

# Task-Specific Editors

This section covers the following topics:

- The Description Editor
- The Subquery Editor
- The Rule Editor
- The Link Editor

## The Description Editor

### Functional Scope

The description editor is used for editing

- extended descriptions
- extended description skeletons
- check expression/triggers of SQL tables
- Predict help texts (for more information see the section Maintain Predict Help Texts under Special Functions in the Predict Administration documentation)

The escape character for line commands can be changed in the session profile or with the SET ESC command of the editor.

### Description Editor Specific Commands

The following commands are available in the Description Editor in addition to the general editor commands.

Editor Commands	Description
C[HECK]	Checks if parts of the default extended description skeleton have been deleted by mistake.
DIS[PLAY]	Display a description as it would be displayed using the retrieval output mode Display or the maintenance Display function. If Con-form is installed and the parameter Use Con-form is set to Y in the current session profile, Con-form instructions in the extended description will be executed. Con-form options are described in the section Predict User Interface in the Introduction to Predict documentation.

### Upper and Lower Case in Descriptions when using the Natural Editor

Various Predict and Natural profile parameters determine whether an extended description entered in with the Natural editor is converted to upper case. The following rules apply:

- The Predict profile parameter Upper/lower case > Description is specified in function Defaults > General Defaults > Miscellaneous. This setting is made by the Dictionary Administrator and applies to all Predict users.
- The Natural profile parameters Editing in lower case and Dynamic conversion of lower case can be defined by each user individually.
- The Predict setting has priority over the Natural parameter settings: If Predict default Upper/lower case > Description is set to Upper case, descriptions are stored in upper case irrespective of the settings in your Natural profile.
- If Predict default Upper/lower case > Description is set to Lower case, descriptions are stored in upper or lower case depending on the settings in your Natural profile.

## Overview

The following table shows the effects of the possible combinations of parameter settings.

Profile Parameters			Result	Remarks
Predict	Natural			
Upper/lower case > Description	Editing in Lower Case	Dynamic Conversion of Lower Case		
Upper case	N	N	upper	
	Y	N	upper	Message "UC forced by DDA"
	N	Y	upper	
	Y	Y	upper	Message "UC forced by DDA"
Lower case	N	N	upper	
	Y	N	lower	
	N	Y	upper	
	Y	Y	lower	No dynamic conversion

## Recommended Natural Editor Settings

Recommend settings for editing extended descriptions with the Natural editor:

- **Predict**  
Set Default parameter Upper/lower case >Description to Lower case
- **Natural**  
Set profile parameters Editing in lower case and Dynamic conversion of lower case to Y.

## Upper and Lower Case Settings for the Software AG Editor

A combination of Predict and Natural profile parameters determine whether text edited with the Software AG Editor is converted to upper case. The following rules apply:

- The Predict profile parameter Upper/lower case > Description is specified in function Defaults > General Defaults > Miscellaneous. This setting is made by the Dictionary Administrator and applies to all Predict users.
- The SAG profile parameter CAPS can be defined by each user individually.
- The Predict setting has priority over the Software AG Editor setting: If Predict default Upper/lower case > Description is set to Upper case, descriptions and other attributes are stored in upper case irrespective of the settings in your Natural profile.
- If Predict default Upper/lower case > Description is set to Lower case, descriptions are stored in upper or lower case depending on the settings in your Software AG Editor profile.

The following table shows the effects of the possible combinations of parameter settings.

Profile Parameters		Result	Remarks
Predict	SAG		
Upper/lower case > Description	CAPS		
Upper case	ON	upper	
	OFF	upper	Message "UC forced by DDA" given
	PGM	upper	
Lower case	ON	upper	
	OFF	lower	
	PGM	upper	Dynamic conversion

### Recommended Software AG Editor Settings

Recommended settings for editing extended descriptions with the Software AG Editor:

- **Predict**  
Set Default parameter Upper/lower case >Description to Lower case
- **SAG**  
Set profile parameter CAPS to PGM

## The Subquery Editor

### Functional Scope

The Subquery Editor is used to edit the following:

- subquery clause of an SQL view
- derived field expression

### Subquery Editor Specific Commands

The following commands are available in the Subquery Editor in addition to the general editor commands.

Editor Commands	Description
C[HECK]	Syntax checks are performed on the subquery clause or derived field expression. For example, referenced tables in a subquery clause must exist in Predict.
CAT, SA[VE][R][ETURN]]	Save and quit the editing session. If RETURN is specified: Save and stay in the editing session. Additional syntax checks are performed.

## The Rule Editor

The Rule Editor is used to edit the rule of a verification.

### Rule Editor-Specific Commands

Editor Commands	Description																																																																
CAT [[FREE] R[ETURN]], SA[VE] [[FREE] R[ETURN]]	Save a text as a rule of a verification and leave the editor.  <b>Note:</b> Note: that the SAVE or CAT command does not perform a syntax check. The syntax is checked however, when cataloging a map that uses a rule.																																																																
Option FREE	Catalog/save the edited rule as a free rule. This command is only available when creating new rules and when editing conceptual rules.																																																																
Option RETURN	Save a text as a rule of a verification and stay in editor.																																																																
C[HECK]	Check whether the edited rule's Natural syntax is valid and report errors.																																																																
GEN[ERATE]	Generate a rule from the values defined in the verification and add it to the end of the Natural source in the rule editor. This command is not available for verifications of type U. The table below shows which Natural statements are generated for the different types of verifications: <table border="1" data-bbox="478 880 1441 1986"> <thead> <tr> <th data-bbox="485 889 560 958">Code</th> <th data-bbox="560 889 711 958">Type of Rule</th> <th data-bbox="711 889 828 958">No. of Values</th> <th data-bbox="828 889 1115 958">Generated Natural Statements</th> <th data-bbox="1115 889 1441 958">Generated SQL Clause</th> </tr> </thead> <tbody> <tr> <td data-bbox="485 958 560 1093" rowspan="2">E</td> <td data-bbox="560 958 711 1093" rowspan="2">Equal to</td> <td data-bbox="711 958 828 1008">1</td> <td data-bbox="828 958 1115 1008">IF NOT ( &amp;= value )</td> <td data-bbox="1115 958 1441 1008">&amp; = value</td> </tr> <tr> <td data-bbox="711 1008 828 1093">0 or 1</td> <td data-bbox="828 1008 1115 1093">IF NOT &amp; /* for format logical</td> <td data-bbox="1115 1008 1441 1093">&amp; = value</td> </tr> <tr> <td data-bbox="485 1093 560 1142">G</td> <td data-bbox="560 1093 711 1142">Greater than</td> <td data-bbox="711 1093 828 1142">1</td> <td data-bbox="828 1093 1115 1142">IF &amp; LE value</td> <td data-bbox="1115 1093 1441 1142">&amp; &gt; value</td> </tr> <tr> <td data-bbox="485 1142 560 1191">L</td> <td data-bbox="560 1142 711 1191">Less than</td> <td data-bbox="711 1142 828 1191">1</td> <td data-bbox="828 1142 1115 1191">IF &amp; GE value</td> <td data-bbox="1115 1142 1441 1191">&amp; &lt; value</td> </tr> <tr> <td data-bbox="485 1191 560 1352" rowspan="2">N</td> <td data-bbox="560 1191 711 1352" rowspan="2">Not equal to</td> <td data-bbox="711 1191 828 1272">n</td> <td data-bbox="828 1191 1115 1272">IF ( &amp;= value1 OR= value2 ... )</td> <td data-bbox="1115 1191 1441 1272">&amp; ^= value 1</td> </tr> <tr> <td data-bbox="711 1272 828 1352">0 or 1</td> <td data-bbox="828 1272 1115 1352">IF &amp; /* for format logical</td> <td data-bbox="1115 1272 1441 1352">&amp; ^= value 2 ...</td> </tr> <tr> <td data-bbox="485 1352 560 1433">R</td> <td data-bbox="560 1352 711 1433">Range of values</td> <td data-bbox="711 1352 828 1433">2</td> <td data-bbox="828 1352 1115 1433">IF NOT ( &amp;= value1 THRU value2 )</td> <td data-bbox="1115 1352 1441 1433">&amp; between value1 and value2</td> </tr> <tr> <td data-bbox="485 1433 560 1514">T</td> <td data-bbox="560 1433 711 1514">Table of values</td> <td data-bbox="711 1433 828 1514">n</td> <td data-bbox="828 1433 1115 1514">IF NOT ( &amp;= value1 OR= Value2 ...)</td> <td data-bbox="1115 1433 1441 1514">&amp; in ( value1, value2...)</td> </tr> <tr> <td data-bbox="485 1514 560 1563">U</td> <td data-bbox="560 1514 711 1563">User routine</td> <td data-bbox="711 1514 828 1563"></td> <td data-bbox="828 1514 1115 1563"></td> <td data-bbox="1115 1514 1441 1563"></td> </tr> <tr> <td data-bbox="485 1563 560 1823" rowspan="2">B</td> <td data-bbox="560 1563 711 1823" rowspan="2">Range, but not</td> <td data-bbox="711 1563 828 1675">3</td> <td data-bbox="828 1563 1115 1675">IF NOT ( &amp;= value1 THRU value2 BUT NOT value3 )</td> <td data-bbox="1115 1563 1441 1675">&amp; between value1 and value2 and &amp; ^=value3</td> </tr> <tr> <td data-bbox="711 1675 828 1823">4</td> <td data-bbox="828 1675 1115 1823">IF NOT ( &amp;= value1 THRU value2 BUT NOT Value3 THRU value4 )</td> <td data-bbox="1115 1675 1441 1823">&amp; between value1 and value2 and &amp; not between value3 and value4</td> </tr> <tr> <td data-bbox="485 1823 560 1904">I</td> <td data-bbox="560 1823 711 1904">Not in range</td> <td data-bbox="711 1823 828 1904">2</td> <td data-bbox="828 1823 1115 1904">IF ( &amp;= value1 THRU value2 )</td> <td data-bbox="1115 1823 1441 1904">&amp;not between value1 and value2</td> </tr> <tr> <td data-bbox="485 1904 560 1984">blank</td> <td data-bbox="560 1904 711 1984">(none) - no rule defined</td> <td data-bbox="711 1904 828 1984"></td> <td data-bbox="828 1904 1115 1984"></td> <td data-bbox="1115 1904 1441 1984"></td> </tr> </tbody> </table>	Code	Type of Rule	No. of Values	Generated Natural Statements	Generated SQL Clause	E	Equal to	1	IF NOT ( &= value )	& = value	0 or 1	IF NOT & /* for format logical	& = value	G	Greater than	1	IF & LE value	& > value	L	Less than	1	IF & GE value	& < value	N	Not equal to	n	IF ( &= value1 OR= value2 ... )	& ^= value 1	0 or 1	IF & /* for format logical	& ^= value 2 ...	R	Range of values	2	IF NOT ( &= value1 THRU value2 )	& between value1 and value2	T	Table of values	n	IF NOT ( &= value1 OR= Value2 ...)	& in ( value1, value2...)	U	User routine				B	Range, but not	3	IF NOT ( &= value1 THRU value2 BUT NOT value3 )	& between value1 and value2 and & ^=value3	4	IF NOT ( &= value1 THRU value2 BUT NOT Value3 THRU value4 )	& between value1 and value2 and & not between value3 and value4	I	Not in range	2	IF ( &= value1 THRU value2 )	&not between value1 and value2	blank	(none) - no rule defined			
Code	Type of Rule	No. of Values	Generated Natural Statements	Generated SQL Clause																																																													
E	Equal to	1	IF NOT ( &= value )	& = value																																																													
		0 or 1	IF NOT & /* for format logical	& = value																																																													
G	Greater than	1	IF & LE value	& > value																																																													
L	Less than	1	IF & GE value	& < value																																																													
N	Not equal to	n	IF ( &= value1 OR= value2 ... )	& ^= value 1																																																													
		0 or 1	IF & /* for format logical	& ^= value 2 ...																																																													
R	Range of values	2	IF NOT ( &= value1 THRU value2 )	& between value1 and value2																																																													
T	Table of values	n	IF NOT ( &= value1 OR= Value2 ...)	& in ( value1, value2...)																																																													
U	User routine																																																																
B	Range, but not	3	IF NOT ( &= value1 THRU value2 BUT NOT value3 )	& between value1 and value2 and & ^=value3																																																													
		4	IF NOT ( &= value1 THRU value2 BUT NOT Value3 THRU value4 )	& between value1 and value2 and & not between value3 and value4																																																													
I	Not in range	2	IF ( &= value1 THRU value2 )	&not between value1 and value2																																																													
blank	(none) - no rule defined																																																																
GEN[ERATE] N	Generates a rule for Natural Construct from a verification of status documented (D). The status of the verification will be changed to N.																																																																

Editor Commands	Description																		
GEN[ERATE] S	Generates an SQL clause for all verification types except user routine. When the rule is saved, the status of the verification is changed to S.																		
GLOBALS SM=OFF	Switch to the reporting mode of Natural.																		
GLOBALS SM=ON	Switch to the structured mode of Natural.																		
RENUM[BER], N	Renumber the source lines in steps of 10 and renumber references to them accordingly.																		
RUN, CHECK	<p>Check the edited rule. If no errors are found, a map is produced with which the user can test the rule by entering input values. The following rules apply:</p> <ul style="list-style-type: none"> <li>- Length and format of the input field are derived from the rule format. Whereas CHECK derives the format without further notification, RUN displays an additional window where the derived field length is displayed and can be overwritten.</li> </ul> <table border="1" data-bbox="480 786 1286 1081"> <thead> <tr> <th>Rule Format</th> <th>Format of the derived field</th> <th>Length of the derived field</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>A</td> <td>66</td> </tr> <tr> <td>B</td> <td>B</td> <td>33</td> </tr> <tr> <td>D</td> <td>D</td> <td></td> </tr> <tr> <td>L</td> <td>L</td> <td>1</td> </tr> <tr> <td>N</td> <td>N</td> <td>27</td> </tr> </tbody> </table> <ul style="list-style-type: none"> <li>- RUN tests a rule of format K (function key) without input data.</li> <li>- For a rule of format L (logical), a blank space means false and any other input value means true.</li> <li>- The stack must not be changed.</li> <li>- The contents of the source area must not be changed.</li> </ul> <p><b>Note:</b> All variables used except the ampersand (&amp;) must be defined within the code.</p> <ul style="list-style-type: none"> <li>- The variable names SYSDIC-C1 and SYSDIC-C2 are used for internal purposes and must not be used within the rule.</li> <li>- The source will be renumbered.</li> </ul>	Rule Format	Format of the derived field	Length of the derived field	A	A	66	B	B	33	D	D		L	L	1	N	N	27
Rule Format	Format of the derived field	Length of the derived field																	
A	A	66																	
B	B	33																	
D	D																		
L	L	1																	
N	N	27																	

## The Link Editor

The Link Editor is a modified Natural Editor. The functionality offered by this editor is not available with the Software AG Editor.

- Functional Scope,
- Calling the Link Editor,
- Standard Link Editor commands,
- Type-specific List Editor Functions,

## Functional Scope

The Link Editor is used to edit the following lists:

- child objects, for example fields of a file
- owners to which an object is assigned
- entry points of a program.

The Link Editor displays a formatted list of related objects.

For each object in the list, certain attributes are displayed. For user-defined object types, the data dictionary administrator specifies which attributes of the object are to be displayed in the Link Editor. For an explanation of the columns in an object list, see the corresponding section in the Predefined Object Types in Predict documentation.

The example below shows the field list of a file.

>		> + Fi: JCA-FI1				L: 1		S: 3			
Ty	L	Field ID	F	Cs	Length	Occ	D	U	DB	S	All
*- -	-	-----	*-	*	-----	-----	*	*	-	-	*
	1	JCA-EL1	A		2.0				AA	N	
	1	JCA-1	F		4.0				AB	N	
SB	1	JCA-DER	A		2.0				AC	N	

## Calling the Link Editor

The Link Editor is called using one of the following methods:

- Enter Y in one of the following fields in any Add, Copy or Modify screen:
  - EDIT: Owner to edit the owner list of the current object.
  - EDIT: <active association>, for example *Uses PR concept.* in a system maintenance screen to edit the program list of a system
  - EDIT: Entries in a program maintenance screen to edit the entry point list of a program.
- Call an Edit or Link function in any maintenance menu, for example Edit entry points in the program maintenance menu to edit the entry point list of a program.
- Enter an EDIT or LINK command, for example LINK FILE ELEMENT <file-id> to edit the field list of the specified file.

## Link Editor Commands for all Object Types

The following commands are valid for all predefined and user-defined object types in Predict. Type-specific Link Editor commands are given below.

Editor Commands	Description
C[HECK]	Performs various validation checks, for example that no IDs in the list are duplicated.
SEL	Displays a list of objects for selection. A screen appears in which type-specific criteria can be entered to limit the scope of the selection. Objects can then be inserted in the current list either at the end (mark with +) or at the beginning (mark with -). To insert an object at a given position, enter the line command .H simultaneously. Objects will then be inserted after the line marked with .H. <b>Note:</b> This command allows you to include fields from another file in the field list currently being edited. It can be used to pull forward fields from a standard file to a master file or from a master file to a related user view.
SORT	Sorts the list of object IDs alphabetically. When sorting field lists containing structures or redefined fields, the following applies: A CHECK command must be executed before a field list containing a structure/redefinition that has been changed is sorted. Fields not on level 1 are not sorted, so group structures are not changed. <b>Note:</b> This command does not apply to verification lists.

Line Commands	Description
.E	Skip to the Add or Modify screen for the object on the current line so that the dictionary object can be created or changed, then return to this editor screen.
.E(n)	Display screens for the next n objects in the list.

## Type-specific List Editor Functions

### Database

The following commands can be used for the file list of a database.

Editor Commands	Description
SORT LOG	Sort the list of file IDs by logical file number.
SORT PHY	Sort the list of file IDs by physical file number.

Line Commands	Description
.A	Calls the Modify Adabas Attributes screen for the file.
.E	Calls the Add file screen for files that have just been added to the file list of the database (dummies) or the Modify file screen for files that already exist.
.T	Calls the Modify Vista element screen for the file.

### File - all File types

The following commands can be used for the field list of a file:

Editor Commands	Description
ADA	Generate field short names for fields that do not already have a short name.
NU[LL]	Only applicable for new fields: If the suppression option of a field is blank, it is reset as follows: - For SQL tables: the suppression option is set to U (null allowed). - For other files, the suppression options is set to N (null value suppression).
READA	Delete any existing field short names and generate new ones for all fields. Only applicable to fields with attribute Field short name.
SORT ADA	Sort the field alphabetically by field short name. Fields not on level 1 are not sorted, so group structures are not changed. Only applicable to fields with attribute Field short name.
SORT	Sort the fields alphabetically by field ID. Fields not on level 1 are not sorted, so group structures are not changed.
SET ADA [ON]	Apply future SCAN commands to field short names instead of field IDs.
SET ADA OFF	Cancel the command SET ADA [ON].

Line Commands	Description
.E	The Add or Modify Field screen is called for the current field.
.E(n)	The Add or Modify Field screen is called for the next n fields in the list.

### File - SQL File Types

The following commands can be used to change the layout of input fields in the link lists of SQL file types, for example DB2 table or ORACLE view.

FLIP C IDs of source fields and columns of derived SQL views are shown in full length (32 characters):

Ty L Field ID	from Field ID
*_ - -----	-----

FLIP T IDs of source fields and tables of derived SQL views are shown in full length (32 characters):

Ty L Field ID	from Table/View ID
*_ - -----	-----

FLIP Return to normal editor input:

Ty L Field ID	from Table/View ID	Field ID
*_ - -----	-----	-----

## Owner

Line Command	Description
.E	The system behavior depends on whether the owner is already assigned to a user: - If an owner is not already assigned to a user, a screen is displayed in which you can enter a user ID. Then the Add a user screen is shown for entering additional information. - If the owner is assigned to one user only, the Modify user screen is shown. - If the owner is assigned to more than one user, a selections screen is displayed in which you can mark a user for modification.

## Program

The following commands can be used for file, module, report, program or entry point lists.

Editor Commands	Description
ACTIVE	Insert information from XRef data into the object list. Mark objects that are used with <active, and mark objects that are not used with <unused.  XRef data without a corresponding documentation object is marked as *NOT DOCUMENTED*. An object ID can then be entered and the .E command can be used to create a Predict object corresponding to the XRef data. The implementation pointer for the new object is derived from XRef data and automatically inserted into the input fields of the Add menu.
UPDATE	Update active reference data in the object list. Mark used objects with <active and delete unused objects from the list. Comments on the ACTIVE command (above) also apply to this command.
RESET	Switches back to normal edit mode after ACTIVE or UPDATE have been issued. Information displayed in the right column is no longer derived from XRef data but is taken from the Predict objects. All lines marked *NOT DOCUMENTED* are removed from the list.  X and Y marks and scan values specified with the SCAN command are reset (as with the RESET command in any other list editor).