PREDICT

Administration

Version 4.3.1

software AG

This document applies to Predict Version 4.3.1 and to all subsequent releases. Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Table of Contents

Administration - Overview

This documentation describes functions normally performed by the Predict administrator to customize and protect a Predict environment.

Defaults	Handling of Predict and the way Predict functions work can be customized with Predict default parameters.
Special Functions	Functions to administrate Predict.
User Exits	Predict functions can be customized using User Exits.
Predict in Batch Mode	An overview of the functions that can be executed in batch mode. Operating system specific information, such as workfiles and additional reports, is provided where required. How to start and finish a Predict/Natural session, and how to enter Predict commands. Sample commands are provided.
Conversion	Conversion of Predict data from Version 4.2.
Metadata Administration	The Predict metastructure (object and association type definitions) can be extended and modified. Retrieval models can be defined to specify precisely the amount of data to be retrieved from the Dictionary.
Application Programming Interface	The functions provided by the Application Programming Interface (API) enable direct read/write access to Predict objects of user-defined types.

Predict Defaults

Parameters determining the way functions work can be modified centrally with functions of the Defaults menu.

This section covers the following topics:

- Defaults Menu
- General Defaults
- Extended Description Skeleton
- DEFAULT Profile
- LIST XREF Default Profile
- Generation Defaults
- Coordinator Defaults
- Defaults for Adabas Native SQL
- Activate User Exits

Defaults Menu

The Defaults menu is called with code D in the Function Main Menu or with the command DEFAULT.

```
13:46:51             *****  P R E D I C T  4.3.1  *****             2003-05-31
Plan  10                        -Defaults -

                      Function

                      D  General defaults
                      E  Extended description skeleton
                      S  DEFAULT profile
                      L  LIST XREF default profile
                      G  Generation defaults
                      C  Coordinator defaults
                      A  Adabas Native SQL defaults
                      U  User exits

Function ..............

Object type ............*
Subtype for skeleton ...*
Language for skeleton ..*

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

Parameters	
Object type	For function *Extended description skeleton* (code **E**): Documentation object type for which the skeleton is to be edited. Mandatory. For function *Generation defaults* (code **G**): The type of external object for which the generation defaults are to be modified. Mandatory.
Subtype for skeleton	For function *Extended description skeleton*: Subtype of Object type for which the skeleton is to be edited. Optional. For example: To edit an extended description skeleton for Adabas Databases, specify Object type=DA and Subtype for skeleton=A.
Language for skeleton	For function *Extended description skeleton*: If Object type=PR, language for which the skeleton is to be edited. Optional.

General Defaults

This section covers the following topics:

- Maintenance Options
- Redocumentation Using Source Code / Using XRef Data
- Protection
- Synonyms
- Suppress Display of Products
- Miscellaneous
- Default Adabas Attributes

The General Defaults menu is called with code D in the Defaults menu. Each of the functions in the menu displays one or more subsequent input screens.

```
13:36:08             *****  P R E D I C T  4.3.1  *****             2003-05-31
Plan  10                   - General Defaults -


                    Function

                    M  Maintenance options
                    R  Redocumentation using source code
                    G  Redocumentation using xref data
                    P  Protection
                    S  Synonyms
                    D  Suppress display of products
                    C  Miscellaneous
                    A  Default Adabas Attributes


Function ..........




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

The table below provides an overview of where to find a specific parameter.

Default Parameter	Function
Adabas D	Suppress display of products
Adabas SQL Server	Suppress display of products
Adabas Vista	Suppress display of products
AOS	Protection
Current network	Miscellaneous
DB2 and SQL/DS	Suppress display of products
Default current Virtual Machine	Miscellaneous
Default device for ASSO, DATA	Maintenance options

Default Parameter	Function
Entire System Server	Suppress display of products
Error transaction in SYSDIC	Miscellaneous
General SQL support	Suppress display of products
Implementation DBnr/Fnr	Maintenance options
Implementation Library	Maintenance options
IMS	Suppress display of products
Informix	Suppress display of products
Ingres	Suppress display of products
Languages Allowed	Miscellaneous
LEASY and ISAM BS2000	Suppress display of products
Modification log	Maintenance options
Modify description	Maintenance options
Modify links	Maintenance options
Modify owners	Maintenance options
Multilingual Support	Miscellaneous
Natural DB2 Defaults	Miscellaneous
Natural IMS Defaults	Miscellaneous
Network support	Miscellaneous
Old mode synonyms	Synonyms
Oracle	Suppress display of products
PC text modifiable on Mainframe	Protection
Position for languages	Synonyms
Position of ’&’	Miscellaneous
Protect current Predict file	Protection
rdb	Suppress display of products
RMS	Suppress display of products
Rule in Map Editor	Protection
Rule in SYSDIC	Protection
Scan texts allowed	Protection
Start in the NDB/Start in logical	Miscellaneous
Static SQL XREF	Miscellaneous
Store user ID of modifier	Maintenance options
Sybase	Suppress display of products
SYSDB2 utility	Protection
SYSDDM utility	Protection

Default Parameter	Function
Unique DBnr/Fnr	Miscellaneous
Upper/Lower-case	Miscellaneous
Use Software AG Editor for outputs	Miscellaneous
Use userview for update	Maintenance options
VSAM	Suppress display of products

Maintenance Options

The Maintenance Options screen is displayed with code M in the General Defaults menu.

```
13:36:33            *****  P R E D I C T  4.3.1  *****            2003-05-31
                          - Maintenance Options -
                                                  Modified 2003-05-31 at 11:28
                                                        by STK

Implementation Library .........* A  Allowed
Implementation DBNR/FNR ........* A  Allowed
Use userview for update ......... Y (Y,N)
Default device for ASSO ........* 3390
Default device for DATA ........* 3390

Modification log
  Modify links .................. Y (Y,N)
  Modify owners ................. Y (Y,N)
  Modify description ............ Y (Y,N)
  Store user ID of modifier ..... Y (Y,N)




For HELP enter '?' in the first field.
```

Parameters	
Implementation Library, DBnr/Fnr	These parameters apply when maintaining programs. **F** Forced: Library and/or DBNR/FNR must be specified. **A** Allowed: Library and/or DBNR/FNR of the implemented program documented with the Predict program object can be specified. **D** Disallowed. Library and/or DBNR/FNR may not be specified. For 3GL library, use *SYSALL*.

<table>
<tr><th colspan="2">Parameters</th></tr>
<tr><td>Use userview for update</td><td>Determines whether a userview is used to update data. This parameter affects the possible format changes to fields within a userview.

For example:

• If a userview is used to update data (UPD), the field format cannot be changed from numeric to alphanumeric; this format change is not update-compatible.
• If a userview is used for read-access only (ACC), it is possible to change the field format from numeric to alphanumeric; this format change is compatible.

Y
Format change to a field in a userview must be update-compatible to the format of the field in the master file. See following table.

<table>
<tr><th>Field Format in Master File</th><th>Permitted Format of Fields in Userview (update-compatible)</th></tr>
<tr><td>B</td><td>B, I, N/U, NS/US, P, PS</td></tr>
<tr><td>N/NS/U/US</td><td>B, I, N/U, NS/US, P, PS</td></tr>
<tr><td>P/PS</td><td>B, I, N/U, NS/US, P, PS</td></tr>
<tr><td>I</td><td>B, I, N/U, NS/US, P, PS</td></tr>
</table>

N
Format change to a field in a userview must be compatible to the format of the field in the master file. See following table.

<table>
<tr><th>Field Format in Master File</th><th>Permitted Format of Fields in Userview (compatible)</th></tr>
<tr><td>D</td><td>D, P6</td></tr>
<tr><td>T</td><td>T, P12</td></tr>
<tr><td>N/NS/U/US</td><td>A, B, I, N/U, NS/US, P, PS</td></tr>
<tr><td>P/PS</td><td>A, B, I, N/U, NS/US, P, PS</td></tr>
<tr><td>I</td><td>A, B, I, N/U, NS/US, P, PS</td></tr>
<tr><td>B</td><td>A, B, I</td></tr>
<tr><td>B 4</td><td>N/U, NS/US, P, PS</td></tr>
</table>
</td></tr>
<tr><td>Default device for ASSO</td><td>If Adabas attributes are added to a Predict file object and the device cannot be taken from the database containing the file, the default device is taken. The default device must have been defined with the special function Adabas Device Types.</td></tr>
<tr><td>Default device for DATA</td><td>See Default device for ASSO above.</td></tr>
</table>

Parameters	
Modification Log	Modifications to Predict objects are recorded in a modification log containing the following information: • ID of the user who created the object • ID of the user who last modified the object • date and time of the last modification. The modification log is updated whenever an attribute of an object is changed unless the scope of the log is limited by setting one or more of the following four parameters to **N**.
Modify links	**N** Modification log will not be updated when the object list of an object is changed.
Modify owners	**N** Modification log will not be updated when the owner list of an object is changed.
Modify description	**N** Modification log will not be updated when the extended description of an object is changed.
Store user ID of modifier	**N** Information on who created or changed an object will not be stored.

Note:
The default setting of all these parameters is **Y**.
The display of this information can be suppressed with the Output Options parameter Display modifier of the session profile. See the section **Predict User Interface** in the **Introduction to Predict documentation**.

Redocumentation Using Source Code / Using XRef Data

The Redocumentation Using Source Code / Using XRef Data screens are displayed with the code R or G in the General Defaults menu.

```
13:38:18             *****  P R E D I C T  4.3.1  *****             2003-05-31
                   - Redocumentation Using Source Code -
                                                      Added 2003-05-31 at 10:23
                                                   Modified 2003-05-31 at 09:49
Mark with 'X' the options which may be modified by the user.
Processing option
X Processing option ..........* L  List only
X Link to system ..............
Naming option
X Program ID prefix ...........
X Lib.name as sec.prefix ...... Y (Y/N)
Contents of documentation                          Implementation pointer
X Abstract ...................* S  Statistics          X Library ..... Y (Y/N)
X Description ................* B  Header comment      X Fnr ......... Y (Y/N)
X Replace/append description .. R (R/A)                X DBnr ........ Y (Y/N)
X Program list ...............* U  Update
X File list ..................* U  Update
X Default owner ...............
X First default keyword .......
  Second default keyword ......

  Handle /* in columns 1+2 as comment or as remark .. R (C/R)
For HELP enter '?' in the first field.
```


These defaults are used by the program maintenance function Redocument program. The individual parameters are described in Specifying the Redocument Parameters of section **Program** in the documentation **Predefined Object Types in Predict**.

Individual input fields can be protected by deleting the preceding X.

Protected default values cannot be overwritten by the user, and protected fields are skipped when positioning the cursor with TAB in the Redocument program function.

Protection

The Protection screen is displayed with code P in the General Defaults screen.

```
13:47:24            *****  P R E D I C T  4.3.1  *****            2003-05-31
                              - Protection -
                                                  Modified 2003-05-31 at 10:30
                                                        by STK

SYSDDM utility ..................* A  Allowed
AOS .............................* A  Allowed
SYSDB2 utility ..................* I  Incorporate
Rule in map editor ..............* N  No check is performed
Rule in SYSDIC ..................* N  No check is performed
Scan texts allowed ..............* A  All allowed
PC text modifiable on Mainframe .. Y (Y,N)
Protect current Predict file ..... Y (Y,N)
  DBnr of NSC file ............... 180
  Fnr of NSC file ................ 55
  Password .......................
  Cipher .........................
```

Parameters	
SYSDDM utility	Determines whether SYSDDM can be used to perform tasks that can be solved with Predict or with SYSDDM. SYSDDM functions **not** available in Predict are always unprotected. **A** Allowed: all SYSDDM functions can be executed. **D** Disallowed: SYSDDM functions that solve tasks which can be solved with Predict cannot be executed. **C** Connected: functions of SYSDDM that solve tasks which can be solved with Predict cannot be executed for DDMs connected to Predict files. **Note:** If SYSDDM utility is set to D or C, SYSDDM rejects attempts to modify DDMs with the message: "Function disallowed according to Predict definition".

Parameters	
AOS	Determines whether AOS can be used to perform tasks that can be solved with Predict or with AOS. AOS functions **not** available in Predict are always unprotected. **A** Allowed: all AOS functions can be executed. **D** Disallowed: functions of AOS that solve tasks that can be solved with Predict cannot be executed. **C** Connected: functions of AOS that solve tasks that can be solved with Predict cannot be executed for physical files which are connected to Predict files. **Note:** If AOS is set to D or C, SYSAOS rejects attempts to modify Adabas files with the message: "AOSPRD01: Function disallowed by Predict's DDA services".
SYSDB2 utility	Determines whether SYSDB2 can be used to perform tasks that can be solved with Predict or with SYSDB2. SYSDB2 functions **not** available in Predict are always unprotected. **A** Allowed: all SYSDB2 functions can be executed. **D** Disallowed: the following SYSDB2 functions cannot be executed: • CREATE DATABASE • CREATE STORAGEGROUP • CREATE TABLE • CREATE TABLESPACE • CREATE VIEW • CREATE INDEX **I** Incorporate: all SYSDB2 functions can be executed outside of Predict. If one of the following statements is submitted to DB2, an automatic incorporation in Predict is performed: • CREATE DATABASE • CREATE STORAGEGROUP • CREATE TABLE • CREATE TABLESPACE • CREATE VIEW **Note:** If SYSDB2 utility is set to D, SYSDB2 rejects CREATE statements with the message: "Statement disabled by Predict".
Rule in Map Editor /Rule in SYSDIC	The parameters Rule in map editor/SYSDIC are used to control who may modify free processing rules (documented as Predict objects of type Verification) and where rules can be modified. Natural Security must be installed if you want to use Rule in map editor options Y and F (see below). The protection mechanism for free rules activated with the parameters Rule in map editor/SYSDIC evaluates the verification attribute modifier. See the section Verification in the **Predefined Object Types in Predict documentation**.

Parameters	
Rule in map editor	Specifies how free rules are protected in the Natural map editor. **Y** Only users specified as modifiers in the Predict verification objects may change a free rule. If no modifier is specified, any user can modify a rule. **N** No check is performed. **F** Force: Predict verifications must have at least one modifier. Only users specified as modifiers may change a rule. **D** Disallowed: Free processing rules may not be modified in the Natural map editor. Modify free rules in SYSDIC and then regenerate them. **Note:** If Rule in map editor is set to D, the Natural map editor rejects attempts to modify rules with the message: "Modification rejected by User Exit".
Rule in SYSDIC	Specifies how free rules are protected in Predict. **Y** Only users specified as modifiers in the Predict verification objects may change a rule. If no modifier is specified, any user may modify a verification object. **N** No check is performed. **F** Force: Predict verifications must have at least one modifier. Only users specified as modifiers may change a rule.

Parameters

Scan texts allowed

A text string can be specified for any selection operation which can be limited by Restrictions. Only objects containing this string in one or more of the following are evaluated:

- abstract
- extended description
- rule
- ID

Default values for restrictions are defined in the section Restriction of the profile. See Restrictions in the section **Predict User Interface** in the **Introduction to Predict documentation**.

The table below lists the possible values for this parameter. A tick indicates that the user has the possibility of scanning Abstracts/Descriptions/Rules/IDs for the specified text string.

Code	Abstracts	Descriptions	Rules	IDs
D				
A	Y	Y	Y	Y
C	Y			
S	Y	Y		
T	Y		Y	
J	Y			Y
N	Y	Y	Y	
E	Y	Y		Y
P	Y		Y	Y
O		Y		
U		Y	Y	
L		Y		Y
Q		Y	Y	Y
R			Y	
M			Y	Y
I				Y

Note:
Full text search can significantly slow down processing. It may therefore be appropriate to switch off this feature for daily use and activate it only when really needed.

PC text modifiable on Mainframe

Y
PC texts for which a binary text exists can also be modified on the mainframe. The binary text is deleted.

Parameters	
Protect current Predict file	**Y** Your Predict environment is to be protected with Predict Security. See the Predict Security documentation for more information. **Note:** This parameter can be set to Y only after all control records have been added in Natural Security with the special function Maintain NSC Definitions > Add NSC Default Definitions. If you execute the function Purge NSC Definitions, this parameter is automatically reset to N. See Maintain NSC Definitions in the section **Special Functions** in this documentation. If this parameter is set to Y and no Fnr or DBnr is specified for the Natural Security file with the parameters below, the user exit U-SEC is called. This user exit is delivered in source form and allows you to define your own security checks. See the section User Exits in this documentation, and also section Protecting Predict with other Security Systems in the **Predict Security documentation**. The following parameters are only applicable if Protect current Predict file is set to **Y**.
DBnr/Fnr of NSC file	Database and file number of the Natural Security file containing the security definitions.
Password/Cipher	Password and cipher of Natural Security file containing the security definitions (if applicable).

Synonyms

The Synonyms screen is displayed with code S in the General Defaults menu.

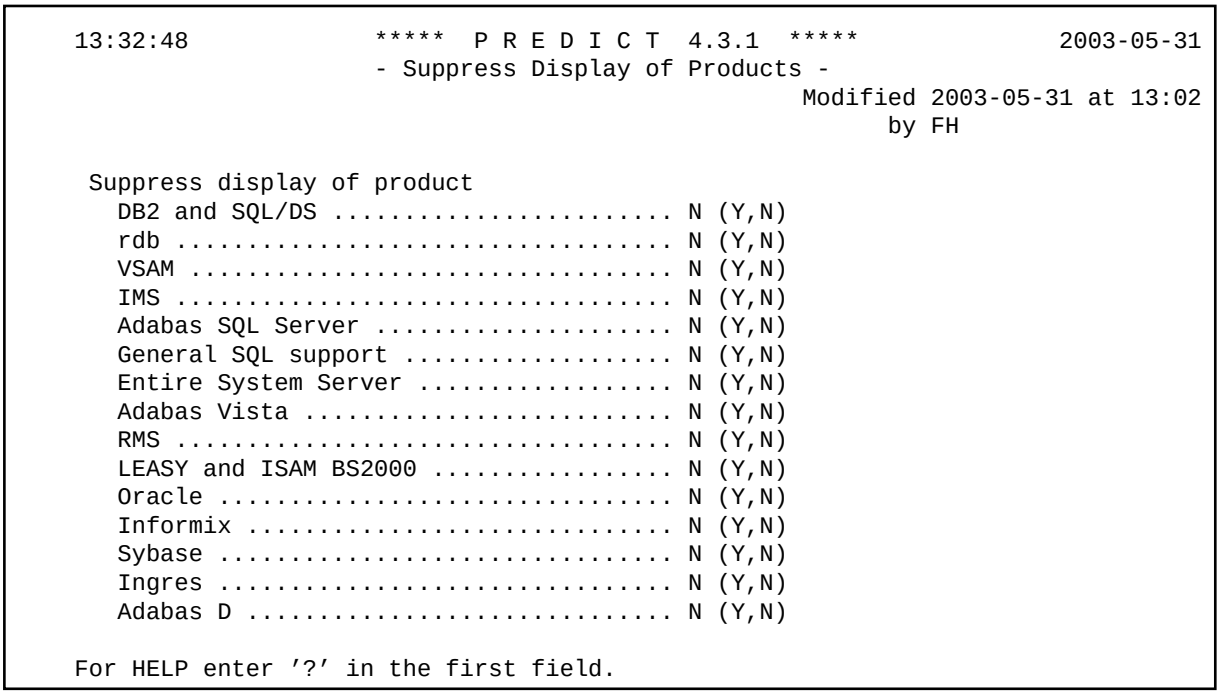
Up to 10 language-specific synonyms can be specified for any Predict object of type field. The order of storage and display of synonyms is as shown in the screen below.

```
13:08:05             *****  P R E D I C T  4.3.1  *****              2003-05-31
                                  - Synonyms -
                                                        Added 2003-05-31 at 08:21
                                                     Modified 2003-05-31 at 19:28

 Old mode synonyms ..... Y (Y,N)


 Position for languages
   Natural ............. 1 (1-9)
   COBOL ............... 2 (1-9)
   PL1 ................. 3 (1-9)
   BAL ................. 4 (1-9)
   FORTRAN ............. 5 (1-9)
   PASCAL .............. 6 (1-9)
   ADA ................. 7 (1-9)
   C ................... 8 (1-9)
   User specific ....... 9 (1-9)
```

Parameters	
Position for languages	The digits in these fields specify the sequence in which the language-specific synonyms of fields are stored and presented in maintenance and retrieval functions. The category User specific can be used to specify synonyms for any other language not contained in the list.
Old mode synonyms	This option is only provided for compatibility with old versions of Predict. **N** Default setting. Compatibility with old versions is not required. **Y** Up to 90 synonyms can be defined as Natural synonyms. These create additional entries in the DDM which have the same attributes as the original object but different names.

Suppress Display of Products

Products that are not installed can be suppressed in Predict selection windows by specifying Y in the respective input field of the screen above. The Suppress display of products screen is displayed with code D in the General Defaults screen.



Rules

- If a Predict object specific to one of the products above exists (for example a dataspace object for DB2 or SQL/DS), a product cannot be suppressed. The input field in the Suppress Display of Products screen is then protected.
- If no product-specific objects exist, the default value for this parameter is Y.
- Even if a product is suppressed, the following functions/utilities can create Predict objects specific to this product.
  - Coordinator
  - Conversion utility

- Predict Case Schema Generator
- Incorporation functions (when accessing a Predict system file using another Natural nucleus where the product is installed)
- Maintenance functions (when accessing a Predict system file using another Natural nucleus where the product is installed)

- If objects for a suppressed product are added to the dictionary with one of the above functions/utilities, the respective Suppress display of products flag is automatically set to N.

Note:
If Adabas Vista is set to Y, Predict assumes that Adabas Vista is not available at your installation. In this case the physical DBnr and the logical DBnr are equal. The Logical distribution type of a file can only be blank (simple file), E (expanded) or N (PROPAGATOR). All databases are isolated or local.
The flag can be set to Y only if Adabas Vista is not installed.

Miscellaneous

The two Miscellaneous screens are displayed with code C in the General Defaults screen.

```
13:05:00            *****  P R E D I C T  4.3.1  *****             2003-05-31
                             - Miscellaneous -
                                                  Modified 2003-05-31 at 13:22
                                                        by GER

Multilingual support                          Upper/lower case
  Multilingual support .. Y (Y,N)               Object ID ...........* U Upper case
  Position of '&' ....... 5 (2-9)               Description .........* L Lower case
                                                Abstract ............* L Lower case
Allowed languages                               Edit mask ...........* L Lower case
  01 Y 02 N 03 N 04 N 05 N 06 N 07 N 08 N       Natural header ......* U Upper case
  09 N 10 N 11 N 12 N 13 N 14 N 15 N 16 N
  17 N 18 N 19 N 20 N 21 N 22 N 23 N 24 N Natural IMS defaults
  25 N 26 N 27 N 28 N 29 N 30 N 31 N 32 N       Start in the NDB ..... NA
  33 N 34 N 35 N 36 N 37 N 38 N 39 N 40 N       Start in logical ..... HA
  41 Y 42 N 43 N 44 N 45 N 46 N 47 N 48 N
  49 N 50 N 51 N 52 N 53 N 54 N 55 N 56 N Natural DB2 defaults
  57 N 58 N 59 N 60 N                           Static SQL XREF  ....* N No

Error Transaction in SYSDIC ..
Use Software AG Editor for outputs ... Y  (Y,N)

For HELP enter '?' in the first field.
```

```
13:59:14            *****  P R E D I C T  4.3.1  *****             2003-05-31
                             - Miscellaneous -                       Page: 2
                                                  Modified 2003-05-31 at 13:22
                                                        by GER

Network support
  Unique DBnr/Fnr ................... N (Y,N)
  Current network ..................* HOME
  Default current virtual machine ..* HOME
```

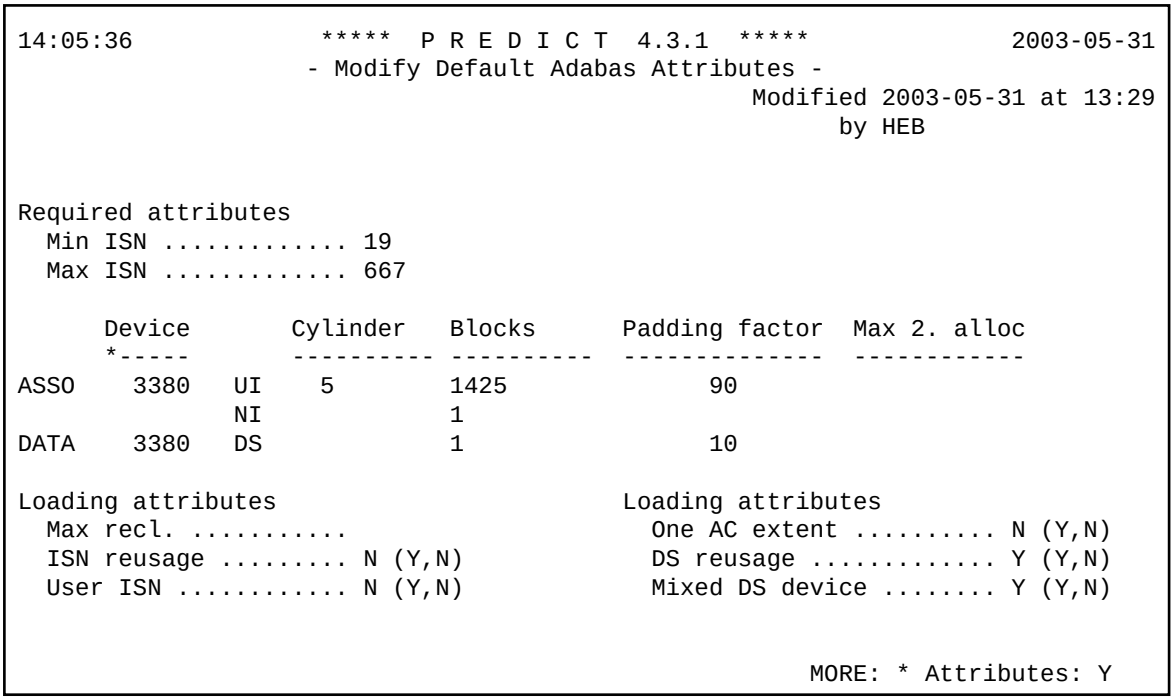
Parameters	
Multilingual Support	
Multilingual Support	**Y** Multilingual Maps (see the section **Natural Map Editor** in your **Natural documentation**) are to be supported by the Predict Retrieval functions. If Multilingual support is set to Y, at least two languages must be allowed.
Position of '&'	The position of the language code in any map name. If 9 is specified, the last character of any map name is interpreted as the language code of the map name.
Allowed languages	Enter Y after the numeric code to activate the languages to be supported. The parameter Multilingual support (see above) must be set to Y, and at least two languages must be allowed. The code corresponds to the Natural system variable *LANGUAGE: **1** English **2** German **3** French **4** Spanish **5** Italian **6** Dutch **7** Turkish **8** Danish **9** Norwegian For a complete list of all supported languages, see **System Variables - *LANGUAGE** in the **Natural System Variables documentation**.
Error transaction in SYSDIC	The transaction (Natural program) specified must exist and be coded according to the rules of a Natural error transaction. Predict will execute the Natural statement *ERROR-TA=transaction-name Even if Natural Security is installed, this transaction will override the error transaction specified in Natural Security!
Use Software AG Editor for outputs	With this version of Predict, output from a retrieval operation is placed in an Software AG Editor session as standard. This editor offers a variety of advantages when handling retrieval output. See the section Editors in Predict in the **Predict Reference documentation**. **Y** Default. Software AG Editor is called for further processing of output from the following functions: • Retrieval, active retrieval: all functions except Select • File implementation: Display implementation plan • File, database, dataspace and storagespace administration: function Display. **N** The result of a retrieval operation is output as in earlier versions of Predict. This setting only makes sense if you previously used the User Exit U-DSP to process your output and have not yet converted this User Exit for use with the Software AG Editor. See the section User Exits in this documentation.

Parameters	
Multilingual Support	
Upper/Lower case	**U** Attribute values are converted to upper case. **L** The following attribute values will be stored in upper/lower case as entered: • Object IDs (this is not generally recommended, but see note for SQL objects below) • Abstract • Extended descriptions and Predict online help texts (both with parameter Description) • Natural headers (hdr-1, hdr-2 and hdr-3) • Edit masks (NAT editm) that are stored for fields. All other types of attributes are always converted to upper case. **Note:** For SQL objects (for example files of type Oracle view or databases of type DB2): If parameter Upper/Lower case is set to L for Object ID, the following are also stored in upper/lower case as entered: • Triggers • DV field expressions • SQL verifications • Check expressions • Constraint names **Note:** If you are using an SQL system that allows names in mixed case, you must set parameter Upper/Lower case to L for Object ID if you wish to incorporate tables and views. **Note:** It is not possible to generate DDMs for objects containing lower case characters.
Natural IMS Defaults	
Start in the NDB	The first field short name that Predict will allocate for an IMS field. For example: if this value is NA, IMS fields are allocated field short names in the range from NA to Z9.
Start in logical	The first field short name that Predict will allocate to a user-defined field for a logical child segment. For example: if this value is HA and Start in the NDB is set to NA, user-defined fields of logical child segments are allocated field short names in the range from HA to M9.
Natural DB2 Defaults	
Static SQL XREF	**Y** XRef data is generated by the Natural DB2 utility SQLGEN. **N** No XRef data is generated. **F** Force. XRef data is generated and the Natural DB2 utility SQLGEN checks whether any program processed is documented by a Predict program object of subtype Q (Static SQL).
Network support	

Parameters	
Multilingual Support	
Unique DBnr/Fnr	**Y** Any DBnr/Fnr specified must be unique throughout a network.
Current network	Network of the Natural environment. All virtual machines linked to the same network can be accessed from the current virtual machine. **Note:** Changing the Current network affects implementation plans as described with the Default current Virtual Machine above.
Default current Virtual Machine	The default current virtual machine is used as follows: • It is used as default parent for databases if a virtual machine is not explicitly specified. • It determines the target environment for Generate/ Incorporate/ Compare functions and some AOS functions called from the Predict Special functions menu. **Note:** After changing the Default Current Virtual Machine, implementation plans which include generation tasks for Adabas objects can only be purged or displayed. See the section File Implementation in the **External Objects in Predict documentation**. The value for Default current Virtual Machine can be overwritten in a session profile. See description of the profile parameter Current Virtual Machine in the section Predict User Interface of the **Introduction to Predict documentation**. **Note:** The current virtual machine can also be set with the command: SET VM<Virtual Machine ID>.

Default Adabas Attributes

The Default Adabas Attributes screen is displayed with code A in the General Defaults screen. The parameters entered here are set as default values in the corresponding file maintenance screen if new Adabas files are added.



For detailed information on the attribute settings see Modifying Adabas Attributes in the section **File** in the **Predefined Object Types in Predict** documentation.

Extended Description Skeleton

A skeleton for extended descriptions can be defined for each subtype of each object type. When an extended description is edited the first time, the extended description skeleton appears. Extended description skeletons are used to ensure consistent format and content of extended descriptions.

Call the Extended Description Skeleton screen with code E from the Defaults menu. The following parameters in the Defaults menu apply to extended description skeletons.

Parameters	
Object type	Two-character Predict object type. The extended description skeleton also applies to all subtypes of an object type for which no individual skeleton has been defined.
Subtype for skeleton	May be entered to determine the subtype of Object type for which the skeleton is to be edited. For example, to edit the extended description skeleton for Adabas databases, specify Object type=DA and Subtype for skeleton=A.
Language for skeleton	Only applies to extended descriptions for objects of type PR. May be entered to specify the language for which the skeleton is to be edited.

```
>                    > + DA: *** Edit title ***               L: 1    S: 45
 Top  ....+....1....+.... Extended description ...+....5....+....6....+....7..
                ====================================
                 Data base additional description
                ====================================
             Database-number.:  ...
            Optional-01
            Optional-02
            Optional-03
            Optional-04
```

If Con-form, Software AG's text formatting facility, is installed, Con-form statements can be used. See screenshot in the section Output Options in the section **Predict User Interface** in the **Introduction to Predict documentation**.

Protecting Parts of an Extended Description Skeleton

Text in an extended description skeleton can be protected by enclosing it in a pair of special characters. The characters are defined with the parameters Start/End character protect extended desc. in the Maintenance Options screen when modifying a session profile.

See Maintaining User-specific Profiles in the section **Predict User Interface** in the **Introduction to Predict documentation**.

If a header or footer is specified, the lines for the header or footer must also be specified with the commands .HS number and .FS number.

When using this option, the output option Use CON-FORM must be set to Y. See Output Options in the section **Predict User Interface** in the **Introduction to Predict documentation**.

Note:
Changes to Con-form variables with the Con-form command .OP are not recognized by Predict. If, for example, the page number sign # has been substituted, page numbers will not be displayed.

DEFAULT Profile

Parameters determining the way functions work that may be changed by any Predict user are specified in the session profile. A user-specific profile can be defined for any user defined in Predict. If no user-specific profile is activated, the Predict DEFAULT profile is active.

The Predict DEFAULT profile can be modified with the function Default profile. This function is called with code S in the Defaults menu or with the command DEFAULT PROFILE.

All profile parameters are explained in the section Maintaining User-specific Profiles in the section **Predict User Interface** in the **Introduction to Predict documentation**.

LIST XREF Default Profile

The LIST XREF Default Profile can be modified by selecting code L in the Defaults menu or with the command DEFAULT XREF.

See Maintaining a LIST XREF Profile in the section **LIST XREF for Natural** in the **Predict Reference documentation**.

Generation Defaults

Default generation values are set at installation and are displayed in the input screen of any generation function.

Generation functions require that generation default values have been specified.
Generation defaults can be changed with code G in the Defaults menu (or with the command DEFAULT) and the external object code. See Generation Defaults in the section **Generation** in the **External Objects in Predict documentation**.

Note:
The individual parameters are described in the sections Storage of External Objects Owned by Predict and Common Parameters in the section **Generation** in the **External Objects in Predict documentation** and with the different generation functions. Defaults for language ADA, however, are described in the section Generation Defaults for Language ADA below.

The following rules apply for the use of default values for generation parameters:

- Default values of generation parameters can be changed with the function Generation Defaults in the Modify Defaults Menu or the command DEFAULT object code. For example: DEFAULT COBOL displays the Modify COBOL Defaults screen shown below.

```
13:01:58            *****  P R E D I C T  4.3.1  *****             2003-05-31
                        -Modify COBOL Defaults -
                                                    Modified 2003-05-31 at 12:28
                                                          by CHD
Mark with 'X' the options which may be modified by the user.

X Save as member ...........           X Save in library .... COBLIB
X Overwrite option ......... Y  (Y,N)  X Op. system member ..
X Punch / output ..........* N         X List offsets ......* N
X List generated code ...... Y  (Y,N)  X Adabas version ....* I7
X Generate format buffer ..* N         X Field name prefix ..
X Check field names .......* A         X Field name suffix ..
X Start level ..............  1 (0-40) X Validate ........... -
X Level number increment....  1 (1-40) X Truncation .........* R
X Level shift increment .... 3  (0-9)  X With Cond. names ... Y (Y,N)
X Nr. of abstract lines ....  3 (0-16) X Indexed by .........* N
X Generate initial value ..* N         X Literal delimiter .* S
X Synchronized ............* Y         X Decimal character .* P
X Depending on ............. N  (Y,N)  X Redefinition name .* S
X Record buffer name .......
X Format buffer name .......
                                         Compiler ..........* 8
  Preprocessor force ....... N  (Y,N)    Library system ..... 2
```

- Most default values are displayed in the input screen of the respective generation function and can then be overwritten for temporary use. Changes to default values apply to subsequent generation tasks until another Predict function is executed.
- Generation defaults can be protected by blanking out the X preceding the parameter in Modify ... Defaults screens. Protected default values cannot be changed when executing a generation function. These fields are skipped when positioning the cursor with the TAB key.
- Some defaults values are not displayed in the input screen of a generation function and can therefore only be changed using the Modify Generation Defaults functions. These parameters are described under **Presettings** in the descriptions of individual generation functions in the respective parts of the section Generation in the **External Objects in Predict documentation**.

Generation Defaults for Language ADA

Generation defaults for language ADA are used by Adabas Native SQL.

```
13:28:16             *****  P R E D I C T  4.3.1  *****             2003-05-31
                          - Modify ADA Defaults -
                                                      Added 2003-05-31 at 10:23
                                                   Modified 2003-05-31 at 13:49



Mark with 'X' the options which may be modified by the user.


X Adabas version ......* I7              X Field name prefix ....
X Truncation ..........* L               X Field name suffix ....
X Validate ............. *
X Record buffer name ...

  Preprocessor force ..: N  (Y,N)
```

Parameters

<table>
<tr><td>Adabas version</td><td>The version of Adabas for which the copy code is to be generated.

<table>
<tr><th>Code</th><th>Version</th><th>Description</th></tr>
<tr><td>I1</td><td>V 5.1 for IBM/Siemens</td><td>Applicable to all languages.
When generating copy/include code, sub/superdescriptors are not included in the record buffer layout.</td></tr>
<tr><td>I3</td><td>V 5.3 for IBM/Siemens</td><td>As above.</td></tr>
<tr><td>I7</td><td>V 7.1 for IBM/Siemens</td><td>As above.</td></tr>
<tr><td>O4</td><td>V 4.1 for IBM/Siemens</td><td></td></tr>
<tr><td>V2</td><td>V 2.1 for VMS</td><td></td></tr>
<tr><td>V3</td><td>V 3.1 for VMS</td><td></td></tr>
<tr><td>V4</td><td>V 3.2 for VMS</td><td></td></tr>
<tr><td>V5</td><td>V 4.1 for VMS</td><td></td></tr>
<tr><td>R1</td><td>V 5.1 for IBM/Siemens</td><td>Only applicable to generation of Copy/Include Code.
Sub/superdescriptors are included physically in the record buffer layout.

Note:
Code generated with this Adabas version cannot be used for update statements.</td></tr>
<tr><td>R3</td><td>V 5.3 for IBM/Siemens</td><td>As above.</td></tr>
<tr><td>R7</td><td>V 7.1 for IBM/Siemens</td><td>Similar to I7, sub/super and collation descriptors are included physically in the record buffer layout.</td></tr>
</table>
</td></tr>
<tr><td>Field name prefix</td><td>The prefix to be used for each field name generated.</td></tr>
<tr><td>Truncation</td><td>Specifies which characters are deleted if a generated field name is too long:

L
truncate from the left

R
truncate from the right

M
truncate from the middle</td></tr>
<tr><td>Field name suffix</td><td>The suffix to be used for each field name generated.</td></tr>
</table>

Validate	Determines how invalid characters are handled. **blank** Invalid characters in a field name will result in an error message but will not be modified. **rep.char** Invalid characters in a field name are replaced by this character. Valid replace characters: letters A-Z, digits 0-9 and underscore (_). ***** Invalid characters in a field name are deleted.
Record buffer name	Specifies the name of the record buffer in the generated structure. If omitted, the file ID is used.
Preprocessor force	**Y** Adabas Native SQL checks that the program to include the generated code is documented. If no Predict object documenting the program is found, the generation task is not executed and a message is given. **N** No check is performed. Default setting when Predict is installed.

Coordinator Defaults

The Coordinator Defaults screen is used to determine the following:

- the database and file number of the Coordinator FDIC
- the virtual machine to which databases of type A, P and V are linked (parameter Default Virtual Machine).
- new names/codes of UDEs defined in an earlier version of Predict, if these names/codes are now reserved. See list of reserved codes and names in Reserved Metadata for Coordinator Defaults.

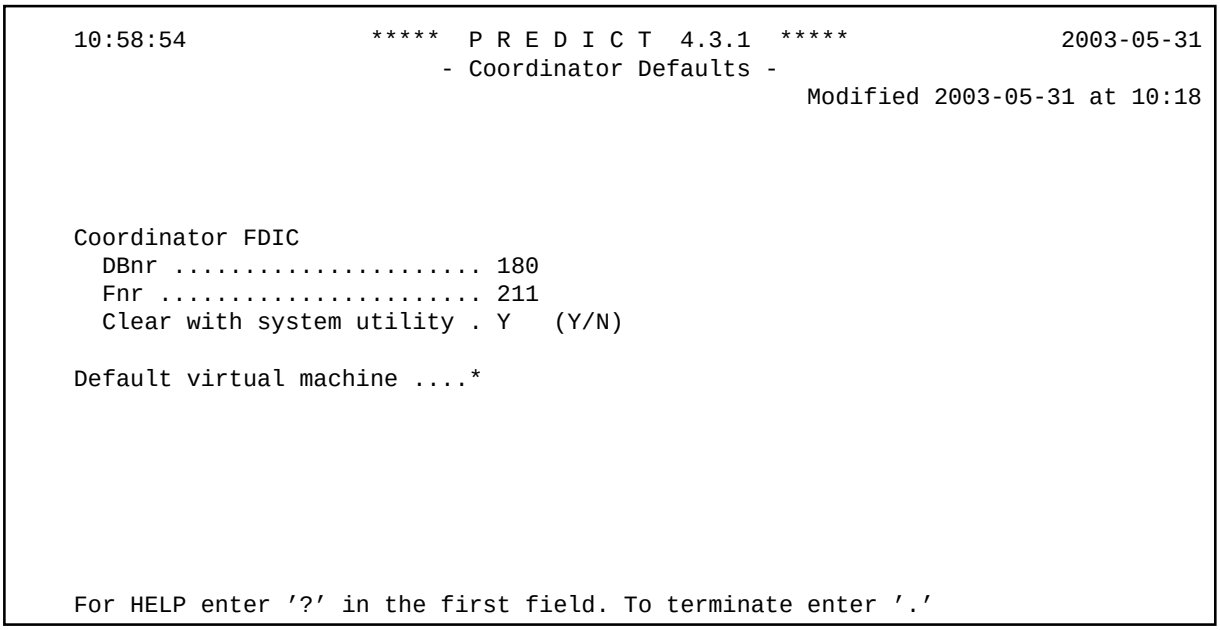
This section covers the following topics:

- First Screen
- Second Screen

First Screen

The first Coordinator Defaults screen is called with Code C in the Defaults Menu or with the command DEFAULT COORDINATOR.

In case you want to make changes to the first screen only, you can alternatively use the command ASSIGN COORDINATOR <DBNR> <FNR> <UTILITY-CLEAR> <DEFAULT-VM>.



Parameters	
Coordinator FDIC	The Coordinator FDIC file is used as a transfer medium by the Coordinator function Import. See the Predict Coordinator documentation for more information.
DBnr	Number of the database containing the Coordinator FDIC.
Fnr	Number of the Coordinator FDIC file.
Clear with system utility	**Y** System utilities (for example AOS in a mainframe environment) are used to delete the contents of the Coordinator FDIC after a CLEAR command, a successful import operation or after Special Function Refresh Coordinator FDIC. Setting this parameter to Y improves performance if your Coordinator FDIC contains large amounts of data.
Default Virtual Machine	Virtual Machine to which Databases of type A, P or V are to be linked. ● Enter **HOME** if all Databases are to be linked to the default Virtual Machine HOME. ● Enter ***OPSYS** if Databases are to be linked to different Virtual Machines depending on the attribute Operating system type. ● Enter an **asterisk** to display a list of Virtual Machines for selection.

Second Screen

Press ENTER in the first Coordinator Defaults screen to call the following screen.

This screen contains the codes that are reserved in Predict Version 4.2. If you defined UDEs with codes from this list in earlier versions, you must enter a new code with this function.

Warning:
If your data contains UDEs with reserved codes and you do not enter a new code in this table, the Load function of the Predict Coordinator will terminate.

```
13:13:57              *****  P R E D I C T  4.3.1  *****              2003-05-31
                           - Coordinator Defaults -
                                                        Added 2003-05-31 at 13:06
                                                           by HNO
Object type code   since Version          Object type code   since Version
  O4 ....               4.3                RC ....               3.3
  TR ....               4.2                RT ....               3.3
  IE ....               4.1                SC ....               3.3
  MD ....               4.1                SN ....               3.3
  PY ....               4.1                SV ....               3.3
  AT ....               3.3                XF ....               3.3
  BF ....               3.3                YF ....               3.3
  CR ....               3.3
  DC ....               3.3
  ET ....               3.3
  JF ....               3.3
  LS ....               3.3
  NO ....               3.3
  OF ....               3.3
  PG ....               3.3

For HELP enter '?' in the first field. To terminate enter '.'
```


Parameters	
Object type code, Object type name	The table contains all codes and names that are used in Predict for predefined object types that are new in Version 4.3. If you created a user-defined object type with a code or name that is now reserved, enter a new code or name here.

Reserved Metadata for Coordinator Defaults

Press ENTER in the second Coordinator Defaults screen to call the following screen.

This screen contains the names that are reserved in Predict Version 4.3. If you defined UDEs with names from this list in earlier versions, you must enter a new name with this function.

```
13:13:21            *****  P R E D I C T  4.3.1  *****            2003-05-31
                         - Coordinator Defaults -
                                                  Added 2003-05-31 at 13:06
                                                     by HNO
Object type name                      since Version
  OS4-FILE ........                        4.3
  TRIGGER .........                        4.2
  VISTA-DA ........                        4.2
  VISTA-FI ........                        4.2
  INTERFACE .......                        4.1
  METHOD ..........                        4.1
  PROPERTY ........                        4.1
  ESD-TABLE .......                        3.3
  EXTRACT .........                        3.3
  FILERELATION ....                        3.3
  INFORMIX-TABLE ..                        3.3
  INGRES-TABLE ....                        3.3
  LIBRARYSTRUCTURE                         3.3
  NODE ............                        3.3
  OBJECTS .........                        3.3

For HELP enter '?' in the first field. To terminate enter '.'
```

Press ENTER in the third Coordinator Defaults screen to call the following screen.

This screen contains the retrieval models that are reserved in Predict Version 4.3. If you defined a retrieveal model from this list in earlier versions, you must enter a new name with this function.


```
13:13:21            *****  P R E D I C T  4.3.1  *****            2003-05-31
                         - Coordinator Defaults -
                                                  Added 2003-05-31 at 13:06
                                                     by HNO
Retrieval model                       since Version
 AP (SY) ..                               4.2









For HELP enter '?' in the first field. To terminate enter '.'
```

Press ENTER in the fourth Coordinator Defaults screen to call the following screen.

This screen contains the associationss that are reserved in Predict Version 4.3. If you defined an association from this list in earlier versions, you must enter a new name with this function.

```
13:13:21            *****  P R E D I C T  4.3.1  *****            2003-05-31
                         - Coordinator Defaults -
                                                  Added 2003-05-31 at 13:06
                                                     by HNO
             Active     Passive        since
Association  code       code           Version
  (FI->TR)     TR ..      FI ..          4.2
  (PR->FI)     IN ..      IP ..          4.2
  (PR->FI)     RE ..      RS ..          4.2
  (SY->SY)     CS ..      CS ..          4.2
  (SY->PR)     CP ..      CP ..          4.2
  (SY->VE)     CV ..      CV ..          4.2
  (SY->FI)     CF ..      CF ..          4.2
  (SY->SY)     LI ..      LI ..          4.2
  (PR->PR)     MS ..      MS ..          4.2









For HELP enter '?' in the first field. To terminate enter '.'
```


Defaults for Adabas Native SQL

The Modify Adabas Native SQL Defaults screen is called with code A in the Defaults menu or with the command DEFAULT SQL.

```
13:37:23             *****  P R E D I C T  4.3.1  *****             2003-05-31
                    -Modify Adabas Native SQL Defaults -
                                                        Added 2003-05-31 at 10:23
                                                     Modified 2003-05-31 at 13:19
Adabas Native SQL defaults
  Non-descriptor search allowed .... Y (Y,N)
  Use of soft-coupling allowed ..... Y (Y,N)

Adabas Version 5 open options
  ISN held incore ..................
  Max. number of held ISNs .........
  Nonactivity time limit (TNAE) ....
  Transaction time limit (TT) ......
  Maximum time for an Sx command ...
  Maximum active command ids .......

Used global format ID counter ......

Database number of FDIC ............ 180
File number of FDIC ................ 220
```

Parameters	
Adabas Native SQL defaults	
Non-descriptor search allowed	Determines whether the Adabas non-descriptor search used by Adabas Native SQL is allowed. **Y** The ADASQL NONDE clause can be specified as Y or D. **N** The NONDE clause can only be set to N. The parameter NONDE controls the use of a non-descriptor field in the WHERE clause of the SELECT statement. NONDE=Y implies that at least one of the fields contained in the WHERE-clause is a descriptor.
Use of soft-coupling allowed	Determines whether the Adabas soft-coupling facility used by Adabas Native SQL is allowed. **Y** The ADASQL SOFT clause can be set to Y. **N** The ADASQL SOFT clause can only be set to N. The parameter SOFT controls the use of the soft-coupling facility between two different files in the WHERE clause of the SELECT statement.
Adabas Version 5 open options	These values are the upper limits for the corresponding parameters of the Adabas CONNECT statement and are described in the section **Adabas Native SQL Statements (Statement CONNECT)** in the **Adabas Native SQL Reference documentation**.
Used global format ID counter	Used to form the global format ID. See description of options parameter GFORMAT in the section **Global parameters** in the **Adabas Native SQL Reference documentation**.
Database/File number of FDIC	See description of the parameter SYSFILE in the section **Global Parameters** in the **Adabas Native SQL Reference documentation**.

Activate User Exits

Activate or deactivate user exits connected to different functions of Predict.
The Activate User Exits screens are displayed with function code U in the Defaults menu, or with the command DEFAULT EXIT.

```
10:07:07            *****  P R E D I C T  4.3.1  *****              2003-05-31
                          - Activate User Exits -

--- User exits (subprograms) ---                    Modified 2003-05-31 at 09:21
                                                          by RBN

Maintenance                          Active retrieval ................. Y (Y/N)
  before any function .. N (Y/N)     File implementation .............. N (Y/N)
  add, copy, modify .... N (Y/N)     Comparison ....................... N (Y/N)
  purge, scratch ....... N (Y/N)     Incorporation .................... N (Y/N)
  modify description ... N (Y/N)     Administration ................... N (Y/N)
  redocumentation ...... N (Y/N)     Command processor ................ N (Y/N)
Retrieval                            Defaults/Special functions ....... N (Y/N)
  before any function .. N (Y/N)     Check against naming conventions . N (Y/N)
  after display .......* N
Generation
  before execution ..... N (Y/N)
  after execution ...... N (Y/N)
```

```
   ----------- User exits (programs) -----------    Modified 2003-05-31 at 09:21
                                                          by RBN
                     ACM     PURGE     CAT
   Database .......... N (Y/N)  N (Y/N)
   Elementary field .. N (Y/N)  N (Y/N)  N (Y/N)
   File .............. N (Y/N)  N (Y/N)
   Keyword ........... N (Y/N)  N (Y/N)
   Program ........... N (Y/N)  N (Y/N)
   File relation ..... N (Y/N)  N (Y/N)
   System ............ N (Y/N)  N (Y/N)
   User .............. N (Y/N)  N (Y/N)
   Verification ...... N (Y/N)  N (Y/N)
   Owner ............*                  N


For HELP enter '?' in the first input field. To leave enter '.'
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

The tables below give an overview of how to activate user exits. A comprehensive description of all user exits including detailed information on how to activate them is contained in the section User Exits of this documentation.

Note:
The user exits U-MODEL and U-XREF are activated in the Metadata Administration. U-PGMLA, used to define new programming languages to Predict, is always active.

Parameters (First Screen)	Activates User Exit(s)
Maintenance	
before any function	U-MNT and U-MNT1
add, copy, modify	U-ACM and U-ACMR
purge, scratch	U-PUR
redocumentation	U-REDOC
modify description	U-DESC
Retrieval	
after display	U-DSP (option N, O and F)
before any function	U-RET
Generation	
before execution	U-GEN
after execution	U-GEN1
Other	
Active retrieval	U-ACT
File implementation	U-IMPL
Comparison	U-COM
Incorporation	U-INC
Administration	U-MIP
Command Processor	U-CMD
Defaults/Special functions	U-SPEC
Check against naming conv.	U-OBJID

Parameters (Second Screen)	Activates User Exit(s)
Database	ACMDAEX, PURDAEX, CATDAEX
Field	ACMELEX, PURELEX, CATELEX
File	ACMFIEX, PURFIEX, CATFIEX
File Relation	ACMRLEX, PURRLEX, CATRLEX
Keyword	ACMKYEX, PURKYEX, CATKYEX
Program	ACMPREX, PURPREX, CATPREX
System	ACMSYEX, PURSYEX, CATSYEX
User	ACMUSEX, PURUSEX, CATUSEX
Verification	ACMVEEX, PURVEEX, CATVEEX
Owner	U-OW and CATOWEX (option N, Y, B and U)

Overview of Activating User Exits

User Exit	Activate with Parameter
U-ACM and U-ACMR	Parameter add, copy, modify
U-ACT	Parameter Active retrieval
U-AM-A	Parameter add, copy, modify
U-CMD	Parameter Command Processor
U-COM	Parameter Comparison
U-DESC	Parameter Maintenance / modify description
U-DSP	Parameter Retrieval / after display
U-GEN	Parameter Generation / before execution
U-GEN1	Parameter Generation / after execution
U-IMPL	Parameter File implementation
U-INC	Parameter Incorporation
U-OBJID	Parameter add, copy, modify
U-MIP	Parameter Administration
U-MODEL	Parameter Userexits / U-MODEL in the Metadata Administration. See User exits.
U-MNT and U-MNT1	Parameter Maintenance / before any function.
U-OW	Parameter Owner
U-PGMLAN	always active
U-PUR	Parameter Maintenance / purge, scratch
U-REDOC	Parameter Maintenance / redocumentation
U-RET	Parameter Retrieval / before any function
U-XREF	Parameter Userexits / U-XREF in the Metadata Administration. See User exits.

Special Functions

Functions in the Special Functions menu offer means to administrate Predict. These functions can be protected with Predict Security. See the section Natural Security Entities in the **Predict Security documentation** for more information.

To call the Special Functions menu, select function code S in the Predict Main Menu or enter the command SPECIAL.

This section covers the following topics:

- Adabas Device Types
- Delete Old Sets
- Maintain Predict Help Texts
- Reposition Implementation Data
- Mass Grant in NSC
- Maintain NSC Definitions
- Recover
- Security for Adabas Online Services
- Consistency of Predict
- Maintain Active References
- Maintain Standard Fields
- Refresh Coordinator FDIC
- Mass Delete of Report Listings

Adabas Device Types

Device types for Adabas (including user-defined device types) to be referenced in Predict documentation objects must be defined with this function.

During installation of Predict, a device type is specified depending on the operating system:

Device type	Operating system
3380	VSE, OS/390, FACOM
2000	BS2000/OSD
2564	WANG/VS
RA81	VMS
2KB, 4KB	UNIX

Predict stores size information of Adabas files and databases in units of blocks. Computation of sizes into units of cylinders is based on the values specified with this function for a device type.

The Maintenance Adabas device types screen is invoked with code A in the Special Functions menu or with the command SPECIAL DEVICE.

```
10:07:31             *****  P R E D I C T  4.3.1  *****             2003-05-31
Plan  10             - Adabas Device Type Maintenance -



                    Function

                    L  List defined device types
                    A  Add device type
                    M  Modify device type
                    P  Purge device type

Function ..........

Device type .......

Database number ... 180    (1-65535)



Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

Parameters	
Device type	Up to four characters identifying the device. See table above for valid values.
Database number	Database by which the device table is read if Adabas Online Services are installed.

Functions

List defined device types - Code L

Lists all device types currently defined in Predict

Add device type - Code A

A new device type definition can be added.
If the device type is a standard SAG device type, all standard values are shown. We recommend not changing these values. If Adabas Online Services are installed, Predict reads the device table of the specified Adabas nucleus.

Modify device type - Code M

An existing device type definition can be modified.
If Adabas Online Services is installed, Predict reads the device table of the specified Adabas nucleus and shows the differences if the device is defined differently in Adabas.

Purge device type - Code P

Purges device type definitions.
A device type can be purged only if it is not specified as the default device type for Predict with the function General defaults.
A device type that has been purged is replaced with the default device type in all databases and files.

Delete Old Sets

Delete sets created with the Natural LIST XREF command or Predict Active Retrieval / LIST XREF for 3GL functions.

The Delete Old Sets screen is invoked with function code D in the Special Functions menu or with the command SPECIAL SET.

```
   13:04:06            *****  P R E D I C T  4.3.1  *****            2003-05-31
                                - Delete Old Sets -



            All sets created before the date specified will be deleted.

            To date: 2003-05-31 (YYYY-MM-DD)
            There are       337 record(s) to inspect.



            Enter . to return to menu
               or ? for help









   Command ===>
   Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
         Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

Parameters	
To date	All sets (for all users in all libraries) which are older than the date specified here are deleted. The value to enter must not be later than the current date minus seven days.

Deleting Old Sets in Batch Mode

This function can also executed in batch mode with the command SPECIAL SET. The only keyword is TO-DATE in the line following the command.

Note:
As a protection measure, no TO-DATE is accepted that is later than the current date minus seven days.

Example

To delete sets created before 31 July 2002, code the command:

```
SPECIAL SET
TO-DATE=2003-05-31
```


Maintain Predict Help Texts

Predict online help texts can be changed with this function. Help texts are edited with a Natural-based editor. See the section Editors in Predict in the **Predict Reference documentation**.

Each Predict function has a corresponding help text. This help text explains what a function does and how it is used. An online help text is displayed by entering a question mark (?) in the first input field of a function.

The Maintain help texts screen is displayed with code H in the Special Functions menu or with the command SPECIAL HELP.

```
13:09:29              *****  P R E D I C T  4.3.1  *****            2003-05-31
Plan  10                  - Maintenance help texts -


                    Function

                    D  Display help text
                    M  Modify help text
                    S  Select help text from a list

Code ..............


Natural program-name (help text) .. H-
Natural library ................... SYSDIC
Natural program-type .............. T (Text)
Natural system file number ........ 54
Natural system database number .... 180


Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

Parameters	
program name (help text)	Name of the Natural text member containing the help text. Starts with H- for standard Predict help texts and with T- for user-defined help texts.
library	Library of Predict in the Natural system file (FNAT). Read-only field.
program type	Always T for text member. Read-only field.
system file number	File number of the Predict system file (FNAT). Read-only.
system database number	Number of the database containing the Predict system file. Read-only.

Functions

Display help text - Code D

Display a Predict help text as it appears online. Highlighted text which is marked intensified is displayed intensified.

Modify help text - Code M

Invokes the Predict help text editor, a Natural-based editor, which comprises standard editor functions and some features specifically designed for editing help texts. See the section Editors in Predict in the **Predict Reference documentation**. Help text specific functions are described below:

FORMAT

Blanks are inserted between words to make text right and left-justified

UNFORMAT

Additional blanks between words are removed to make text left-justified only.

Select help text from a list - Code S

Displays a list of all help texts available for selection. Mark the required help text with any non-blank character and press enter. The name of this help text appears at Natural program name (help text) in the Maintenance help texts Menu.

Creating a Help Text

To add a new help text, enter the command SAVE with the new name as parameter; or add the help text via Natural directly (object type must be TEXT). Use Natural commands for purging a help text.

Intensifying Text

If part of a help text line (for example, a word) is to be displayed intensified, enclose that part in a hash sign (#) and a ’paragraph’ sign (§) as used on German keyboards.

Using Topic Lists

To use topics as search criteria, each line which is to be used as a topic must contain ’ >>’ in its first four columns and ’<<’ in its last two columns. The corresponding line in which the information pertaining to this topic starts must be enclosed in both a hash sign (#) and a ’paragraph’ sign (§) as used on German keyboards.

Protecting User Help Texts

During an INPL for any new version or SM of Predict, all standard Predict help texts are overwritten with new texts from the tape. Changes made to standard Predict help texts since the last INPL are then lost.

The following method is used to protect user-defined help texts:

- User-defined help texts and standard Predict help texts are stored in help members with identical names but different prefixes:
  Member names for user-defined help texts have the prefix T-, whereas the names of standard Predict help texts have prefix H-.
  For example: H-MNTFI is name of the standard Predict help text to be displayed for the File Maintenance menu. A user-defined help text for this menu must be stored in a member T-MNTFI.
  The names of standard help texts for user-defined entities always start with H-#, for example H-#MNT for a maintenance help text.
- Whenever help information is requested, Predict looks for an appropriate T-name text. If no T-name text is available, the H-name text is displayed.

To prevent changes to help texts being overwritten, modify a standard Predict help text as required and then save the help text with the prefix T-.

Reposition Implementation Data

If the database/file number of a user system file is changed by an Adabas utility, XRef data points to a wrong file, and the implementation pointer of the documentation may point to a wrong database/file number. Database/file numbers and implementation pointers can be corrected with the function Reposition implementation data.

The Reposition Implementation Data function is invoked with code I in the Special Functions menu or with the command SPECIAL IMPLEMENTATION.

If a duplicate implementation pointer is found and the parameter Add to workplan is set to Y, a MODIFY command is added to the workplan for the respective documentation object.

```
13:13:47            *****  P R E D I C T  4.3.1  *****             2003-05-31
Plan  10            - Reposition implementation data -



Old environment                              New environment
  Database number.........                     Database number....... 180
  File number.............                     File number........... 54
  Library.................. *                   Library............... *

What to convert
  Active References ....... Y (Y/N)
  Documentation ........... Y (Y/N)

Processing options
  Fill in documentation...* D  Mem, Lib, FNR, DBNR
  List actions ...........* A  All actions
  Add to workplan ......... Y (Y/N)

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

Note:
It is possible that large number of records are read and updated with this function. We therefore recommend running it in batch mode. See Reposition Implementation Data in Batch Mode.

Parameters	
Old environment	
Database number	Old database number (source).
File number	Old file numbers (source).
Library	Old library name (source). It is possible to specify the current library with an asterisk, but we strongly recommend specifying the library explicitly to prevent accidental deletion.
New environment	
Database number	Database number of the current environment which is always taken as the new environment. Cannot be overwritten.
File number	File number of the current environment which is always taken as the new environment. Cannot be overwritten.

Library	New library name (target). If an asterisk is specified in this field, the library name of the old environment is kept.
What to convert	
Active References	**Y** Reposition XRef data. The system checks if an object module exists for the new file. If this is the case and the processing option Del. wrong XRef data is set to Y, the XRef data is moved from Old environment to New environment and the old XRef data is deleted. This means that the following Adabas functions must be executed first: • unload FUSER • load FUSER to another file or renumber the Adabas file. **N** XRef data is not repositioned.
Documentation	**Y** Documentation is repositioned.
Processing options	
Fill in documentation	Determines the scope of the implementation pointer. Valid values: **D** Member, library, file number, database number **F** Member, library, file number **L** Member, library **M** Member **Note:** It is not always necessary to specify all four parameters to create a unique implementation pointer.
List actions	Determines the scope of information displayed. **A** All objects found under Old environment are listed. **N** No actions are listed, information appears briefly on screen.
Add to workplan	Only applicable if duplicate implementation pointer is found, in which case no change is performed and an error message is given. **Y** The command MOD <object type><name> is added to the workplan. The values for <object type are either PR or SY. **N** No command is added to the workplan.

Reposition Implementation Data in Batch Mode

This function can also executed in batch mode with the command SPECIAL IMPLEMENTATION. If duplicate implementation pointers are found, no changes are performed, an error message is given and the command MOD <object type><name> is added to the workplan.

Note:
When working online, MOD <object type><name> commands are put in your own workplan. In batch mode, the workplan must be specified explicitly, whereby different considerations apply if Natural Security is installed. See table below.

Adding Commands to Workplan in Batch Mode	
With Natural Security	**Without Natural Security**
Logon user ID Profile user ID	JOB name Profile User ID of JOB
Adding commands to workplan where profile ID = logon user ID	
Commands are added to the workplan of the logon user ID. If the user is not defined in Predict, an error message is given.	Give the job the same name as the ID of the user whose workplan is to be appended. This allows you to use the profile of another user If naming conventions at your installation prevent you using a user ID as job name, you must code the following statement: PROFILE <user> This results in the user ID being stored in the parameter profile user ID. The system checks first whether the job name can also be used as user ID. If not, the workplan of the user defined in profile user ID is appended.
Using the profile of another user	
To use the profile of another user, code the following statement: PROFILE <other-user>	Give the job the same name as the ID of the user whose workplan is to be appended. To use the profile of another user code the following statement: PROFILE<other-user> If naming conventions at your installation do not permit using a user ID as job name, it is not possible to use the profile of another user.

Parameters for the SPECIAL IMPLEMENTATION command can be entered in positional or keyword form. The table below gives a list of keywords, the corresponding field in the Reposition implementation data screen and the relative position of the keywords.

Keyword	Field	Position
OLD-DBNR	Database number	01
OLD-FNR	File number	02
OLD-LIB	Library	03
NEW-LIB	Library	04
ACTIVE	Active References	05
DOC	Documentation	06
DEL-XREF (not used)	Delete wrong XRef data	07
FILL	Fill in documentation	08
LIST	List actions	09
ADD-TO-WP	Add to workplan	10

Example

In the following example, references are changed from database number 10, file number 5 to the current FUSER file number. All updates performed are listed on screen, all incorrect XRef data is deleted.

```
SPECIAL IMPLEMENTATION
OLD-DBNR=10, OLD-FNR=5, ACTIVE=Y, DOC=Y, FILL=D, LIST=A,%
ADD-TO-WP=Y
```

or in positional form:

```
SPECIAL IMPLEMENTATION
10,5,*,*,Y,Y,D,A,Y
```


Mass Grant in NSC

With the special function Mass Grant in NSC you can create security definitions in Natural Security for all Predict objects contained in an Extract.

See the Predict Security documentation for more information.

This section covers the following topics:

- Prerequisites
- Calling the Function
- Example

Prerequisites

The following prerequisites must be met before you can use this function:

- Natural Security must be installed.
- Default values for the Natural Security file must be specified under Defaults > General Defaults > Protection > DBnr/Fnr of NSC file. For this function, the current Natural Security file must be the same as the Natural Security file used when starting the Natural session.
- Default definitions must already have been added in Natural Security with the special function Maintain NSC Definitions > Add NSC Default Definitions. See Add NSC Default Definitions - Code A.
- An Extract containing all objects to which the user or user group is to have access must exist. See the section Extract in the **Predefined Object Types in Predict documentation**.
- If you wish to grant access to a user or group, this user or group must already be defined in Natural Security.

When to use this Function

This function only makes sense if you create an Extract using a Restriction such as Owner or Keyword.

If you wish to protect a range of Predict objects (for example, all Files that start with ABC), it is better to create a corresponding definition in Natural Security using asterisk notation.

Security Evaluation using Owners and Keywords

In earlier Predict versions, information such as Keywords or Owners was evaluated to restrict access to objects. Many customers used this method to adapt their environment to the particular security requirements of their company.

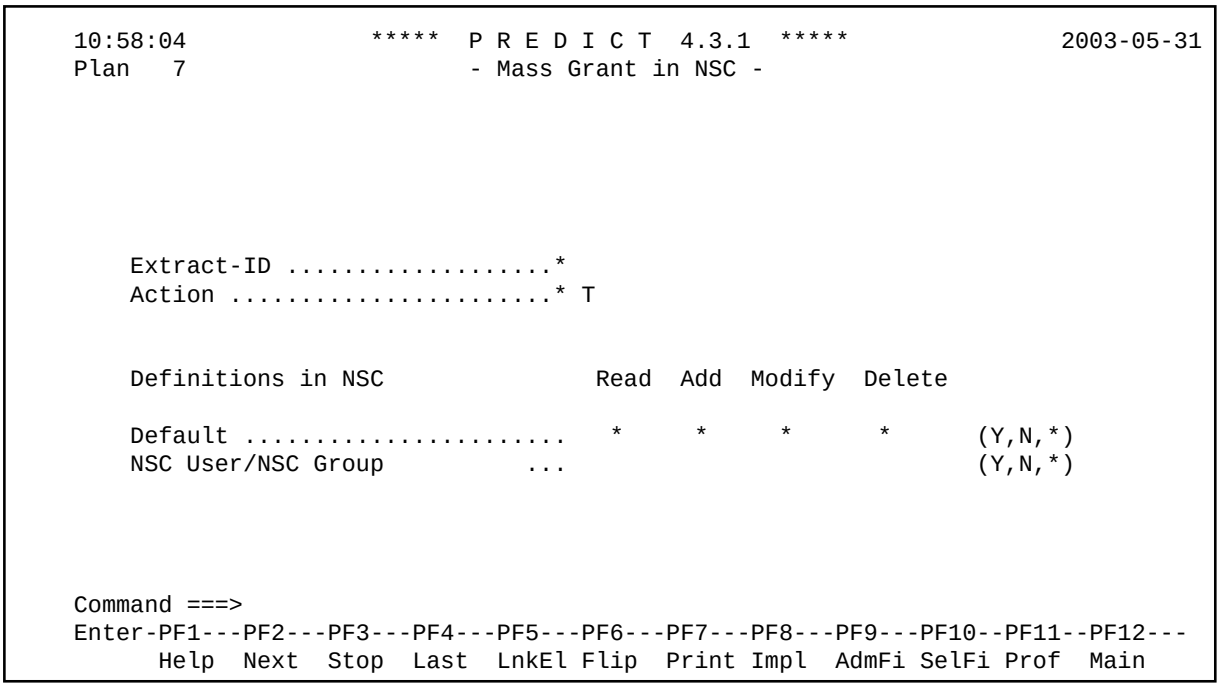
Predict Security gives you a whole range of new possibilities for defining your security environment which makes most of these customer solutions obsolete. We therefore recommend the following:

- Everything a user may do or may not do should be defined centrally in Natural Security.
- Create an Extract and enter as Restrictions the keyword or owner information, for example, that you evaluated in earlier versions of Predict.
- Execute the function Mass Grant in NSC for this extract.

For more information see the Predict Security documentation.

Calling the Function

The Mass Grant screen is called with code M in the Special Functions menu or with the command SPECIAL MASS.



Parameters	
Extract ID	ID of the extract containing the objects for which a security profile is to be maintained.
Action	**S** Save. NSC definitions are added as specified for the objects contained in the extract. **T** Test. NSC definitions are listed for the objects contained in the specified extract, but no data is written to Natural Security.
Default	Each security object in Natural Security has a default definition. Permission is added as defined in this screen if no definition already exists. Existing security definitions are not overwritten. Possible values: **Y** Access is granted for the respective access mode. **N** Access is denied for the respective access mode. ***** Inherit. The security definition of the higher level object is taken if appropriate.
NSC User/NSC Group	Enter the ID of the user or group for which you wish to create a security definition. The user/group must already exist in Natural Security. A link is created between the user/group and the objects contained in the extract. If a link already exists, it will be overwritten.

Example

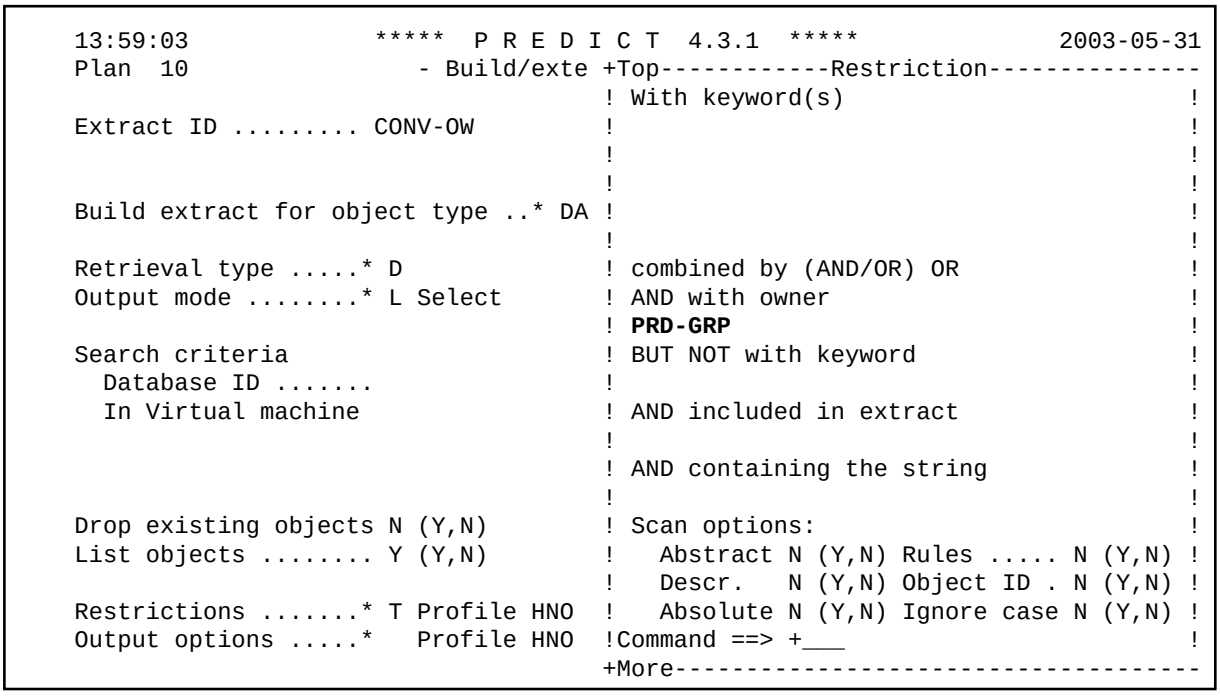
In an earlier version of Predict, only users of the group PRD-GRP were allowed to access Files and Databases in Predict. Access to these objects was controlled by evaluating Owner information with a User Exit. Now Administrator ADMIN-1 wants to add this information as security definitions in Natural Security.

Step 1 Create an Extract

There are various methods of creating an extract containing all files and databases with the owner PRD-GRP. The methods available depend on the editor you are using and are described in detail in the section Extract in the **Predefined Object Types in Predict documentation**.

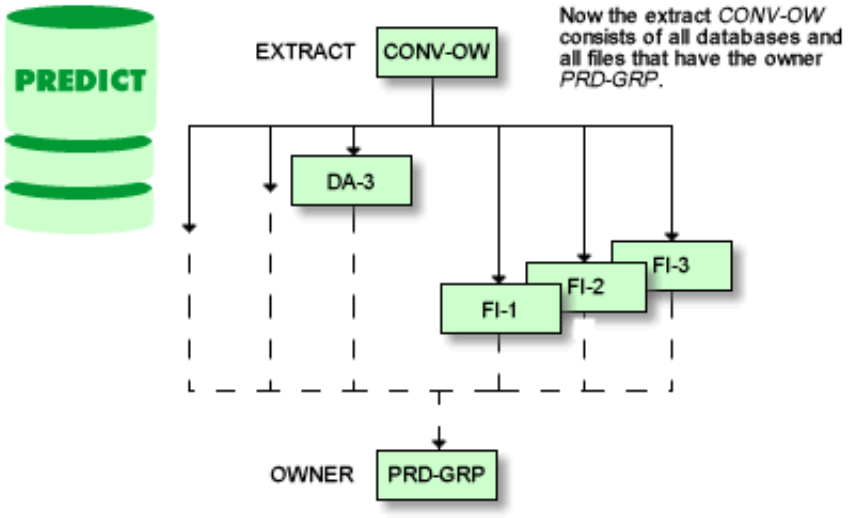
You must execute two Retrieval functions: one for files, the other for databases. The screen below shows the function Build / extend an extract for object type DA.

Enter code D for Retrieval type and code L for Output mode without specifying any other search criteria. Enter T in the field Restrictions for a temporary profile, and enter PRD-GRP in the field AND with owner. See below.



This function will add all databases with the owner PRD-GRP to the extract CONV-OW.

Perform another retrieval operation for object type file with the same temporary profile.

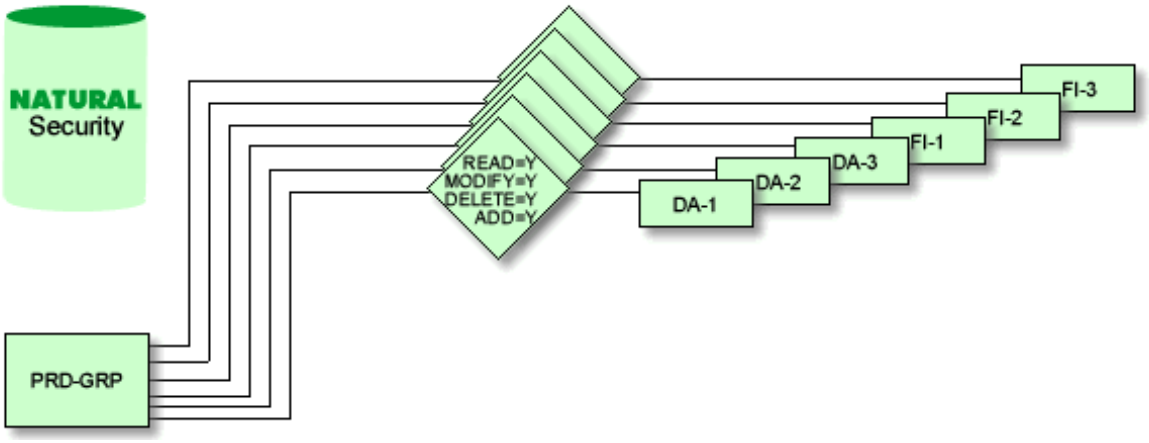


Step 2 Execute the Special Function Mass Grant in NSC

Call the special function Mass Grant in NSC with Code M in the Special Function Menu. Enter the values as shown in the screen below:

```
13:43:44              *****  P R E D I C T  4.3.1  *****              2003-05-31
Plan   7                      - Mass Grant in NSC -



    Extract ID ...................* CONV-OW
    Action .......................* S


    Definitions in NSC               Read  Add  Modify  Delete

    Default ......................    n     n      n       n   (Y,N,*)
    NSC User/NSC Group prd-grp  ...   y     y      y       y   (Y,N,*)
```

A link is created in Natural Security for group PRD-GRP to all objects contained in the extract.
If a link already exists, it will be overwritten.



Maintain NSC Definitions

General Information

The special function Maintain NSC Definitions provides the following functions for maintaining security definitions in Natural Security.

- Add NSC Default Definitions
- Check NSC Definitions
- Display NSC Definitions
- Purge NSC Definitions

The function Add NSC Default Definitions must be executed if you wish to protect Predict objects and functions with Predict Security. See the Predict Security documentation for more information.

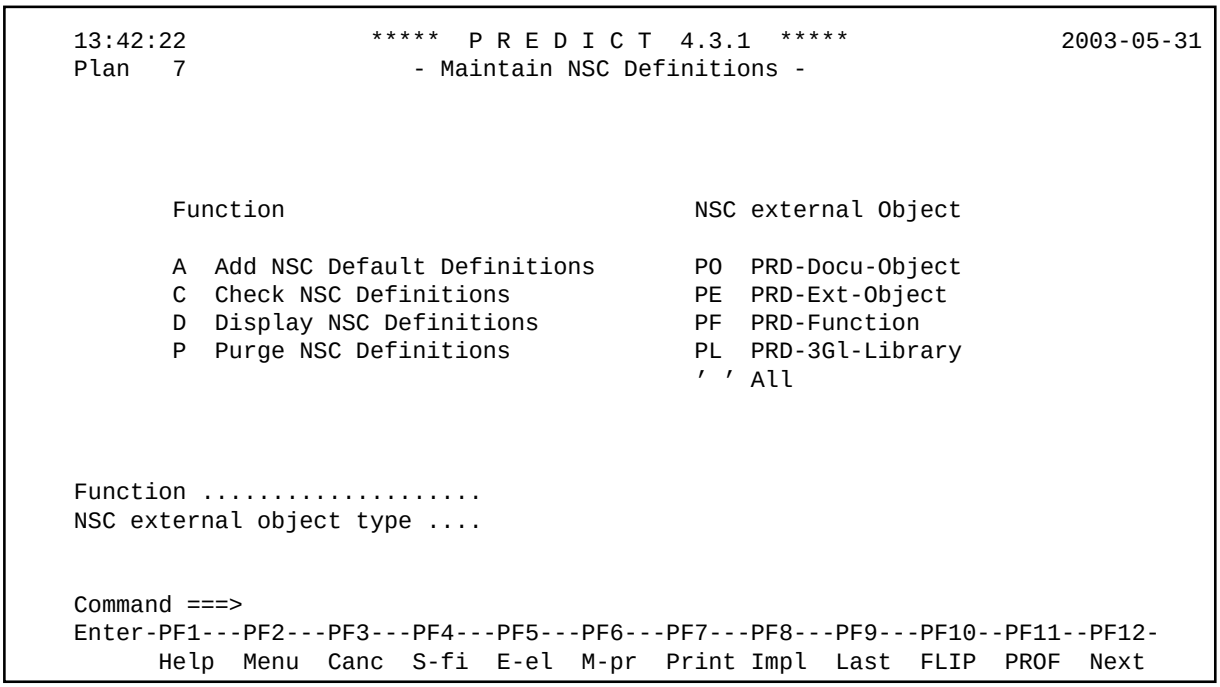
Prerequisites

The following prerequisites must be met before you can use this function:

- Natural Security must be installed.
- Default values for the Natural Security file must be specified under Defaults > General Defaults > Protection > DBnr/Fnr of NSC file. For this function, the current Natural Security file must be the same as the Natural Security file used when starting the Natural session.

Calling the Function

The Maintain NSC Definitions screen is called with code N in the Special Functions menu or with the command SPECIAL NSC-DEFINITIONS.



Parameters	
Function	Enter one of the function codes displayed in the menu.
NSC external object	Enter one of the two-letter codes displayed in the menu to limit the function to a specific NSC external object type, or leave this field blank to execute the function for all NSC external object types.

Functions

Add NSC Default Definitions - Code A

This function is used to add the NSC external object types and default values for these. If an NSC external object type is specified, a window appears in which you can define the access modes for this type. If no NSC external object is specified, a series of windows appears in which you can define the access modes for all four types.

```
13:34:12            *****  P R E D I C T  4.3.1  *****          2003-05-31
Plan   7               - Maintain NSC Definitions -




   +-------------------------------------+   NSC External Object Type
   ! Default for NSC ext. object type PL !
   !                                     !   PO  PRD-Docu-Object
   !  Read     Y (Y,N)                  !   PE  PRD-Ext-Object
   !  Add      Y (Y,N)                  !   PF  PRD-Function
   !  Modify   Y (Y,N)                  !   PL  PRD-3Gl-Library
   !  Delete   Y (Y,N)                  !   ' ' All
   +-------------------------------------+
```

When the access mode values have been specified, a report list is created of default definitions to be added. Confirm this list to add the definitions to Natural Security.

Note:
Only objects contained in the report list will be added. We recommend paging to the end of this list to make sure it is complete.

Note:
This function can be run again at any time, for example after adding new user-defined object types in metadata administration. This function adds default definitions for the new object types.

Changing the Default Definitions

Existing default definitions cannot be overwritten. To change the defaults, proceed as follows:

- delete existing values with function Purge NSC Definitions
- reexecute the function Add NSC Default Definitions.

Setting the Parameter Protect current Predict file

Only when all control records have been added in Natural Security can you set the parameter Defaults > General Defaults > Protect current Predict file to Y. This parameter is automatically reset to N if you execute the function Purge NSC Definitions.

Check NSC Definitions - Code C

This function is used to check the consistency of the security definitions in Natural Security. For example:

- If the user has access to file objects starting with ABC*, he must also have access to NSC external object type FI.

Parameters	
Check for user	**D** Default definitions are checked. **blank** Security definitions are checked for all Natural Security users. **user-id** Enter a user ID to check the security definitions of an individual user in Natural Security.
Check for object type	This parameter is only applicable to NSC external object type PO or blank. Either • enter the two-character code of a predefined or user-defined object type in Predict to limit the function to security objects of a particular type, or • leave this field blank to check security definitions for all Predict object types.
List all	**N** Only inconsistent security definitions are listed. **Y** All security definitions for the specified user and object type are listed.

Inconsistent security definitions are written to an Software AG Editor session. If no inconsistencies are found, a corresponding message is given.

See sample screen below:

```
10:57:38            *****  P R E D I C T  4.3.1  *****              2003-05-31
                           - Check NSC Definitions -

 NSC external object ... PO  *PRD-Docu-Object
 ------------------------------------------------------------------------------
 Definitions for user ... GMA

 Cnt   NSC object                                Read Add Modify Delete

 1     DA    ARH*                                 Y    Y     Y     N
           >>> Superior definition: MODIFY is not allowed <<<
 2     DA-A                                       Y    Y     Y     N
           >>> Superior definition: MODIFY is not allowed <<<
 3     DA-A  A*                                   Y    Y     Y     Y
           >>> Superior definition: DELETE is not allowed <<<
 4     DA-A  GMA-DA                               Y    N     N     Y
           >>> Superior definition: DELETE is not allowed <<<
 5     FI    A*                                   N    Y     N     N
           >>> Read must be allowed if others are allowed <<<
```


Display NSC Definitions - Code D

With this function you can list the security definitions in Natural Security which are different to higher-level definitions. This function combines default definitions and any definitions for the individual user, so you can see exactly what permission the current user has.

Display for user	**D** Default definitions are displayed. **blank** Security definitions are displayed for all Natural Security users. **user-id** Enter a user ID to display the security definitions of an individual user in Natural Security.
Display for object type	This parameter is only applicable to NSC external object type PO or blank. Either • enter the two-character code of a predefined or user-defined object type in Predict to limit the function to security objects of a particular type, or • leave this field blank to display security definitions for all Predict object types.

See sample screen below.

```
13:02:46             *****  P R E D I C T  4.3.1  *****            2003-05-31
                          - Display NSC Definitions -

NSC external object ... PO  *PRD-Docu-Object
------------------------------------------------------------------------------
Default definitions

Cnt   NSC object                                 Read Add Modify Delete

1     >>> Default value <<<                       Y    Y     Y     N
2     DA                                         Y    N     N     N
3     DA-A  GMA-DA                               Y    Y     Y     Y
```

Purge NSC Definitions - Code F

With this function you can delete security definitions in Natural Security. You must delete **all** security definitions for a particular NSC external object type, otherwise inconsistencies and a large amount of obsolete data could result if the default value were changed.

A window appears which indicates how many NSC external objects and how many links to users/groups will be affected by the purge.

Parameter Protect current Predict file

The parameter Defaults > General Defaults > Protect current Predict file is automatically reset to N if you execute the function Purge NSC Definitions. If you subsequently add new NSC definitions, you will be able to reset this parameter to Y.

Recover

File descriptions are locked while the following functions are performed on them:

- Rename/Renumber file
- Copy file
- Resequence file
- Catalog elements of a file
- Generate DDM

These functions issue an intermediate ET (end of transaction) to prevent a hold queue overflow. Until the function finishes successfully, the file descriptions are in an inconsistent state, so Predict sets them in locked status. After successful execution the files are unlocked automatically.

A locked file description must be recovered before it can be used again (although purging is always allowed). The Recovery function unlocks it.

The Recover screen is invoked with code R in the Special Functions menu or with the command SPECIAL RECOVER.

```
    13:38:06           *****  P R E D I C T  4.3.1  *****            2003-05-31
    Plan   3                        - Recover -

       Some Predict functions issue intermediate 'ET's to prevent a hold queue
    overflow. Until the functions finish successfully, their objects (always file
    descriptions) are in an inconsistent state so Predict sets them in locked
    status.
       A locked file description must be recovered before it can be used again
    (although purging is always allowed); the recovery function unlocks is.


                           Function

                           R  Recover
                           S  Select

    Function ..............

    File ID ...............

    Command ===>
    Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
          Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

Parameter	
File ID	ID of the file to be recovered. Internal ID must be entered for Recover function. This field is ignored for function Select.

Functions

Recover - Code R

Predict determines whether it is possible to complete the updating of the affected files, or whether partially completed transaction groups must be backed-out. In either case, the files are left in a consistent condition. The user should carefully check the files and reapply updates which were in process at the time of the system failure and could not be completed by the recovery process.

Select - Code S

Lists the locked files. Mark with any non-blank character in column M the files to be recovered.

Security for Adabas Online Services

Predict/AOS profiles can be added/modified/displayed or purged. See Protecting Adabas Databases and Files in the section **Protecting External Objects in Predict with Natural Security** in the **Predict Security documentation** for a description of the concepts and handling this option.

The Security for Adabas Online Services screen is invoked with code S in the Special Functions menu or with the command SPECIAL SECURITY.

```
 13:51:39            *****  P R E D I C T  4.3.1  *****             2003-05-31
                   - Security for Adabas Online Services -             DDAAOSM3


                      Code    Function
                      ------  ---------------
                        A     Add new Profile
                        D     Display Profile
                        M     Modify Profile
                        P     Purge Profile
                        S     Select Profile
                        ?     Help
                        .     Terminate
                      ------  ---------------

           Enter Code : _____
             File No. : _____   To File No.:
         Database ID  : _____
         Predict-user : ________

   or direct command:
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Term  S-fi  E-el  M-pr  Print Impl  Let   Flip  Prof  Menu
```

For more details see the section **Adabas Online Services** of the **Adabas DBA Reference documentation**.

Consistency of Predict

The Consistency of Predict screen is invoked with code P in the Special Functions menu or with the command SPECIAL CONSISTENCY.

Data can become logically inconsistent as a result of errors. Some of these errors can be rectified with this function.

For Predict data to be modified, the Update option must be set to Y.
If set to N (default), inconsistencies will only be listed.

```
13:43:39            *****  P R E D I C T  4.3.1  *****             2003-05-31
                    - Check Consistency of Predict data -

                 Function

                 B   Check database records
                 D   Check the extended descriptions
                 E   Conversion of EDIT MASKS in Field Entries
                 F   Check consistency of files and fields
                 K   Check consistency of keywords
                 P   Check entries for programs
                 R   Conversion of free and automatic rules
                 V   Check verifications
                 N   Check Naming conventions
                 X   Check XREF data



Function .......


Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

Functions

Check database records - Code B

Checks the validity and consistency of database and file attributes:

- virtual machine of the database exists
- database type valid
- Run mode valid
- DBnr valid and unique
- Vista access only parameter valid
- Vista key valid (see Vista Key in the section **Vista** in the **Predict and Other Systems documentation**)
- IMS values
- File types conform with database type
- File numbers in valid range
- Physical distribution type of files valid
- Vista elements exist (if necessary) and valid

Check the text attributes - Code D

The following checks are performed:

- The invalid character H'00' in text attributes is found. If Update is set to Y, all H'00' are changed to blanks.
- If Check description in the object type administration is set to Y, default text in descriptions is checked for correctness and completeness.
- Text attributes without object are deleted.
- Invalid records (created during conversion) are corrected.

Conversion of EDIT MASKS in Field Entries - Code E

The Adabas edit masks are transformed into Natural edit masks

Check consistency of files and fields - Code F

Performs a check whether a master file is available for all user view files. The consistency of the file and field definitions of user views are checked after the consistency of the master file was checked. Field definitions of user views are not checked separately. The inconsistencies are listed so that the errors are comprehensible with the exception of the field attributes SUPER-USAGE, PE-INDICATOR, VERIFICATION-TYPE and INTERNAL FILE NUMBER. These attributes do not appear on any screen. In the user-view SYSDIC-EL, which is normally available on the Predict file, these attributes are described

Check consistency of keywords - Code K

This function checks if keywords are used which are not defined within Predict and lists all references to such keywords. A keyword must be defined in Predict before it can be used. There are two possibilities to get rid of these inconsistencies:

- execute this function with the UPDATE option set to Y.
- add the keywords using Predict Maintenance functions.

Check entries for programs - Code P

Deletes entry points for which no program exists.

Conversion of free and automatic rules - Code R

Predict rules which were stored in Predict V2.n are transformed to the new required format.

Check verifications - Code V

The consistency of the following records is checked:

- SYSDIC-VE-ACT Active rules of verifications of type free and automatic verifications.
- SYSDIC-VE-EL Connection records between DDMs and active rules of automatic verifications.
- SYSDIC-DESC Passive rules of conceptual and automatic verifications
- SYSDIC-VE All verification objects
- SYSDIC-FI All file objects.

Check naming conventions - Code N

This function checks the IDs of all objects of a specified type against the naming conventions defined with Metadata Administration function Add/Modify object type definition. See Add / Modify Object Type - Code A, M. All objects which do not conform to these naming conventions are listed.

If no object type is specified, all objects of all types are checked against default naming conventions. See Defaults Administration in the section **Metadata Administration** in this documentation.

Check XRef data - Code X

This function lists invalid XRef data. The scope of the function may be restricted by database number, file number, library and program name. Asterisk notation may be used for library and program name.

If the Update option is set to Y, corrupted data is corrected whenever possible. If an ID is specified with parameter User ID used to create set, all programs containing XRef data that cannot be corrected automatically are put in a set. See Using Sets in the section **LIST XREF for Natural** in the **Predict Reference documentation**.

Maintain Active References

Functions in the Maintain Active References screen are used to delete XRef data and to delete control information written by the Predict preprocessor.

Deleting XRef Data

XRef data for implemented members is normally deleted if the implemented members are deleted. In the following cases, however, XRef data has to be deleted manually:

- After 3GL programs were deleted (ADA, Assembler, COBOL, FORTRAN, PL/I programs or SQL-DBRMs). XRef data for 3GL programs is not deleted automatically. XRef data for 3GL is deleted with the function Delete 3GL data.
- If XRef data for Natural members could not be deleted together with the members
  - because the members were deleted with an Adabas utility or
  - the XRef data for the purged members is stored in a Predict system that was not active when the Natural members were deleted.

  XRef data for Natural is deleted with the function Delete Natural data.

Deleting Preprocessor Control Information

Predict writes control information during execution of the preprocessor. If the preprocessor terminates abnormally, this control information remains on the Predict system file. It can be deleted with the function Delete Preprocessor abend data.

Calling the Function

The Maintain Active References screen is invoked with code X in the Special Functions menu or with the command SPECIAL XREF.


```
13:51:39            *****  P R E D I C T  4.3.1  *****              2003-05-31
Plan   3             - Maintenance Active References -



                    Function

                    A  Delete Preprocessor abend data
                    G  Delete 3GL data
                    N  Delete Natural data



Function ..........

Member ............                    User system file number ...... 54
Library ...........                    User system database number .. 180
                                       User system password .........
                                       User system cipher code ......


Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

Parameters	
Program	Limits the function to data in the program specified. Asterisk notation can be used when deleting 3GL data. If a program is entered when deleting preprocessor or Natural data, Library must be entered too.
Library	Limits the function to data in the library specified. Leave blank to deletes XRef data in all libraries. **Note:** When deleting 3GL data: if an asterisk is entered as first character here, it is taken literally (it is not interpreted as a generic character). Example: to delete all XRef data in libraries beginning with *SYS, enter *SYS*.
User system file number	Only applicable when deleting Natural data: Must be entered to identify the file and database containing the Natural programs.
User system database number	
User system password	
User system cipher code	

Functions

Delete Preprocessor ABEND data - Code A

Deletes all protocol data from previous day that was created if the preprocessor terminated abnormally. Only protocol data from today and yesterday is kept.

Delete 3GL data - Code G

Lists XRef data for programs written in 3GL languages or SQL-DBRMs. Mark these for deletion at your own risk, as Predict cannot adequately supervise non-Natural object programs.

Delete Natural data - Code N

Deletes XRef data for Natural programs. XRef data will be deleted only if no corresponding module is found. The file number and database number of the file where the Natural programs are stored must be specified.

Asterisk notation can be used for Library and Program to specify a range of XRef records to be deleted.

Maintain XRef Data in Batch Mode

This function can be executed in batch mode with the command SPECIAL XREF. Parameters can be entered in positional or keyword form. The table below gives a list of keywords, the corresponding field in the Maintenance Active References screen and the relative position of the keywords.

Keyword	Field	Position
FUNCTION	Function	1
PR	Program	2
LIB	Library	3
FNR	User system file number	4
DBNR	User system database number	5
Password	User system password	6
Cipher	User system cipher code	7

Example

To delete the XRef data for a PL/I program with name TEST in library PLILIB code the command:

```
SPECIAL XREF
FUNCTION=G, PR=TEST, LIB=PLILIB
```

or in positional form

```
SPECIAL XREF
G,TEST,PLILIB
```


Maintain Standard Fields

Overview

The Special Function Maintain standard fields allows you to identify duplicate standard fields and assign standard fields to another standard file. This functionality was introduced for the following reasons:

- Coupled fields no longer need to have the same ID.
  This special function enables you to consolidate standard fields within your organization.
- Duplicated standard fields may not be transferred to Natural Engineering Workbench. See Recognizing and Resolving Conflicts with Duplicated Standard Fields. This special function identifies duplicated standard fields and offers means of resolve this duplication.
- In Predict, data element types from Natural Engineering Workbench and Predict Case are transferred as field objects and are linked to the default standard file SAG-DT. These fields can be reassigned with this special function. See Reassign Data Element Types from PCA and NEW.

Functional Scope

- List duplicate standard fields
  This function lists standard fields with the same ID and identical standard attributes.
- Select duplicate standard fields
  This function lists standard fields with the same ID and identical standard attributes for selection. You can use this list to put commands in the workplan to resolve the duplication.
- Move field to other standard file
  This function assigns the standard field to another standard file.
- Reassign standard relationships
  This function transfers all references from one standard field to another standard field, either in the same file or in another standard file.

Calling the Function

Call the menu Maintain standard fields with code Z in the Special Functions Menu or with the command SPECIAL STANDARD.


```
09:33:02              *****  P R E D I C T  4.3.1  *****              2003-05-31
                         - Maintain Standard Fields -



                   Function

                   L   List duplicate standard fields
                   S   Select duplicate standard fields
                   M   Move field to other standard file
                   R   Reassign standard relationships


Function ........

from File ID ...*
    Field ID ...*
to   File ID ...*
    Field ID ...*

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

Enter the following parameters for the functions Move field to other standard file and Reassign standard relationship. The functions List/Select duplicate standard fields require only the first parameter.

Parameters	
from File ID	ID of the file containing the field to be processed.
from Field ID	ID of the field to be processed. For function Move field to other standard file: If this field is contained in the target file, an error message is given. **Warning**: This object is deleted after the Move function is executed. For function Reassign standard relationship: References to this standard field are to be transferred to another field, but the object will still exist after the function has been executed.
to File ID	The target of a Move / Reassign function.
to Field ID	With function Reassign standard relationship: Standard attributes must be identical to from field ID. See below. If asterisk notation is used, only fields with identical standard attributes are displayed for selection. If no fields with identical attributes are found, an error message is given. This parameter is ignored with function Move field to other standard file.

List Duplicate Standard Fields - Code L

This function lists standard fields that have the same ID and identical standard attributes. These attributes are:

- type
- format
- length
- descriptor type

- unique option
- suppression / null value option

In addition, the same

- linked verifications

must be present in both fields. The order in which the verifications are listed is not important.

Calling the Function

Enter code L in the menu Maintain standard fields.

Use the parameter Standard file ID to limit the scope of the function. Asterisk notation is possible. If you leave this field blank, all duplicated fields in all standard files will be listed.

```
09:38:18            *****  P R E D I C T  4.3.1  *****            2003-05-31
                    - List duplicate standard fields -

--------------------------------------------------------------------------------
Cnt   Ty L Field ID                          F    Length D File ID
    1 PE 1 PE-GR                                         PD-Z-001
      PE 1 PE-GR                                         PD-ZZZ
      PE 1 PE-GR                                         PD-Z1
    2    2 EL1-IN-PE                         A     10.0    PD-Z-001
         2 EL1-IN-PE                         A     10.0    PD-ZZZ
         2 EL1-IN-PE                         A     10.0    PD-Z1
    3    2 EL2-IN-PE                         A     10.0    PD-Z-001
```

Alternatively, you can enter the command DUPL-LIST ELEMENT <standard-file-id>.

Select Duplicate Standard Fields - Code S

This function produces a list of standard fields with duplicate IDs and identical standard attributes. See standard attributes list in the section List Duplicate Standard Fields. This list can be used to place a command in the workplan to resolve the duplication.

Calling the Function

Enter code S in the menu Maintain standard fields.

Use the parameter Standard file ID to limit the scope of the function. Asterisk notation is possible. If you leave this field blank, all duplicated fields in all standard files will be listed for selection.

Enter an asterisk in the column Cmd to list all valid commands.


```
09:39:31            *****  P R E D I C T  4.3.1  *****            2003-05-31
Plan   7           - Select duplicate standard fields -

Cmd Ty L Field ID                          F    Length D File ID

 __ PE 1 PE-GR                                          PD-Z-001
 __ PE 1 PE-GR                                          PD-ZZZ
 __ PE 1 PE-GR                                          PD-Z1
 __    2 EL1-IN-PE                         A     10.0    PD-Z-001
 __    2 EL1-IN-PE                         A     10.0    PD-ZZZ
 __    2 EL1-IN-PE                         A     10.0    PD-Z1
 __    2 EL2-IN-PE                         A     10.0    PD-Z-001
```

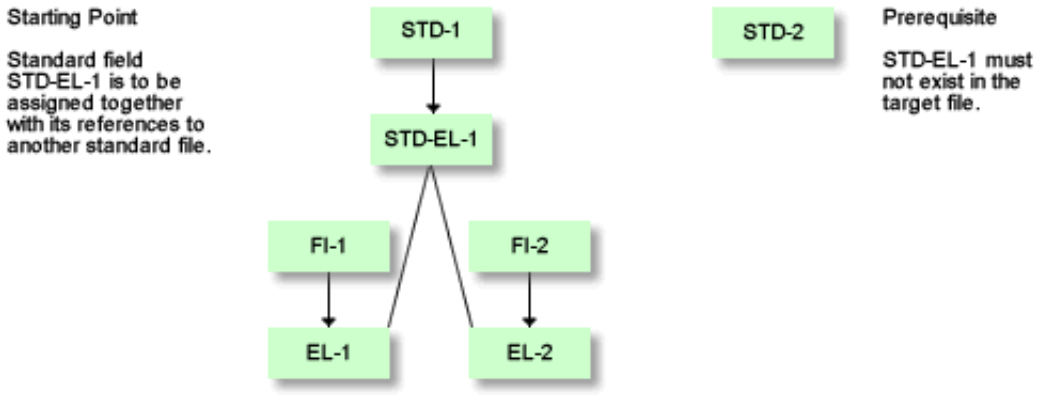
Alternatively, you can enter the command DUPL-SELECT ELEMENT <standard-file-id>.

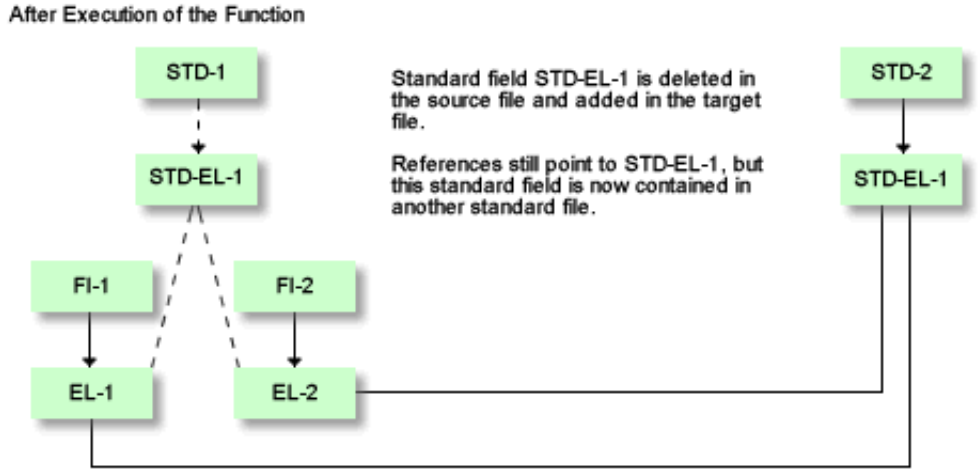
Move Field to other Standard File - Code M

This function assigns a standard field to another standard file. The field specified under From field ID will be deleted and added as To field ID. All references to the old standard field are transferred to the new standard field.

Command: SHIFT ELEMENT <file 1> <field 1> <file 2>

Example





Reassign Standard Relationships - Code S

This function reassigns all references from one standard field to another standard field, either within the same file (Example 1) or in another file (Example 2 without rename of standard field and Example 3 with rename).

A prerequisite for the successful execution of this function is that the standard attributes in source and target fields are identical. See list of standard attributes in the section List Duplicate Standard Fields.

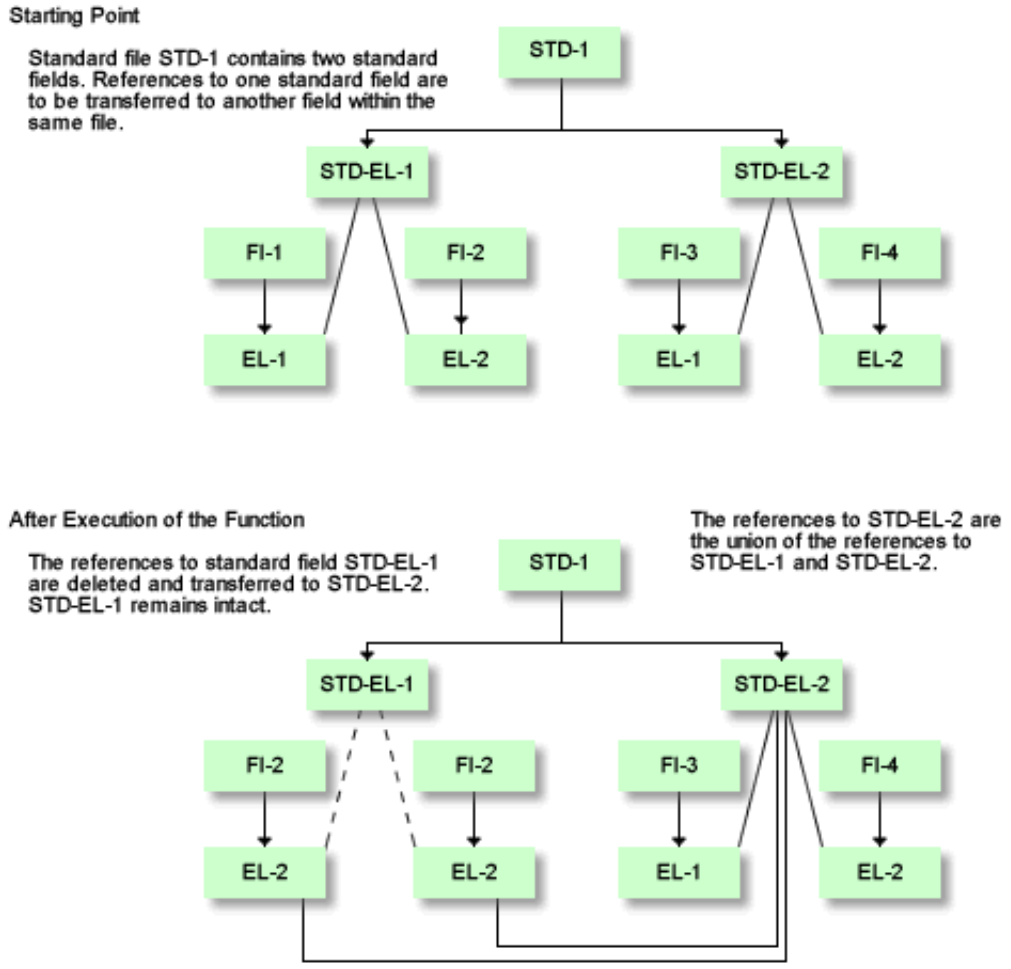
If you specify for to Field ID a field where the standard attributes are not identical, a screen similar to the one below appears.

```
13:48:34              *****  P R E D I C T  4.3.1  *****            2003-05-31
                         - Maintain standard fields -

   Ty L Field ID                          Fo Length   D U N
   -- - ------------------------------- -- -------- - - -
      1 NUM_5_2                           NS 5.02         U
   Verifications:


   Ty L Field ID                          Fo Length   D U N
   -- - ------------------------------- -- -------- - - -
      1 NUM-4-2                           NS 4.02
   Verifications:

DIC1602 STANDARD ATTRIBUTES ARE DIFFERENT.
```

In this example the values for Length and Suppression are different.

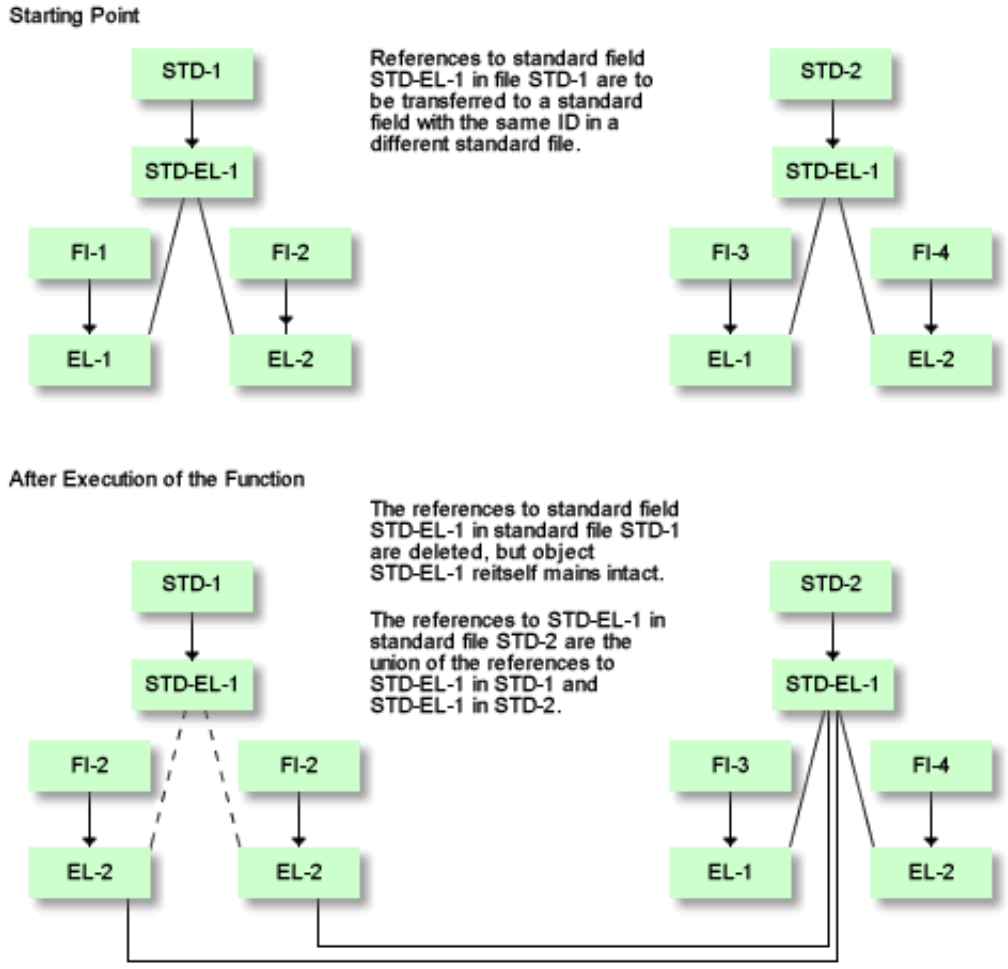
Example 1: Merge in the same File

Command: REASSIGN ELEMENT <file 1> <field 1> <file 1> <field 2>



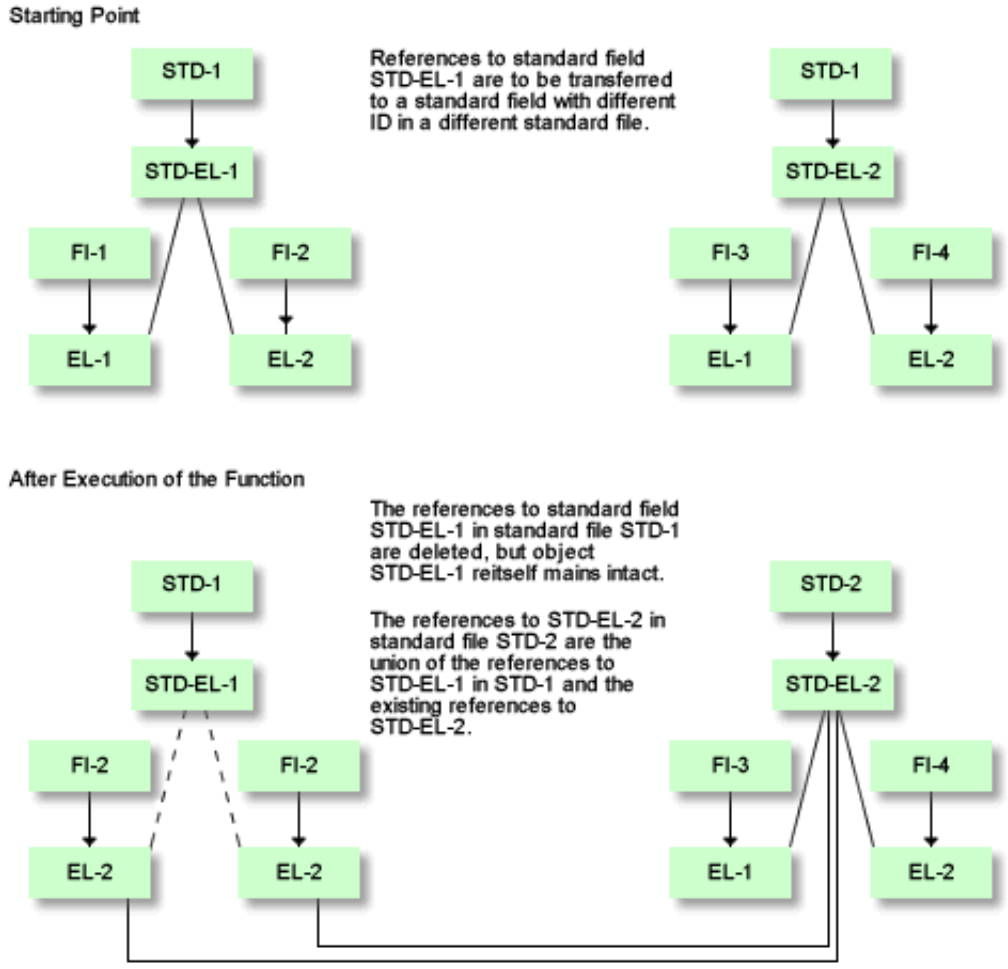
Example 2: Merge over two Files without Rename

Command: REASSIGN ELEMENT <file 1> <field 1> <file 2> <field 1>



Example 3: Merge over two Files with Rename

Command: REASSIGN ELEMENT <file 1> <field 1> <file 2> <field 2>



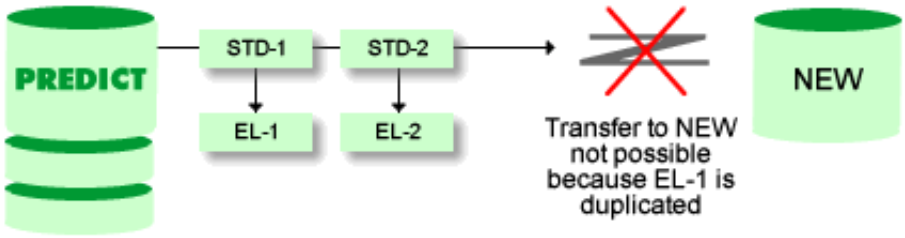
Using the Function Maintain Standard Fields

Below are two examples of where to use this special function:

- Conflicts with duplicated standard fields. See below
- Reassign Data Element Types from PCA and NEW

Recognizing and Resolving Conflicts with Duplicated Standard Fields

Standard fields in Predict are stored in Natural Engineering Workbench (NEW) as Data Element Types. In Predict, a standard field only has to be unique within the standard file. In NEW, on the other hand, a standard field has to be unique within the entire system. This can lead to conflicts when transferring data from Predict to NEW.



Use the functions offered in the menu Maintain standard fields to identify duplicate standard field IDs and resolve the situation.

Steps

- Execute the function List duplicate standard fields.

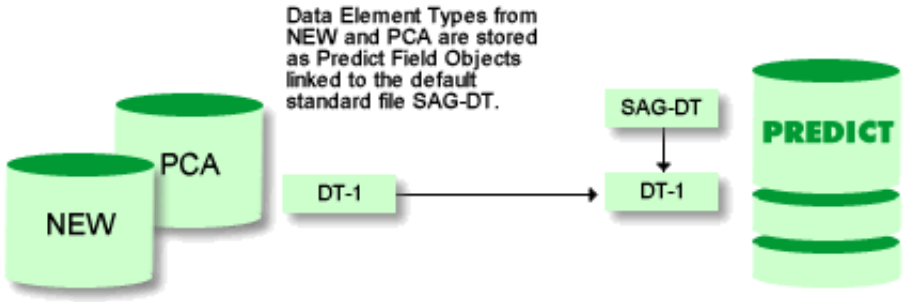
This produces a list of all duplicate standard fields.

- If the standard attributes are identical (See List duplicate standard fields), we recommend the following:
  - Execute the function Reassign standard relationships to consolidate the references to these two fields in the target standard field.
  - Delete the other standard field.
- If the standard attributes are not identical, rename one of the duplicate standard fields.

Now you can transfer your data to NEW.

Reassign Data Element Types from PCA and NEW

Data Element Types from Natural Engineering Workbench or Predict Case are stored in Predict as standard fields linked to the default standard file SAG-DT. This file is delivered as standard.



Use the function Reassign standard relationships or Move field to other standard file to reassign the standard fields from NEW or Predict Case to another standard file.

Refresh Coordinator FDIC

This function clears the Coordinator FDIC if the user who started an load/import operation is not able to clear the Coordinator FDIC himself.

The Refresh Coordinator FDIC screen is invoked with code F in the Special Functions menu or with the command SPECIAL REFRESH.

Refreshing the Coordinator FDIC in Batch Mode

This function can be executed in batch mode with the command SPECIAL REFRESH. This command has no parameters.

Mass Delete of Report Listings

This function deletes multiple report listings in one operation.

The Mass delete of report listings screen is invoked with code L in the Special Functions menu or with the command SPECIAL DELETE.

```
09:19:10            *****  P R E D I C T  4.3.1  *****            2003-05-31
                      Mass delete of Report listings



Report listing ID ....
Extract ID ..........*

With update .......... N (Y/N)
```

Parameters	
Report listing ID	The ID of the report listing to be deleted. Asterisk notation is possible.
Extract ID	If an extract ID is specified, the scope of the function is limited to report listing linked to the specified extract.
With update	**Y** Report listings are displayed and deleted. **N** Report listings are displayed only.

Deleting Report Listings in Batch Mode

This function can be executed in batch mode with the command SPECIAL DELETE. Parameters can be entered in positional or keyword form. The table below gives a list of keywords, the corresponding field in the Mass delete of report listings screen and the relative position of the keywords.

Keyword	Field	Position
RT-ID	Report listing ID	1
ET-ID	Extract ID	2
UPD	With update	3

Example

To delete the report listing EXA-RT linked to extract EXA-ET, code the command:

```
SPECIAL DELETE
RT-ID=EXA-RT, ET-ID=EXA-ET, UPD=N
```

or in positional form


```
SPECIAL DELETE
EXA-RT,EXA-ET,N
```

User Exits

User exits are Natural subprograms which are delivered as source members along with Predict. They can be used to enforce standards for documenting applications or to customize Predict functions. Examples for the use of user exits are:

- implementing security checks
- enforcing naming conventions
- logging changes to Predict data
- integrating third-party products into Predict

Note:
The subprograms U-EXPSC, U-EXPSRP, U-IMPSC, and U-IMPSRP are also delivered as source members. They are used for transferring text between Predict and an external environment and are called automatically by the respective functions. See Exporting Text to an External Target and Importing Text from an External Source of section **Editors in Predict** in the **Predict Reference documentation**. These user exits cannot be controlled by the Activate user exits function and are therefore not described in this section.

This section covers the following topics:

- Concept of User Exits
- New User Exits
- Old User Exits

Note:
All user exits are documented with Predict objects of type program. To obtain additional information on user exits, apply retrieval functions to these objects. Program objects documenting user exits are named SAG-PRD-userexitname (for example, SAG-PRD-U-MODEL is the name of the program object documenting the user exit U-MODEL).

Concept of User Exits

This section contains:

- Distinction between Old and New User Exits
- General Rules
- Format of Description of User Exits

Distinction between Old and New User Exits

The concept of user exits was redesigned with Version 3.3 of Predict. Although user exits of previous versions will be supported in future versions (to ensure compatibility) you are recommended to use the new user exits.

Old user exits are:

- ACMxxEX - Add/Copy/Modify objects
- CATELEX - Catalog fields
- CATOWEX - Catalog owners
- PURxxEX - Purge objects

General Rules

The following applies to both old and new user exits. Rules applying to either old or new user exits are described later in this section (see Old User Exits or New User Exits).

- **User exits are either called before or after execution of a function**
  User exits called before execution may, for example, be used to check naming conventions for object IDs; those called after execution may be used to perform security checks depending on attribute values. Old user exits are always called **after** execution of a function.
- **User exits are delivered as source members together with Predict**
  After Predict has been installed, the user exits delivered with Predict are located in the library SYSDIC.

  Warning:
  You are recommended to keep copies of the Natural source programs for your own user exits in a separate library to prevent them from being overwritten during installation of any new Predict version or software maintenance level.

- **Message texts can be stored in the Natural message file**
  Most user exits can display an online message. Message texts can be stored in the Natural system file, identified by a message number. For library SYSDIC, message numbers from 9000 to 9999 are reserved for user exit messages. Message texts are maintained with the Natural utility SYSERR.
- **Using Version 3.4 User Exits under Predict Version 4.2**
  See User Exit Routines in the section **Considerations Prior to Upgrading** in the **Predict Installation documentation** for information on how to use Version 3.4 user exits with Version 4.2.
- **Restrictions**
  - Predict modules must not be used in user exits.
  - The stack must not be changed.
  - If any of the user exits issue BACKOUT TRANSACTION statements, the Natural parameter OPRB must be set to NOOPEN. This is not required under UNIX.
    For further details, see the respective **Natural Operations documentation**.
  - The contents of the source area must not be changed.

Format of Description of User Exits

All descriptions of user exits are in the following format:

Activate with:

Name of the parameter in one of the Activate User Exits screens that activates or deactivates the user exit, or - if the user exit is not activated in an Activate User Exits screen - the method to activate the user exit.

Applies to:

Object types and/or functions the user exit applies to.

Called where and when:

Where and when the user exit is executed (if active).

Effect of MSG-NR:

How the parameter MSG-NR is evaluated. Only for new user exits. See also Process Control Using MSG-NR.

+SYSDIC-FUNCTION:

How +SYSDIC-FUNCTION is evaluated. Only for old user exits.
See also Process Control Using +SYSDIC-FUNCTION.

Additional remarks:

Additional notes on how to use the user exit.

Any category that does not apply is omitted from the description.

New User Exits

New user exits are of type subprogram and are called using the CALLNAT statement.

This section contains:

- Process Control Using MSG-NR
- Using the Parameter Data Area
- Activating New User Exits
- User Exit for Maintenance Functions
  - U-MNT and U-MNT1 - before any function
  - U-ACMR - add, copy, modify for all object types
  - U-ACM - add, copy, modify for user-defined object types
  - U-AM-A - add, modify for user-defined associations with attributes
  - U-PUR - purge, scratch
  - U-DESC - modify description
  - U-OW - Owner
  - U-REDOC - redocumentation
- User Exits for Retrieval Functions
  - U-RET - before any function
  - U-DSP - after display
  - U-MODEL - execute retrieval model
  - U-XREF - execute retrieval model
- User Exits for Generation Functions
  - U-GEN - before execution
  - U-GEN1 - after execution
- User Exit for Other Function Groups
  - U-ACT - Active Retrieval
  - U-IMPL - File Implementation
  - U-COM - Comparison
  - U-INC - Incorporation
  - U-MIP - Administration Implementation
  - U-CMD - Command Processor
  - U-SPEC Defaults / Special Functions
  - U-OBJID - add, copy, rename, incorporation, redocumentation
  - U-PGMLAN - Define New Program Language
- SUMPRDEX - Maintain Rule in Natural Map Editor

Process Control Using MSG-NR

The parameter MSG-NR determines how a Predict function that called a user exit continues after the user exit processing terminated:

- If MSG-NR is zero, the function continues as normal. An END-OF-TRANSACTION statement is issued.
- If MSG-NR is non-zero, the function either quits or issues a BACKOUT-TRANSACTION statement.
  In either case the screen from which the Predict function was called is displayed, and the message identified by MSG-NR is displayed.

  Note:
  Some programs do not evaluate MSG-NR.

How the parameter MSG-NR is evaluated is described under **Effect of MSG-NR** in the descriptions of the individual user exits below.

Using the Parameter Data Area

Note:
If a user exit has to be cataloged under Predict 4.2, then the corresponding Parameter Data Area (PDA) has to be cataloged before since it is delivered with Predict 4.2 in source form only.

Naming Conventions

The names for the parameter data areas to be used by user exits are composed by concatenating the name of the user exit and a "P" (for example, U-MNTP for the user exit U-MNT).

Use of Parameters

Depending on the user exit, some dozens of parameters may be passed to a user exit. Most of these parameters cannot be modified (Natural CALLNAT parameter AD=O).
The following parameters can be modified:

MSG-NR (I2)
Used for process control and identification of a message. See the section Process Control Using MSG-NR above.

MSG-TEXT1 (A34)
To be inserted for the wild card :1: in the message identified by MSG-NR.

MSG-TEXT2 (A34)
To be inserted for the wild card :2: in the message identified by MSG-NR.

SYSDIC-USER (A250)
Reserved for users; represents the global data area variable +SYSDIC-USER.

Activating New User Exits

With the exception of the user exits U-MODEL and U-XREF, new user exits are activated in the first Activate User Exits screen:
Main Menu -> Defaults -> Activate User Exits ->User Exits (subprograms).


```
10:02:37            *****  P R E D I C T  4.3.1   *****            2003-05-31
                          - Activate User Exits -

--- User exits (subprograms) ---                    Modified 2003-05-31 at 10:02
                                                           by STK

Maintenance                          Active retrieval ................. N (Y/N)
  before any function .. N (Y/N)     File implementation .............. N (Y/N)
  add, copy, modify .... N (Y/N)     Comparison ....................... N (Y/N)
  purge, scratch ....... N (Y/N)     Incorporation .................... N (Y/N)
  modify description ... N (Y/N)     Administration ................... N (Y/N)
  redocumentation ...... N (Y/N)     Command processor ................ N (Y/N)
Retrieval                            Defaults/Special functions ....... N (Y/N)
  before any function .. N (Y/N)     Check against naming conventions . Y (Y/N)
  after display .......* N
Generation
  before execution ..... N (Y/N)
  after execution ...... N (Y/N)


For HELP enter ? in the first input field. To leave enter .
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

Y activates a user exit and **N** deactivates it (with the exception of U-DSP).

Note:
If the user exit does not exist in the Predict system file, the respective parameter cannot be set to Y.

User Exits for Maintenance Functions

U-MNT and U-MNT1 - Before Any Function

Activate with:

parameter before any function

Applies to:

U-MNT applies to **all object types except fields**, U-MNT1 applies **only to fields**.

Note:
When the editor command SEL is used, the parameter +SYSDIC-FUNCTION must be set to T. See Selecting Text from another Source within Predict in the section **Editors in Predict** in the **Predict Reference documentation**.

Called where and when:

- from Maintenance menus before a function is executed,
- before a Predict editor is called,
- before an object is added or modified with the .E command (in an editor),
- before the Select function is executed in an editor,
- before the Purge function with option SCRATCH is executed. The user exit is performed before dependent objects and references to the child objects are deleted. If, in this case, parameter MSG-NR is set, the dependent objects or references to the child object are not deleted and a message text is displayed (or printed in batch mode).
  No possibility for re-input is given here!
  See the respective section of the Predefined Object Types in Predict documentation for a list of objects that

are deleted with the Purge function.

- **File and Database-Specific**
  - in case of add or copy after subtypes have been entered (second screen).
- **Field-Specific**
  - during saving of an edited field list and before a field is added, copied, modified, renamed or deleted.

Effect of MSG-NR:

If non-zero, the Predict function is not executed and the message MSG-NR is displayed.

U-ACMR - Add, Copy, Modify for All Object Types

Activate with:

add, copy, modify

Applies to:

all object types

Called where and when:

- From the add, copy, modify function **after** attribute values have been written to the dictionary, before END OF TRANSACTION.
- Before the user exits ACMxxEX are executed.

Effect of MSG-NR:

If non-zero, BACKOUT TRANSACTION is issued and the message MSG-NR is displayed.

U-ACM - Add, Copy, Modify for User-Defined Object Types

Activate with:

add, copy, modify

Applies to:

user-defined object types

Called where and when:

From the add, copy, modify function **before** attribute values are written to the dictionary.

Effect of MSG-NR:

If non-zero, the Predict function is re-executed and the message MSG-NR is displayed. Attribute values can then be changed in the input screen of the function.

U-AM-A - Add, Modify for User-Defined Association Types with Attributes

Activate with:

add, modify

Applies to:

user-defined associations with attributes

Called where and when:

From the add or modify function **before** attribute values are written to the dictionary.

Effect of MSG-NR:

If non-zero, the Predict function is re-executed and the message MSG-NR is displayed. Attribute values can then be changed in the input screen of the function.

U-PUR - Purge, Scratch

Activate with:

purge, scratch

Applies to:

all object types

Called where and when:

When the option Delete or Scratch has been specified after confirmation of the deletion.

Effect of MSG-NR:

If non-zero, the Predict function is re-executed and the message MSG-NR is displayed. Attribute values can then be changed in the input screen of the function.

U-DESC - Modify Description

Activate with:

modify description

Applies to:

all object types

Called where and when:

- After an extended description was saved and before an END TRANSACTION statement is issued.
- Additional remarks:
  If this user exit is activated, a copy of it must reside in library SYSLIB of the current Natural system file (FNAT). This is necessary to provide the same kind of processing of the description when added or modified using the Natural command edit description.

Effect of MSG-NR:

If non-zero, BACKOUT TRANSACTION is issued and the message MSG-NR is displayed.

U-OW - Owner

Activate with:

U (only U-OW is performed) or

B (U-OW and CATOWEX are performed).

Applies to:

each owner in the list

Called where and when:

Before owners in an owner list are actually updated or deleted by the CAT command.

Effect of MSG-NR:

If non-zero, the editor is called again and the message MSG-NR is displayed.

U-REDOC - Redocumentation

Activate with:

redocumentation

Applies to:

programs

Called where and when:

after parameters for the function Redocument Program have been specified.

Effect of MSG-NR:

If non-zero, the Predict function is not executed and the message MSG-NR is displayed.

User Exits for Retrieval Functions

U-RET - Before Any Function

Activate with:

Retrieval / before any function

Applies to:

all object types

Called where and when:

from each retrieval menu after input parameters have been specified and before the function is executed.

Effect of MSG-NR:

If non-zero, the Predict function is not executed and the message MSG-NR is displayed.

U-DSP - After Display

Activate with:

O
Optional: U-DSP can be (de)activated by the user by setting the output option User exit or

F
Force: U-DSP is active and cannot be deactivated by an output option.

Applies to:

all object types

Called where and when:

after an object has been displayed and before the extended description is displayed.

Effect of MSG-NR:

No effect.

Note:
With this version of Predict, retrieval output is placed in an Software AG Editor session by default. See the section Editors in Predict in the **Predict Reference documentation**. If you used this user exit in a former version, you must convert it before using it with the Software AG Editor (see Converting the User Exits U-DSP and U-MODEL).
If you do not wish to convert this user exit, you must set the parameter Use Software AG Editor for Retrieval output (under General > Miscellaneous) to N. In this case, retrieval output is handled as in former versions of Predict. See Software AG Editor in the section **Editors in Predict** in the **Predict Reference documentation**.

U-MODEL - Execute Retrieval Model

Activate with:

parameter Use U-MODEL in Add/Modify Retrieval Model definition screen when adding or modifying retrieval models

- **Y**
  Objects found by the retrieval operation are processed exclusively by U-MODEL.
- **N**
  U-Model is deactivated.
- **A**
  Objects found are first displayed by the retrieval operation and then processed by the U-MODEL.

Applies to:

all object types specified in the retrieval model

Called where and when:

- for any object found by a retrieval operation with retrieval type Execute retrieval model.
  Further processing depends on the output mode:
  - With output mode structured list (code T), any object found by the retrieval operation can be processed by U-MODEL (for example: be displayed in a user-defined format). The scope of a report can be restricted using the parameter REJECT. If REJECT is set to TRUE for an object, processing of the retrieval model skips all objects on lower levels and continues with the next object on the same or a higher level.

  For example: Information on a Natural application using utilities implemented in COBOL are retrieved. Any information on internal structures of the COBOL utilities can be excluded from the report by setting REJECT=TRUE whenever a COBOL member is found.

- With output mode cross reference (code X), the report can be restricted using the parameter REJECT. With this output mode, immediate further processing of object information collected by the retrieval operation is not possible: object data is written to a sort buffer (source area) before it is displayed. For further processing, U-XREF must be used.

Effect of MSG-NR:

No effect.

Note:
From the current Predict version, retrieval output is by default placed in an Software AG Editor session (see the section Editors in Predict in the **Predict Reference documentation**.). If you used this user exit in a former version, you must convert it before using it with the Software AG Editor (see Converting the User Exits U-DSP and U-MODEL).

If you do not wish to convert this user exit, you must set the parameter Use Software AG Editor for Retrieval output (under General > Miscellaneous) to N. In this case, retrieval output is handled as in former versions of Predict. See Software AG Editor in the section **Editors in Predict** in the **Predict Reference documentation**.

U-XREF - Execute Retrieval Model / Cross Reference

Activate with:

parameter Use U-XREF in Add/Modify Retrieval Model definition screen when adding or modifying retrieval models

Applies to:

all object types specified in the retrieval mode

Called where and when:

after object information has been read from the sort buffer (source area) for display (retrieval type Execute retrieval model and output mode cross reference).

Effect of MSG-NR:

No effect.

Converting the User Exits U-DSP and U-MODEL

If these user exits have been used in versions of Predict 3.3 or below, they must be converted before they can be used in the current version. This section describes the changes and enhancements which must be considered:

- The parameter data areas U-DSPP and U-MODELP have been enhanced by the parameter SYSDIC-SESSION-NO. This parameter specifies the session number which is used for output; it must not be changed.
- The subprogram N-WRTEDT has been included in the user exits U-DSP and U-MODEL. This subprogram passes the output lines to the editor. N-WRTEDT has the following parameters:

Parameter	Meaning
SYSDIC-SESSION-NO	current session number used for output
TABLE-LINE-OUT (A79/1:15)	table containing the output lines
CHAR-DATE (A1/1:15)	defines the type of output data (subheader, header, etc., see example below)
INTENSIVE-CHAR-BEG	user-defined character to mark the beginning of high-lighted text
INTENSIVE-CHAR-END	user-defined character to mark the end of highlighted text
SYSDIC-MSG-GR	message indicating whether processing will continue or whether the function is terminated
SYSDIC-TRACE	for internal use only

Programming Hints

- 'AT TOP OF PAGE' control is no longer needed since it is performed by the editor program.
- Each WRITE statement must be replaced by a MOVE or COMPRESS statement.
- The entries in the tables TABLE-LINE-OUT and CHAR-DATE refer to each other: An entry in CHAR-DATE determines the type of the data contained in the corresponding entry of TABLE-LINE-OUT.

Example:

Index	TABLE-LINE-OUT	CHAR-DATE
1	header1	H
2	header2	H
3	subheader	S
4	actual data	

Explanation:

- The first line to be displayed is a header (indicated by 'H' in CHAR-DATE) with the text 'header1' (corresponding entry in TABLE-LINE-OUT).
- The second line is also part of the header and contains the text 'header2'.
- The third line is a subheader (indicated by 'S' in CHAR-DATE) with the text 'subheader1'.
- The forth line contains the data (indicated by blank in CHAR-DATE).
- The lines must be passed to the subprogram N-WRTEDT in the appropriate format.
- After N-WRTEDT has been called, MSG-NR must be checked. If MSG-NR is greater than zero, an ESCAPE ROUTINE statement must be performed.

- Possible values of CHAR-DATE:

<table>
<tr><th colspan="2">Possible Values</th></tr>
<tr><td>P (Newpage)</td><td>User-defined page break; for this entry in CHAR-DATE, there is no corresponding entry in TABLE-LINE-OUT.</td></tr>
<tr><td>H (Header)</td><td>A header can consist of one or more lines. The separation line between header and data is controlled by the editor program.</td></tr>
<tr><td>S (Subheader)</td><td>Subheaders are highlighted and are not repeated on subsequent pages.</td></tr>
<tr><td>T (Column header)</td><td>A column header: is repeated on any subsequent page; may consist of more than one line; is highlighted; is deactivated when a new header (CHAR-DATE = H) or a new column header is to be displayed.</td></tr>
<tr><td>X (Permanent subheader)</td><td>Subheader which is to be repeated on any subsequent page. It is deactivated when a new header (CHAR-DATE = H) or a new subheader (CHAR-DATE = S or T or X) is to be displayed.</td></tr>
<tr><td>D (Output using PM=C)</td><td>For lines to be displayed using the session parameter PM=C, the following must be entered in the tables (before the lines containing the data) to activate this parameter:

<table><tr><th>TABLE-LINE-OUT</th><th>CHAR-DATE</th></tr><tr><td>PM=C</td><td>D</td></tr></table>
To deactivate the session parameter, the following must be specified:

<table><tr><th>TABLE-LINE-OUT</th><th>CHAR-DATE</th></tr><tr><td>PM=</td><td>D</td></tr></table>
The output of headers cannot be controlled using PM=C.</td></tr>
</table>

User Exits for Generation Functions

U-GEN - Before Execution

Activate with:

before execution

Applies to:

all object types processed by generation functions

Called where and when:

after generation parameters have been specified and before the generation is executed.

Effect of MSG-NR:

If non-zero, the Predict function is not executed and the message MSG-NR is displayed.

Additional remarks:

U-GEN is active after installation. U-GEN as delivered rejects the Generate Adabas file function in order to prevent data from being overwritten inadvertently.
Before the function Generate Adabas file can be used, U-GEN must be modified **or** before execution must be set to N.

Note:
Adabas files can also be protected by means of Natural Security. For more information see the section Protecting External Objects in Predict with Natural Security in the **Predict Security documentation**.

U-GEN1 - After Execution

Activate with:

after execution

Applies to:

all object types processed by generation functions

Called where and when:

after successful execution of generation (a member must have been saved).

Effect of MSG-NR:

No effect.

User Exits for Other Function Groups

U-ACT - Active Retrieval

Activate with:

Active retrieval

Applies to:

object types processed by active retrieval functions

Called where and when:

from each active retrieval menu after input parameters have been specified and before the function is executed.

Effect of MSG-NR:

If non-zero, the Predict function is not executed and the message MSG-NR is displayed.

U-IMPL - File Implementation

Activate with:

File implementation

Applies to:

File implementation

Called where and when:

Before an implementation plan is added, extended, listed, modified, purged or executed.

Effect of MSG-NR:

If non-zero, the Predict function is not executed and the message MSG-NR is displayed.

U-COM - Comparison

Activate with:

Comparison

Applies to:

all object types processed by comparison functions

Called where and when:

after comparison parameters have been specified and before the function is executed.

Effect of MSG-NR:

If non-zero, the Predict function is not executed and the message MSG-NR is displayed.

U-INC - Incorporation

Activate with:

Incorporation

Applies to:

all object types processed by incorporation functions

Called where and when:

after incorporation parameters have been specified and before incorporation is executed. Exception: With the functions Incorporate Natural Security User and Incorporate Super Natural User, the user exit is called before specification of the parameters and before execution, but all selected object types are incorporated during the same run.

Effect of MSG-NR:

If non-zero, the Predict function is not executed and the message MSG-NR is displayed.

U-MIP - Administration Implementation

Activate with:

Administration

Applies to:

all external objects processed by administration functions

Called where and when:

from the Administration Implemented Object menu after parameters have been specified and before the function is executed.

Effect of MSG-NR:

If non-zero, the Predict function is not executed and the message MSG-NR is displayed.

Additional remarks:

U-MIP is active after installation. U-MIP as delivered inhibits execution of the functions Purge/Refresh Adabas file to prevent accidental loss of data.
Before the functions Purge/Refresh Adabas file can be used, U-MIP has to be modified or Administration must be set to N.

- **Note:**
  Adabas files can also be protected by means of Natural Security. For more information see the section Protecting External Objects in Predict with Natural Security in the **Predict Security documentation**.

U-CMD - Command Processor

Activate with:

Command Processor

Applies to:

Predict commands

Called where and when:

after the syntactical analysis and before interpretation and execution of a command. A command not recognized as a Predict command is passed to the user exit with SYSDIC-COMMAND-TYPE 'US'. The user exit can be used

- to interpret user-defined commands
- to inhibit commands.

Effect of MSG-NR:

If non-zero, the Predict command is not executed and the message MSG-NR is displayed.

Additional remarks:

U-CMD allows implementation of user-specific command codes. An example is contained in U-CMD.

U-SPEC - Defaults / Special Functions

Activate with:

Defaults / Special functions

Applies to:

all Predict administration functions contained in the Defaults / Special functions menus

Called where and when:

before any of the functions in the Defaults or Special Functions screen is displayed

Effect of MSG-NR:

If non-zero, the calling of the Defaults or Special Functions screen is rejected and the message MSG-NR is displayed.

Additional remarks:

U-SPEC can be used to protect Predict Administration functions contained in the Defaults / Special functions menus.

U-OBJID - Add, Copy, Rename, Incorporation, Redocumentation

Activate with:

add, copy, rename, incorporation, redocumentation

Applies to:

all object types

Called where and when:

after the object ID has been checked (first character is a letter, length and disallowed characters defined under Metadata Administration) and before END TRANSACTION is executed.

Effect of MSG-NR:

If non-zero, BACKOUT TRANSACTION is issued and the message MSG-NR is displayed.

U-PGMLAN - Define New Program Language

Activate with:

U-PGMLAN is always active

Applies to:

program objects

Called where and when:

Whenever Predict functions display a selection window for programming languages and if changes to the attribute language are validated.

Effect of MSG-NR:

No effect.

Additional remarks:

Up to ten new languages can be defined.

SUMPRDEX - Maintain Rule in Natural Map Editor

SUMPRDEX does not fit into the ’old/new’ scheme we use to categorize user exits. It is in fact an old user exit because it already existed in Predict V2.3, is a Natural program (not a subprogram) and is activated in the User exits (programs) screen. But unlike other old user exits, SUMPRDEX cannot be substituted by a new user exit and is therefore contained in this section on new user exits.

Note:
See also Protecting Processing Rules in the section **Protecting External Objects in Predict with Natural Security** in the **Predict Security documentation**.

User Exit SUMPRDEX

Activate with:

Y
for verification in the column ACM of the Activate user exits - User exits (programs) screen. See Activating Old User Exits.
If you specify Y in this field, the user exit ACMVEEX is activated, too.

Applies to:

Free rules.

Called where and when:

Immediately after a rule has been added or modified with the Natural map editor, the program SUMPRDM in library SYSLIB is called. This program issues a CALLNAT SUMPRDEX statement and passes the variables listed in the table below.

SUMPRDEX uses SYSDIC-RSP for process control.

Additional remarks:

No global data area is available. SUMPRDEX has to be stored in library SYSLIB.

Variable	Meaning
VERIFICATION-ID(A32)	ID of the variable currently being processed.
SYSDIC-ISN(P8)	ISN of the record just processed.
SYSDIC-FUNCTON(A1)	Current function: **A** add **M** modify
SYSDIC-RSP(N1)	Response code: **0** ok **2** BACKOUT TRANSACTION

Old User Exits

All old user exits are of type program. Old user exits are still supported to ensure compatibility with earlier versions of Predict. You are recommended to use new user exits.

This section contains:

Overview of Old User Exits

ACMxxEX

The Predict functions Add, Copy, Modify and Rename call the user exits ACMxxEX.

xx stands for one of the object type codes: DA, EL, FI, KY, PR, RL, SY, US and VE.

For example: the user exit ACMFIEX is called before a file is added, copied or modified.

PURxxEX

The Purge function calls the user exits PURxxEX before the deletion is executed.

xx stands for one of the object type codes: DA, EL, FI, KY, PR, RL, SY, US and VE.

For example: the user exit PURELEX is called before a field is purged.

CATALEX and CATOWEX

The CAT command calls the user exit CATELEX (only valid for the fields of a file) or CATOWEX (only valid for the owners of an object).

Process Control Using +SYSDIC-FUNCTION

With old user exits, the parameter +SYSDIC-FUNCTION determines how a Predict function that called a user exit continues after the user exit processing is completed:

- If +SYSDIC-FUNCTION is not blank, the program continues as normal.
- If +SYSDIC-FUNCTION is blank, the Predict function issues a BACKOUT-TRANSACTION statement.

If +SYSDIC-FUNCTION is not reset and therefore no BACKOUT TRANSACTION issued, an END OF TRANSACTION statement **must not** be coded.

Using the Global Data Area

The following variables in the block ROOT of the global data area PGDA contain information which may be of use in user exits.

Variable	Meaning
+SYSDIC-ID(A32)	With the rename function: old ID of the object. With the catalog function: ID of the object (field or owner) currently being processed.
+SYSDIC-ISN(P10)	ISN of the record just processed.
+SYSDIC-OLD-ISN(P10)	With the copy function: ISN of the record that was copied.
+SYSDIC-FUNCTION(A2)	When this variable is passed by the Predict function, its first character indicates the function: **A** add **C** copy **M** modify **N** rename **P** purge **O** edit owners **W** edit description +SYSDIC-FUNCTION must be reset to issue a BACKOUT TRANSACTION statement. (See Process Control Using +SYSDIC-FUNCTION above).
+SYSDIC-USER(A250)	Reserved for the user.

Using the Global Data Area with Type-Dependent User Exits

- Old user exits must be cataloged with the Predict global data area PGDA using the following statements:

```
DEFINE DATA GLOBAL USING PGDA WITH ROOT
END-DEFINE
```

- With the exception of +SYSDIC-FUNCTION and +SYSDIC-USER, variables in the global data area **must not be changed**.
- +SYSDIC-ID, +SYSDIC-ISN and +SYSDIC-OLD-ISN deliver values that may be needed for user exits (see table above).

Activating Old User Exits

Old user exits are activated in the second Activate User Exits screen:
Main Menu > Defaults > Activate User Exits >User Exits (subprograms) > User Exits (programs).


```
10:36:25            *****  P R E D I C T  4.3.1   *****            2003-05-31
                          - Activate User Exits -

   ----------- User exits (programs) -----------   Modified 2003-05-31 at 10:32
                                                          by STK
                      ACM     PURGE     CAT
   Database .......... Y (Y/N)  N (Y/N)
   Elementary field .. N (Y/N)  N (Y/N)  N (Y/N)
   File ............... N (Y/N)  N (Y/N)
   Keyword ............ N (Y/N)  N (Y/N)
   Program ............ N (Y/N)  N (Y/N)
   File relation ..... N (Y/N)  N (Y/N)
   System ............. N (Y/N)  N (Y/N)
   User ............... N (Y/N)  N (Y/N)
   Verification ...... N (Y/N)  N (Y/N)
   Owner .............*                  N


For HELP enter '?' in the first input field. To leave enter '.'
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

Old user exits are activated by entering Y in the respective line and column. In the example above, ACMDAEX is activated; all other old user exits are deactivated.

Note:
Setting ACM/Verification to Y activates the user exits SUMPRDEX and ACMVEEX.

ACMxxEX - Add/Copy/Modify Objects

Activate with:

Y for any object type in the ACM column.

Applies to:

object types DA, EL, FI, KY, PR, RL, SY, US and VE.

Called where and when:

depending on the function and the object type.

- **Add, Copy and Modify**
  The user exit is executed before the editors (for extended description, owner list, sub-object list) are activated.
- **Add, Copy and Modify Field**
  The user exit for fields (ACMELEX) is executed before rippling takes place.
- **Rename**
  For all object types except files, the user exit is executed **after** the object has been renamed.
  With the function Rename/renumber file, the user exit is executed **before** the File is renamed and before the fields and related files are processed.

+SYSDIC-FUNCTION:

To issue a BACKOUT TRANSACTION statement, +SYSDIC-FUNCTION must be reset.

Additional remarks:

If an object list is edited and several .E(n) line commands have been entered, the appropriate user exit (ACMxxEX) is invoked for each object.
If a BACKOUT TRANSACTION is issued by the user exit, processing does not continue with the next object in the list, and Predict returns to the object list editor.

With the function Browse through fields of a file, a BACKOUT TRANSACTION in the user exit ACMELEX causes Predict to return to the Field Maintenance menu.

CATELEX - Catalog Fields

Activate with:

Y for field in the CAT column.

Applies to:

Fields

Called where and when:

After storing, updating and rippling (including delete rippling) have taken place and before deletion takes place.

+SYSDIC-FUNCTION:

To issue a BACKOUT TRANSACTION statement, +SYSDIC-FUNCTION must be reset.

Additional remarks:

CATELEX is invoked after the CAT command if a field in the field list has been modified, unless it has been modified using the .E editor line command. If the .E editor line command was used, the user exit ACMELEX is invoked instead.

Unless rejected by the user exit ACMELEX, the editor line command .E is executed and the dictionary is updated.

CATOWEX - Catalog Owners

Activate with:

Y , **B** or **U** for field in the CAT column.

- **Y**
  only CATOWEX is executed
- **B**
  CATOWEX and U-OW are executed
- **U**
  only U-OW is executed

Applies to:

Owners

Called where and when:

After an owner list has been edited and a CAT command has been issued, CATOWEX is executed before storing/updating or deletion takes place. It is invoked once for each owner in the list.

+SYSDIC-FUNCTION:

To issue a BACKOUT TRANSACTION statement, +SYSDIC-FUNCTION must be reset.

Additional remarks:

Unless rejected by the user exit ACMUSEX, the editor line command .E is executed and the dictionary is updated.

PURxxEX - Purge Function

Activate with:

Y for any object type in the PURGE column.

Applies to:

object types DA, EL, FI, KY, PR, RL, SY, US and VE.

Called where and when:

before the confirmation screen of the function purge appears.

+SYSDIC-FUNCTION:

To issue a BACKOUT TRANSACTION statement, +SYSDIC-FUNCTION must be reset.

Using Predict in Batch Mode

Most Predict functions can be used either online or in batch mode. Batch mode is useful

- when processing a large number of objects
- when redocumenting entire libraries
- when reports are created regularly, for example with the same restrictions and output options.

This section covers the following topics:

- Where Predict Commands are Described
- Batch Mode Options
- Batch Mode Input
- Condition Codes
- Natural Workfiles and Additional Reports in Batch Mode
- Examples for Batch Mode Input

Where Predict Commands are Described

- Most commands are described in the section Predict Commands in the **Predict Reference documentation**.
- For generation, incorporation and comparison functions:
  the valid keywords depend on the type of external object and are listed in the appropriate section of the respective sections of this documentation.
- For Coordinator functions:
  in the section Data Transfer Examples in the **Predict Coordinator documentation**.
- For conversion functions:
  in the section Converting Dictionary Data in Batch Mode of section **Conversion** in this documentation.

Keywords are also listed in the help text displayed by entering ? in the first input field of a particular function, for example Object ID for generation functions, or Retrieval type for retrieval functions. Command descriptions can easily be found using the index.

Batch Mode Options

This section contains:

- Overview of Possible Functions
- Using Batch Mode with Different Operating Systems

Overview of Possible Functions

The following functions can be performed in batch mode.

- **Predict Functions (Library SYSDIC)**
  - All generation functions. See the section Generation in the **External Objects in Predict documentation**.
  - Punch out previously generated code command: PUNCH or WRITE. See the section Predict Commands in the **Predict Reference documentation**.
  - All comparison functions. See the section Comparison in the **External Objects in Predict documentation**.
  - All incorporation functions except Incorporate DDM. See the section Incorporation in the **External**

Objects in Predict documentation.
    - Produce retrieval reports. See the sections Retrieval, Active Retrieval and LIST XREF for Natural of the **Predict Reference documentation**.
    - Execute the preprocessor. See the section Preprocessor in the **External Objects in Predict documentation**.
    - File Implementation. See the section File Implementation Plans in the **External Objects in Predict documentation**.
    - The following special functions. See the section Special Functions in this documentation.
        - Delete old sets
        - Reposition implementation data
        - Recover
        - Consistency of Predict
- **Conversion Functions (Library SYSDICCO)**
  Convert Predict data. See the section Conversion in this documentation.
- **Coordinator Functions (Library SYSDICBE)**
  Unload/load Predict data. See the **Predict Coordinator documentation**.
- **Metadata Administration Functions (Library SYSDICMA)**
  All metadata administration functions must be performed online.

Using Batch Mode with Different Operating Systems

Not all functions are available on all operating systems. This section lists the operating systems in which Predict can be run and lists any restrictions that apply to the particular operating system.

- **OS/390, VSE, BS2000/OSD, CMS**
  All Predict commands available in batch mode can be used.

  For some functions, work files or reports must be specified. The names of these files and the record format/length depend on the operating system and are given in the section Natural Workfiles and Additional Reports in Batch Mode.

- **UNIX**
  See the Natural documentation on installation and operation.

Batch Mode Input

This section contains:

- Prerequisites
- Recommendation
- Starting Batch Input
- Terminating Batch Input
- Entering Predict Commands in Batch Mode

Prerequisites

To execute Predict functions in batch mode, a Natural batch session must be started. The following Natural parameters should always be set:

```
MAXCL=0
CC=OFF
INTENS=1
LANG=1
MADIO=0
```

Recommendation

System Variable *DEVICE

When entering commands in batch mode, do not specify any profile parameters which influence the system variable *DEVICE.

For example: If you specify PC=ON, the system variable *DEVICE=PC will be set.

Starting Batch Input

Predict functions that can be performed in batch mode are located in libraries SYSDIC, SYSDICBE or SYSDICCO. See Overview of Possible Functions.

Logon to the required library with the command:

LOGON<library>

or, if you are working with Natural Security,

LOGON<library><user-ID><password>

Then code the following command to put Predict into the required state for processing batch commands:

MENU

Terminating Batch Input

There are two commands to terminate your Predict/Natural session. With both commands, global variables are reset, but values in the Predict hot object table remain.

FIN

The FIN command terminates the Predict/Natural session.

END or .

The END or . command will perform a logoff in Natural Security if no command is on the stack, and the input mode is set to forms mode (IM=F). The other parameters are not altered.

Entering Predict Commands in Batch Mode

Any number of Predict commands may be coded between MENU and FIN or END. The batch mode commands are the same as the Predict commands in online mode. The following rules apply:

Format

In OS/390, VSE and BS2000/OSD environments, input is read in fixed-length format of 80-character records. The last 8 characters are ignored.

Input Mode

Input mode is always delimiter mode (Natural profile parameter IM=D).

Continuation Records

If the parameters will not fit in one record, the last parameter in the record must be followed by a percent character (%). Parameters in the next line are then concatenated.

Assign Character

If entering commands in keyword form (see below), an assign character is used to separate the keyword and the parameter value. There must be no blanks between keyword, assign character and parameter value. Throughout the documentations, an equals sign (=) is used as the assign character.

You can change the assign character with the Natural parameter IA.

Input Delimiter

Input parameters may be separated by blanks or by the input delimiter character. Throughout the documentations, a comma is used as the delimiter character.

You can change the input delimiter with the Natural parameter ID.

Comments

Comments in batch jobs must be proceeded with an asterisk.

Entering Parameters in Line Following Command

With most commands the command parameters are entered in the same line as the command word. With the commands listed below, however, parameters are entered in the line(s) following the command.

- GENERATE ...
- COMPARE ... (with I option)
- INCORPORATE ...
- IMPLEMENT FILE
- SPECIAL ...

For example:

```
GENERATE COBOL
FILE-ID=TEST-FILE,REPLACE=Y,PREFIX=NEW-
```

Note:
To generate an external object using the generation defaults, code the file ID in the same line as the command. No other parameters can be specified.

For example:

```
GENERATE COBOL TEST-FILE
```

Entering Command Parameters in Keyword or Positional Form

Most commands can be entered in keyword form, positional form or a mixture of both. Positional form is obtained by omitting the keywords and the Natural assign character.

Example: Keyword Form

```
COMMAND KEY1=value1,KEY2=value2,KEY3=value3
```


or, if parameter 2 is omitted:

COMMAND KEY1=value1,KEY3=value3,KEY4=value4

Example: Positional Form

COMMAND value1,value2,value3

or, if parameter 2 is omitted:

COMMAND value1,,value3,value4

Mixing Keyword and Positional Form

The keyword and positional form of specifying command parameters can be mixed.

COMMAND KEY5=value5, value6,, value8

Condition Codes

Some functions (for example generation, migration, incorporation, active references and preprocessor functions) may return condition codes. A non-zero condition code can terminate processing without executing any subsequent commands.

Code	Meaning
0	Function completed normally.
4	Natural detected an error condition. This condition code does not originate in Predict.
8	Natural could not be started. This condition code does not originate in Predict.
101	Generation warnings. An error condition was detected during the generation of record layouts and/or the format buffer which gave rise to a warning message.
103	Invalid statement found. An invalid Predict preprocessor statement was found. The preprocessor continues.
106	Functions in the implementation plan were executed with errors or terminated abnormally, or the plan contains at least one generation task that already failed or could not be executed (status impossible).
107	External object has not been generated or no code has been saved/generated (the requested code has not been included in the output file). For example, the generation function fails or the Copy Code must be regenerated before it can be included by the preprocessor.
109	Storage overflow in extended buffer. Increase the Natural parameter ESIZE.
110	The data dictionary administrator has specified that all programs must be documented in Predict. The preprocessor detected that a Predict program object documenting the program does not exist. The program is not processed.
111	The preprocessor could not be started. The specified input file could not be found.
112	Program name not found by preprocessor.
115	An error occurred. Detailed message is written to report 0.
117	Invalid command, or function not allowed in batch mode (FILOB, PUNCH).

Evaluating Predict Condition Codes under BS2000/OSD

Predict condition codes can be evaluated under BS2000/OSD.
The following is required:

- The BS2000/OSD facility "Job Variables" must be installed.
- The SET parameter "&JV" in the Natural BS2000/OSD batch driver NATBS2 must be set.
- The job variable used for the condition code must be declared in the Predict batch job with the link name *NATB2JV.
  The necessary DECLARE statement looks as follows:

```
/DCLJV NATBJV,LINK=*NATB2JV
```

Note:
See the **Natural Operations documentation for Mainframes** for more information.

Natural Workfiles and Additional Reports in Batch Mode

For some batch commands, workfiles or additional reports must be defined.

This section contains:

- Workfile Required for Generation Commands
- Workfiles/Reports Required for Preprocessor Commands
- Workfiles/Reports Required for Incorporate COBOL Command
- Workfiles/Reports Required for Coordinator Commands

Workfile Required for Generation Commands

The following commands write output to Natural workfile 1:

- GENERATE (if Punch/output is set to Y), also if generation commands are called from an implementation plan.
- PUNCH / WRITE

			Record	
Description	**DD Name**	**BS2000/OSD Link Name**	**Format**	**Length**
Generated code	CMWKF01	W01	F or FB	80
			V or VB	min. 80

Note:
When generating ADAFDU definitions with function Generate ADACMP/ADAWAN definitions in batch mode with parameter Punch / Output set to Y for one of the following Adabas versions:

- any version starting with U,
- V3 and above or
- P2 and above,

we recommend setting the record format to V or VB with a record length of 250. Otherwise data may be truncated.

Workfiles/Reports Required for Preprocessor Commands

The PREPROCESS command requires the first three Natural workfiles:

Description	DD Name	BS2000/OSD Link Name	Record	
			Format	Length
Input dataset for Preprocessor	CMWKF01	W01	F or FB	80
Output dataset	CMWKF02	W02	F or FB	80
Temporary workfile	CMWKF03	W03	F or FB	91

The preprocessor produces three types of output on the first three Natural reports:

Description	DD Name	BS2000/OSD Link Name	Record	
			Format	Length
Report whether run was successful or not	CMPRT01	P01	F or FB	91
A list of preprocessor commands found	CMPRT02	P02	F or FB	133
A list of errors	CMPRT03	P03	F or FB	133

Workfiles/Reports Required for Incorporate COBOL Command

The INCORPORATE COBOL command requires the first two Natural workfiles:

Note:
If INPUT-TYPE=I, the copy code is read from the Natural command input stream and written to workfile 1.
If input type=W (default), the workfile must contain the copy code to be incorporated.

Description	DD Name	BS2000/OSD Link Name	Record	
			Format	Length
Input dataset	CMWKF01	W01	F, FB, V or VB	80
Temporary dataset	CMWKF02	W02	V or VB	at least 44

The Incorporate COBOL command produces three types of output on the first three Natural reports:

Description	DD Name	BS2000/OSD Link Name	Record	
			Format	Length
Input dataset (COBOL Copy Code)	CMPRT01	P01	F or FB	91
COBOL Copy Code with error messages	CMPRT02	P02		
Predict file description	CMPRT03	P03		

Workfiles/Reports Required for Coordinator Commands

The Coordinator uses the Natural workfiles 1, 5, 6, and 7. See the **Predict Coordinator documentation**.

Transfer Medium	DD Name	BS2000/OSD Link Name	Record	
			Format	Length
Migrate format	CMWKF01, CMWKF05, CMWKF06 or CMWKF07	W01, W05, W06 or W07	V or VB	Rec length at least 1796, block size at least 1800
ALF format	CMWKF01, CMWKF05, CMWKF06 or CMWKF07	W01, W05, W06 or W07	V or VB	Rec length at least 110, block size at least 114

Examples for Batch Mode Input

The examples throughout the documentation assume that the input delimiter character is set to a comma with the Natural parameter

ID=,

and the input assign character is set to equals with the Natural parameter

IA==

This section contains:

- Retrieval
- Generation
- Preprocessor
- Punch Output
- File Implementation
- Redocument Natural Program
- COBOL Copy Code Incorporation
- Conversion
- Coordinator
- LIST XREF, LIST XREF for 3GL

Retrieval

- List all child and parent object of all types of files starting with ’TEST’ and having keyword NEW.
  60 lines per page are printed, the report will start with a cover page, and for each object up to 10 abstract lines are displayed.
  The options will stay in effect for all subsequent retrieval functions.

```
LOGON SYSDIC
MENU
SET RESTRICTION KEY=NEW
SET OUTPUT PS=60,COVER=Y,ABSTRACT-CUR=10
CHILD FILE ALL TEST*
PARENT FILE ALL TEST*
FIN
```

  Note:
  `COMMENT-CUR` and `COMMENT-REL` can be used as synonyms for `ABSTRACT-CUR` and `ABSTRACT-REL` respectively, so that batch jobs created for earlier versions are still valid.

- Perform retrieval model EX (standard Predict explode) with output mode Cross reference on all Files starting with 'TEST'

```
LOGON SYSDIC
MENU
EXECMODEL FILE TEST* MODEL=EX,MODE=X
FIN
```

Note:
In batch mode, you must specify output mode X (Cross reference) or T (Structured list).

Generation

Generate ADACMP cards for the file TEST-FILE in database TEST-DA, save them in member TEST in library WANLIB and punch them to workfile 1.

```
LOGON SYSDIC
MENU
GENERATE,ADACMP
FILE-ID=TEST-FILE,DB=TEST-DA% <-- Continuation record follows
PUNCH=Y,MEM=TEST,LIB=WANLIB
```

Generate COBOL copy code for file TEST-FILE using default options.

```
GENERATE,COBOL,TEST-FILE
FIN
```

Preprocessor

Process the COBOL source program on workfile 1 with the preprocessor and write XRef data under member name COBTEST in library RBN.

```
LOGON SYSDIC
MENU
PREPROCESS,COBOL,COBTEST,RBN
FIN
```

Punch Output

Punch the previously generated copy code member TEST from library WANLIB to workfile 1.

```
LOGON SYSDIC
MENU
PUNCH TEST,WANLIB
FIN
```

File Implementation

Execute and then display the implementation plan IMP-TEST.

```
LOGON SYSDIC
MENU
IMPLEMENT FILE
FUNCTION=X,PLAN-ID=IMP-TEST
IMPLEMENT FILE
D,IMP-TEST
FIN
```


Redocument Natural Program

Create Predict program objects for all members of the Natural library NATTEST and replace existing program objects.

Construct the program IDs from the prefix NAT-, the library name and the member name.

Put all comment lines from the source into the extended description.

Assign to all redocumented program objects the owner NEWREDOC and the keyword NEW.

Link all redocumented programs to the system NATTEST-LIBRARY. Use the library structure TREE to determine the programs used.

```
LOGON SYSDIC
MENU
REDOCUMENT PROGRAM
PROCESS-OPT=R,LIBRARY=NATTEST,STRUCTURE=TREE,DESCRIPTION=A%
DEFAULT-OWNER=NEWREDOC,DEFAULT-KEY1=NEW%
SYSTEM=NATTEST-LIBRARY,PREFIX=NAT-
FIN
```

COBOL Copy Code Incorporation

Create a Predict sequential file object from the data definitions in the COBOL copy code member on workfile 1 with prefix PRD-, decimal character set to comma, literal delimiter set to single quote. An existing Predict file with the same name is replaced.

```
LOGON SYSDIC
MENU
INCORPORATE,COBOL
PREFIX=PRD-,DECIMAL-CHAR=C,DELIMITER=',STORE=Y,%
REPLACE=Y,INPUT-TYPE=W
FIN
```

Create a Predict sequential file object from the data definitions in the COBOL Copy Code member with prefix PRD-, decimal character set to comma, literal delimiter set to single quote. An existing Predict file with the same name is replaced.

```
LOGON SYSDIC
MENU
INCORPORATE,COBOL
PREFIX=PRD-,DECIMAL-CHAR=C,DELIMITER=',STORE=Y,%
REPLACE=Y,INPUT-TYPE=I
000100 01 FI-S-INCCOB-1
000110    02 PE-1 OCCURS 30 TIMES.
000120       03 PE-EL-1 PIC XX.
000130       03 PE-EL-2 PIC X(6).
000140 01 FI-S-INCCOB-2
000150    02 PERSON.
000160       03 CHR-NAME PIC X(32).
000170       03 LST-NAME PIC X(32).
000170    02 DUTY PIC X(25) OCCURS 5.
END
FIN
```


Conversion

Convert the data on the Predict system file from version 4.1 to version 4.2 format.

Convert the Predict objects of type report starting with "REP" and with keyword OLD to objects of type program.

Convert the Predict objects of type module with the prefix MOD* and with keyword OLD to objects of type program.

```
LOGON SYSDICCO
MENU
CONVERT VERSION42
CONVERT REPORT
REP*,OLD,*,N
CONVERT MODULE
MOD*,*,OLD,*,N
FIN
```

Coordinator

See the Predict Coordinator documentation for more information.

LIST XREF, LIST XREF for 3GL

```
LOGON SYSDIC
MENU
ACTIVE XREF *SYSPLI* PROGRAM USING VIEW
```


Conversion - Converting Predict Data From Previous Versions

Conversion - Converting Predict Data From Previous Versions

To use Predict Version 4.3 to maintain data created with previous Predict versions, the data **must** be converted to Version 4.3 format.

The conversion functions are used to convert dictionary data created with Predict Version **4.2**.
These functions are described in this section.

Note:
You can also use the Coordinator to transfer data from a 4.2 to a 4.3 environment. However, this procedure takes longer and offers no advantages.

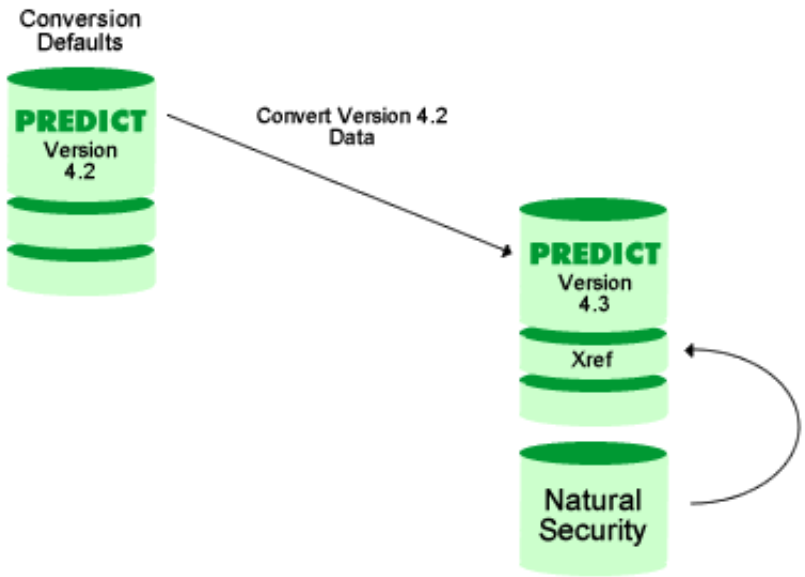
This section covers the following topics:

- General Information
- Calling Conversion Functions
- Conversion Defaults
- Version 4.2 Data
- Add Natural Security XRef Data
- Convert XRef Data
- Converting Dictionary Data in Batch Mode

General Information

Overview of Conversion Functions

This diagram shows you the scope of the Conversion Utility and the order in which the individual functions must be performed.

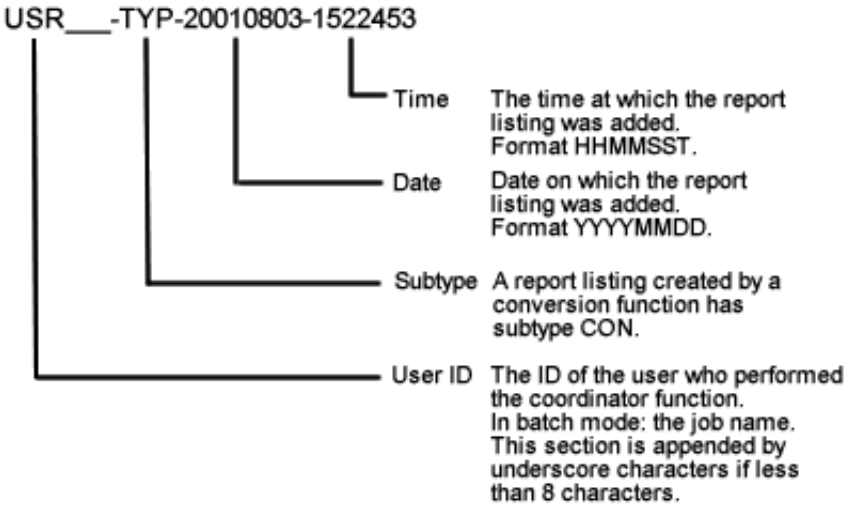


Logging Conversion Functions

With some conversion functions, an object of type report listing is created in your new Predict file. This report listing contains the following:

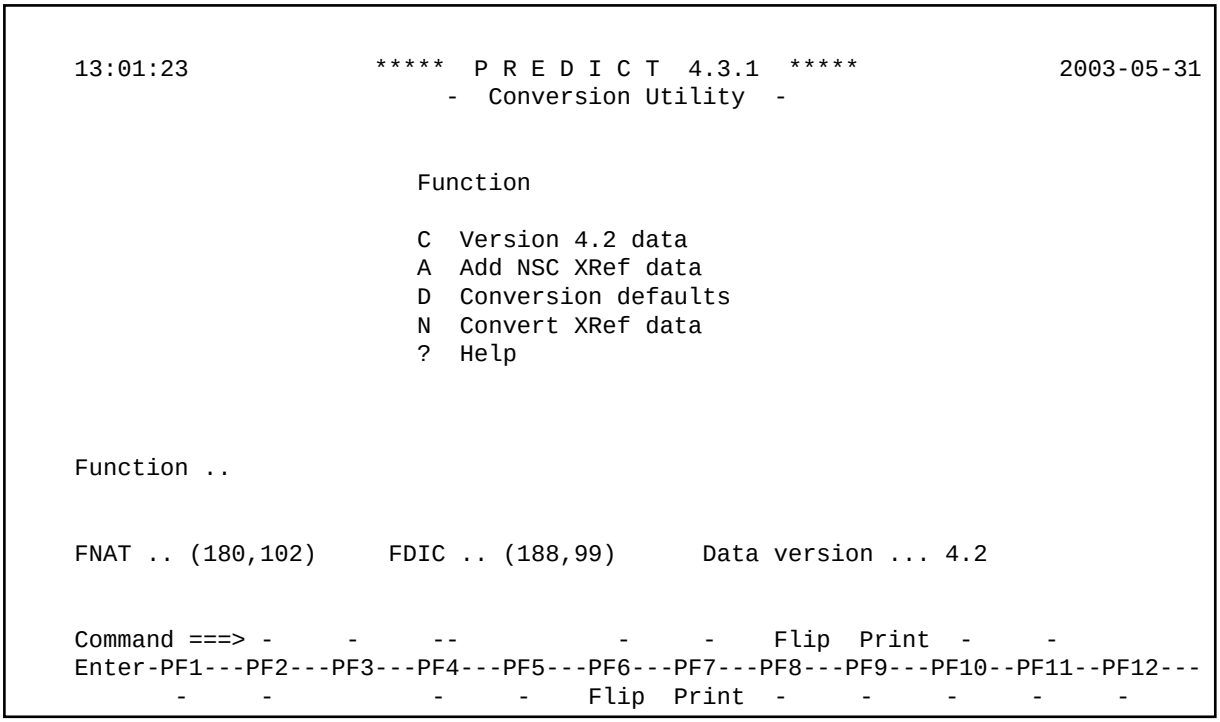
- an overview of how the function ran
- all error messages that occurred during conversion.

The ID of this object is composed as follows:



Calling Conversion Functions

The Conversion Utility Menu is called with the command MENU at the NEXT prompt in the Natural Library SYSDICCO.



This screen contains all conversion functions and provides the following additional information:

- Adabas DBID and FNR of both the Predict system file (FDIC) and the Natural system file (FNAT)
- the format of data in the Predict dictionary.

Conversion Defaults

If you defined user defined metadata (user defined entities (UDEs), retrieval models and associations) in an earlier version of Predict and the metadata is now reserved in this version of Predict, you can use the function Conversion Defaults to specify how the existing metadata is to be renamed in the new version.

If you try to execute the function Version 4.2 data and a UDE name, UDE code, retrieval model or association is contained in the data to be converted, an error message will be given.

The following names and codes are reserved in Version 4.3.

Reserved Metadata for Conversion Defaults

Reserved Object Type	
Names	**Codes**
OS4-FILE	O4

Reserved Association		
Object Type	**Active Code**	**Passive Code**
(SY->SY)	LL	LL
(SY->PR)	SO	SO

Calling the Function

The function Conversion Defaults is called with function code D in the Conversion Utility Menu.

This function can only be executed online.

Example

In version 4.2 you defined a UDE with object type name OS4-FILE and Code OF. In version 4.3, OS4-FILE is the name of a predefined object type and is not permitted as an UDE.

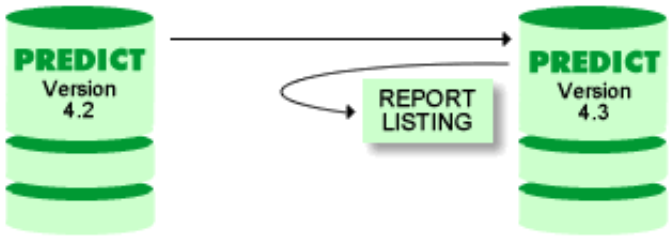
If you call the function Conversion defaults, the following screen appears:

```
13:41:46              *****  P R E D I C T  4.3.1  *****            2003-05-31
                           - Conversion Defaults -
                                                      Added 2003-05-31 at 13:03
                                                         by GER
 Object type code            Object type name              Retrieval model
                               OS4-FILE .......
```

The function lists all UDEs which are in conflict with the new version. Because code OF is not reserved, this does not appear in the list and does not need to be changed. You do, however, have to change the object type name OS4-FILE before you can convert your data from version 4.2. Enter under Object type name a new name (which is not reserved) for your UDE and press ENTER.

Version 4.2 Data

With the function Version 4.2 data, all data from your 4.2 environment are transferred to your new 4.3 environment.



Calling the Function

The function Version 4.2 data is called with function code C in the Conversion Utility Menu. It can be executed both online and in batch mode (see Converting Version 4.2 Data in Batch Mode).

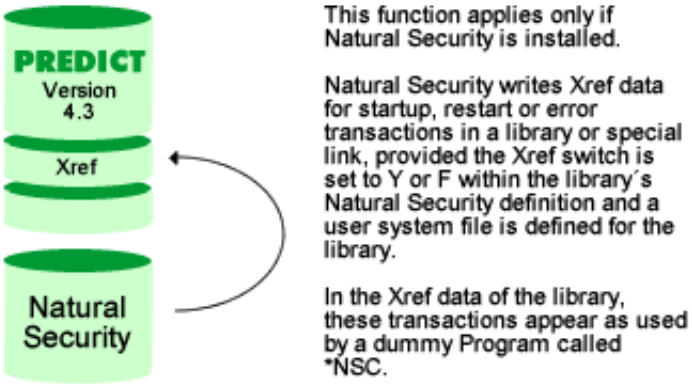
The function Version 4.2 data can only be executed once. It is logged by an object of type report listing. See Logging Conversion Functions.

If a conversion function should be interrupted for any reason, it can be restarted at any time. Information on the file version and the objects already converted is written to the Report Listing mentioned above.

The dictionary data cannot be modified until the entire file has been converted to the new version.

Note:
The Coordinator FDIC file must be empty during the conversion because data in the Coordinator file are not converted. For details see the Predict Coordinator documentation.

Add Natural Security XRef Data



The function creates the XRef data for the first time. When both products have been installed, this XRef data will be maintained automatically whenever the Natural Security definition of a library or special link is changed.

The libraries for which XRef data have been written are listed.

If the function is repeated later, the Natural Security XRef data will always be set to a state corresponding to the current Natural Security definitions of startup, restart or error transactions.

Calling the Function

This function should be executed **once** in one of these cases:

- after the installation of Predict 4.3 if Natural Security is already installed,
- after the installation of Natural Security if Predict 4.3 is already installed.

Call the function with code A in the Conversion Utility Menu. The function can also be executed in batch mode. See Adding Natural Security XRef Data in Batch Mode.

Convert XRef Data

XRef data format is converted automatically using the function Version 4.2 data. If for some reason there are XRef data in old format layout on your FDIC file, use this function to convert this data to current format layout.

Calling the Function

You can use this function any time.

Call the function with code N in the Conversion Utility Menu. The function can also be executed in batch mode. See Converting XRef Data in Batch Mode.

Converting Dictionary Data in Batch Mode

Converting Version 4.2 Data in Batch Mode

```
LOGON SYSDICCO
MENU
CONVERT VERSION43
FIN
```

Adding Natural Security XRef Data in Batch Mode

```
LOGON SYSDICCO
MENU
ADD XREF
FIN
```

Converting XRef Data in Batch Mode

```
LOGON SYSDICCO
MENU
CONVERT XREF
FIN
```


Metadata Administration

Predict's metadata structure can be changed with the functions of the Metadata Administration.

- The number of object types and association types which can be added to Predict's metadata structure is limited. It can be any value between 1,000,000 to 4,999,999. Assign a unique number from this value range to each object type, association type, object type attribute and association type attribute.
- Defaults for predefined object types can be defined.
- Retrieval models can be defined.

New object and association types are referred to as User-Defined Entities (UDEs) or User-Defined Object Types, User-Defined Associations respectively.

Note:
Metadata Administration functions cannot be performed in batch mode.

This section covers the following topics:

- Starting and Quitting the Metadata Administration
- Object Type Administration
- Association Administration
- Retrieval Model Administration
- Defaults Administration

Starting and Quitting the Metadata Administration

To display the Metadata Administration menu, enter the following command at the NEXT prompt:

LOGON SYSDICMA
MENU

To quit the Metadata Administration, enter a period (.).

```
09:26:43             *****  P R E D I C T  4.3.1  *****             2003-05-31
                        - Metadata Administration -




                       Meta objects

                       O  Object type
                       A  Association
                       R  Retrieval model
                       D  Defaults



Meta object ..........




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       -     -    Canc   -     -     -    Print  -     -    FLIP   -    Menu
```

Note:
For details of the internal data structure of user-defined object types and associations, and the way they are handled, see the file descriptions whose names start with "SYSDIC-UDE".

Object Type Administration

The standard Predict metadata structure contains the following object types:
database, dataspace, extract, field, file, file relation, interface, keyword, library structure, method, network, node, packagelist, program, property, report listing, server, storagespace, system, trigger, user (owner), verification and virtual machine.

In addition to these predefined types, user-defined object types can be added to the Predict metadata structure.

User-defined object types are created and maintained with functions of the Object Type Administration Menu. Some administration functions of this menu can also be applied to predefined object types.

The Object Type Administration Menu is called by entering code O in the Metadata Administration main menu.

```
09:55:00             *****  P R E D I C T  4.3.1  *****             2003-05-31
                   - Object Type Administration Menu -


                      Function

                      A  Add an object type
                      D  Display object type
                      S  Select object type
                      M  Modify object type
                      N  Rename object type
                      P  Purge object type
                      T  Modify attribute names and numbers
                      X  Cross reference object type

Function ..............

Object type code ....*



Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       -     -    Stop   -     -    Flip  Print  -     -     -     -    Main
```

Parameters	
Function	Single-character code to select one of the functions available. The functions are described in the following sections: • Add / Modify Object Type - Codes A, M • Display Object Type - Code D • Select an Object Type - Code S • Rename an Object Type - Code N • Purge an Object Type - Code P • Modify Attribute Number - Code T • Cross Reference Object Type - Code X
Object type code	Identifies the object type in Predict menus or direct commands, such as DA (database) or FI (file).

Add / Modify Object Type - Codes A, M

The Add and Modify functions use the same screens.

When an object type is added, a default extended description and help texts for retrieval and maintenance are generated. The default extended description can be changed using the function Modify Defaults which is called from the Function Main Menu with code D.

This section contains:

- Rules For Modifying Object Types
- First Screen
- Defining Attributes
- Defining Edit Masks
- Defining Verifications
- Defining the Header Layout

Rules For Modifying Object Types

When an object type is modified the following rules apply:

- If an attribute is deleted, the attribute will be deleted in all existing objects of the object type.
- If an attribute is added or an attribute name is changed, the change will be applied to all existing objects of this type.
- The type of an attribute cannot be changed.
- For an attribute, the total number of positions may only be increased.
- If the number of positions is increased and the attribute has a verification or edit mask, the verification or edit mask will be deleted. This rule does not apply to the data types D (date) and T (time).
- If the verification or edit mask of an attribute is changed, the corresponding object data remain unchanged.

First Screen

```
13:54:15            *****  P R E D I C T  4.3.1  *****             2003-05-31
                    - Modify object type definition -

Object type code ... CE                           Modified 2003-05-31 at 10:46
                                                        by MSZ

Object type attributes                 Default association
  Internal code ... 1000005              Parent association .*
  Name ............ CHNG-ENHANCEMENT      Child association ..* DA
  Title ........... Chng-enhancement
  Object type no .* 1012000

Edit owner ..............* D   Disallowed
Edit description ........* F   Force
Check description ......... Y   (Y,N)
Object ID length .......... 32  (1-32)
Disallowed characters .....
Natural naming convention . N   (Y,N)

Abstract   * Zoom: N
  Object-type to document
  change enhancements for pro-
Screen number .... 1  of 1  (H=Header)                 Free attributes:  70
```


Note:
The parameters Internal code, Name and Title cannot be changed using the Modify function. To modify the Name or Title, use the function Rename object type.

Parameters	
Object type attributes	
Internal code	This internal code is assigned automatically and cannot be changed. It is used by Predict to administrate object types. This field is empty when an object type is added.
Name	Name of the object type. A name • must have at least three characters and start with a letter (a-z, A-Z). • must not contain blanks. • must be unique, and must not be an abbreviated form of an existing object type (for example, Key is not permitted because it could be confused with the predefined object type Keyword). • must not start with an object type that already exists (for example, Filecard is not permitted).
Title	String that is used in Predict menus and output screens for the respective object type. Can be up to 17 characters long and contain numbers, special characters and blanks. If no title is specified, the object type name with the first character capitalized will be taken as title.
Object type no	Number of the object type. When an object type is created, it is assigned a number between 1,000,000 and 4,999,999. In the Modify function, this number can be modified. You can use the asterisk notation to select a number which has not yet been assigned to an object type, an association type or an attribute.
Default association **These parameters can be changed for user-defined object types.**	
Parent association / Child association	The associations specified apply as default values in Predict functions for the in object type parameter (for example, in system parameter in maintenance screens of the user-defined object type change enhancement). Only user-defined association types can be specified.

Note:
The following parameters can be modified for both predefined and user-defined object types.

Parameters	
Edit owner	Determines the handling of the owner list: • **F** Force: Only owners which are assigned to a user can be specified. At least one owner must be specified. • **A** Allow: the owner list can be edited. • **D** Disallow: the owner list must not be edited. **Note:** For the object type User, the parameter Edit owner can only be set to Disallow if it has not been set to Force for any other object type. If the parameter Edit owner is set to Force for any object type, Predict does not check whether all objects of this type have an owner in their owner lists.
Edit description	Determines the handling of the extended description: • **F** Force: at least three lines must be specified. • **A** Allow: the extended description can be modified. • **D** Disallow: the extended description must not be modified.
Check description	Parts of an extended description skeleton can be protected (see Extended Description Skeleton). If this parameter is set to Y, the protected text must be contained in the extended description.
Object ID length	Enter a value from 1 - 32. The value entered here is the maximum length of an ID for objects of this type.
Disallowed characters	Up to 20 characters that are not allowed in IDs for objects of this type can be entered here. **Note:** Object ID length and disallowed characters can be checked with Special Function Consistency of Predict > Check naming conventions. See Check Naming Conventions.
Natural naming convention	**N** Default. Predict naming conventions apply. See Naming Conventions in the **Predefined Object Types in Predict documentation**. **Y** Natural naming conventions apply to object IDs. These conventions support use of double-byte character sets (DBCS). See your **Natural documentation** for further information.

Note:
The parameters above can be modified for existing object types, but checks are only performed on newly created objects or objects imported with the Predict Coordinator.

For example:
You can change the parameter Object ID length for object type database to 30 even if databases exist with an ID of 32 characters.

Parameters	
Abstract	An abstract of up to 16 lines can be specified. Each line can contain up to 30 characters.
Screen number	Up to five maintenance screens can be defined for each object type. When a number between 1 and 5 is entered, the corresponding screen for attribute definition is selected. Only an existing screen or the next free screen can be selected. If zero is entered, the Metadata Administration menu appears. If 'H' is entered, the header is displayed.
Free Attributes	The remaining number of free attributes is displayed - a maximum of 80 for each object type.

Defining Attributes

Up to 80 attributes can be defined for each object type - a maximum of 40 in one maintenance screen. The maximum numbers of attributes for the individual attribute types are:

alphanumeric: 40; numeric: 40; date, time, logical, and literal 20 each.

```
13:04:08            *****  P R E D I C T  4.3.1  *****              2003-05-31
                    - Modify object type definition -

Name ........... CHNG-ENHANCEMENT                  Modified 2003-05-31 at 10:46
Screenname .... Attributes                               by MSZ
Screennumber .. 1
                                                    Positions of
                                                     Title    Value
   Attribute Name                      Fmt Len Case  Li/Col   Li/Col  Lit Ext Ver
                                                                        *
1  Customer                             A   58   L   1   1    1   20
2  Title                                A   58   L   2   1    2   20
3  Version                              A    6   U   4   1    4   20           T
4  Subproduct                           A   20   L   3   1    3   20
5  Status                               A    1   U   6   1    6   20           R
6  Test                                 A    1   U   8   1    8   20           S
7  Closing-date                         N  8.0       7   1    7   20
8  CE-Nr                                N  6.0       5   1    5   20
9  time                                 T            10  1    10  20
10 date                                 D            12  1    12  20

 Test screen layout: N                       More:   Attributes: N
```

Parameters	
Screen name	Name of the corresponding maintenance screen. If several maintenance screens are existing, the screen names are displayed in a selection window. Up to 17 characters are allowed. Default value for the first screen name is Attributes.
Attributes	Names of the attributes which are to be displayed in the corresponding maintenance screen for this object type. Up to 32 characters are allowed.

Fmt	Format of the attribute: **A** alphanumeric **D** date **L** logical **N** numeric **T** time **X** literal
Len	The length of an attribute value depends on the format: **A** 1 to 78 **N** 1 to 27 places before the decimal point, and 0 to 7 places after the decimal point, where the total number of places must not exceed 27
Case	This field is only valid for alphanumeric attributes (type A). **U** The attribute value is converted to capital letters. **L** No conversion to capital letters.
Position of Title	The position refers to a window of 15 lines and 78 columns: • line ($1 \leq$ line ≤ 15) • column ($1 \leq$ column ≤ 78) The attribute name is not displayed if both values are 0. The position must be defined for type X (literal).
Position of Value	The position refers to a window of 15 lines and 78 columns: • line ($1 \leq$ line ≤ 15) • column ($1 \leq$ column ≤ 78) For type X (literal), this field must remain empty.
Lit	SMR

Ext	Extended attribute description. Subsequent screens for the definition or modification of edit masks and verifications for the attribute are displayed. **E** edit mask (only for types D, L and T) **T** table of verification values **R** range of verification values **S** external verification (special check) For type X (literal) neither edit masks nor verifications can be defined; for type L (logical) no verifications can be defined. The following default values apply to edit masks: **L** F/T **D** YYYY-MM-DD **T** HH:II:SS Changing the verification type (T, R or S) is possible. In this case, the old verification is deleted. Before deletion, a warning is displayed. The corresponding object data remain unchanged. **Note:** Entries in the Ext fields can only be processed for one attribute. If values for several attributes are entered, an error message is displayed.
Ver	An existing verification type (T, R or S) is displayed for the attribute. **Note:** The verification is not performed by the link editor, if the attribute name is not defined in the header layout (as described in the section Defining the Header Layout).
Test screen layout	**Y** The maintenance screen is displayed to test the screen layout.
More Attributes	**Y** Each definition screen can contain ten attributes. If more attributes are to be defined (up to 40), this field is used to display a subsequent screen.

Defining Edit Masks

For the attribute types L, D and T (logical, date and time) edit masks are defined. Enter E in the field Ext to display a selection window where you can select an appropriate edit mask for the attribute. The following values are possible:

Attribute Type	Value
L (Logical)	FALSE/TRUE
	OFF/ON
	NO/YES
	N/Y
	* F/T
D (Date)	DD/MM/YY
	DD.MM.YY
	YY-MM-DD
	DD/MM/YYYY
	DD.MM.YYYY
	* YYYY-MM-DD
	MM/DD/YYYY
	YYYYMMDD
T (Time)	* HH:II:SS
	HH:II:SS:T
	HH:II:SS AP
	HH:II:SS:T AP

Asterisks indicate the active edit mask for the attributes.

Defining Verifications

There are three methods of performing plausibility checks for object attributes:

- using a Table of Values
- using a Range of Values
- using an External Routine.

Defining a Table of Values

Enter T in the field Ext to define a table of values.


```
13:09:01            *****  P R E D I C T  4.3.1  *****             2003-05-31
                         - Add a Table Definition -

Name ............. CHNG-ENHANCEMENT                Modified 2003-05-31 at 10:46
Attribute name .. datum                                 by MSZ

EDIT MASK ....... YYYY-MM-DD

Required ........ N (Y/N)
 1 1993-01-01   2 1994-01-01
 3 1995-01-01   4
 5              6
 7              8
 9             10
11             12
13             14
15             16
17             18
19             20
21             22
23             24
```

Parameters	
EDIT MASK	Active edit mask for the attribute (only for types D, L and T).
Required	**Y** Attributes for which a verification rule with a table of values has been defined must be filled in maintenance screens (mandatory fields). Attributes with a verification rule that allows blank values need not be filled explicitly. Use the following 24 fields to define the verification values.

Defining a Range of Values

Enter R in the field Ext to define verifications with ranges of values.

```
13:13:48            *****  P R E D I C T  4.3.1  *****             2003-05-31
                         - Add a Range Definition -

Name ............. CHNG-ENHANCEMENT                Modified 2003-05-31 at 10:46
Attribute name .. date                                    by MSZ

EDIT MASK ....... YYYY-MM-DD
Required ........ N (Y/N)

Range of values *:

 GT 1994-01-01                      AND  LT DATE____

                                    OR_

 GT 1993-01-01                      AND  LE 1993-03-31

                                    ___

 __ ________                        ___  __ ________

                                                      Error code: ____
```


Parameters	
EDIT MASK	Active edit mask for the attribute (only for types D, L and T).
Required (Y/N)	**Y** Attributes for which a verification rule with a range of values has been defined must be filled in maintenance screens (mandatory fields). Attributes with a verification rule that allows blank values need not be filled explicitly.
Range of values	In the following fields, you can define ranges of values according to the following syntax: AA Value_1 BBB AA Value_2 CCC AA Value_3 BBB AA Value_4 CCC AA Value_5 BBB AA Value_6 For any defined Value_X, **AA** is a mandatory field with one of the following possible values: **EQ** (equal), **NE** (not equal), **LT** (less than), **LE** (less than or equal), **GT** (greater than), **GE** (greater than or equal). If you enter an asterisk, the possible values are displayed in a selection window. Value_X are the verification values. Up to six verification values can be defined. For type **A** (alphanumeric), the length is restricted to 32 - independent of the length of the attribute itself. For type **N** (numeric), the following applies: a maximum of 20 digits before and 7 digits after the decimal point can be entered. If DATE is entered for type **D** (date), the verification value is the current date. **BBB** is one of the possible values AND or OR which combines the two elementary conditions of one line in a boolean expression. **CCC** combines the boolean expressions defined in each line with the operand AND or OR. The logical expressions are always processed line by line. **Note:** The logical consistency of the verification rule is not checked.
Error code	User-defined, verification-related number of the error message which appears if an invalid attribute value has been entered. Range of values: 9000 - 9999.

Note:
With attributes of type A (alphanumeric) only the number of positions defined in the verification rule are considered.

Using an External Routine

Enter S in the field Ext to define that an external verification rule is to be called.


```
13:34:17            *****  P R E D I C T  4.3.1  *****             2003-05-31
                          - Add a Special Check -

Name ............ CHNG-ENHANCEMENT                Modified 2003-05-31 at 10:46
Attribute name .. Test                                    by MSZ


Subprogram ... N-TEST__


Attribute names *
 Customer_____________________________
 Version______________________________
 _____________________________________
```

Parameters	
Subprogram	Name of the subprogram which is called to verify the attribute value.
Attributes	In the following fields, you can define up to three object attributes, which are defined in one maintenance screen, as input parameters for the subprogram. If you enter an asterisk, the names of the object attributes of this maintenance screen are displayed in a selection window. Attributes of type X (literal) cannot be defined as input parameters for the subprogram. The values of these attributes are passed to the defined subprogram in alphanumeric format (length 78) as follows: • type A: left aligned with leading blanks. If Case = U, letters are converted to capital letters. • types D and T: right aligned according to their internal representation. • type L: left aligned; "1" for TRUE and "0" for FALSE. • type N: right aligned without decimal point. All places after the decimal point (according to the attribute definition) are passed. The number of places after the decimal point is passed by an additional parameter. With negative values, the sign is passed as well. Leading zeros are suppressed. The following CALLNAT statement is generated: **CALLNAT 'subprogram'** **PAR1 PAR2 PAR3** **RETURN-CODE RETURN-TEXT1 RETURN-TEXT2** **PAR1 PAR2 PAR3**: values of the attributes (format A78); **RETURN-CODE**: number of a Predict error message (format I2); RETURN-TEXT1 and RETURN-TEXT2: texts as parameters for this error message (format A34). In the subprogram, attribute values of the types D, N, and T can be converted using the Natural function VAL. The verification rules are processed screen by screen. Independent of the sequential numbers of the attributes to be verified, the verification rules with external verification routines are processed last - they are only processed after all verification rules with tables of values and ranges of values have been processed.

Defining the Header Layout

```
13:43:42            *****  P R E D I C T  4.3.1  *****             2003-05-31
                          - Modify Header Layout -

Object type ..... CHNG-ENHANCEMENT



        Number of:  Column   Header                              Length  Format
        Attr. Scr.
 1.     2      1      1      Title                                  19   A   58
 2.     7      1      21     C-date                                 9    N  8.0
 3.
 4.

Layout:                                ....+....1....+....2....+....3....+....4

      Chng-enhancement                 Title               C-date
      -------------------------------  ------------------- ---------
```

This screen appears after object attributes have been defined. Here you can define a table layout to be used by the maintenance functions List, Select and Link children.

The following rules apply:

- A maximum of four rules can be specified.
- The total length of all attribute values must not exceed 30 characters (including blanks to separate the columns).
- The text of the header lines can be modified as desired.
- The display length can only be modified for alphanumeric attributes.

Parameters	
Number of Attr.	Number of the attribute in the definition screen.
Number of Scr.	Number of the definition screen of this attribute.
Column	Position in the table screen.
Header	Text of the header line in the table screen.
Length	Length with which the value is displayed. **Note:** If the display length is less than the length of the header text, the attribute value is displayed with the length of the header text.

Display Object Type - Code D

Displays object type definitions in an output format similar to the screens used for adding or modifying object type definitions. For a description of all output fields, see Add / Modify Object Type.

Select an Object Type - Code S

A list of predefined and user-defined object types is displayed for selection. Press ENTER to display the next page.

```
13:36:14            *****  P R E D I C T  4.3.1  *****           2003-05-31
                      - Select Object Type Definition -

Mark Code  Title             Object type


      CH   Child             CHILD
      CE   Chng-enhancement   CHNG-ENHANCEMENT
      DA   Database          DATABASE                          predefined
      DK   Dokument          DOKUMENT
      EL   Elementary field  ELEMENT                           predefined
      EY   Elementary field  EMPTY
      FI   File              FILE                              predefined
      UP   File              FILL
      HI   Verif             HIT
      HO   Keyword           BWG1
      JB   Owner             JOB
      KY   Keyword           KEYWORD                           predefined
      NO   Node              NODE                              predefined



Hit the corresponding PF-KEY to STOP, Enter 'X' to ABORT or 'T' for TOP : _
```

Rename an Object Type - Code N

```
09:44:42            *****  P R E D I C T  4.3.1  *****             2003-05-31
                    - Rename Object Type Definition -

Name ............ CHNG-ENHANCEMENT               Modified 2003-05-31 at 07:32
Object type ..... CE                                    by ARH

Object type attributes
  Intern code.... 1000005
  Title.......... Chng-enhancement

New attributes
  New name ......... CHNG-ENHANCEMENT   ( '.' to return to menu )
  New object type .. CE
  New title......... Chng-enhancement


          ----------------- ATTENTION ------------------
          Contents of help-texts and default description
          will not be adapted.
```

The name of the object type, the external code, and the title can be changed.

Purge an Object Type - Code P

Deletes a user-defined object type and all data dictionary objects of this type. The following rules apply:

- Object types that are linked to another object type by an association cannot be purged.
- The default extended description and the help texts for retrieval and maintenance are deleted.

Modify Attribute Number - Code T

```
13:22:39             *****  P R E D I C T  4.3.1  *****             2003-05-31
                        - Modify attribute number -

Name ........ Parent_of_H1                          Modified 2003-05-31 at 13:30
                                                          by HEB

Cnt  Number   Attribute name                   title
---  *------  -------------------------------- --------------------------------
 1   2000082  SCREEN_1                         Screen 1
 2   2000077  BLAB01                           Blab01
 3   2000078  BLAB02                           Blab02
 4   2000079  BLAB03                           Blab03
 5   2000080  BLAB04                           Blab04
 6   2000081  BLAB05                           Blab05
 7   2000130  SCREEN_2                         Screen 2
 8   2000001  ALPHA_2-1                        Alpha_2-1
 9   2000003  ALPHA_2-10                       Alpha_2-10
10   2000131  SCREEN_3                         Screen 3
11   2000002  ALPHA_3-1                        Alpha_3-1
12   2000005  ALPHA_3-12                       Alpha_3-12
13   2000132  SCREEN_4                         Screen 4
14   2000010  SS                               ss
```

Number of the attribute

When an attribute is added, it is assigned a number between 1,000,000 and 4,999,999. This value can be modified. You can use the asterisk notation to select a number which has not yet been assigned to an object type, to an association or to an attribute.

Cross Reference Object Type - Code X

Displays the following information on the associations of the object type:

- all object types to which the object type is linked
- all retrieval models defined for the object type
- all retrieval models that report on links to the object type.

Association Administration

Since Predict Version 4.2 defining more than just one association between two object types is possible. To ensure unique identification of these associations an identifier, the Association code is necessary.

The terms active and passive association describe the different views of parent and child object on the same association. The term active association describes the view of the child object, while the term passive association describes the view of the parent object.

This section covers the following topcs:

- Predefined Associations
- Association Administration Menu
- Add / Modify Association - Code A,M
- Display Association - Code D
- Purge Association - Code P
- Select Association - Code S
- Modify Attribute Number - Code T
- Cross Reference of an Association - Code X

Predefined Associations

The following associations are predefined in Predict:

Parent	Association	Child	Code
Database	Contains FI	File	FI
	Contains DC	Dataspace	DC
Dataspace	Contains FI	File	FI
Field	Is verified by VE	Verification	VE
	Triggered by TR	Trigger	TR
File	Has Fields	Field	EL
	Has TR	Trigger	TR
Interface	Contains MD	Method	MD
	Contains PY	Property	PY
Keyword	Decomp. into KY	Keyword	KY
Library Structure	Contains SY	System	SY
Node	Contains SV	Server	SV
Network	Contains NO	Node	NO
	Uses VM	Virtual Machine	VM
Packagelist	Uses PR	Program	PR
Program	Uses FI concept.	File	FI
	Uses PR concept.	Program	PR
	Defines IE	Interface	IE
	Invokes MD	Method	MD
	Input FI	File	IN
	Returns FI	File	RE
Report Listing	Uses ET	Extract	ET
Server	Uses PR	Program	PR
System	Uses PR concept.	Program	PR
	Uses PG	Packagelist	PG
	Has subappl. SY	System	CS
	Has component PR	Program	CP
	Has component VE	Verification	CV
	Has component FI	File	CF
	Has library SY	System	LI
Virtual Machine	Contains DA	Database	DA

Child	Association	Parent	Code
Database	Belongs to VM	Virtual Machine	VM
Dataspace	Located in DA	Database	DA
Field	Belongs to FI	File	FI
Extract	Contained in RT	Report Listing	RT
File	Contained in DA	Database	DA
	Ref. by PR	Program	PR
	Contained in DC	Dataspace	DC
	Input to PR	Program	IP
	Result of PR	Program	RS
	Is comp. of SY	System	CF
Interface	Defined in PR	Program	PR
Keyword	Composed by KY	Keyword	KY
Method	Belongs to IE	Interface	IE
	Invoked by PR	Program	PR
Node	Contained in NW	Network	NW
Packagelist	Contained in SY	System	SY
Program	Belongs to SY	System	SY
	Used by PR	Program	PR
	Contained in PG	Packagelist	PG
	Used by SV	Server	SV
	Is comp. of SY	System	CP
Property	Defined in IE	Interface	IE
Server	Contained in NO	Node	NO
System	Contained in LS	Library Structure	LS
	Is subappl. of SY	System	CS
	Is library of SY	System	LI
Trigger	Triggers FI	File	FI
	Triggers EL	Field	EL
Verification	Verifies EL	Field	EL
	Is comp. of SY	System	CV
Virtual Machine	Belongs to NW	Network	NW

In addition to the predefined associations, user-defined associations can be created. User-defined associations can define a relationship between predefined as well as user-defined object types. With the exception of the object types Owner and Field, all objects types can be linked in any combination.

Association Administration Menu

This menu is called by entering code A in the Metadata Administration menu.

```
09:45:32             *****  P R E D I C T  4.3.1  *****             2003-05-31
                     - Association Administration Menu -


                       Function

                       A  Add an Association
                       D  Display Association
                       M  Modify Association
                       P  Purge Association
                       S  Select Association
                       T  Modify attribute names and numbers
                       X  Cross reference of an Association

Code ................

Object type code ....*
Association code ....*
Link direction ......*


Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       -     -    Stop   -     -    Flip  Print  -    Let    -     -    Main
```

Parameters	
Code	To select one of the functions available. The functions are described on the following pages.
Object type code	Code of the object type the association belongs to.
Association code	Unique association identifier.
Link direction	**A** Active or **P** Passive **Note:** If you add a new association (function code A), **A** (active) is obligatory.

Add / Modify Association - Code A,M

The left pane of the screen (Active Association) displays the view of the parent object, the right pane of the screen (Passive Association) displays the view of the child object.

Defining the Association

The Add and Modify functions use the same screens.

```
13:13:30            *****  P R E D I C T  4.3.1  *****              2003-05-31
                           - Add Association -



  Active Association                   Passive Association
    Code ...... PR                       Code ...... SY
    Name ...... PROGRAM                  Name ...... SYSTEM
    Title ..... Uses PR concept.         Title ..... Belongs to SY


                        Object types

                          Parent ...... SY System
                          Child ......* PR Program




Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       -     -    Stop   -     -    Flip  Print  -     -     -     -     -
```

The association codes must be unique concerning object types and the link direction. Referring to the above screen this implies that:

- among all active associations of object type SY (System), the code PR is allowed only once and
- among all passive associations of object type PR (Program), the code SY is allowed only once.

This ensures that an association can be identified uniquely with the object type, association code and and the link direction.

```
   13:58:05            *****  P R E D I C T  4.3.1  *****              2003-05-31
                             - Add Association -

   PR System Uses PR concept.                              Added 2003-05-31 at 13:12
   SY Program Belongs to SY                                   by HNO

   Codes                               Default associations
     Parent object type ..... SY         Active for System ................ N
     Child object type ...... PR         Passive for Program .............. N
     Association internal ... 2000146    Unload for System ................ N
     Association type no. ..* 4000146

   Abstract     Zoom: N









   Screen number .... 1        (H=Header)                  Free attributes:  80
```


The fields Screen number and Free attributes are described in First Screen in the section **Object Type Administration**.

Defining Attributes

Attributes of associations and attributes of object types are defined in the same manner.

Defining the Header Layout for the Link Editor Screen

```
13:07:12            *****  P R E D I C T  4.3.1  *****              2003-05-31
                         - Modify header layout -

Parent name ..... CHNG-ENHANCEMENT                Modified 2003-05-31 at 10:53
Child name ...... HEB-TEST                              by HNO

     Number of:
     Attr. Scr.  Column                                      Length  Format
---- ----------  ------  Association attribute header ----- ------- --------
 1.   1     1      1      attr1                                 5      N 1.0
 2.
---- ----------  ------  Child attribute header ----------- ------- --------
 1.   1     1      10     A                                     2      N 1.0
 2.
---- ----------  ------  ---------------------------------- ------- --------


Layout:                              ....+....1....+....2....+....3....+....4

      Heb-test                       attr1    A
      ------------------------------ -----    --
```

This screen appears after the attributes have been defined. Here you can define the layout of the Link Editor screen.

Note:
The attributes of the child object type can only be defined if the child object type is a user-defined object type.

The following rules apply:

- The total length of all attribute values must not exceed 30 characters (including blanks to separate the columns).
- The text of the header lines can be modified as desired.
- The attributes of the association must be specified in their full length and must be positioned to the left of the attributes of the child object type in the Link Editor screen.
- Only the alphanumeric attributes of the child object type can be displayed in a length less than the length of the attribute type.
- In the field Length, a length greater than the attribute length can be defined for the header line text.

Parameters	
Number of Attr.	Number of the attribute in the definition screen.
Number of Scr.	Number of the definition screen of this attribute.
Column	Position in the Link Editor screen.
Assoc. attr. header	Header text for the attribute of the association in the Link Editor screen.
Child attr. header	Header text for the attribute of the child object type in the Link Editor screen.

Display Association - Code D

Displays the association definition in a format similar to the screens that are used to add or modify the association definition.

```
13:58:05            *****  P R E D I C T  4.3.1  *****             2003-05-31
                          - Display Association -

PR System Uses PR concept.                              Added 2003-05-31 at 13:12
SY Program Belongs to SY                                   by HNO

Names                                                   Abstract
  Active  Association .. PROGRAM
  Passive Association .. SYSTEM

Codes
  Parent object type ..... SY SYSTEM
  Child object type ...... PR PROGRAM
  Association internal ... 2000004
  Association type no. ..* 2002000

Defaults
  Default child for System ............ Y
  Default parent for Program .......... Y
  Default download Association ........ Y
```

Purge Association - Code P

Deletes all associations between objects of the parent type and objects of the child type as well as all retrieval models that report on the association.

Select Association - Code S

Displays a list of associations. If both fields Object type code of parent/child are left blank, all association types are displayed. Enter a valid object type code for parent or child to restrict the selection to associations with the specified parent/child.

Sample Output


```
 13:35:34            *****  P R E D I C T  4.3.1  *****            2003-05-31
                       - Association Selection Menu -

Mark   Parent              Association           Child            Predef.

  _    SY System           PR Uses PR concept.   Program          yes
  _                        PG Uses PG            Packagelist      yes
  _                        CS Has subappl. SY    System           yes
  _                        CP Has component PR   Program          yes
  _                        CV Has component VE   Verification     yes
  _                        CF Has component FI   File             yes
  _                        LI Has library SY     System           yes
  _                        FI Uses File          File



Hit the corresponding PF-KEY to STOP, Enter 'X' to ABORT or 'T' for TOP : _
```

Modify Attribute Number - Code T

The functions Modify attribute number of associations and Modify attribute number of object types are defined as described above.

Cross Reference of an Association - Code X

Enter codes in fields Object type code, Association code and Link direction to display retrieval models which use the specified association.

Retrieval Model Administration

This section covers the following topics:

- Retrieval Models - The Basic Concepts
- Defining and Maintaining a Retrieval Model
- Add / Modify Retrieval Model - Codes A, M
- Copy a Retrieval Model - Code C
- Display Retrieval Model - Code D
- Rename a Retrieval Model - Code N
- Purge a Retrieval Model - Code P
- Select a Retrieval Model - Code S
- Cross Reference a Retrieval Model - Code X

Retrieval Models - The Basic Concepts

Information on the relationships of objects and the structure of an information system as a whole can be retrieved with the function Execute retrieval model. Defining a retrieval model involves:

- defining a retrieval structure
- defining a layout for the reports to be created.

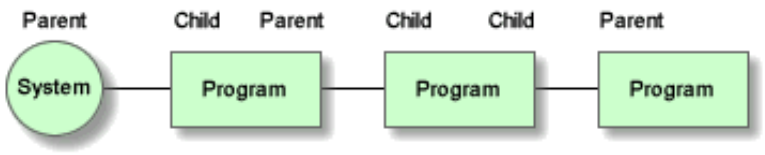
The basic concepts of retrieval models are described in this section. A more detailed description of the various facilities is then given with the description of the parameters in the section Defining and Maintaining a Retrieval Model.

What is a Retrieval Structure ?

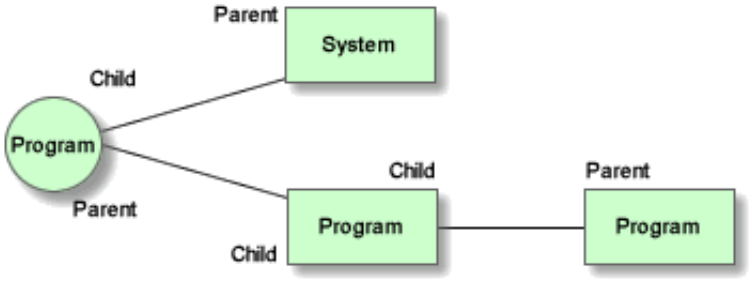
The function Execute retrieval model does not report on the metadata model of an information processing system as a whole, but reports on any part of the metadata model which matches the retrieval structure defined in the retrieval model.

Two simple examples for such a retrieval structure are:

- Programs that belong to a system:



- Programs contained in systems, files used by the programs and databases these files belong to:



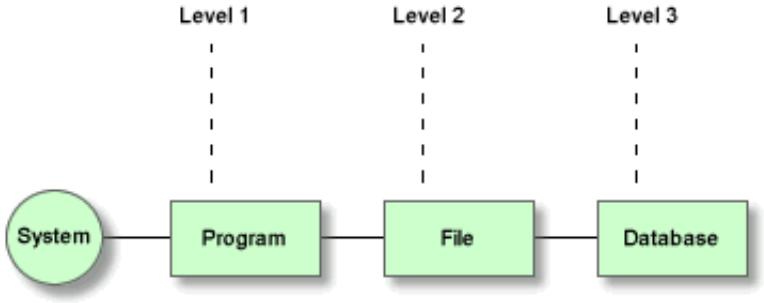
Rules Applying to Retrieval Structures

Retrieval Structures are Type-Specific

All retrieval structures are defined for a specific object type. When applying a retrieval model, an object of this type is used as the start object for scanning the metadata model for the specified structure and reporting the results.

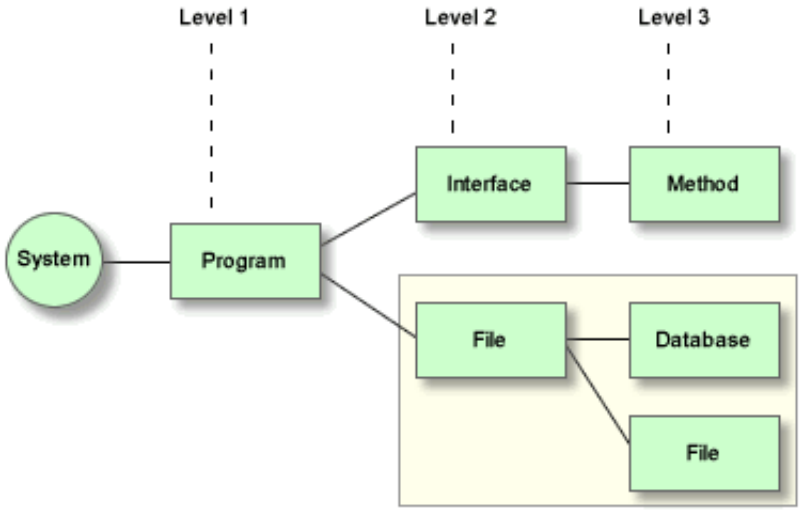
Retrieval Structures are Defined as Hierarchical Tree Structures

A retrieval structure is defined as a hierarchical tree structure. Each object type is specified with respect to another object type on a higher level.



Note:
Predict metadata models are entity relationship models, and hierarchical levels in a retrieval structure do **not** therefore reflect any top-down hierarchy of the metadata model. Using hierarchical levels in a retrieval structure is simply a way of expressing which object types on lower levels are linked to which object types on higher levels.

Retrieval Structures can be Divided into Sub-structures

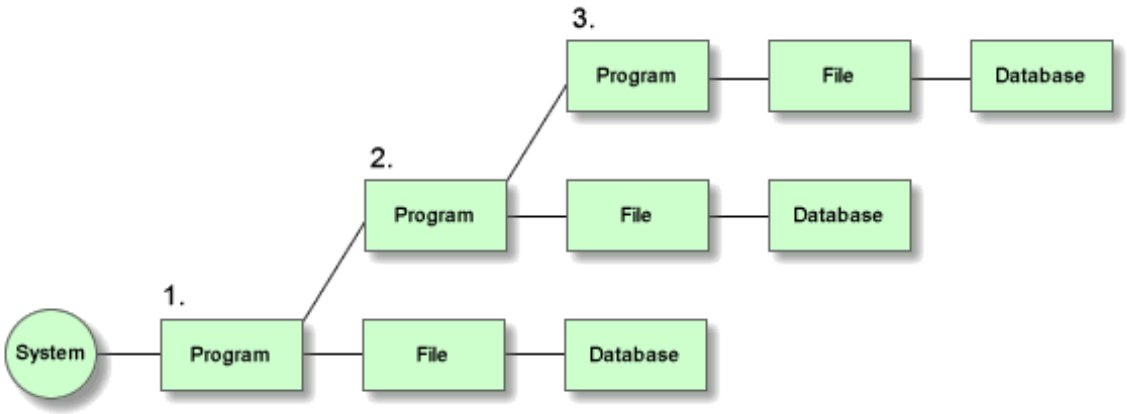


A retrieval structure can be seen as logically divided into sub-structures. A sub-structure starts at any object type and includes all subsequent object types on lower levels. In the example above, one possible sub-structure (file-database-file) is highlighted.

Retrieval Structures and Sub-structures can be Executed Recursively

A retrieval (sub-)structure can be executed several times to report on matching (sub-)structures starting with a recursive association, thus building up a nested structure.

If, for example, the substructure program-file-database in the first of the two examples in What is a Retrieval Structure? was executed with depth three, the retrieval structure shown in the diagram below results. The repetitions start with a recursive association of type Program-Program.



Defining and Maintaining a Retrieval Model

Enter code R in the Metadata Administration menu to display the Retrieval Model Administration Menu. This menu provides functions to create and maintain retrieval models.


```
13:13:02            *****  P R E D I C T  4.3.1  *****              2003-05-31
                  - Retrieval Model Administration Menu -


                     Function

                     A  Add a retrieval model
                     C  Copy retrieval model
                     D  Display retrieval model
                     M  Modify retrieval model
                     N  Rename retrieval model
                     P  Purge retrieval model
                     S  Select retrieval model
                     X  Cross reference retrieval model

Function ...........

Model ..............    for type ..*
Copy model .........

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       -     -    Canc   -     -     -    Print  -    Let   FLIP   -    Menu
```

Parameters	
Function	To select one of the functions of the Retrieval Model Administration Menu. The functions are described in the following sections.
Model	Two alphanumeric characters identifying the retrieval model. Must start with a letter and be unique for each object type. For example: a retrieval model ZZ can be defined for systems, and another retrieval model ZZ can be defined for files.
for type	The retrieval model is to be applied to this object type (for example, FI for file).
Copy Model	A new retrieval model can be defined by copying and modifying a retrieval model that already exists. Enter new model code here.

Note:
Defining a retrieval model is not a trivial task. You are recommended to outline the retrieval structure on paper before you start defining it in the system. When you define the layout of reports that a retrieval model will produce, Predict assists you by immediately showing the effect of the layout parameters.

Add / Modify Retrieval Model - Codes A, M

The Add and Modify functions of the Retrieval Model Administration Menu use the same three screens. These screens are described in the following sections:

- First Screen - General Settings
- Second Screen - Defining the Retrieval Structure
- Third Screen - Specifying the Amount of Information and the Layout

First Screen - General Settings

In the first screen, general characteristics of a retrieval model are defined and the retrieval model can be described.

```
13:13:17           *****  P R E D I C T  4.3.1  *****            2003-05-31
                   - Add Retrieval Model Definition -
 Model............ JP
 Retrieval for.... Database


 Model description               Userexits
                                   Use U-MODEL.....: N   (Y/N/A)
                                   Use U-XREF......: N   (Y/N)



 Abstract     Zoom: N
```

Parameters	
Model description	Up to three lines of 17 characters can be entered. This description is displayed in any Select retrieval model window and by the Select retrieval model function.
Abstract	Up to 16 lines of 30 characters each can be entered.
User exits	Two user exits can be activated: U-MODEL and U-XREF. U-MODEL applies to retrieval type Execute retrieval model with output modes Structured list and Cross reference. Valid values: **Y** The data collected will be handled exclusively by the user exit. **N** The user exit will not be called (default). **A** A report as defined in the retrieval model will be created and the data will be handled by the user exit additionally. U-XREF applies to retrieval type Execute retrieval model with output mode Cross reference. Valid values: **Y** The user exit will be called. **N** The user exit will not be called (default). Both user exits are described in the section section User Exits in this documentation.

Second Screen - Defining the Retrieval Structure

In the second screen, the retrieval structure is defined.


```
13:13:44            *****  P R E D I C T  4.3.1  *****             2003-05-31
                    - Add Retrieval Model Definition -
Model ........... LP
Retrieval for ... Program


                                    Association                  Related
  No  Level   Object type  Link*  A/P Code* Title       Depth  Object type
   1   1          PR        C         PR   Uses PR       3         PR
   2   2          PR        C         FI   Uses PR       __        FI
   3   3          FI        C         TR   Has TR        __        TR
   4   __                   _         __                __
   5   __                   _         __                __
   6   __                   _         __                __
   7   __                   _         __                __
   8   __                   _         __                __
   9   __                   _         __                __
  10   __                   _         __                __
  11   __                   _         __                __
  12   __                   _         __                __
  13   __                   _         __                __
  14   __                   _         __                __

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      -     -    Stop   -     -    Flip  Print        -     -     -     -
```

A line in this screen represents an association between two object types. A relation between the various lines (associations) is established via the level number:

On the first level the object type for the whole retrieval model is entered. In the above example PR (Program).

The object type on the second level results from the related type of the last line on level one. In the above example PR (Program).

The object type on the third level results from the related type of the last line on level two. In the above example FI (File).

Up to 20 levels can be defined that way.

An association can be identified uniquely with the object type, association code and and the link direction.

This has the following effects on the retrieval models:

1. The object type always results from the level number of the previous line.
2. Link and code must be entered and define the association.
3. The title and the related object type result from the chosen association.

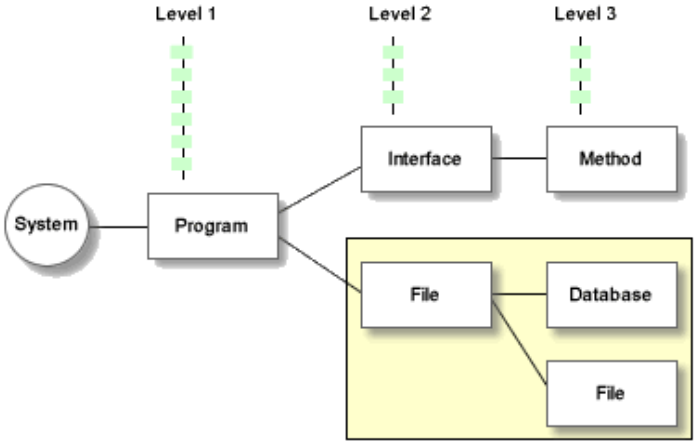
Editing the Retrieval Structure with Line Commands

The following line commands can be entered in the first position of the Depth column to edit the retrieval structure.

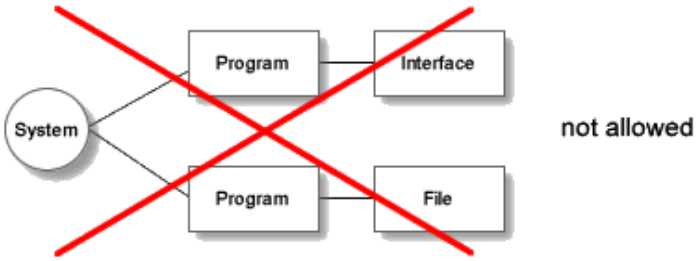
.I[n]
Inserts n lines. If n is not specified, three lines will be inserted. Blank lines will be deleted automatically.

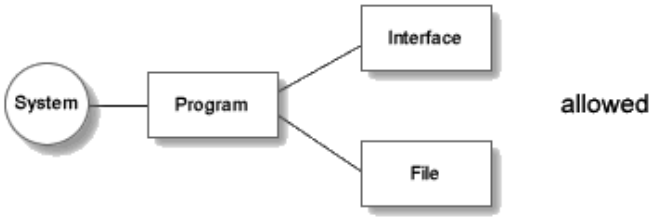
.D
Deletes the current line.

Parameters	
Depth	Number of times a substructure is executed. See Retrieval Structures can be Executed Recursively. Enter a leading C or P to determine if the recursive executions are to be applied to objects that are linked as children or as parents.
Level	A retrieval structure is essentially defined by specifying object types on different levels. Object types on lower levels are linked to an object type on the next higher level. In the example below, the object types Interface and File on level 2 are linked to the object type Program on level 1.

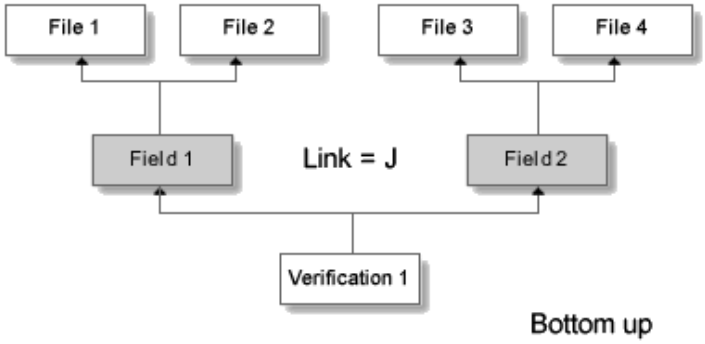


Note:
An object type **must not** occur twice on the same level and refer to the same object type on the next higher level.





Link	nk types. The following



R real files

F files of a file relation

S file relations of the file

I verifications which are linked to a file via fields (see figure below)

J files which are linked to a verification via fields (see figure below)

K databases in which a storagespace is used

L storagespaces used by database

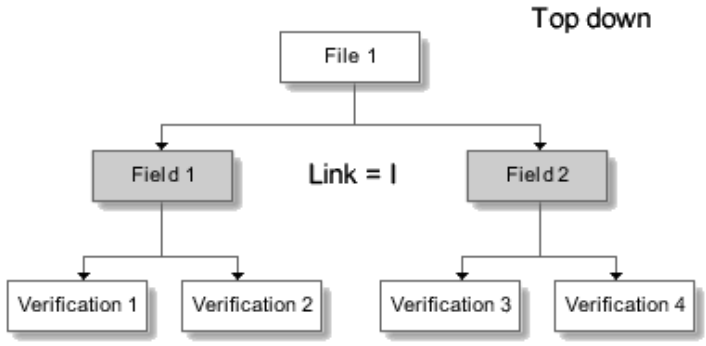
N dataspaces in which a storagespace is used

O storagespaces used by dataspace

U subcollections of a packagelist

T total collections of a packagelist

Enter an asterisk in any field in this column to display a window with all valid link types.

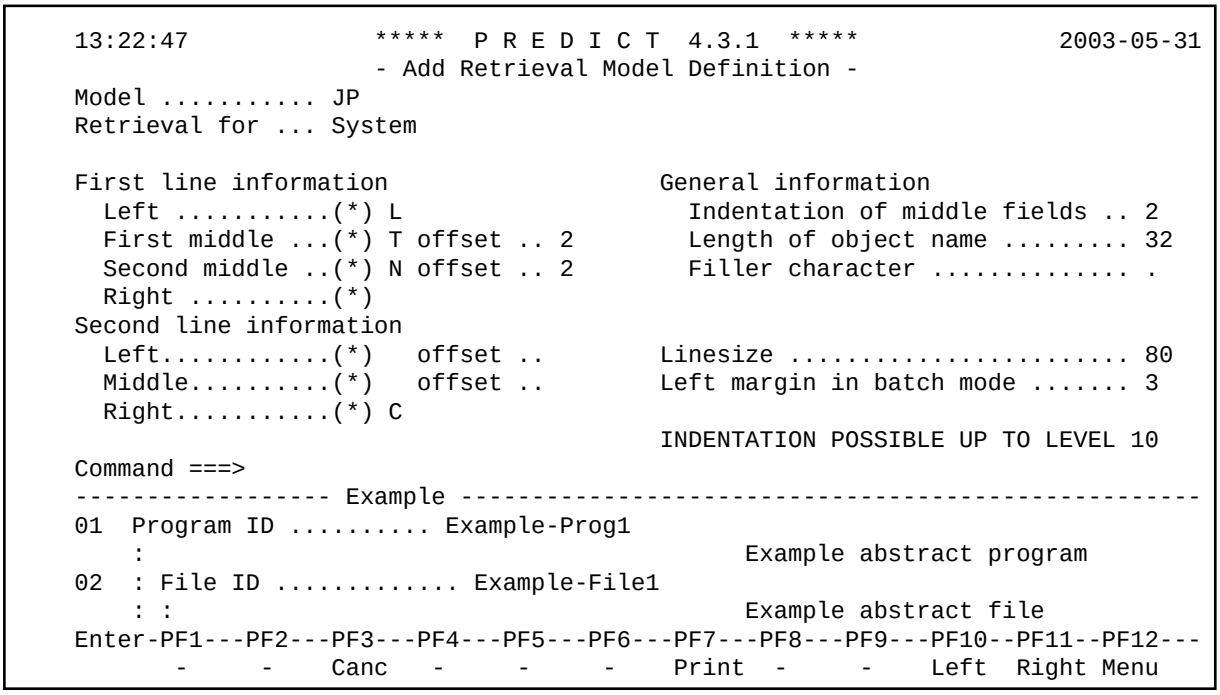


Code	Type of predefined or user-defined object to be included in the retrieval structure. Enter an asterisk in any field in this column to display a window with all valid object types.

Third Screen - Specifying the Amount of Information and the Layout

This screen is used to specify the information to be displayed on each object and the layout of a report created by the Execute retrieval model function when used with output mode Structured list.

Note:
In batch mode, a line size greater than the maximum line size of 80 characters used in online mode can be defined.

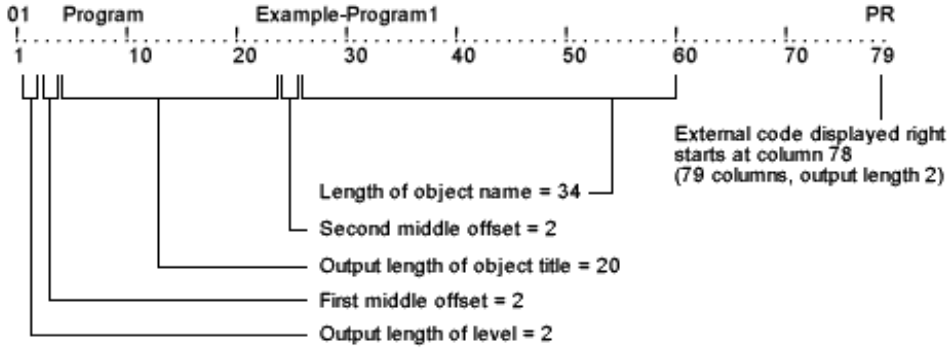


The screen consists of three parts:

- The top part displays status information.
- The middle part determines the information to be displayed on each object and the layout of the retrieval model. The parameters that can be specified are described below.
- The lower part of the screen displays the impact that changes to parameters will have on the layout of a report. The values displayed here (program, file, etc.) are "dummies". They are not derived from actual data stored in Predict.

Specifying the Amount of Information to be Reported on an Object	
First /Second line information	In reports created by Execute retrieval model, up to seven information categories out of a choice of ten can be displayed on each object, four as first line information (Left, First middle, Second middle, Right) and three as second line information (Left, Middle, Right). If the information that is displayed does not fit into one line, the first and second line information can actually be several lines long.
Left / First middle / Second middle / Middle	Right By specifying a code in one of these fields you determine, that information on a specific attribute of an object is to be included in a report and where it is displayed. Possible codes are: • **L** Level number.The level of the object type in the retrieval structure. The output length is 2. • **E** External codeof the object type (for example FI for Files). The output length is 2. • **T** Title of the object type. String used in Predict menus and output screens for the respective object type. The output length is 20 (including 3 bytes for the object ID). • **S** Subtype of the object type (for example Conceptual File). The output length is 17. • **N** Object name. Object names can be up to 32 characters long. The output length is 34, because two characters are reserved for the optional output of a character marking implemented objects or dummies. The output length can be truncated with the parameter Length of object name (see below). • **D** Link type. The output length is 1. • **A** Information contained in the last 32 columns of object lists is included. The layout of object lists is described in the respective sections of the Predefined Object Types in Predict documentation. • **K** Keywords. The output length is 32. • **O** Owners. The output length is 32. • **C** Abstract. The output length is 30. • **X** Extracts. The output length is 32. • **B** Association attributes. The output length is 32. A selection window containing all types of information can be displayed by entering an asterisk (*****)

Specifying the Layout	
The positions for the display of the various pieces of information are determined as follows:	
Left	Information displayed Left will always start in the first column if it is displayed online. In batch mode, the left margin can be specified explicitly (see parameter Left margin in batch mode below).

First middle / Second middle / Middle	The position of information displayed in a middle field is primarily determined with an offset value (see parameter offset below). The position is indented for lower level objects if a value is specified with the parameter Indentation of middle fields (see below).
Right	Information is positioned according to its length starting at the right margin. If, for example, the object name is displayed on the right hand side and its output length (parameter Length of object name) is set to 20, the name is displayed starting in column 60 (79 columns minus the length of the information). This position is calculated only once. Either the first line information or the second line information is used for the calculation, whichever is the longer.
Offset	Distance between the first character of a middle output field and the last character of its preceding field. The last character of the preceding field is determined by the output length of the respective piece of information. 
Indentation of middle fields	Information on objects that are defined on lower levels of the retrieval structure are indented the given number of columns (optional). The indentation refers to the starting position of the First Middle field (first line information) or the Middle field (second line information).
Length of object name	Object names (up to 32 characters) are truncated to the given length (optional).
Filler character	If a value displayed in the field First middle is shorter than the output length of the respective type of information, the remaining space is filled with the character specified. A filler character can only be used if a Second middle field is used. If, for example, a hyphen is specified, the two middle fields are displayed as follows: `File-------------: Example-File1`
Linesize	In batch mode, the line size can be greater than in online mode. The value of this parameter is compared with the Natural parameter *LINESIZE. If it is less than or equal to *LINESIZE it is used as line size in batch mode; otherwise an error message is displayed.
Left margin in batch mode	The left margin can be indented for use in batch mode (optional).
INDENTATION POSSIBLE UP TO LEVEL:	Information on objects that are specified on lower levels in the retrieval model can be indented in reports created by the Execute Retrieval Model function. How many lower levels can be indented depends on the output length of the different pieces of information and on the values specified with the parameters Offset and Indentation of Middle Fields. The maximum indentation is displayed in this field. Levels lower than the maximum indentation level are displayed without further indentation.

Saving, Resetting, Quitting the Retrieval Model Administration	
Maintaining a retrieval model can be quit with one of the following commands:	
SA[VE], CAT	All changes will be saved.
SA[VE] R, CAT R	Saves all changes without leaving the function.
Q[UIT],	All previous changes to the retrieval model (in all three screens) are lost. A window appears, asking for confirmation of the quit function.
REFRESH	Resets all changes of parameters in the third screen without leaving the function. Previous changes to the retrieval structure and the general settings (first screen) remain.

Note:
Before saving the parameters of a retrieval model, Predict checks whether the line size is appropriate for online display. If this is not the case, Predict checks whether the line size is appropriate for batch mode. If so, a message appears indicating that the retrieval model can be used in batch mode only; otherwise, the definition of the retrieval model is rejected.

Copy a Retrieval Model - Code C

A simple method of creating a new retrieval model is to copy an existing model and modify the copy. The model to be copied is identified by its name (Model) and the Object type code.

Display Retrieval Model - Code D

Displays a retrieval model definition in several subsequent screens. For a description of the individual output fields refer to the attribute descriptions in the sections above.

Rename a Retrieval Model - Code N

Retrieval models can be renamed with this function. A second screen is displayed to enter the new name.

Purge a Retrieval Model - Code P

Retrieval models can be purged with this function. Additional confirmation of the purge operation is requested to protect you from purging a retrieval model mistakenly.

Select a Retrieval Model - Code S

Retrieval models can be displayed for selection. The scope of the list can be determined by specifying a model name in asterisk notation and/or by specifying an Object type code.

Cross Reference a Retrieval Model - Code X

Displays a list of all object types, associations and special links that are used by the retrieval model.

Defaults Administration

Enter code D in the Metadata Administration main menu to display the Defaults screen. In this screen, you can define the default values to be displayed in the fields Object ID length and Disallowed characters of the Add/Modify object type definition screen (see First Screen).

These values can be checked with Special Function Consistency of Predict > Check naming conventions. See Check Naming Conventions - Code N in the section **Special Functions**.

```
13:28:47            *****  P R E D I C T  4.3.1  *****             2003-05-31
                               - Defaults -
                                                  Modified 2003-05-31 at 11:07
                                                        by RBN

Object ID length ........ 32 (1-32)
Disallowed characters ... * & F ?









Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       -     -    Stop   -     -    Flip  Print  -     -     -     -    Main
```

Parameters	
Object ID length	Maximum length for object IDs. Specify a number between 1 and 32.
Disallowed characters	Up to 20 characters that are not to be allowed in object IDs can be entered here.

Application Programming Interface

The functions provided by the Application Programming Interface (API) are used to read or modify Predict objects of user-defined types. With API program USR1051N you can also access predefined as well as user-defined object types.

This section covers the following topics:

General Rules

This section contains:

Applying Predict Defaults

When applying API functions, settings in session profiles or the General defaults menu are evaluated as if standard Predict functions were being performed.
Information is taken from the Predict system profile or a user-specific profile if a program using API functions is called by a user and the profile of this user is activated.

The following defaults are evaluated:

- Upper/lower case conversion: Object IDs, abstracts, extended descriptions and owners are converted to upper case if the respective parameters in the General defaults menu are set to U.
- The force owner/description parameter in the Metadata Administration is evaluated.
- The modification log (date, time and user of modification) is updated if this function is activated in the session profile.
- A complete maintenance check is performed if Y is specified with the parameter Full maintenance check in the session profile.

Executing User Exits

User exits that have been activated with the Activate user exits function of the Modify Defaults menu are executed in combination with programs using the API functions. The field SYSDIC-USER is defined for all user exists and the API. It can be used to pass values from user programs to user exits and vice versa.

The following user exits can be executed together with API functions:

- U-MNT before a maintenance function is executed.
- U-MNT1 before a maintenance function for an elementary field is executed.
- U-ACM before updated attributes are stored in Predict.
- U-PUR before an object is purged or scratched.
- U-DESC after an extended description has been modified.
- U-OW before the owner list is modified.
- U-ACMR after an Add, Copy, Modify or Rename function has been executed.
- U-OBJID after the object ID has been checked.

The user exit CATOWEX is not supported.

Validation Checks

Predict checks if values specified with function calls have been specified correctly. The following checks are applied:

- Are object types and object IDs specified correctly?
- Does any of the alphanumeric or numeric values specified exceed the length of the respective field definitions in the metadata definitions of the object type?
- Are more attribute values specified than defined in the metadata definitions of the object type?
- Is any keyword specified that is not yet defined in the data dictionary?
- If verification rules for attributes have been defined in the Metadata Administration, these rules are evaluated.

Rules and Hints

The following rules apply when using API functions:

- An END OF TRANSACTION statement must be executed within the user programs; API functions do not execute this statement.

- The field #OBJECT-COMPLETE contains a return code specifying whether an object has been defined completely or - in the case of an END OF TRANSACTION - the definition of an object is incomplete. Consistency of data can be ensured by evaluating this field within user programs modifying Predict objects of user-defined types.

Possible values of field #OBJECT-COMPLETE:

- 0 Complete
- 1 Owner missing (force)
- 2 Description missing (force)
- 3 Owner and description missing (force)

- Numeric variables are stored in the field #NUM-VALUES without decimal places. The number of decimal places is determined by the format definition of a field. For example: the value 10 will result in numeric values depending on different format definitions as shown below.

Format definition	Numeric value
N4.3	10000
N4	10

- The variable #INTERNAL determines whether General default or profile parameters are evaluated. #INTERNAL should not be changed with RESET or by assigning a value to prevent Predict from reading the default values unnecessarily often.

Using API / Subprogram PALUD42

The subprogram PALUD42 is delivered with Predict as standard and is stored in library SYSDIC. It contains parameter data area PALPUD42, which is used for accessing user-defined object types in 4.2 format.

If you wish to access data still in 3.2 format, see Compatibility with Predict Version 3.2.

If you wish to access data still in 3.3, 3.4 or 4.1 format, see Compatibility with Predict Version 3.3, 3.4 and 4.1.

This section contains:

- Parameter Data Area PALPUD42
- Executing an API Function
- Condition Codes
- Sample Program PALTEST

Parameter Data Area PALPUD42

If API functions are to be called from a program, the parameter data area PALPUD42 must be included in the data definition:

```
            :
            :
   DEFINE DATA
          LOCAL USING PALPUD42     /* API CALLNAT parameter
          LOCAL
            :
            :
```

Field Definitions in PALPUD42

Some fields defined in the parameter data area PALPUD42 can be used both as input or as output fields. These fields may be read and modified by both the Predict program PALUD42 (that executes API functions) and user programs. Other fields may only be used as either input or output fields.

The description of the different functions in the following sections contains lists of the fields that are used as input fields and fields that are used as output fields by the program PALUD42.

Many of the fields defined in the parameter data area PALPUD42 are defined by redefining the field API-INFO(*).

Executing an API Function

From within any program, an API function is executed by calling the program PALUD42 with a CALLNAT statement and the field #FUNCTION containing the code of the function that is to be executed (GA, GV, GO, etc.). All fields of the parameter data area PALPUD42 must be specified with the CALLNAT statement:

```
CALLNAT 'PALUD42'
#FUNCTION
#OBJECT-TYPE
#CHILD-TYPE
#OBJECT-ID
#OBJECT-COMPLETE
```



```
#MAINTENANCE-ACTIVITY
#RESP-GR
#SYSDIC-USER
#INTERNAL
#API-INFO(*)
```

When calling a function, make sure that all the values needed to execute a function are passed.

Condition Codes

After execution of PALUD42, a response code is returned in the field #RESPONSE-GR.

Field Name	Format	Description
#RESPONSE-GR		
#RESP-CODE	I2	Predict error message number. The full text of the respective message can be displayed by entering ? DICnnnn in the command line of Predict (where nnnn is the message number).
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

Sample Program PALTEST

The sample program PALTEST, which uses all API functions, is delivered with Predict. Enter PALTEST at the NEXT prompt in your Predict environment to see what PALTEST does.

Retrieval Functions

This section covers the following topics:

- GA - Get Attributes of Object Type
- GL - Get Linked Objects
- GO - Get Owners of Object
- GV - Get Values of Attributes
- HO - Get Owners of Object
- HV - Hold Values
- PV - Return Predict Version

GA - Get Attributes of Object Type

The function returns the metadata definitions of the object type. The following information is retrieved for different attributes:

- Attribute name,
- will upper/lower case translation be performed for alphanumeric fields,
- the data type (alphanumeric or numeric) of attributes,
- the length of attribute fields (for numeric fields including the decimal places),
- decimal places of numeric fields.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#ATTR-NAME (1:80)	A32	Name of the attribute
#UP-LOW (1:80)	A1	Upper/lower case conversion
#FORMAT (1:80)	A1	Data type of attribute field
#LENGTH (1:80)	N2	Length of attribute field
#DEC-POS (1:80)	N1	Number of decimal places
#IN-COL (1:80)	N8	Line/column for attribute name and value
#SCR-NUM (1:80)	N2	Number of the screen in which the attribute is used
#ED-MASK (1:80)	N2	Type of the edit mask
#RULE (1:80)	A2	Type of the verification rule
#OCC (1:80)	N2	Position of the attribute within the record
#MAINTENANCE-DATE (1)	T	Date and time of creation
#MAINTENANCE-ACTION(1)	A8	ID of user who created the object
#MAINTENANCE-DATE (2)	T	Date and time of modification
#MAINTENANCE-ACTION(2)	A8	ID of user who modified the object
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

GL - Get Linked Objects

This function retrieves all objects that are linked to a specified object via an association code. All links to an object are stored in link records. Up to 50 objects can be stored in one link record. A maximum of 999 link records per object for one association code can exist.

Each call of the function reads one link record. An input loop must be coded to read more than one link record.

The fields #CHILD-NUMBER and #CHILD-COUNTER are used to specify how many lines of the extended description are read (see example below).

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#ASSO-TYPE	A2	Predict Association Code
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) Predict Object ID of a file (only used for elementary fields)
#SYSDIC-USER	A250	User area
#CHILD-COUNTER	N4	Must be incremented for each call of the GL function (if called more than once; see example below). Note that #CHILD-COUNTER=1 after the first record of linked objects has been read.

Output Fields

Field Name	Format	Description
#CHILD-NUMBER	N4	Number of existing link records; is returned after the first execution of the function. Can be used as end value of a loop if more than one child record is to be read.
#CHILD-NAME (1:50)	A64	consists of: Object ID of linked child (A32) Object ID of a file (A32) (only used for elementary fields)
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

Example


```
DEFINE DATA LOCAL
       USING PALPUD42
END-DEFINE
*
#FUNCTION      = 'GL'
#OBJECT-TYPE   = 'OO'
#OBJECT-ID     = 'ULH-OO-FILLED'
*
REPEAT
  CALLNAT 'PALUD42' #FUNCTION #OBJECT-TYPE #CHILD-TYPE #OBJECT-ID
                    #OBJECT-COMPLETE #MAINTENANCE-ACTIVITY #RESP-GR
                    #SYSDIC-USER #INTERNAL #API-INFO(*)
*
   DISPLAY #CHILD-NAME(*)
   ADD 1 TO #CHILD-COUNTER
   IF #CHILD-COUNTER > #CHILD-NUMBER
      ESCAPE BOTTOM
LOOP
END
```

GO - Get Owners of Object

The function reports all owners of an object.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Object Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) — Predict Object ID #OBJECT-FILE-NAME (A32) — Predict Object ID of a file (only used for elementary fields)
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#OWNER (1:99)	A32	Owner IDs
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

GV - Get Values of Attributes

The function reports attribute values of an object. The following values are retrieved:

- Numeric attribute values.
- Alphanumeric attribute values.
- Abstract and keywords.
- Whether owners or descriptions are specified for the object.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Object Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) Predict Object ID of a file (only used for elementary fields)
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#ALPHA-VALUES (1:80)	A78	Values of alphanumeric fields
#NUM-VALUES (1:80)	P27	Values of numeric fields
#SHORT-COMMENTS (1:16)	A30	Abstract
#KEYWORDS (1:32)	A32	Keywords
#OWNER-YN	A1	Y if owner is specified, else N
#DESC-YN	A1	Y if description is specified, else N
#MAINTENANCE-DATE (1)	T	Date and time of creation
#MAINTENANCE-ACTION(1)	A8	ID of user who created the object
#MAINTENANCE-DATE (2)	T	Date and time of modification
#MAINTENANCE-ACTION(2)	A8	ID of user who modified the object
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

HO - Get Owners of Object

This function reports all owners of an object. The object is put into HOLD while the function is executed.

If the owner list of an object is to be to be modified, this function should be called first. There are two reasons for this recommendation:

- The object is put in HOLD.
- The modify function MO always overwrites the complete owner list. If for example only one owner is added or changed, all other owners would be overwritten with blanks. Calling HO before modifying any owner ensures that all owners are stored in the respective fields of PALPUD42 and therefore kept when old definition is overwritten.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Object Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) — Predict Object ID #OBJECT-FILE-NAME (A32) — Predict Object ID of a file (only used for elementary fields)
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#OWNER (1:99)	A32	Owner IDs
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

HV - Hold Values

The function reports attribute values of an object and puts the object into HOLD while the function is executed. The following values are retrieved:

- Numeric attribute values.
- Alphanumeric attribute values.
- Abstract and keywords.
- Whether owners or descriptions are specified for the object.

If attribute values of an object are to be to be modified, this functions should be called first. There are two reasons for this recommendation:

- The object is put in HOLD.
- The modify function MV always overwrites the complete attribute definition of an object. If only one attribute is changed, all other attributes would be overwritten with blanks. Calling the function HV before modifying any attribute values ensures that the old attribute values are stored in the respective fields of PALPUD42 and therefore kept when the old definition is overwritten.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) not used
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#ALPHA-VALUES (1:80)	A78	Values of alphanumeric fields
#NUM-VALUES (1:80)	P27	Values of numeric fields
#SHORT-COMMENTS (1:16)	A30	Abstract
#KEYWORDS (1:32)	A32	Keywords
#OWNER-YN	A1	Y if owner is specified, else N
#DESC-YN	A1	Y if description is specified, else N
#MAINTENANCE-DATE (1)	T	Date and time of creation
#MAINTENANCE-ACTION(1)	A8	ID of user who created the object
#MAINTENANCE-DATE (2)	T	Date and time of modification
#MAINTENANCE-ACTION(2)	A8	ID of user who modified the object
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

PV - Return Predict Version

The function PV returns the current Predict Version.

Input Fields

None.

Output Fields

Field Name	Format	Description
#Predict-VERSION	N3	Current Predict version

Maintenance Functions

This section contains:

- AV - Add Object
- D - Delete an Object
- LO - Link Object
- ML - Modify Links
- MO - Modify Owners
- MV - Modify Values of Attributes
- N - Rename Objects
- ST - Find Objects of Different Types with Same Name
- UO - Unlink Object

AV - Add Object

This function adds an object with its attribute values. The following parameters are returned by PALUD42 because their values may have been converted to upper case: #OBJECT-ID, #ALPHA-VALUES, #NUM-VALUES, #SHORT-COMMENTS and #KEYWORDS.

Note:
If return code 1, 2 or 3 is returned in the field #OBJECT-COMPLETE, inconsistent data has been added to the dictionary. The functions MO or MD must then be called to add an owner or extended description to the object. See also Rules and Hints.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Object Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) not used
#ALPHA-VALUES (1:80)	A78	Values of alphanumeric fields
#NUM-VALUES (1:80)	P27	Values of numeric fields
#SHORT-COMMENTS (1:16)	A30	Abstract
#KEYWORDS (1:32)	A32	Keywords
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#OBJECT-COMPLETE	A1	Whether object has been defined completely. Valid values: • 0 - Complete • 1 - Owner missing (force) • 2 - Desc missing (force) • 3 - Owner and description missing (force)
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) not used
#ALPHA-VALUES (1:80)	A78	Values of alphanumeric fields
#NUM-VALUES (1:80)	P27	Values of numeric fields
#SHORT-COMMENTS (1:16)	A30	Abstract
#KEYWORDS (1:32)	A32	Keywords
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

D - Delete an Object

This function deletes an object, all links to child objects, and all links from parent objects to the deleted object.

Note:
The function D (Delete) of Version 3.4 corresponds to the function S (Scratch) of Version 3.2. The Delete functionality of Version 3.2 has not been available since Predict Version 3.3.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) not used

Output Fields

Field Name	Format	Description
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

LO - Link Object

This function adds one entry to the child list of an object. When the list is sorted, the new entry is inserted in its proper place. If an object with the given #CHILD-NAME already exists, a message is issued.

#CHILD-NAME is returned by PALUD42 because the child name (ID) may have been converted to upper case.

Note:
Attributes for associations cannot be maintained with this function, i.e. the values of the attributes are not checked.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#CHILD-TYPE	A2	Predict Child Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) Predict Object ID of a file (only used for elementary fields)
#CHILD-NAME (1:1)	A64	consists of: Object ID of linked child (A32) Object ID of a file (A32) (only used for elementary fields)
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#CHILD-NAME (1:1)	A64	consists of: Object ID of linked child (A32) Object ID of a file (A32) (only used for elementary fields)
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

ML - Modify Links

The function modifies the child list of objects. #CHILD-NAME will be returned by PALUD42 because the child names (IDs) may have been converted to upper case. The following rules apply:

- Before PALUD42 is called with the function ML for the first time, the parameter #CHILD-INFO must be set to F. This has the effect that
  - the old child records are deleted, and
  - the first child record is stored.

  We therefore recommend assigning at least one child to the parameter #CHILD-NAME before PALUD42 is called for the first time (see example below).
- After the first call of PALUD42, #CHILD-INFO must be reset (as shown in the example below).
- Up to 50 child objects can be specified in one call of ML. If there are more entries in the child list, the function must be called again.
- After the new child objects have been transferred (or while transferring the last new child objects), the parameter #CHILD-INFO must be set to L (last record). This ensures that - if there are attributes for the link and if child objects have been deleted from the list - the corresponding attribute records are purged.

Note:
Attributes for associations cannot be maintained with this function, i.e. the values of the attributes are not checked.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#CHILD-TYPE	A2	Predict Child Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) Predict Object ID of a file (only used for elementary fields)
#CHILD-INFO	A1	F: first access to the child records L: last access to the child records
#CHILD-NAME (1:50)	A64	consists of: Object ID of linked child (A32) Object ID of a file (A32) (only used for elementary fields)
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#CHILD-COUNTER	N4	Number of current child records
#CHILD-NAME (1:50)	A64	consists of: Object ID of linked child (A32) Object ID of a file (A32) (only used for elementary fields)
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

Example

```
MOVE 'F' TO #CHILD-INFO /* first record
MOVE 'child1' TO #CHILD-NAME(1)
CALLNAT 'PALUD42' 'ML'  #CHILD-INFO ... #CHILD-NAME(*) ....
RESET #CHILD-INFO
MOVE 'child51' TO #CHILD-NAME(1)
CALLNAT 'PALUD42' 'ML'  #CHILD-INFO ... #CHILD-NAME(*) ....
MOVE 'child101' TO #CHILD-NAME(1)
MOVE 'L' TO #CHILD-INFO /* letzter Record
CALLNAT 'PALUD42' 'ML'  #CHILD-INFO ... #CHILD-NAME(*) ....
        /*Delete records of deleted link records
END TRANSACTION
```


MO - Modify Owners

This function modifies owners of an object. The parameter #OWNER is returned by PALUD42 because owner names may have been converted to upper case.

If the owner list of an object is to be modified, the function HO should be called first. There are two reasons for this recommendation:

- The object is put in HOLD.
- The modify function MO always overwrites the complete owner list. If - for example - only one owner is added or changed, all other owners would be overwritten with blanks. Calling HO before modifying any owner ensures that all owners are stored in the respective fields of PALPUD42 and therefore kept when the old definition is overwritten.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) Predict Object ID of a file (only used for elementary fields)
#OWNER (1:99)	A32	Owner ID
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#OWNER (1:99)	A32	Owner ID
#OBJECT-COMPLETE	A1	Whether object has been defined completely. Valid values: • 0 Complete • 1 Owner missing (force) • 2 Description missing (force) • 3 Owner and description missing (force)
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

MV - Modify Values of Attributes

This function changes values of attributes. The following parameters are returned by PALUD42 because their values may have been converted to upper case: #OBJECT-ID, #ALPHA-VALUES, #NUM-VALUES, #SHORT-COMMENTS and #KEYWORDS.

If attribute values of an object are to be modified, the function HV should be called first. There are two reasons for this recommendation:

- The object is put in HOLD.
- The modify function MV always overwrites the complete attribute definition of an object. If only one attribute is changed, all other attributes would be overwritten with blanks. Calling the function HV before modifying any attribute values ensures that the old attribute values are stored in the respective fields of PALPUD42 and therefore kept when the old definition is overwritten

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) not used
#ALPHA-VALUES (1:80)	A78	Values of alphanumeric fields
#NUM-VALUES (1:80)	P27	Values of numeric fields
#SHORT-COMMENTS (1:16)	A30	Abstract
#KEYWORDS (1:32)	A32	Keywords
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#OWNER-YN	A1	Y if owner is specified, else N
#DESC-YN	A1	Y if owner is specified, else N
#OBJECT-COMPLETE	A1	Whether object has been defined completely. Valid values: • 0 Complete • 1 Owner missing (force) • 2 Description missing (force) • 3 Owner and description missing (force)
#ALPHA-VALUES (1:80)	A78	Values of alphanumeric fields
#NUM-VALUES (1:80)	P27	Values of numeric fields
#SHORT-COMMENTS (1:16)	A30	Abstract
#KEYWORDS (1:32)	A32	Keywords
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

Example

```
CALLNAT 'PALUD42' 'HV' ... alpha-value
MOVE 'abc' TO alpha-value
CALLNAT 'PALUD42' 'MV' ... alpha-value
```

N - Rename Objects

The function renames objects. The parameter #NEW-OBJECT-ID is returned by PALUD42 because the new object ID may have been converted to upper case.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) not used
#NEW-OBJECT-ID	A32	New object ID for rename function

Output Fields

Field Name	Format	Description
#NEW-OBJECT-ID	A32	New object ID for rename function
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

ST - Find Objects of Different Types with Same Name

Predict objects of different types can have the same name (ID). The function ST finds all objects of different types having the same name.

Input Fields

Field Name	Format	Description
#OBJECT-ID	A32	Object ID

Output Fields

Field Name	Format	Description
#NUMBER-OF-TYPES	N3	Number of objects of different types having the same ID
#OBJECT-TYPES (1:130)	A2	Object type code
#OBJECT-TITLE (1:130)	A17	Titles of objects types in upper case
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

UO - Unlink Object

The function removes one entry of the child list of an object. If an object with the given #CHILD-NAME is not found, a message is issued.

#CHILD-NAME is returned by PALUD42 because the child name (ID) may have been converted to upper case.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#CHILD-TYPE	A2	Predict Child Type
#OBJECT-ID	A64	consists of: #OBJECT-NAME (A32) Predict Object ID #OBJECT-FILE-NAME (A32) Predict Object ID of a file (only used for elementary fields)
#CHILD-NAME (1:1)	A64	consists of: Object ID of linked child (A32) Object ID of a file (A32) (only used for elementary fields)
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#CHILD-NAME (1:1)	A64	consists of: Object ID of linked child (A32) Object ID of a file (A32) (only used for elementary fields)
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

Compatibility with Predict Version 3.2

To ensure that existing user programs continue to be executable without changing the parameter data area, the program PALUD is still available (along with the parameter data area PALPUD). If you wish to use PALUD, you must ensure that the metadata of the corresponding user-defined object has the format of version 3.2.

Note:
The program PALUD returns a maximum of 24 keywords for one object ID.

API Functions of Version 3.2

The following functions are still supported with Predict Version 4.2 to ensure compatibility with Version 3.2:

- GD - Get Extended Description
- MD - Modify Description
- S - Scratch an Object

GD - Get Extended Description

This function retrieves the extended description of an object. An extended description can contain more than one record. Each call of the function reports one record. A loop must be coded to read more than one record.
The fields #DESC-NUMBER and #DESC-COUNTER are used to specify how many lines of the extended description are read (see description below).

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#OBJECT-ID	A32	Predict Object ID
#SYSDIC-USER	A250	User area
#DESC-COUNTER	N4	Must be incremented for each call of the GD function if GD is called more than once (see example below). Note that #DESC-COUNTER is 0 after the first record of an extended description has been read.

Output Fields

Field Name	Format	Description
#DESC-NUMBER	N4	Number of existing description records. #DESC-NUMBER is returned after the function has been executed first. Can be used as end value in a loop if more than one record of an extended description is to be read.
#EXT-DESCRIPTION (1:27)	A76	27 lines of description
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

Example

```
DEFINE DATA LOCAL
       USING PALPUD
END-DEFINE
*
#FUNCTION      = 'GD'
#OBJECT-TYPE   = 'OO'
#OBJECT-ID     = 'ULH-OO-FILLED'
*
REPEAT
  CALLNAT 'PALUD' #FUNCTION #OBJECT-TYPE #CHILD-TYPE #OBJECT-ID
                  #OBJECT-COMPLETE #MAINTENANCE-ACTIVITY #RESP-GR
                  #SYSDIC-USER #INTERNAL #API-INFO(*)
*
   DISPLAY (ES=ON) #EXT-DESCRIPTION(*)
   ADD 1 TO #DESC-COUNTER
   IF #DESC-COUNTER > #DESC-NUMBER
      ESCAPE BOTTOM
LOOP
END
```

MD - Modify Description

This function deletes the existing extended description of an object and adds a new extended description. #EXT-DESCRIPTION are returned by PALUD because the extended description may have been converted to upper case.
The following rules apply:

- Before PALUD is called for the first time with the function MD, #DESC-INFO must be set to F. This has the effect
  - that the old extended description is deleted, and
  - that a first record of the new extended description is stored.

  We therefore recommend assigning at least one line of an extended description to #EXT-DESCRIPTION before PALUD is called for the first time (see example below).
- After the first call of PALUD, #DESC-INFO must be reset (as shown in the example below).
- Before any new records are stored, the old extended description is purged.
- Up to 27 lines of description can be specified in one call of MD.
  If more lines are to be added, the function must be called again.
- After transferring a new extended description, the parameter #DESC-INFO must be set to L (last line) and an additional call of PALUD must be performed before the END OF TRANSACTION. This ensures:
  - the status information of the respective object is updated
  - the user exit U-DESC is executed (if activated).

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#OBJECT-ID	A32	Object ID
#DESC-INFO	A1	Specifies first (F) and last (L) access of extended description
#EXT-DESCRIPTION (1:27)	A76	Up to 27 lines of description can be specified in one call of MD
#SYSDIC-USER	A250	User area

Output Fields

Field Name	Format	Description
#DESC-COUNTER	N4	Number of current description record
#EXT-DESCRIPTION (1:27)	A76	Up to 27 lines of description can be specified in one call of MD
#OBJECT-COMPLETE	A1	Whether object has been defined completely. Valid values: • 0 Complete • 1 Owner missing (force) • 2 Description missing (force) • 3 Owner and description missing (force)
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

Example

```
MOVE 'F' TO #DESC-INFO /* First record
MOVE 'abc' TO #EXT-DESCRIPTION(1)
CALLNAT 'PALUD' 'MD'  #DESC-INFO ... #EXT-DESCRIPTION(*) ....
RESET #DESC-INFO
MOVE 'def' TO #EXT-DESCRIPTION(1)
CALLNAT 'PALUD' 'MD'  #DESC-INFO ... #EXT-DESCRIPTION(*) ....
MOVE 'L' TO #DESC-INFO /* Last record
CALLNAT 'PALUD' 'MD'  #DESC-INFO ... #EXT-DESCRIPTION(*) ...
     /* ensures update of status info and execution of U-DESC (if activated)...
END TRANSACTION
```

S - Scratch an Object

This function deletes an object, all links to child objects, and all links from parent objects to the deleted object.

Note:
The function D (Delete) of Version 3.3 corresponds to the function S (Scratch) of Version 3.2. The Delete functionality of Version 3.2 has not been available since Predict Version 3.3.

Input Fields

Field Name	Format	Description
#OBJECT-TYPE	A2	Predict Object Type
#OBJECT-ID	A32	Object ID

Output Fields

Field Name	Format	Description
#SYSDIC-USER	A250	User area
#RESP-CODE	I2	Predict error message number
#RESP-TEXT1	A34	Text for parameter 1 used in the error message
#RESP-TEXT2	A34	Text for parameter 2 used in the error message

Compatibility with Predict Version 3.3, 3.4 and 4.1

To ensure that existing user programs continue to be executable without changing the parameter data area, the program PALUD33 is still available (along with the parameter data area PALPUD33).

Using API / Subprogram USR1051N

Using API / Subprogram USR1051N

With this subprogram you can perform the follwing tasks:

- Read/write extended description or abstract of any Predict predefined or user-defined object
- Select any predefined or user-defined object
- Read field objects

Calling the Subprogram

Enter command SYSEXT at the Natural prompt. This command invokes the library SYSEXT, which contains various Natural user exits. The following is provided for each user exit:

- a user exit subprogram (in object form)
- a sample program (in source form) of how to invoke the subprogram
- a text member explaining the function of the user exit.

A screen similar to the one below will appear. Page to subprogram USR1051N if necessary.

```
13:13:43               ***** NATURAL USER EXITS *****              2003-05-31
User: HNO                          - Menu -                  Library: SYSEXT

Cmd  Source    User Exit  Comment                                        Prod
 _   USR1034P  USR1034N   Display NTTF file table                         NAT
 _   USR1035P  USR1035N   Maintain a NATURAL member in the source area     NAT
 _   USR1036P  USR1036N   Maintain the user profile of the Software AG Editor      NAT
 _   USR1037P  USR1037N   Information about NATURAL ABEND data
 _   USR1051P  USR1051N   Interface to various PRD data                   PRD
```

Enter a question mark in the Cmd column to display a list of possible actions. Enter D in this column to display a description of this subprogram.

When you add an extended description, specify a negative value in SL-NUM after the last line you passed.

Using API / Subprogram USR2033N

With this subprogram you can perform the follwing tasks:

- Read all sets of a user in a library
- Read all members of an LX set

Calling the Subprogram

Enter command SYSEXT at the Natural prompt. This command invokes the library SYSEXT, which contains various Natural user exits. The following is provided for each user exit:

- a user exit subprogram (in object form)
- a sample program (in source form) of how to invoke the subprogram
- a text member explaining the function of the user exit.

A screen similar to the one below will appear. Page to subprogram USR2033N if necessary.

```
13:13:43               ***** NATURAL USER EXITS *****              2003-05-31
User: HNO                          - Menu -                 Library: SYSEXT

Cmd  Source    User Exit  Comment                                        Prod
 _   USR1034P  USR1034N   Display NTTF file table                         NAT
 _   USR1035P  USR1035N   Maintain a NATURAL member in the source area    NAT
 _   USR1036P  USR1036N   Maintain the user profile of the Software AG Editor      NAT
 _   USR1037P  USR1037N   Information about NATURAL ABEND data
 _   USR1051P  USR1051N   Interface to various PRD data                   PRD
 _   USR2033P  USR2033N   Information about PRD LX sets                   PRD
```

Enter a question mark in the Cmd column to display a list of possible actions. Enter D in this column to display a description of this subprogram.

Using API / Subprogram USR3005N

With this subprogram you can perform the following task:

- Read and maintain all documentation objects defined in Predict (predefined and user-defined).

This section contains:

Calling the Subprogram

Enter the command SYSEXT at the Natural prompt. This command invokes the library SYSEXT, which contains various Natural user exits. The following is provided for each user exit:

- a user exit subprogram (in object form)
- a sample program (in source form) of how to invoke the subprogram
- a text member explaining the function of the user exit
- several example programs that explain in detail how to read and maintain documentation objects.

A screen similar to the one below will appear. Page to subprogram USR3005N if necessary.

```
13:13:43                ***** NATURAL USER EXITS *****              2003-05-31
User: HNO                           - Menu -                  Library: SYSEXT

Cmd  Source    User Exit  Comment                                          Prod
 _   USR1034P  USR1034N   Display NTTF file table                           NAT
 _   USR1035P  USR1035N   Maintain a NATURAL member in the source area      NAT
 _   USR1036P  USR1036N   Maintain the user profile of the Software AG Edit NAT
 _   USR1037P  USR1037N   Information about NATURAL ABEND data
 _   USR1051P  USR1051N   Interface to various PRD data                     PRD
 _   USR2033P  USR2033N   Information about PRD LX sets                     PRD
 _   USR3005P  USR3005N   Process documentation objects                     PRD
```

Enter a question mark in the Cmd column to display a list of possible actions. Enter D in this column to display a description of this subprogram.

Predict DTD

```
<!ELEMENT Predict (Request | Result)>
<!ATTLIST Predict Version CDATA #IMPLIED>
<!ELEMENT Request (Select | Update | Purge | Add | Lock | Unlock)>
<!ELEMENT Select (Search, Return)>
<!ELEMENT Update (Search, Set)>
<!ELEMENT Purge (Search)>
<!ELEMENT Add ((Object-Type | Link), Set)>
<!ELEMENT Lock (Search)>
<!ELEMENT Unlock (Search)>
<!ELEMENT Result (Row * , Message)>
<!ELEMENT Message EMPTY>
```



```
<!ATTLIST Message number CDATA #REQUIRED
                  text CDATA #REQUIRED
                  type CDATA #REQUIRED
                  invalid-attribute CDATA #IMPLIED
                  index-in-invalid-attribute CDATA #IMPLIED
                  additional-msg-number CDATA #IMPLIED
                  additional-msg-text CDATA #OPTIIONAL>
<!ELEMENT Search ((Object-Type | Link | Multi-Link), Attribute*)>
<!ATTLIST Search  extract CDATA #IMPLIED
                            from-date CDATA #IMPLIED
                  key CDATA #IMPLIED
                  key2 CDATA #IMPLIED
                  key3 CDATA #IMPLIED
                  key4 CDATA #IMPLIED
                  key5 CDATA #IMPLIED
                  key-not CDATA #IMPLIED
                  key-op (AND | OR) #IMPLIED
                  owner CDATA #IMPLIED
                  scan-value CDATA #IMPLIED
                  scan-abstracts (Y | N) #IMPLIED
                  scan-desc (Y | N) #IMPLIED
                  scan-rules (Y | N) #IMPLIED
                  scan-ignore-case (Y | N) #IMPLIED
                  scan-obj-id (Y | N) #IMPLIED
                  scan-absolute (Y | N) #IMPLIED>
<!ELEMENT Return (Field*)>
<!ELEMENT Set (Row *)>
<!ELEMENT Object-Type EMPTY>
<!ATTLIST Object-Type value CDATA #REQUIRED
                      default CDATA #IMPLIED>
<!ELEMENT Link EMPTY>
<!ATTLIST Link source-object-type CDATA #REQUIRED
               association CDATA #REQUIRED
               direction (ACTIVE | PASSIVE) #REQUIRED>
<!ELEMENT Multi-Link EMPTY>
<!ATTLIST Multi-Link source-object-type CDATA #REQUIRED
                     association CDATA #REQUIRED
                     direction (ACTIVE | PASSIVE) #REQUIRED
                     default CDATA #IMPLIED>
<!ELEMENT Attribute EMPTY>
<!ATTLIST Attribute name NMTOKEN #REQUIRED
                    value CDATA #REQUIRED
                    display-only CDATA #OPIONAL>
<!ELEMENT Field EMPTY>
<!ATTLIST Field name NMTOKEN #REQUIRED>
<!ELEMENT Row (Attribute | Structure)*>
<!ELEMENT Structure (Group* | Attribute *)>
<!ATTLIST Structure name NMTOKEN #REQUIRED
                    display-only CDATA #OPIONAL >
<!ELEMENT Group (Attribute *)>
```

Input to and Response from API / Subprogram USR3005N

Input to and response from the API uses XML documents. These XML documents must comply with the rules of the Predict DTD (document type definition) above.

Input to the API uses documents of the type Request while a response from the API uses documents of the type Result as defined in the first element of the Predict DTD. All data passed is represented as attribute(s) of an element. This method is used to reflect the dynamic extension of the Predict metastructure.

To analyze the result, you might use the copy code PARSER_X which is delivered in library SYSEXXT. The appropriate local data area PARSER-X also exists in library SYSEXXT.

Example

Search all Adabas files having the owner XYZ and a name starting with EMP and return some attributes.

```
<Predict>
   <Request>
      <Select>
         <Search>
           <Object-type value="FILE-A" Owner="XYZ"/>
           <Attribute name="ID" value="EMP*"/>
         </Search>
         <Return>
           <Field name="ID"/>
           <Field name="CREATED-ON"/>
           <Field name="CREATED-BY"/>
           <Field name="CHANGED-ON"/>
           <Field name="CHANGED-BY"/>
           <Field name="ABSTRACT"/>
         </Return>
      </Select>
   </Request>
</Predict>
```

The select request is used to read data from Predict. The `<Search>` element comprises a specification of the data to be retrieved (in the above example `<Object-type value="FILE-A"/>`) and, if needed, a set of common search attributes (`Owner="XYZ"`, `keyword`, etc.) and optional additional search conditions formulated via the `<Attribute>` element (in the example `<Attribute name="ID" value="EMP*"/>`).

All attributes that are to be returned in the result must be contained in the `<Return>` element. References to attributes not belonging to the addressed object type or association will not result in an error message but are simply skipped.

The result may look like this:

```
<Predict
   <Result>
     <Row>
       <Attribute Name="ID" Value="EMPLOYEES-FILE"/>
       <Attribute Name="CREATED-ON" Value="199910130916489"/>
       <Attribute Name="CREATED-BY" Value="PRD411"/>
       <Attribute Name="CHANGED-ON" Value="199911161626520"/>
       <Attribute Name="CHANGED-BY" Value="NATQA"/>
       <Structure Name="Abstract">
         <Attribute Name="Abstract-line" Value="CONTAINS UNIQUE DESCRIPTIONS"/>
         <Attribute Name="Abstract-line" Value="OF EMPLOYEES OF AN INTERN."/>
         <Attribute Name="Abstract-line" Value="ENTERPRISE (COMPAR. TO SAG)"/>
       </Structure>
     </Row>
     <Row>
       <Attribute Name="ID" Value="EMPLOYEES-ABOVE"/>
       <Attribute Name="CREATED-ON" Value="199807071515436"/>
       <Attribute Name="CREATED-BY" Value="NATQA"/>
       <Attribute Name="CHANGED-ON" Value="199807141258276"/>
       <Attribute Name="CHANGED-BY" Value="RRI"/>
       <Structure Name="Abstract">
         <Attribute Name="Abstract-line" Value="This file was incorporated"/>
         <Attribute Name="Abstract-line" Value="from FDT on 98-07-07"/>
       </Structure>
```



```
    </Row>
   <Message number="2517" text="DIC2517 Function TERMINATED SUCCESSFULLY." type="Success"/>
  </Result>
</Predict>
```

The `type` attribute of the `<Message>` tag indicates whether the request could be successfully processed. Every object in the result document is delimited by a `<Row>` element.

Formulating Requests

Select Requests

You specify an object type with the `<Object-type>` element of the `<Search>` element. Additional search conditions can be given with the `<Attribute>` elements of the `<Search>` element, by specifying the attribute ID and the corresponding search value. For alphanumeric attributes, asterisk notation is allowed to search for all objects that have an attribute value which starts with the given search value.

Attribute values are transferred as strings. Both the quote character and the double-quote character can be used as string delimiters. Quotes within values must be coded as '''.

All replacements:

&	&amp;
'	&apos;
"	&quot;
<	&lt;
>	&gt;

With the return element, the IDs of implicitly or explicitly defined attributes are specified, for which the values are to be returned. Only attributes belonging to the given object type may be specified, the others are ignored.

Note:
When reading or modifying periodic groups, multiple value fields or simple groups additional rules should be considered. Special restrictions apply for fields with format logical or numeric. See Object Type #ATTRIBUTE for further information.

Modification Requests

To allow updating or deleting of an object, it must first be locked. An update request does not unlock the object. This has to be requested explicitly. A successful delete request will also remove the lock for the object, whereas an unsuccessful delete request will keep the lock.

Links and multiple links are considered to be attributes of the parent object. This implies that the parent object of a link or multiple link has to be locked before the link/multiple link can be modified.

Update and delete requests may only refer to one object at a time. That means that the result of the search operation consists only of one object.

Files have an attribute named `Element List`. With this attribute, the base attributes of all fields belonging to a file are transferred. There is no Add function for fields of a file. To add new fields to a file, the complete element list has to be updated.

There is no explicit rename request. Renaming must be done by updating the attribute ID. The object must be unlocked with the new ID.

To add exactly one link to a link list, use the `<Add>` element in in your search request.

To delete exactly one link from a link list, use the `<Purge>` element in in your search request.

If you want to add or modify the association attributes of only a single link in a link list by using the `<Update>` element, then your search request must include a fully qualified attribute of type `Target-Object`.

If you want to modify the complete link list of the associated object (including all association attributes) by using the `<Update>` element, then a fully qualified attribute of type `Target-Object` is not necessary in your search request.

Link a program to a system.

```
<Predict>
  <Request>
    <Add>
      <Link source-object-type="SYSTEM" association="PROGRAM" direction="ACTIVE"/>
    <Set>
      <Row>
        <Attribute name="SOURCE-OBJECT" value="SYSTEM1"/>
        <Attribute name="TARGET-OBJECT" value="PROGRAM1"/>
      </Row>
    </Set>
    </Add>
  </Request>
</Predict>
```

Example

Update some Adabas attributes of a file.

```
<Predict>
  <Request>
    <Update>
      <Search>
        <Multi-Link source-object-type="DATABASE-A" association="FILE" direction="ACTIVE"/>
        <Attribute name="SOURCE-OBJECT" value="DB180"/>
        <Attribute name="TARGET-OBJECT" value="CHD-A-FORMATE"/>
      </Search>
    <Set>
      <Row>
        <Attribute Name="MAX-ISN" Value="800"/>
        <Attribute Name="ASSO-DEVICE-TYPE" Value="3390"/>
        <Attribute Name="DATA-DEVICE-TYPE" Value="3390"/>
        <Attribute Name="ASSOPFAC" Value="91"/>
        <Attribute Name="CIPHERED" Value="Y"/>
      </Row>
    </Set>
    </Update>
  </Request>
</Predict>
```

Fields can only be added by modifying the element list of a file. Modifications to the field attributes `type`, `level`, `format` and `length` can also be made in the element list only.

The element list has an attribute named `EL-UNIQUE-ID`. When updating the element list, the value of this attribute indicates whether a field has been added, modified or renamed.

General

With this version it is not possible to change the sub-type of an object (e.g. change a conceptual file into an Adabas file). It is not possible to maintain the Predict metadata.

Predict security checks are done as usual.

The Predict user exits U-ACMR, U-DESC, U-MNT, U-MNT1, U-OW, U-PUR and U-SEC are invoked as defined.

Global and Special Attributes

The following global attributes are available for objects of all object types defined on the FDIC. The notation (S) after an attribute name means that this attribute can be used in a search condition:

- ID (S)
- CREATED-BY
- CREATED-ON
- CHANGED-BY
- CHANGED-ON
- ABSTRACT (representing a structure with the attribute ABSTRACT-LINE)
- KEYWORDS (representing a structure with the attribute KEYWORD)
- OWNERS (representing a structure with the attribute OWNER)
- DESCRIPTION (representing a structure with the attribute DESCRIPTION-LINE)
- OBJECT-STATUS (possible values are OBJECT, DUMMY and PLACEHOLDER)
- #OBJECT-TYPE. The value of this attribute represents the specialization of the object type (for example, "FILE-A"). If there is no specialization, the value of the base type is represented.
- #BASE-TYPE. The value of this attribute represents the name of the base type (for example "FILE").
- #OBJECT-TITLE. The value of this attribute represents the title of the specialization of the object type (for example, "Adabas file").
- #BASE-TITLE. The value of this attribute represents the title of the base type (for example, "File").

Object Type #ATTRIBUTE

To provide attribute definitions specific to an object type or to an association, the special object type #ATTRIBUTE exists. The notation (S) after an attribute name means that this attribute can be used in a search condition.

Objects of the type #ATTRIBUTE can have the following attributes:

- ID (S)
- TITLE
- NAMESPACE (S)
- NAMESPACETYPE (S) (either #OT-DEFINITION or #ASSOCIATION)
- NAMESPACE-START-OBJECTTYPE (S) (only for Namespacetype="#ASSOCIATION")
- NAMESPACE-DIRECTION (S) (only for Namespacetype="#ASSOCIATION")
- BELONGS-TO-SCREEN (S)
- IS-SCREEN (S) (Valid values: Y or N)
- LEVEL (Used to group attributes)
- FORMAT
  - A - Alphanumeric
  - N - Numeric
  - D - Date
  - T - Time

  - L - Logic
  - X - Literal
  - E - Text
  - F - Frame (Group frame, if level = 1, it denotes a tab/screen)
  - S - Structure
  - M - Field length with unit specification
- LENGTH
- MULTIPLE (N5)
- SEARCHABLE (S) (Indicates that this attribute can be used in search conditions.)

Example

Get all attributes of the object type application library.

```
<Predict>
  <Request>
    <Select>
      <Search>
        <Object-type value="#ATTRIBUTE"/>
        <Attribute name="NAMESPACETYPE" value="#OT-DEFINITION"/>
        <Attribute name="NAMESPACE" value="SYSTEM-A"/>
      </Search>
      <Return>
        <Field name="TITLE"/>
        <Field name="ID"/>
        <Field name="FORMAT"/>
        <Field name="LENGTH"/>
      </Return>
    </Select>
  </Request>
</Predict>
```

The result may look like this:

```
<Predict>
  <Result>
    <Row>
      <Attribute Name="TITLE" Value="Implementation Pointer"/>
      <Attribute Name="ID" Value="IMPLEMENTATION-POINTER"/>
      <Attribute Name="FORMAT" Value="F"/>
      <Attribute Name="LENGTH" Value="0"/>
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="Library"/>
      <Attribute Name="ID" Value="LIBRARY"/>
      <Attribute Name="FORMAT" Value="A"/>
      <Attribute Name="LENGTH" Value="8"/>
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="User system Fnr"/>
      <Attribute Name="ID" Value="FUSER-FNR"/>
      <Attribute Name="FORMAT" Value="N"/>
      <Attribute Name="LENGTH" Value="5"/>
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="User system DBnr"/>
      <Attribute Name="ID" Value="FUSER-DBID"/>
      <Attribute Name="FORMAT" Value="N"/>
      <Attribute Name="LENGTH" Value="5"/>
    </Row>
```



```
    <Row>
      <Attribute Name="TITLE" Value="Type"/>
      <Attribute Name="ID" Value="SUBTYPE"/>
      <Attribute Name="FORMAT" Value="A"/>
      <Attribute Name="LENGTH" Value="1"/>
    </Row>
   <Message number="2517" text="Function TERMINATED SUCCESSFULLY." type="Success"/>
  </Result>
</Predict>
```

If an attribute defined with the format S (Structure) or F (Frame) includes more than one attribute and `Multiple` has a value > 0, then it is transferred with the `<Structure>` element and the included attributes have to be grouped with the `<Group>` element. The structure represents a periodic group. The sequence of the `<Group>` elements represents the occurrences.

If an attribute defined with the format S (Structure) or F (Frame) includes exactly one attribute and `Multiple` has a value > 0, then it is transferred with the `<Structure>` element and no `<Group>` elements are transferred. The structure or the included attribute represents a multiple value field.

If an attribute defined with the format S (Structure) or F (Frame) includes more than one attribute and `Multiple` has the value 0, it is transferred with the `<Structure>` element and the included attributes are not grouped. The structure represents a simple group.

Values for attributes with the format L (Logic) are transferred as string with content ’Y’ or ’N’.

Numeric values with decimal precision (Format N and M) are transferred without a decimal point. For example, the value 75.3 for an attribute defined with Length 5 and Precision 1 is transferred as ’753’. For an attribute defined with Length 5 and Precision 2, the value ’7530’ has to be specified.

Object Type #OT-DEFINITION

To provide all object types defined on the FDIC, the special object type #OT-DEFINITION exists. The notation (S) after an attribute name means that this attribute can be used in a search condition.

Objects of type #OT-DEFINITION can have the following attributes:

- ID (S)
- TITLE
- TYPE (S) (Object type or specialization type. If the value "ALLOBJECTTYPE" is used, the result list includes the object type Field.)
- CREATED-BY
- CREATED-ON
- CHANGED-BY
- CHANGED-ON
- INSTANCE-EXIST (Indicates if objects of the specified type exist.)

Objects of type #OT-DEFINITION have an indicator named "Type" that shows whether this object is a definition of a base object type (e.g. Database, File, Program, etc.) or of a specialization type (e.g. Adabas database, DB2 database, Adabas file, DB2 table, Subprogram, Subroutine, etc.).

Base object types have only those attributes common to all objects of this type. Specialization types additionally have those attributes that are specific to objects of the specified sub-type.

Example

Search for all object types defined on the FDIC and, for each, return the title and the ID.

```
<Predict>
  <Request>
    <Select>
      <Search>
        <Object-type value="#OT-DEFINITION"/>
        <Attribute name="TYPE" value="OBJECTTYPE"/>
      </Search>
      <Return>
        <Field name="TITLE"/>
        <Field name="ID"/>
      </Return>
    </Select>
  </Request>
</Predict>
```

The result may look like this:

```
<Predict>
  <Result>
    <Row>
      <Attribute Name="TITLE" Value="Keyword"/>
      <Attribute Name="ID" Value="KEYWORD"/>
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="Network"/>
      <Attribute Name="ID" Value="NETWORK"/>
    </Row>
    .
    .
    .
    <Row>
      <Attribute Name="TITLE" Value="File"/>
      <Attribute Name="ID" Value="FILE"/>
    </Row>
   <Message number="2517" text="Function terminated successfully." type="Success"/>
  </Result>
</Predict>
```

Object Type #ASSOCIATION

To provide all associations defined on the FDIC, the special object type #ASSOCIATION exists.

Objects of type #ASSOCIATION can have the following attributes:

- ID (S)
- Title
- SOURCE-OBJECT-TYPE (S)
- SOURCE-TYPE-TITLE
- TARGET-OBJECT-TYPE (S)
- TARGET-TYPE-TITLE
- DIRECTION (S) (Indicates the direction of the association. Valid values: ACTIVE or PASSIVE.)
- MANDATORY (S)
- CREATED-BY
- CREATED-ON
- CHANGED-BY

- CHANGED-ON
- ABSTRACT
- ADDITIONAL-ATTRIBUTES (Valid values: Y or N.)
- MAINTAINABLE (by User)
- MULTI-LINKS-POSSIBLE

Example

Search for all active associations of Adabas files:

```
<Predict>
  <Request>
    <Select>
      <Search>
        <Object-type value="#ASSOCIATION"/>
        <Attribute name="SOURCE-OBJECT-TYPE" value="FILE-A"/>
        <Attribute name="DIRECTION" value="ACTIVE"/>
      </Search>
      <Return>
        <Field name="TITLE"/>
        <Field name="ID"/>
        <Field name="TARGET-OBJECT-TYPE"/>
        <Field name="TARGET-TYPE-TITLE"/>
      </Return>
    </Select>
  </Request>
</Predict>
```

The result may look like this:

```
<Predict>
  <Result>
    <Row>
      <Attribute Name="TITLE" Value="Has Fields"/>
      <Attribute Name="ID" Value="ELEMENT"/>
      <Attribute Name="TARGET-OBJECT-TYPE" Value="ELEMENT"/>
      <Attribute Name="TARGET-TYPE-TITLE" Value="Field"/>
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="Linked to VE"/>
      <Attribute Name="ID" Value="LINKED_TO_VE"/>
      <Attribute Name="TARGET-OBJECT-TYPE" Value="VERIFICATION"/>
      <Attribute Name="TARGET-TYPE-TITLE" Value="Verification"/>
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="Has Direct Views"/>
      <Attribute Name="ID" Value="HAS_DIRECT_VIEWS"/>
      <Attribute Name="TARGET-OBJECT-TYPE" Value="FILE"/>
      <Attribute Name="TARGET-TYPE-TITLE" Value="File"/>
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="Has Sequence EL"/>
      <Attribute Name="ID" Value="HAS_SEQUENCE_EL"/>
      <Attribute Name="TARGET-OBJECT-TYPE" Value="ELEMENT"/>
      <Attribute Name="TARGET-TYPE-TITLE" Value="Field"/>
    </Row>
   <Message number="2517" text="Function terminated successfully." type="Success"/>
  </Result>
</Predict>
```


LINK and MULTI-LINK

To allow searching for links between objects, the `<Link>` and `<Multi-Link>` elements exist.

Links can have the following attributes:

- SOURCE-OBJECT (S)
- TARGET-OBJECT (S)
- SOURCE-OBJECT-TYPE
- SOURCE-TYPE-TITLE
- TARGET-TYPE-TITLE
- TARGET-OBJECT-TYPE
- TARGET-OBJECT-STATUS
- MULTIPLE-LINK (Valid values: Y or N. Indicates whether additional attributes are provided via link or multi-link.)
- MULTIPLE-ALLOWED (Valid values: Y or N. Indicates whether a child object can be linked more than once to the same parent object via the same association. Currently used for FI-ADA only.)

In addition, links from fields have the following searchable attribute:

- SOURCE-OBJECT-NAMESPACE (S)

In addition, links to fields have the following searchable attribute:

- TARGET-OBJECT-NAMESPACE (S)

Furthermore, links can have additional attributes that have been defined for the corresponding association.

When searching for links, if the name of a specialization is used as input for SOURCE-OBJECT-TYPE, than the sub-type is ignored and the base object type is used.

For example, it is possible to search for links to files from the conceptual database A-C with the following search request:

```
<Search>
  <Link source-object-type value="DATABASE-A" association="FILE" Direction="ACTIVE"/>
  <Attribute name="SOURCE-OBJECT" value="A-C"/>
</Search>
```

Example

Search for all programs linked to the conceptual system SAG-PRD-OTHER-PGMS via the active association PROGRAM:

```
<Predict>
  <Request>
    <Select>
      <Search>
        <Link source-object-type="SYSTEM-C" association="PROGRAM" direction="ACTIVE"/>
        <Attribute name="SOURCE-OBJECT" value="SAG-PRD-OTHER-PGMS"/>
      </Search>
      <Return>
        <Field name="TARGET-OBJECT"/>
        <Field name="TARGET-OBJECT-STATUS"/>
        <Field name="TARGET-OBJECT-TYPE"/>
        <Field name="TARGET-TYPE-TITLE"/>
        <Field name="MULTIPLE-LINK"/>
        <Field name="MULTIPLE-ALLOWED"/>
```



```
      </Return>
    </Select>
  </Request>
</Predict>
```

The result may look like this:

```
<Predict>
  <Result>
    <Row>
      <Attribute Name="TARGET-OBJECT" Value="SAG-PRD-SUMPRDEX"/>
      <Attribute Name="TARGET-OBJECT-STATUS" Value="OBJECT"/>
      <Attribute Name="TARGET-OBJECT-TYPE" Value="PROGRAM-N"/>
      <Attribute Name="TARGET-TYPE-TITLE" Value="Subprogram"/>
      <Attribute Name="MULTIPLE-LINK" Value="N"/>
      <Attribute Name="MULTIPLE-ALLOWED" Value="N"/>
    </Row>
    <Row>
      <Attribute Name="TARGET-OBJECT" Value="SAG-ADABAS"/>
      <Attribute Name="TARGET-OBJECT-STATUS" Value="OBJECT"/>
      <Attribute Name="TARGET-OBJECT-TYPE" Value="PROGRAM-E"/>
      <Attribute Name="TARGET-TYPE-TITLE" Value="External program"/>
      <Attribute Name="MULTIPLE-LINK" Value="N"/>
      <Attribute Name="MULTIPLE-ALLOWED" Value="N"/>
    </Row>
   <Message number="2517" text="Function terminated successfully." type="Success"/>
  </Result>
</Predict>
```

Multi-links can have the following attributes:

- SOURCE-OBJECT (S)
- TARGET-OBJECT (S)
- SOURCE-TYPE-TITLE
- TARGET-TYPE-TITLE
- CREATED-BY
- CREATED-ON
- CHANGED-BY
- CHANGED-ON

Note:
Multi-links must have one additional identifying attribute of the associated link. For example, the link *Has files* for the object type Database (FI-ADA) has the additional identifying attribute P-FNR.

Using API / Subprogram N-CHKCOO

Using API / Subprogram N-CHKCOO

The subprogram is delivered in library SYSDIC. It checks the status of the coordinator file.

Calling the Subprogram

Copy N-CHKCOO to your application libray. The following example shows

The example program below checks the status of the coordinator file assigned to the specified FDIC file. If the coordinator file can be used for load, the example program executes some direct commands. During the execution of the direct commands, no additional input is necessary. If the coordinator file cannot be used for load, the returned messages are printed.

The example shows that it is possible to initiate a load process from an application library and to return control to the application.

```
0010 /* Example program for online loading from application
0020 /*
0030 DEFINE DATA LOCAL
0040    01  #DBNR   (N5) /* i - DBNr of FDIC (if not specified current FDIC
0050                  /*     is used)
0060    01  #FNR    (N5) /* i - FNr of FDIC (if not specified current FDIC
0070                  /*     is used)
0080    01  #PASS   (A8) /* i - password of FDIC (optional)
0090    01  #CIPH   (N8) /* i - cipher code of FDIC (optional)
0100    01  #USER   (A8) /* o - user who is active on coordinator FDIC
0110    01  #TID    (A8) /* o - terminal ID of active user
0120    01  #L-DB   (N5) /* o - DBNr of FDIC coordinator is used for
0130    01  #L-FN   (N5) /* o - FNr of FDIC coordinator is used for
0140    01  #MSG    (A80)/* o - information about error
0150    01  #MSG2   (A80)/* o - information about error
0160    01  #RESP   (I4) /* o - 0 coordinator file is empty
0170                  /*  ne 0 coordinator/FDIC file is used/invalid.
0180 END-DEFINE
0190 *
0200 CALLNAT 'N-CHKCOO' #DBNR #FNR #PASS #CIPH #USER #TID #L-DB #L-FN
0210                    #MSG #MSG2 #RESP
0220 IF #RESP NE 0
0230    PRINT '=' #RESP
0240      /   '=' #USER
0250      /   '=' #TID
0260      /   '=' #L-DB
0270      /   '=' #L-FN
0280      /   '=' #MSG
0290      /   '=' #MSG2
0300 ELSE
0310    RELEASE STACK
0320    STACK TOP DATA 'END'
0330    STACK TOP DATA 'LOAD OBJ ALL REPLACE=Y'
0340    STACK TOP DATA
0350     'SET MEDIUM-TYPE=D MEDIUM-ID=EMPLOY MEDIUM-DBNR=180 MEDIUM-FNR=200'
0360    STACK TOP DATA 'SET 188 32 '
0370    STACK TOP COMMAND 'CMD'
0380    STACK TOP COMMAND 'LOGON SYSDICBE'
0390    STACK TOP COMMAND 'SETUP *,I'
0400 END-IF
0410 END
```

