

# Defining Map Fields

The fields which are to comprise a map definition can be specified in any of the ways described in this section.

- Defining Fields Directly on the Screen
- Selecting Definitions from Other Objects
- Using System Variables in a Map Definition

## Defining Fields Directly on the Screen

Map fields are defined directly in the map editing area by entering a delimiter character followed by the number of positions to be allocated for the field. The following characters can be used:

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

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

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

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

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

## Selecting Definitions from Other Objects

A map field can be defined by selecting a data definition from another Natural object. Data definitions are either field definitions in a DDM or variable definitions. For a list of objects that can be used to select data definitions, see *Object Types and Codes*.

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

### Restrictions of Selection:

You cannot select an item preceded by a period (.), such as a group.

Input and modifiable (input/output) fields can only be specified once in the map editing area.

This section covers the following topics:

- Listing and Copying Data Definitions
- Checking and Correcting References to DDMs
- Object Types and Codes

### Listing and Copying Data Definitions

#### To list and copy definitions from other Natural objects

1. Switch on split-screen mode, if required.
2. In the **Ob** (Object) input fields in the upper left screen section, enter a valid object code (see also *Object Types and Codes*) followed by an object name, for example, P TEST as shown in the example in Step 4. If required, delete any remaining characters in this line.

Or:

Leave the first input field (denoted by an underscore) of the upper left **Ob** input fields blank and enter a question mark (?) in the second input field or press PF1 (Help). If required, delete any remaining characters in this line.

When you press ENTER, the **Select an Object** screen appears with a list of all objects contained in the current library. You can press ENTER to scroll down the list. In the **Select** field, enter the number that corresponds to the object you want to select.

3. Press ENTER.

The data definitions in the `DEFINE DATA` section of the Natural program (in the following example, program TEST) are displayed in the upper left section of the map editing screen.

To scroll in the list of data definitions, in the **Ob** input fields, overwrite the existing entries (in the following example, P TEST) with any of the following positioning commands and leave the remaining characters:

Command	Explanation
+	Scrolls down one page in the list.
++	Scrolls to the end of the list.
-	Scrolls up one page in the list.
--	Scrolls to the beginning of the list.
+n	Scrolls down n lines.
-n	Scrolls up n lines.

4. To display the data definitions of another object, in the **Ob** input fields in the upper right screen section, enter a valid object code (see *Object Types and Codes*) followed by an object name and delete any remaining characters in this line.

For example:

Ob <b>P TEST</b>		Ob <b>L LDA01</b>	
1 #NAME-START	A20	A #NAME-START	A20
2 #NAME-END	A20	B #NAME-END	A20
3 #MARK	A1	C #PERS-ID	A8
. EMPLOYEES-VIEW	*V1	D #MAKE	A20
4 PERSONNEL-ID	A8	E #MODEL	A20
5 NAME	A20	.	
001 --010---+-----+-----030---+-----+-----050---+-----+-----070---+-----			

Or:

Select an object from a list by proceeding as described earlier in Step 2.

5. Copy data definitions into the map editing area:
- In the map editing area, in any position of a blank line, enter a delimiter character followed by the sequential number or letter displayed next to the data definition required. For example:

(3

Selects the data definition assigned to the number 3 (in the example above, #MARK).

:C

Selects the data definition assigned to the letter C (in the example above, #PERS-ID).
  - Press ENTER.

The characteristics of the data definitions (name, format, length or array definitions) are copied into the map.

You can display or modify the copied values by using the map editor's field editing functions described in the sections *Extended Field Editing* and *Array and Table Definitions*.

## Checking and Correcting References to DDMs

For each map field copied from a DDM, an INCDIR statement is automatically generated into the map source. The INCDIR statement contains the name of the DDM file and field referenced by the map field. You can check whether an INCDIR statement contains an invalid DDM reference by switching on the CHKRULE option of the COMPOPT system command. Natural then issues an appropriate error message when you catalog the map definition. For detailed information, see *CHKRULE - Validate INCDIR Statements in Maps* in the *System Commands* documentation.

From the map editing area, you can list all INCDIR statements and change the name of a DDM if required.

### ▶ To list and update single or multiple INCDIR statements

1. In any line of the editing area, enter the following line command:

```
..L
```

A **Modify INCDIR Statements of Map** screen similar to the example below appears with a list of all INCDIR statements generated into the source of the current map definition:

```

16:24:23          - Modify INCDIR Statements of Map  MAPTEST  -          2008-05-14
                                                    Top of List

      DDM Name                Field Name
AUTOMOBILES                MAKE
AUTOMOBILES                OWNER-PERSONNEL-NUMBER
FINANCE                    BANK
AUTOMOBILES                MODEL
EMPLOYEES                  PERSONNEL-ID
AUTOMOBILES                BODY-TYPE
EMPLOYEES                  FIRST-NAME
AUTOMOBILES                NUMBER-OF-CYLINDERS
EMPLOYEES                  MIDDLE-I
AUTOMOBILES                HORSEPOWER
EMPLOYEES                  NAME
AUTOMOBILES                PISTON-DISPLACEMENT
EMPLOYEES                  MIDDLE-NAME
AUTOMOBILES                WEIGHT
EMPLOYEES                  MAR-STAT
AUTOMOBILES                COLOR

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

```

The list contains the names of all DDM files and DDM fields from which you copied a map field.

2. If required, change the DDM name for a INCDIR statement by replacing the name(s) in the **DDM Name** column.

Or:

Press PF5 to open the **Generic Change** window. In the **New DDM Name** field, enter the name of the new DDM to be used by all INCDIR statements that reference this DDM.

The DDM name(s) in the corresponding INCDIR statement(s) are updated when you save the map definition by using either the **Save Map** or **Stow Map** function.

## Object Types and Codes

Listed below are the object types and the corresponding object-type codes that can be used to select data definitions:

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

## Using System Variables in a Map Definition

Natural system variables can also be specified in a map definition. For information about system variables, see *System Variables and System Functions* in the *Programming Guide* and the detailed variable descriptions in the *System Variables* documentation.

A system variable must be preceded by an output delimiter as indicated in the following examples:

```
( *TIME
( *DATE
( *APPLIC-ID
```