:</p> <p>"(<PF2>=Edit Rank if Array) ".</p> <p>The name of a variable which contains the attributes to be used for this field. This variable must be defined with Natural data 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>
DY	<p>Dynamic string attributes; you are prompted:</p> <p>"(Y,<PF2>=Edit)".</p> <p>This parameter is used to define certain characters contained in the text string of an alphanumeric variable to control the attribute setting. For detailed information on valid input values, see <i>DY - Dynamic Attributes</i> in the <i>Parameter Reference</i>.</p>

Field Attribute	Explanation
HE	<p>The HE option is used to assign a helproutine or a help map to the map field. A helproutine or help map is then invoked at execution time when a help request is made for the map field. For detailed information, see the description of the HE session parameter in <i>HE Helproutine</i> in the <i>Parameter Reference</i>.</p> <p>If you enter a "Y" in the HE field or place the cursor in the field and press PF2, a window prompts you to enter the name of a helproutine or help map and parameters to be passed to this helproutine or help map.</p> <p>The syntax that applies to specifying names and parameters in the HE field corresponds to the syntax of the HE session parameter described in <i>HE Parameter Syntax</i> (<i>Parameter Reference</i>). In addition to the syntax explanations provided there, the following applies when using the map editor:</p> <p><i>operand1:</i></p> <ul style="list-style-type: none"> ● If a variable name is specified which corresponds to the name of a map field, this field must be in the Natural data format/length A8. ● If a variable name is specified for which no map field yet exists, a map parameter with that name is automatically defined in the Natural data format/length field A8. <p><i>operand2:</i></p> <ul style="list-style-type: none"> ● If a variable name is specified for which no map field yet exists, a map parameter with that name is automatically defined in the Natural data format/length N7. <p>Removing a parameter from the HE field implies that the parameter is also removed from the map, unless the parameter is a map field or is associated with any other map field as a help parameter or "Starting from" value.</p>
EM	Edit mask to be used for the field.

Three of the above items are of special interest:

- Rule Editing - Processing Rules
- Array Editing
- AD - Attribute Definition

Rule Editing - Processing Rules

Field-Related Processing Rules

Three types of processing rules can be defined:

- Inline processing rules

- Predict free rules
- Predict automatic rules

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

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*.

Predict free rules have a name assigned and are stored in the Predict Dictionary. You cannot modify an existing Predict free 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.

You can access Predict free rules that are stored in either a local or a remote Predict environment.

Predict free rules that are stored in a remote environment on a UNIX, an OpenVMS or a mainframe host can be accessed by using a Natural RPC server.

For information on how to connect to a Predict server, see the profile parameter USEDIC (*Parameter Reference*) and *Dictionary Server Assignments* in the section *Overview of Configuration File Parameters* (*Configuration Utility* documentation).

Inline rules can become Predict free 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 from the data definition in another Natural object, 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.

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

Examples:

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

Function-Key-Related Processing Rules

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

- Inline processing rules

- Predict free rules

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

Note:

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

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:  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Address
-
State      :  XX
Zip        :  99999
City       :  XXXXXXXXXXXXXXXXXXXX
Street/Number: XXXXXXXXXXXXXXXXXXXXXXXX  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 Predict free 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 Editing Processing Rules*.

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 Editing Processing Rules*.

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 Editing Processing Rules

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

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

Note:

The map editor does not support X-arrays (eXtensible arrays).

The Array Definition window contains the following entries:

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 the data definition in another Natural object, 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 the data definition in another Natural object, the dimensions of the map array must not exceed the dimensions as defined in the Natural program.</p>
Dimensions	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.
Occurrences	The number of occurrences to be defined for a dimension.
Starting From	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.
Spacing	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 Natural data 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_	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.

•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: XXXXXXXXXXXX           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:

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 Data Definitions

A map field can be defined by selecting a data definition from another Natural object. Data definitions are either field definitions in a Natural DDM (data definition module) or variable definitions.

To select a data definition, first select the object class. Valid object classes are:

Object Class	Description
A	Parameter Data Area
G	Global Data Area
H	Helproutine
L	Local Data Area
M	Map
N	Subprogram
P	Program
S	Subroutine
V	View (DDM)

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

For demonstration purposes, a Natural object of type DDM 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          · : XXXXXXXXXXXXXXXXXXXXXXXX
I· G Global Data Area   ·      ·ACTION          · : 999999
F· H Helproutine       ·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          ·XXXXXXXXXXXXXXXXXXXXXXXXXXXX
A· U User Defined      ·      ·GEN_CODE          ·
-· V View Defined      ·      ·HILFSDAT         ·
S· 1 Parm Defined     ·      ·MAP              ·
Z· 2 Local Defined     ·      ·OBJ              ·
C.....·XXXX·OBJECTTYPE          ·
Street/Number: XXXXXXXXXXXXXXX·PERSONNEL      ·
                ·PERSONNEL-FILE   ·
                ·.....LIB= SYSTEM·
.....

Take variable definition from view
    
```

Select the desired DDM (for example, PERSONNEL) and press ENTER. The selected DDM 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 PERSONNEL-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 DDM 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 fields required for your map have been defined.

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.

The following topics are covered below:

- Parameter Definitions
- Local Data Definitions

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                Helproutine .....:                .
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.

The following topics are covered below:

- Map Profile Settings
- Filler Characters

Map Profile Settings

The following map profile settings can be used:

Entry	Explanation
Page Size	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.
Line Size	The number of map columns to be edited (5 - 249).

Entry	Explanation
Layout dynamic	<p>The name of a map source definition which contains a predefined layout.</p> <p>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.</p> <p>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.</p> <p>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.</p>
Zero Print	<p>Y - displays a field value of all zeros as one zero only.</p> <p>N - displays a zero value as blanks.</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>
Upper Case	<p>Y - indicates that all input entered for fields at map execution time is to be converted to upper case.</p> <p>N - indicates that no lower to upper case conversion is to be performed.</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 <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.</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	<p>The character to be used as the decimal notation character. This character can only be changed with the GLOBALS command.</p>
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.</p>
Right Justify	<p>The type of field justification to be used for numeric and alphanumeric fields taken from data definitions in other Natural objects:</p> <p>Y - right justified N - left justified</p>

Entry	Explanation
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).</p> <p>I - indicates inverse print direction.</p> <p>I C - indicates standard print direction. 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 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>
Field Sensitive	<p>Y - 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.</p> <p>N - 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>
WRITE Statement	<p>Y - Marking this field with Y 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.</p> <p>N - Marking this field with N 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>
Helproutine	<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 <i>Modifying a User-Defined Variable - Field Editing</i>.</p>
Help Parameter	<p>The help parameter which is invoked at execution time when the help function is invoked.</p> <p>Note: A maximum of 20 help parameters are possible. If you enter more, they are ignored.</p>
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.</p>
Help Text	<p>Y - specifies that this map is actually help text; default = "N".</p>
Position Line Column	<p>The position where the help map is to appear on the screen at execution time.</p>
AutoRuleRank	<p>The rank (priority) assigned to Predict automatic rules when they are linked to the map during field definition. Default is 1.</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:

Field Type	Explanation
Optional Partial	Is not mandatory, need not be completely filled.
Required Partial	Mandatory, need not be completely filled (AD=E).
Optional Complete	Is not mandatory, must be completely filled (AD=G).
Required Complete	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 the screen layout of a map may be assigned the field name and field attributes of a DDM field definition 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 from a DDM must be redefined by using the appropriate DDM field definition from this DDM or a DEFINE DATA definition. However, this is only possible if it is the same database field.

Post assignment of a DDM field definition can only be done by deleting the respective field definition from the map and selecting a new DDM field definition from a DDM or a DEFINE DATA definition (see also *Selecting Data Definitions*).

Post assignment cannot be used for view arrays (in a DDM field definition or a DEFINE DATA definition) 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.

The following topics are covered below:

- Advantages of Field-Sensitive Processing
- Defining a Map as Field-Sensitive

Advantages of Field-Sensitive Processing

The advantages of field-sensitive processing are described in the following section.

- Dynamic Fillings 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 (Based on User Input)

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.

Pressed Key	*PF-KEY
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.