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
- The Exit Function
- Defining Globally Unique IDs in the Local and Global Data Area Editors

---

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 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 Leng Index/Init/EM/Name/Comment
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.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 is used to display 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 items of information are 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	Label Indicator. Information field supplied by the editor. This column is not modifiable by the user. Possible entries are: E indicates that a definition error has been detected. I indicates that an initial value has been defined via the ".E" line command. M indicates that an edit mask has been defined via the ".E" line command. S indicates that both an initial value and an edit mask have been defined via the ".E" line command. Parameter Data Areas only (see also Edit Fields): blank indicates the parameter specification BY REFERENCE (default). V indicates the parameter specification BY VALUE. R indicates the parameter specification BY VALUE RESULT.
T	Type. Possible types are: B Data block C Constant (user-defined variable only) or Counter field (database field only) * Comment G Group M Multiple-value field O Handle of object P Periodic group R Redefinition U Globally Unique Identifier (GUID) V View
L	Level number (1 - 9). 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.
NAME	Name of the variable, block or view. Instead of specifying a variable name, the filler option (*n*X) can be used. With the filler option, *"n"* filler bytes can be denoted within a field or variable being redefined, where *"n"* can be in the range from 1 to 253. The definition of trailing filler bytes is optional.
F	Format. Any format supported by Natural can be used.
LENG	Length. No length is permitted for formats C, D, T and L.

Field	Explanation
INDEX/INIT/EM/ NAME/COMMENT	This field can be used to define 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. 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) See the Natural Parameter Reference documentation for further information on the PM session parameter. Since this field may be too short to make all necessary or desired specifications, an additional Edit Fields facility is provided with the ".E" line command. **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.

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
CATALOG [*name*]	This command catalogs the data area definition currently in the edit area.
CHECK	This command checks the data area definition currently located in the edit area. It also orders the entries INDEX/INIT/EM/NAME/COMMENT in the sequence shown on the editor screen.
CLEAR	This command clears the edit area.
CREATE GLOBALS	This command can only be used for a library created using Natural Version 1.2. It collects all global variables contained in cataloged objects (not saved objects) of the current library and places them in a global data area named "COMMON". The asterisk notation can be used to restrict processing to only those objects whose names begin with the specified value. For example: CREATE GLOBALS * (all objects) CREATE GLOBALS ABC* (all objects beginning with ABC)
EXIT	With this command you leave the data area editor.
GENERATE [*name*]	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 *name* is entered, the generated copycode is saved under this name.
PROFILE [*name*]	This command displays the current editor profile.
READ *name*	This command reads an existing data area definition into the edit area.
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.
SET PREFIX <prefix>/off	This command allows you to specify a prefix for field names. 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. 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.
SET SCAN COMMENT|NAME	If SET SCAN is set to COMMENT, you can scan for a value in the "Comment" column. If SET SCAN is set to NAME, you can scan for a value in the "Name" column. You cannot scan in both columns simultaneously; the default is NAME.

Command	Function
SET SIZE ON|OFF	If SET SIZE is set to ON, the size of the data area is displayed at the bottom information line of the editor screen.
SET STAY ON|OFF	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. If STAY is set to OFF, pressing ENTER positions to the next screen.
SET TYPE	This command sets the data area object type: G Global data area L Local data area P Parameter data area
STOW [*name*]	This command saves and catalogs the data area definition currently in the edit area.

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	This command deletes one or more lines. When entered for an individual field, only that field definition is deleted. 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. **Note:** In the data area editor, the ".D" command works differently from the program editor.
.D(*nnnn*)	This command deletes *nnnn* lines, beginning with the line in which you enter the command. Unlike ".D" (see above), ".D(*nnnn*)" affects only the number of lines specified, regardless of any hierarchical structure.
.E	This command invokes a separate screen for the definition of initial values and edit masks. If ".E" is executed for a DDM field, the Edit Mask screen is invoked immediately, since only edit masks (and no initial values) can be defined.
.F(*file-name*)	This command includes a Predict file (applicable to file types: Conceptual, Standard, Sequential, Other).
.I(*n*)	This command adds *n* empty lines, where *n* can be in the range from 1 to 9. If *n* is not (or not correctly) specified, 10 lines (5 lines in split-screen mode) are added by default. 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.

Command	Function
.I(*obj*)	This command includes a Natural object. Apart from data areas, the following object types can be specified: programs, subprograms, subroutines, helproutines, maps. If the object specified as *obj* 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: - 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. 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. If a variable redefinition results in more than one variable, each variable is incorporated as one individual redefinition using filler bytes where appropriate. If the specified object has been cataloged using the Natural Optimizer Compiler, initial values and constants cannot be incorporated.
.I(*obj*,*ssss*,*nnnn*)	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. The "*ssss*" entry can be used to indicate at which line the insertion is to begin. For example, when setting "*ssss"* to 20, the insertion begins with the 20th line of the data area. The "*nnnn*" entry can be used to indicate the number of lines to be inserted. If *"ssss"* and/or *"nnnn" i*s specified for an object other than a data area (see the .I(obj) command), the specified value(s) are ignored.
.R	This command redefines a field or variable. With the filler option (*n*X), *n* filler bytes can be denoted within a field or variable being redefined. The definition of trailing filler bytes is optional.

Command	Function
.V [(*view-name*[,NOFL])]	This command defines a view. A view (DDM) layout is displayed. You then select the fields from the view which are to be used in the program. If no view name is specified, the view currently in the split screen is included. If ".V *view-name*" is specified within a view of the same name as specified for *view-name*, the selected fields are included in this view and no new view is defined. If NOFL is specified, the selected fields are included without format and length specification. 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. If Predict is active, Predict redefinitions and comments are incorporated, too. **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.
.*	This command generates a C* variable for multiple-value fields or fields within a periodic group.
number [(*nnn*,*m*)]	This command is available in split-screen mode and with a view in the split-screen area only. 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 *number* is entered inside a view) or as user variables. If the selected field has the same name as the field for which the command was entered, it is substituted instead of inserted. Multiple lines can be obtained from the split screen using the "*nnn*" notation where *nnn* is the number of lines to be included. The "*m*" notation can be used to specify a level number to be assigned to the field or group to be inserted.

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

Edit Fields

To invoke the Initial Values and Edit Mask menu

- Enter the ".E" line command in front of a specific field.

This feature is not available for redefined fields.

```
17:11:57                *****  EDIT FIELD  *****                    2000-07-12
                     - Initial Values and Edit Mask -

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
           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 of the Initial Values and Edit Mask screen is changed from "no" to "yes".

The following functions are available:

Code	Function
S	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. 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.
F	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).
E	This function enables you to define an edit mask and/or header for the specified field according to the Natural rules for edit mask specification. 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. If ".E" has been executed for a DDM field, this function is invoked immediately, since only edit masks (and no initial values) can be defined for DDM fields.
D	This function enables you to delete, at a stroke, all definitions made via the "S", "F" and "E" functions. Any "yes" status messages are changed to "no".
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. See also Parameter-Data-Definition in the DEFINE DATA section of the Natural Statements documentation.

Any definitions made within the Initial Values and Edit Mask function are immediately incorporated into the data area currently in your data area editor.

Special Commands Available within the Edit Fields Function

The following commands can be entered in the command line of any edit field subfunction:

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.

The ".E" Line Command with Control Variables

When the ".E" line command is entered in front of a control variable, the Define Attributes screen is invoked, where attributes and colors can be specified as initial values for control variables.

For details on attributes and colors, see the session parameters AD and CD in the Natural Parameter Reference documentation.

The 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.

Defining Globally Unique IDs in the Local and Global Data Area Editors

The definition of Globally Unique IDs (GUIDs) requires NaturalX and is possible under TSO and OS/390 Batch only.

To define a GUID, enter "U" in the T(ype) column and fill the L(evel) and the Name columns.

Format and length will be inserted automatically when you confirm your entry.

If it is possible to generate the GUID in the current Natural environment, it will be inserted as a protected const in the free mode editing section.

If the GUID cannot be generated, the information will be displayed in the comment field and the free mode init field that it has not been inserted. In this case, you can insert the GUID into the source of the data area by using the interface USR2022 in the library SYSEXT.

