

Data Area Editor

The Natural data area editor is used to define and maintain definitions for global, local and parameter data areas.

A data area definition can consist of user-defined variables, database views and global data blocks (a collection of variables and/or views).

This section covers the following topics:

- Invoking the Data Area Editor
- Top Information Line
- Bottom Information Line
- Editor Command Line
- Editing a Data Area
- Editor Commands
- Line Commands
- Exit Function
- Sharing Data Area Sources by Natural Version 3.1 and 4.1

Invoking the Data Area Editor

You invoke the data area editor with the system command EDIT, specifying a data area type (GLOBAL, LOCAL or PARAMETER) or the name of a data area with the command (for details, see the system command EDIT as described in the Natural System Command Reference documentation). If you specify the name of a data area, it is read into the edit area of the data area editor.

The data area editor screen appears with a local data area in the edit area:

```

Local      TEST1      Library SAGTEST      DBID 10 FNR 32
Command
I T L Name          F Length      Miscellaneous      > +
All -- -----
* LDA for new application
  1 INCOME          A          20 (1:3,1:5) INIT ALL<'0'>
  1 PERSON
  2 SEX             A           6
  2 AGE             N           3
  1 NAME            A          24
R 1 NAME            /* REDEF. BEGIN : NAME
  2 FIRST-NAME      A          10
  2 MIDDLE-INIT     A           2
  2 LAST-NAME       A          10
C 1 DOLLAR          A           5 CONST<' $US '>
V 1 FINANCE-VIEW   FINANCE
  2 PERSONNEL-NUMBER N          8.0
P 2 MAJOR-CREDIT   (1:1) /* PERIODIC GROUP
  3 CREDIT-CARD     A          18 (EM=XXX.XXX.XXX.XXX.XXX)
  3 CREDIT-LIMIT    N           4.0
  3 CURRENT-BALANCE N           4.0
-- Current Source Size: 625 Free: 61408 ----- S 12 L 1

```

Top Information Line

The top information line of the editor screen displays the type and name of the data area currently in the editor, as well as the library, database ID and file number to which you are currently logged on.

Bottom Information Line

In the bottom information line of the editor screen, the following information is displayed:

| | |
|---------------------|---|
| Current Source Size | Size (number of characters) of the current object. As source lines are stored in variable length in the work area, trailing blanks within a source line are not counted; leading and embedded blanks are counted. This information is only displayed if the "Source Size Information" parameter in the editor profile defaults is set to "Y". |
| Free | The number of characters still available in the work area. This information is only displayed if the "Source Size Information" parameter in the profile defaults is set to "Y". |
| S | Size (number of lines) of the object being edited. |
| L | The number of the source line currently displayed as the top line. |

Editor Command Line

The second line of the data editor screen is the edit command line. In this line, you can enter:

- a Natural system command (for example, EDIT, CHECK, SAVE),
- one or more editor commands,
- the name of a Natural data area to be executed.

In addition, the direction indicator can be set to control the direction of several editor and line commands. The value "+" indicates **after** and the value "-" indicates **before**. The exact interpretation is described with the relevant command description.

Editing a Data Area

The editor screen of the data area editor is divided into columns of fields with the following possible entries:

| Field | Explanation |
|-------|---|
| I | <p>Label Indicator. Information field supplied by the editor. This column is not modifiable by the user.</p> <p>+ indicates that more than one of the entries listed below has been defined using the ".E" line command.</p> <p>Possible entries are:</p> <p>E indicates that a definition error has been detected.</p> <p>A indicates that array bounds have been defined using the ".E" line command.</p> <p>I indicates that an initial value has been defined using the ".E" line command.</p> <p>M indicates that an edit mask has been defined using the ".E" line command.</p> <p>S indicates that both an initial value and an edit mask have been defined using the ".E" line command.</p> <p>Parameter Data Areas only (see also Extended Field Definition Editing):</p> <p>blank indicates the parameter specification BY REFERENCE (default).</p> <p>V indicates the parameter specification BY VALUE.</p> <p>R indicates the parameter specification BY VALUE RESULT.</p> <p>O indicates the parameter specification OPTIONAL.</p> |
| T | <p>Type. Possible types are:</p> <p>B Data block</p> <p>C Constant (user-defined variable only) or Counter field (database field only)</p> <p>* Comment</p> <p>G Group (within a view only)</p> <p>M Multiple-value field</p> <p>O Handle of object</p> <p>P Periodic group</p> <p>R Redefinition</p> <p>U Globally Unique Identifier (GUID)</p> <p>V View</p> |
| L | <p>Level number (1 - 99). Variables which are not within a hierarchical structure must be assigned level 1. View definitions must be assigned level 1. Level numbers cannot be used with data block definitions.</p> |
| Name | <p>Name of the variable, block or view.</p> <p>Instead of specifying a variable name, the filler option (<i>n</i>X) can be used. With the filler option, "<i>n</i>" filler bytes can be denoted within a field or variable being redefined, where "<i>n</i>" can be up to 10 digits (smaller than 1 GB). The definition of trailing filler bytes is optional.</p> |
| F | <p>Format. Any format supported by Natural can be used.</p> |

| Field | Explanation |
|---------------|---|
| Length | Length. No length is permitted for formats C, D, T and L. You can define dynamic variables by specifying "DYNAMIC" in the length field. |
| Miscellaneous | <p>This field can be used to define the bounds for an array, to supply initial values for a variable or to supply an edit mask for a variable; for a view definition, the name of a DDM from which this view is derived must be entered; for a block definition, the name of the parent block must be entered; and a comment can be entered. See also the examples below.</p> <p>Together with an edit mask, also a field header (HD) and the print mode (PM) can be defined: (HD='Name' EM=XXX.XXX.XX PM=N)</p> <p>See the Natural Parameter Reference documentation for further information on the session parameter PM .</p> <p>As this field may be too short to make all necessary or desired specifications, the ".E" line command is provided for extended field definition editing.</p> <p>Note: When defining a view, the name of the DDM from which this view is derived can be modified. However, this is only possible if all fields of the view are also contained in the DDM with the modified name.</p> |

Examples of Array Definitions:

```
(2,2) (2 dimensions, 2 occurrences)
(2,2,2) (3 dimensions, 2 occurrences)
(1:10,2)
(-1:3,2)
```

Examples of Initial Value Assignments:

```
INIT<3>
INIT<'ABC'>
INIT<'H'FF'>
CONST<12>
```

Example of an Edit Mask Definition:

```
(EM=999.99)
```

Editor Commands

The following editor commands can be entered in the command line of the data area editor:

| Command | Function |
|----------------------------------|---|
| <u>C</u> ATALOG [<i>name</i>] | This command catalogs the data area definition currently in the edit area. |
| <u>C</u> HECK | This command checks the data area definition currently located in the edit area. It also orders the entries in the Miscellaneous column in the following sequence: - index - initial value - edit mask - name - comment |
| CLEAR | This command clears the edit area. |
| DX, DY, DX-Y | This command deletes a line marked with "X" or "Y", or a block of lines delimited by "X" or "Y". |
| EX, EY, EX-Y | This command deletes: <ul style="list-style-type: none"> ● source lines from the top of the source area to the line marked with "X", but not including this line; or ● source lines from the line marked with "Y", but not including this line, to the bottom of the source area; or ● all source lines in the source area, excluding the block delimited by "X" or "Y". |
| EXIT | With this command you leave the data area editor. |
| <u>G</u> ENERATE [<i>name</i>] | This command generates Natural copycode using the data area definitions currently in the edit area. A DEFINE DATA LOCAL and corresponding END-DEFINE statement are automatically included. If a <i>name</i> is entered, the generated copycode is saved under this name. |
| M +/- | Scrolls the Miscellaneous field. + Scrolls to the right. - Scrolls to the left. |
| PROFILE [<i>name</i>] | This command displays the current editor profile. |
| READ <i>name</i> | This command reads an existing data area definition into the edit area. |
| SAVE | This command stores a source object. The contents of the source work area are not affected. |
| SET ABS [ON OFF] | This command determines whether the SCAN command operates in absolute or non-absolute mode. ON: the SCAN command operates in absolute mode, which means that the value to be scanned need not be delimited by blanks or special characters. OFF: the SCAN command operates in non-absolute mode, which means that the value to be scanned must be delimited by blanks or special characters. The default is OFF. |

| Command | Function |
|----------------------------------|---|
| SET PREFIX <i>prefix</i> OFF | <p>This command allows you to specify a prefix for field names.</p> <p>This prefix is then automatically placed before the value entered in the "Name" column for each line that is entered or modified, unless the name already begins with this prefix.</p> <p>If the concatenated variable is longer than 32 bytes, a message is given and the value in the name field can be shortened. If this is not done, the prefix will not be inserted.</p> |
| SET SCAN COMMENT NAME | <p>If SET SCAN is set to COMMENT, you can scan for a value in the "Comment" column.</p> <p>If SET SCAN is set to NAME, you can scan for a value in the "Name" column.</p> <p>You cannot scan in both columns simultaneously; the default is NAME.</p> |
| SET SIZE ON OFF | <p>If SET SIZE is set to ON, the size of the data area is displayed at the bottom information line of the editor screen.</p> |
| SET STAY ON OFF | <p>If STAY is set to ON, the current screen will stay when ENTER is pressed. Forward and backward positioning can be done by positioning commands only.</p> <p>If STAY is set to OFF, pressing ENTER positions to the next screen.</p> |
| SET TYPE G L P | <p>This command sets the data area object type:</p> <ul style="list-style-type: none"> G Global data area L Local data area P Parameter data area |
| STOW [<i>name</i>] | <p>This command saves and catalogs the data area definition currently in the edit area.</p> |

Line Commands

All line commands described for the Natural program editor (except those which require a line number) can be used in the data area editor as well.

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

In addition, the following line commands are available for the data area editor:

| Command | Function |
|------------------------|---|
| .D | <p>This command deletes one or more lines.</p> <p>When entered for an individual field, only that field definition is deleted.</p> <p>When entered for a part of a hierarchical structure (view, group, redefinition), all subsequent definitions on subordinate levels are also deleted. If, for example, you enter ".D" for a group defined at level 2, everything belonging to that group and with a level number greater than 2 is also deleted up to (but not including) the next level 2 definition. Comment lines (which usually are not assigned a level) are also considered to be at a subordinate level. To avoid the undesired deletion of a comment, assign an appropriate level to it.</p> <p>Note: In the data area editor, the ".D" command works differently from the program editor.</p> |
| .D(<i>nnnn</i>) | This command deletes <i>nnnn</i> lines, beginning with the line in which you enter the command. Unlike ".D" (see above), ".D(<i>nnnn</i>)" affects only the number of lines specified, regardless of any hierarchical structure. |
| .E | <p>Using this command you can define array bounds, initial values, edit masks and parameter attributes. For more information, see the section Extended Field Definition Editing below.</p> <p>If ".E" is executed for a DDM field, the Define Edit Mask / Header screen is invoked immediately, because only edit masks and headers can be defined for DDM fields. It is not possible to define initial values for DDM fields.</p> |
| .F(<i>file-name</i>) | This command includes a Predict file (applicable to file types: Conceptual, Standard, Sequential, Other). |
| .I(<i>n</i>) | <p>This command adds <i>n</i> empty lines, where <i>n</i> can be in the range from 1 to 9. If <i>n</i> is not (or not correctly) specified, 10 lines (5 lines in split-screen mode) are added by default.</p> <p>If the direction indicator is set to "+", the lines are added after the current line of the object being edited; if the direction indicator is set to "-", the lines are inserted before the current line.</p> |

| Command | Function |
|----------------------------|---|
| .I(<i>obj</i>) | <p>This command includes a Natural object. Apart from data areas, the following object types can be specified:</p> <ul style="list-style-type: none"> programs, subprograms, subroutines, help routines, maps. <p>If the object specified as <i>obj</i> is not a data area, it must be available as cataloged object. A window appears in the data area editor screen where you can select one of the following data definitions to be incorporated into your current data area:</p> <ul style="list-style-type: none"> - all local variables and parameters contained in the specified object (including those incorporated from local and/or parameter data areas), - all local variables contained in the specified object (including those incorporated from local data areas), - only those local variables defined within the specified object, - all parameters contained in the specified object (including those incorporated from parameter data areas), - only those parameters defined within the specified object. <p>If you incorporate variable definitions from objects without a DEFINE DATA definition (that is, from objects coded in reporting mode), variable redefinitions (see the REDEFINE statement in the Natural Statements documentation) might be placed to a wrong position; that is, after the wrong variable. So, before compiling your new data area, check all variable definitions and redefinitions for correct positioning.</p> <p>If a variable redefinition results in more than one variable, each variable is incorporated as one individual redefinition using filler bytes where appropriate.</p> <p>If the specified object has been cataloged using the Natural Optimizer Compiler, initial values and constants cannot be incorporated.</p> <p>If the object you want to insert has features the data area editor does not support, a message will be displayed and the relevant line will be marked as a comment line.</p> |
| .I(<i>obj,ssss,nnnn</i>) | <p>This command includes a global, local or parameter data area. This feature is only supported for data areas which do not contain initial values or edit masks.</p> <p>The "<i>ssss</i>" entry can be used to indicate at which line the insertion is to begin. For example, when setting "<i>ssss</i>" to 20, the insertion begins with the 20th line of the data area. The "<i>nnnn</i>" entry can be used to indicate the number of lines to be inserted.</p> <p>If "<i>ssss</i>" and/or "<i>nnnn</i>" is specified for an object other than a data area (see the .I(<i>obj</i>) command), the specified value(s) are ignored.</p> |
| .R | <p>This command redefines a field or variable.</p> <p>With the filler option (<i>nX</i>), <i>n</i> filler bytes can be denoted within a field or variable being redefined. The definition of trailing filler bytes is optional.</p> |

| Command | Function |
|--|---|
| <p>.V [[<i>view-name</i>[,NOFL]]]</p> | <p>This command defines a view.</p> <p>A view (DDM) layout is displayed. You then select the fields from the view which are to be used in the program.</p> <p>If no view name is specified, the view currently in the split screen is included.</p> <p>If ".V <i>view-name</i>" is specified within a view of the same name as specified for <i>view-name</i>, the selected fields are included in this view and no new view is defined.</p> <p>If NOFL is specified, the selected fields are included without format and length specification.</p> <p>When a periodic group or multiple-value field defined - in a DDM generated with Predict - as "PC" or "MC" respectively is included in a data area, a C* variable (internal count of occurrences) for the group or field is automatically generated and placed before the group or field. The index for such a periodic group or multiple-value field is defined with the number of occurrences defined in Predict. If the number of occurrences has not been defined in Predict, the maximum occurrences (191) are used.</p> <p>If Predict is active, Predict redefinitions and comments are incorporated, too.</p> <p>Note: With VSAM views, always the actual number of occurrences is displayed. In addition, VSAM views contain information on subdescriptors and superdescriptors (for further information, see the Natural for VSAM documentation).</p> |
| <p>.*</p> | <p>This command generates a C* variable for multiple-value fields or fields within a periodic group.</p> |
| <p><i>number</i> [(<i>nnn,m</i>)]</p> | <p>This command is available in split-screen mode and with a view in the split-screen area only.</p> <p>To obtain fields and groups from the split-screen area, the level number of the field or group from the split-screen area must be specified in the first column (without a period "."). The field or group is inserted before or after the current line, depending on the setting of the direction indicator ("+" or "-"). Fields and groups from the split-screen area can be included as fields of a view (if <i>number</i> is entered inside a view) or as user variables.</p> <p>If the selected field has the same name as the field for which the command was entered, it is substituted instead of inserted.</p> <p>Multiple lines can be obtained from the split screen using the "<i>nnn</i>" notation where <i>nnn</i> is the number of lines to be included.</p> <p>The "<i>m</i>" notation can be used to specify a level number to be assigned to the field or group to be inserted.</p> |

Note:

".I(obj.)", ".R" and ".*" are available in full-screen mode only, not in split-screen mode.

Extended Field Definition Editing

To invoke Extended Field Definition Editing, enter the ".E" line command in front of a specific field. The following screen will be displayed:

```

17:11:57          ***** EDIT FIELD *****          2003-08-26
                - Extended Field Editing -

Local   SAGAREA   Library SAGTEST          DBID  10 FNR  49

      Code  Function                               Definition
-----  -
      S     Single Value Initialization           no
      F     Free Mode Initialization             no
      E     Edit Mask Definition                 no
      P     Parameter Type                       no
      A     Array Index Definition              no
      D     Delete all Definitions
      ?     Help
      .     Exit
-----  -

Code   ?   for Field: FIELD1(A10/1:2)
    
```

If any initial values or edit masks have been defined, the corresponding status message in the Definition column is changed from "no" to "yes".

The following function codes are available:

| Code | Function |
|------|---|
| S | <p>This function enables you to define an initial value for the specified field. You need only enter the desired field value; any further specifications necessary (including apostrophes for alphanumeric fields) are generated automatically. For an array (multiple-value field), an initial value can (but does not necessarily have to) be defined for each occurrence.</p> <p>With arrays, asterisk notation (*) can be entered in the command line to repeat the value in the last line of the previous page until the end of the current page.</p> <p>For attribute control variables, a screen is displayed where you can select attributes and colors as initial values. For details on attributes and colors, see the session parameters AD and CD in the Natural Parameter Reference documentation.</p> |
| F | <p>This function, too, enables you to define an initial value for the specified field. However, a free-mode editor is provided where you can enter your initial value definitions according to the common Natural syntax definitions. In this way, for example, the same initial value can be assigned to a whole range of field occurrences at a time. During editing, however, the specified values are not checked (unless you enter the CHECK command).</p> |
| E | <p>This function enables you to define an edit mask and/or header for the specified field, according to the Natural rules for edit mask and header specification.</p> <p>If both an edit mask and a header are specified, together they must not exceed 57 characters in length. However, if only an edit mask is specified, it can be up to 63 characters long; if only a header has been specified, it can be up to 58 characters long.</p> <p>If ".E" is entered for a DDM field, this function is invoked immediately, as only edit masks and headers can be defined for DDM fields. It is not possible to define initial values for DDM fields.</p> |
| P | <p>This function only applies to parameter data areas and enables you to specify a parameter BY REFERENCE (default), BY VALUE or BY VALUE RESULT, and OPTIONAL.</p> <p>See also Parameter-Data-Definition in the DEFINE DATA section of the Natural Statements documentation.</p> |
| A | <p>This function enables you to define array bounds for the specified field. A free-mode editor is provided where you can enter your bound definitions in accordance with the common Natural syntax definitions. While you are editing, however, the specified values will not be checked (unless you enter the CHECK command).</p> |
| D | <p>This function enables you to delete, at a stroke, all definitions made via the "S", "F" and "E" functions.</p> <p>Any "yes" status messages are changed to "no".</p> |

Any definitions made within the Initial Values and Edit Mask/Header subfunctions are immediately incorporated into the data area currently displayed in the data area editor.

For an attribute control variable, only the function codes S, F and A are allowed.

For a field that redefines another field, only the function code A is allowed.

After any of the above listed function codes (except function code D) has been entered on the Extended Field Editing screen, a new screen is displayed. There you can enter one of the following commands in the command line:

| Command | Function |
|----------------|---|
| EDIT | This command returns you to your data area editor screen. |
| . | This command returns you to the previous screen to continue processing. |
| -- | This command returns you to the beginning of the initial value specification(s). It is only available for arrays in Single Value Initialization mode. |
| + | This command takes you one page forward. If the last page has been reached or if there is only one page available, you are returned to your data area editor screen. |
| * | This command copies the initial value of the last occurrence of the previous page to all empty fields of the current page. It is only available for arrays in Single Value Initialization mode. |

Exit Function

If the editor default parameter "Prompt Window for Exit Function" is set to "Y", any time you enter the EXIT command in the command line, the EXIT Function prompt window is invoked, offering you the following options:

| Option | Explanation |
|---------------------|---|
| Save and Exit | Leaves the editor and saves all modifications made to the current object. |
| Exit without Saving | Leaves the editor without saving any modification made to the current object since the last SAVE command was entered. |
| Resume Function | Neither leaves the editor nor saves any modifications; the prompt window is closed and the current function is resumed. |

When the parameter "Prompt Window for Exit Function" is set to "N", the EXIT command leaves the editor and saves all modifications made to the current object; no prompt window is displayed.

Sharing Data Area Sources by Natural Version 3.1 and 4.1

The data area editor uses an internal layout to store data area sources in the FUSER system file. New features and definitions, that are available with Natural Version 4.1, require that the data area source is stored in the FUSER system file using a new and extended source format.

Data areas that are stored using this new source format cannot be used or edited with Natural Version 3.1. The Natural Version 4.1 data area editor supports the former and the new data area source format. The editor can read both formats and converts the version 3.1 format internally to the new source format. As long as no new features are used, data areas are stored using the former Natural Version 3.1 compatible source format by default to allow for sharing the data area between a Natural Version 3.1 and Natural Version 4.1 environment.

The V31COMP compiler option may be used to ascertain that a data area that is edited and cataloged with the Natural Version 4.1 data area editor can still be edited and cataloged with Natural Version 3.1.

V31COMP = ON

The setting of V31COMP is checked when one of the editor commands save, check, catalog or stow is executed. With check, catalog or stow all differences between Version 4.1 and 3.1 relevant for compilation are checked. See the section Compilation Relevant Differences between Version 4.1 and 3.1. For the save command only the features, which cannot be stored using the former version 3.1 compatible data structure, are not allowed.

The following features cannot be saved in the former internal format:

- The first position of the level number field contains other values than blank or 0.
- The length definition in the length field contains more than 4 bytes. This includes the definition of dynamic variables.
- Array bounds that are defined by using the Array Index Definition function of the Extended Field Definition Editing.
- Definition of optional parameters.

A corresponding message is displayed and you can select to return to the editor or to save the data area in the new source format to avoid the loss of changes.

No error will be reported if V31COMP=ON and the system commands READ and SAVE are executed from the NEXT prompt for a data area containing one of the above mentioned new features.

V31COMP = OFF

By default, the editor will save the data area in the old source format. If new features are used, the editor will generate the new internal format automatically.

Note:

With Natural Version 4.1, the default format for storing data areas in the FUSER system file is the format compatible with Natural Version 3.1. With the next major release of Natural after Version 4.1, the default will be changed to the new and extended format introduced with Natural Version 4.1.