Predict

Reference

Version 8.5.2

October 2022

ADABAS & NATURAL

This document applies to Predict Version 8.5.2 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1983-2022 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at http://softwareag.com/licenses.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at http://softwareag.com/licenses/ and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at http://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: PRD-REF-852-20221008

Table of Contents

Preface xi
1 About this Documentation 1
Document Conventions 2
Online Information and Support 2
Data Protection 3
I Predict Commands 5
2 How to Use Predict Commands 7
Where to Enter Commands 8
Repeating Command Execution 8
Truncating Commands 8
Syntax of Predict Commands 8
Taking Parameter Values from the Command Context 9
Specifying Commands in Keyword or Positional Form 9
Delimiting Parameters in Predict Commands 10
Metasyntax 10
3 General Predict Commands 11
Displaying Help Information 13
Terminating the Predict Session 13
ACTIVE 13
ADMINISTRATE 14
ASSIGN 14
BACK 14
COMPARE 15
COORDINATOR 15
DEFAULT 15
DUPL-LIST ELEMENT 18
DUPL-SELECT ELEMENT 18
GENERATE 19
IMPLEMENT 20
INCORPORATE 20
LAST 21
LOGOFF 21
LOGON 21
MAIN 21
MAINTAIN 21
MENU 22
NEWGEN1 22
NEWEXGEN 22
NEXT 22
NEXT - 22
NPRLOGON 22
NP2LOGON 23
PLAN 23

PREPROCESS 23
PROFILE 23
PUNCH / WRITE 24
REASSIGN ELEMENT 24
RESETHOT 25
RETRIEVE 25
SET 25
SPECIAL 29
4 Standard Predict Commands 35
Standard Maintenance Commands 36
Standard Retrieval Commands 37
Standard Administration Commands 38
5 Predict Coordinator Commands 41
6 Type-Specific Predict Commands 43
Organization of this Section 45
7 Database 47
Database Maintenance 48
Database Retrieval 49
Administration Implemented Database 50
8 Dataspace 53
Dataspace Maintenance 54
Dataspace Retrieval 54
Dataspace Administration 56
9 Extract 57
Extract Maintenance 58
Extract Retrieval 59
10 Field 61
Field Maintenance 62
Field Retrieval 64
Field Active Retrieval 65
Field-specific Special Functions 66
11 File 69
File Maintenance 70
File Retrieval 72
File Active Retrieval 74
Administration Implemented File 75
12 File Relation 79
File Relation Maintenance 80
File Relation Retrieval 80
13 Interface 83
Interface Maintenance 84
Interface Retrieval 84
14 Keyword 87
Keyword Maintenance 88
Keyword Retrieval 88

15 Library Structure 91
Library Structure Maintenance 92
Library Structure Retrieval 92
16 Member 95
Member Active Retrieval 96
17 Method 99
Method Maintenance 100
Method Retrieval 100
18 Network 103
Network Maintenance 104
Network Retrieval 104
19 Node 107
Node Maintenance 108
Node Retrieval 108
20 Owner 111
Owner Maintenance 112
Owner Retrieval 112
21 Packagelist 115
Packagelist Maintenance 116
Packagelist Retrieval 116
22 Program 119
Program Maintenance 120
Program Retrieval 122
Program Active Retrieval 123
23 Property 125
Property Maintenance 126
Property Retrieval 126
24 Report Listing 129
Report Listing Maintenance 130
Report Listing Retrieval 131
25 Server 133
Server Maintenance 134
Server Retrieval 134
26 Storagespace 137
Storagespace Maintenance 138
Storagespace Retrieval 138
Storagespace Administration 140
27 System 141
System Maintenance 142
System Retrieval 142
System Active Retrieval 144
28 Trigger 147
Trigger Maintenance 148
Trigger Retrieval 149
29 User 151

User Maintenance ... 152
User Retrieval ... 153
30 Verification ... 155
Verification Maintenance ... 156
Verification Retrieval ... 157
Verification Active Retrieval ... 158
31 Virtual Machine ... 161
Virtual Machine Maintenance ... 162
Virtual Machine Retrieval ... 162
32 UDE - User-Defined ... 165
UDE Maintenance ... 166
UDE Retrieval ... 166
II Maintenance in Predict ... 169
33 Maintenance in Predict ... 171
Overview of Maintenance Functions ... 172
Creating Predict Objects ... 176
Selecting and Displaying Objects ... 178
Purging Objects ... 179
Renaming Objects ... 179
Support of Dummy and Placeholder Objects ... 180
Overview of Standard Attributes ... 181
Abstract and Extended Description ... 182
Creating and Modifying Type-Specific Attributes ... 189
III Editors in Predict ... 191
34 General Information ... 193
Defining Editor Defaults ... 194
Calling an Editor in Predict ... 195
Functional Scope of the Editors available in Predict ... 196
35 Natural Editor ... 199
Functional Scope ... 200
Prerequisites ... 200
Presettings ... 200
Profile ... 200
How Editor Commands are Called ... 201
Basic Functions ... 202
36 Software AG Editor ... 209
General Information ... 210
Common Functions of the Software AG Editor ... 212
Troubleshooting ... 213
37 Task-Specific Editors ... 217
Description Editor ... 218
Subquery Editor ... 220
Rule Editor ... 221
Link Editor ... 223
38 Transferring Text ... 229

Selecting Text From Another Source Within Predict 230
Calling the Select Function 231
Importing Text from an External Source 233
Exporting Text to an External Target 242
Maintaining Extended Descriptions with Word for Windows 247
IV Predict Retrieval 253
39 Concepts of Retrieval 255
Types of Information you can Retrieve 256
Printing the Output of Retrieval Operations 257
Executing Retrieval Operations in Batch Mode 258
Using Retrieval Operations to Select Objects for Further Processing 258
Indicating Dummy and Placeholder Objects 258
40 Using Retrieval Options 261
Overview of Retrieval Options 262
Selection Criteria 262
Restrictions 264
Retrieval Type 267
Output Mode 269
Output Options 272
Indicating which Objects are Implemented 280
Displaying Information on the Implementation of Programs 280
Using Restrictions and Output Options from Profiles 282
41 Layout of Reports 283
Header Information 284
Cover Page 285
Body of Report 286
Footer 287
42 Standard Retrieval Types 289
Objects - Code D 290
Objects with Parents - Code B, Objects with Children - Code T 290
Objects with No Parent - Code O , Objects with No Child - Code U 291
Dummy/Placeholder Objects - Code C 292
Execute Retrieval Model - Code E 293
43 Type-Specific Retrieval Types 295
Database-Specific Retrieval Types 296
Extract-Specific Retrieval Types 296
Field-Specific Retrieval Types 297
File-Specific Retrieval Types 298
Keyword-Specific Retrieval Types 299
Network-Specific Retrieval Types 299
Owner-Specific Retrieval Types 300
Storagespace-Specific Retrieval Type 301
User-Specific Retrieval Type 301
Verification-Specific Retrieval Types 301
V Active Retrieval 303

44 Concepts of Active Retrieval Functions 305
How XRef Data is Created 306
What is an Implemented Member 306
Where to Find Comprehensive Information on XRef Data 307
45 Executing Active Retrieval Operations 309
Selecting Objects for Active Retrieval Operations 310
Using Implementation Pointers to Establish a Connection between Documentation and XRef Data 310
46 Saving the Result of Active Retrieval Operations in Sets 313
How and Where Sets are Saved 314
Rules for Using Sets 314
47 Member Active Retrieval 315
Displaying the Member Active Retrieval Screen 316
Retrieval Type, Output Mode and Save Set Option 316
Limiting the Scope of Member Active Retrieval 317
Retrieval Types 318
48 Field Active Retrieval 321
Retrieval Type, Output Mode and Save Set Option 322
Limiting the Scope of Field Active Retrieval 323
Retrieval Types 325
49 File Active Retrieval 327
Retrieval Type, Output Mode and Save Set Option 328
Limiting the Scope of File Active Retrieval 329
Retrieval Types 330
50 Program Active Retrieval 333
Displaying the Program Active Retrieval Screen 334
Retrieval Type, Output Mode and Save Set Option 335
Limiting the Scope of Program Active Retrieval 335
Retrieval Types 336
Applying the Function to Different Program Types 339
51 System Active Retrieval 343
Displaying the System Active Retrieval Screen 344
Retrieval Type and Output Mode 345
Limiting the Scope of System Active Retrieval 345
Retrieval Types 346
52 Verification Active Retrieval 349
Displaying the Verification Active Retrieval Screen 350
Retrieval Type, Output Mode and Save Set Option 350
Retrieval Types 352
VI SYSHELP - Active Help System for Your Application 355
53 SYSHELP - Active Help System for Your Application 357
Active Help System of Predict - Overview 358
Overview of Activities when Using SYSHELP 360
Using Field Help 361
Using Screen Help 363

Interface to the SYSHELP Subprogram ... 363
VII LIST XREF For Natural ... 367
54 Creation of XRef Data For Natural ... 369
55 Maintaining XRef Data ... 371
Using the Natural Utilities SYSOBJH (Object Handler), SYSMAIN, NATLOAD, NATUNLD or INPL ... 372
Using the Natural Commands SCRATCH, UNCAT , DELETE, PURGE and RENAME ... 372
Using Predict Special Functions ... 373
Using the Predict Coordinator ... 373
56 Contents of XRef Data ... 375
57 Types of LIST XREF Functions ... 377
58 Steplib Support ... 379
LIST XREF Main Menu ... 380
Using a Structure Documented in Predict ... 383
Using the Runtime Structure ... 383
Working Without any Structure ... 384
Displaying the Current Environment ... 385
Steplib Support in Batch Mode ... 386
Defining a Default Structure ... 386
59 Multilanguage Support ... 387
60 Calling List XRef Functions ... 389
Processing Several Objects in One Call ... 390
Help Information ... 390
Using the LIST XREF Profile ... 390
61 Retrieving Information For Individual Members ... 391
Two Ways to Retrieve Information on Individual Members ... 392
Information about Indirect Calls ... 393
Which Invocation Methods are Evaluated ... 393
Invoked Programs ... 395
Data Areas and Variables ... 401
Views and Their Fields ... 404
Copy Code ... 407
Error Numbers ... 409
Printers ... 411
Workfiles ... 412
Retained Sets ... 414
Processing Rules ... 415
External Resources ... 417
Report Programs with XRef Data ... 419
62 Verify Application Integrity ... 423
Verify Application Integrity Menu ... 424
Verify Consistency ... 425
Verify Implementation ... 427
Verify References ... 429

Handling Objects that are not Referenced 431
Avoiding Dynamic Invocation 432
Verifying All Aspects of an Application 432
63 Using Sets 435
Functional Scope 436
Examples for the Use of Sets 436
Using Sets to Share Work among Team Members 437
Internal Structure of Sets 437
How are Sets Created? 438
Who can Work with Sets? 438
Working With Sets 438
Creating a New Set via Selection 438
Operate on Sets 439
Application Programming Interface 443
64 Maintaining a LIST XREF Profile 445
Functional Scope 446
Assigning Functions to PF Keys 447
Limiting the Scope of the Report Programs with XRef Data Function 448
Parameter Keep set after cat/stow processing 448
Parameter Default Library Structure 449
Parameter Search Usage in Web Services 449
Parameter Verify Structure 450
Assigning Defaults to the Verify All Function 450
65 LIST XREF Commands 453
General Information 454
Using LIST XREF in Batch Mode 455
Commands for Retrieving Information for Individual Members 457
Command to Report on Program Active References 462
Commands for Verifying the Integrity of an Application 462
Miscellaneous 463
VIII List XRef for Third Generation Languages 465
66 List XRef for Third Generation Languages 467
Creation of XRef Data for 3GL 468
Contents of XRef Data for 3GL 469
Types of LIST XREF Functions for 3GL 470
Calling Functions of LIST XREF for 3GL 471
Retrieving Information for Individual Members 472
Verify Application 479
Operate on Sets 480
Maintaining a LIST XREF Profile 481

Preface

This documentation describes the basic functions of Predict. It contains all the information needed for processing documentation objects. It is organized in the following parts:

Predict Commands	All Predict commands and how they are used. This section provides a summary of all functions and screen input fields. It gives you an overview of the functionality of Predict and shows which object types can be used with the individual functions.
Maintenance	All general maintenance functions. Type-specific maintenance functions are described in the respective sections of the Predefined Object Types in Predict documentation.
Editors in Predict	Editors available for processing attributes of objects in Predict. This section also describes the transfer of Predict text attributes to and from an external environment.
Retrieval	Describes the general retrieval functions. The type-specific functions are described in the respective sections of the Predefined Object Types in Predict documentation.
Active Retrieval	Active Retrieval functions evaluate XRef data together with Predict documentation data to find differences between the implementation and documentation of an application. This section describes all the available options. If you are working with Natural steplibs, see also section Steplib Support in the *Predict and Other Systems documentation*.
SYSHELP	Active Help System for your Application: How to generate online help for your application from data stored with Predict objects.
LIST XREF for Natural	The `LIST XREF` command for Natural evaluates XRef data for Natural members created when cataloging Natural objects. All options are described in this section.
LIST XREF for Third Generation Languages	The Predict `LIST XREF` functions for third generation languages evaluate XRef data created by Adabas Native SQL, the Predict Preprocessor or the `CREATE DBRM` command of Natural for DB2. All options are described in this section.

1 About this Documentation

- Document Conventions 2
- Online Information and Support 2
- Data Protection 3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

In addition, you can also access the cloud product documentation via **https://www.softwareag.cloud**. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at **https://knowledge.softwareag.com**.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

I Predict Commands

All Predict functions can be invoked with commands or from batch jobs. The syntax and the use of all Predict commands are described in these sections.

This description provides a comprehensive summary of all functions and input fields in maps. Hence, the functional scope of Predict can be learned as well as which object types can be processed with which individual functions.

For an overview of all options to call a function (menu, command, workplan) see the section Predict User Interface in the Introduction to Predict documentation.

The use of commands in batch jobs is described under *Predict in Batch Mode*.

How to Use Predict Commands

General Predict Commands

Standard Commands

Predict Coordinator Commands

Type-specific Commands

2 How to Use Predict Commands

- Where to Enter Commands 8
- Repeating Command Execution 8
- Truncating Commands 8
- Syntax of Predict Commands 8
- Taking Parameter Values from the Command Context 9
- Specifying Commands in Keyword or Positional Form 9
- Delimiting Parameters in Predict Commands 10
- Metasyntax 10

Where to Enter Commands

Predict commands can be entered:

- On any Predict screen containing the Command ===> prompt. The line containing this prompt is referred to throughout the Predict documentation as the command line.
- After the Natural `NEXT` prompt, preceded by CMD and separated by blank.
- In batch jobs: Retrieval, active retrieval, generation, incorporation and comparison commands are also available in batch mode, but most maintenance commands are not.

Repeating Command Execution

The command `LAST` returns to the Predict Main Menu and displays the Predict command that was executed last. The command can then be modified or reexecuted.

Truncating Commands

Commands can be truncated provided that the input remains unambiguous. For example, the command `GENERATE` can be entered as G since no other command begins with G. However, the command `LOGON` must be entered in full, since truncations (LOGO, LOG etc.) could be confused with the `LOGOFF` command.

If the truncated command is ambiguous, the Predict command processor checks whether any accompanying object type is unambiguous. If so, it checks what commands can be applied to this object type.

Syntax of Predict Commands

General and standard Predict commands have the following syntax:

```
<command> <parameters>
```

Type-dependent commands have the following syntax:

```
<command><object-type><object-ID><parameters>
```

or

```
<command><object-type><association-type><object-ID><parameters>
```

Meaning	
<command>	The command used to call a specific function, for example `ADD`, `COPY`, `DISPLAY`, `RETRIEVE`. Commands are written in upper case throughout the documentation.
<object-type>	Type of object to be processed, for example system or file.
<association-type>	Type of association to be processed, for example the association type for the command `LINK`.
<object-ID>	ID of the object to be processed.
<parameters>	Additional information needed to carry out the function.

Taking Parameter Values from the Command Context

Wherever possible, Predict takes parameter values that are omitted from the context in which a command is entered:

- If an object type is omitted, the object type of the current menu is taken. If for example the command `MAINTAIN` is entered in the File Retrieval menu, the File Maintenance menu is displayed.
- Other parameter values are taken from the hot object table as far as possible. See *Specifying Parameter Values* in the section *Predict User Interface* in the *Introduction to Predict* documentation.

Specifying Commands in Keyword or Positional Form

Commands can be specified in keyword form, positional form or a mixture of both. Throughout the documentations the command syntax is described in keyword form. The positional form is obtained by omitting the keywords and the Natural assign character (=). The assign character can be changed in Natural.

Keyword Form

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

or, if parameter 2 is omitted:

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

Positional Form

```
COMMAND value1, value2, value3
```

or, if parameter 2 is omitted:

```
COMMAND value1,, value3, value4
```

In the positional form, the omitted parameter is represented by an empty parameter only if followed by other parameters.

Mixing Keyword and Positional Form

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

```
COMMAND KEY5=value5, value6,, value8
```

Delimiting Parameters in Predict Commands

The individual parameters in Predict commands are delimited by the character that is specified as the delimiter character in Natural. Throughout the following descriptions a comma (,) is used as the delimiter character. The delimiter character can be changed in Natural.

Metasyntax

The following notations are used in the command descriptions:

[ ]	Square brackets indicate optional parts of a command that can be omitted.
<>	Parameter values are enclosed in angle brackets. Angle brackets must not be typed. Replace text in lower case with an appropriate value. Parentheses can be omitted only when the text they enclose is omitted.
KEYWORD	Keywords are shown in upper case and can be truncated provided that the input is unambiguous.

3 General Predict Commands

- Displaying Help Information 13
- Terminating the Predict Session 13
- ACTIVE 13
- ADMINISTRATE 14
- ASSIGN 14
- BACK 14
- COMPARE 15
- COORDINATOR 15
- DEFAULT 15
- DUPL-LIST ELEMENT 18
- DUPL-SELECT ELEMENT 18
- GENERATE 19
- IMPLEMENT 20
- INCORPORATE 20
- LAST 21
- LOGOFF 21
- LOGON 21
- MAIN 21
- MAINTAIN 21
- MENU 22
- NEWGEN1 22
- NEWEXGEN 22
- NEXT 22
- NEXT - 22
- NPRLOGON 22
- NP2LOGON 23
- PLAN 23
- PREPROCESS 23
- PROFILE 23
- PUNCH / WRITE 24
- REASSIGN ELEMENT 24
- RESETHOT 25

- RETRIEVE 25
- SET 25
- SPECIAL 29

General commands perform session management tasks.

Displaying Help Information

?	Displays the help screens for Predict commands.
??	Displays the Predict help main menu.
? nnnn	Displays the Natural message number nnnn.
? DIC nnnn	Displays the Predict message number nnnn.

Terminating the Predict Session

END or .	Quits the Predict session.
FIN	Executes the Natural FIN command which causes immediate termination of the Predict and Natural session.

ACTIVE

Syntax

```
ACTIVE <object-type>
```

Displays the Active Retrieval Menu for the specified object type.

Object Type	Code	Kind of Object Type
ELEMENT	EL	documentation object
FILE	FI	documentation object
PROGRAM	PR	documentation object
SYSTEM	SY	documentation object
VERIFICATION	VE	documentation object
MEMBER	ME	external object
XREF	XR	function

ADMINISTRATE

Syntax

```
ADMINISTRATE <object-type>
```

Displays the Administration Implementation Menu for the specified object type.

Object Type	Code
DATABASE	DA
DATASPACE	DC
FILE	FI
PROGRAM	PR
STORAGESPACE	SC

ASSIGN

Syntax

```
ASSIGN COORDINATOR
```

Displays the Coordinator Defaults Menu.

```
ASSIGN COORDINATOR DBNR=n FNR=n UTILITY-CLEAR=Y
```

Assigns the Coordinator FDIC in batch mode with the DBNR and FNR specified.

BACK

Switches from Coordinator FDIC to Main FDIC. Returns to the Main FDIC from which you entered the **COORDINATOR** command.

COMPARE

Syntax

```
COMPARE <object-type>
```

Invokes the Comparison function for the specified external object type.

External Object Type	Code	Command
Adabas Database	AD	ADABAS-DATABASE
Adabas File	AF	FDT
Vista Table	AT	VISTATAB
DDM for Natural	DD	DDM
DB2 Database	D2	DB2-DATABASE
Adabas Table/View	EQ	ESQ
Oracle Table	OF	ORACLE-TABLE
DB2 Storagegroup	SG	STORAGEGROUP
DB2 Tables/View	T2	TABLE
DB2 Tablespace	TS	TABLESPACE

The parameters used depend on the type of external object and are listed under Comparison in the *External Objects in Predict* documentation.

COORDINATOR

Switches from Main FDIC to the Coordinator FDIC. Use the **BACK** command to return to your original environment.

DEFAULT

Syntax

```
DEFAULT[<documentation-type>|<external-type>|<command-word>]
```

Depending on which parameter is specified with the DEFAULT command, different defaults of Predict can be changed.

Without

Syntax

```
Without <documentation-type>|<external-type>|<command-word>
```

If only the command `DEFAULT` is entered, the Defaults Menu is displayed.

With

Syntax

```
With <documentation-type>
```

If the command `DEFAULT` is entered with a documentation object type, the extended description skeleton for that object type can be modified. See table below for list of object types and codes.

Documentation Object Type	Code
DATABASE	DA
DATASPACE	DC
EXTRACT	ET
FIELD	EL
FILE	FI
FILE RELATION	RL
INTERFACE	IE
KEYWORD	KY
LIBRARYSTRUCTURE	LS
METHOD	MD
NETWORK	NW
NODE	NO
PACKAGELIST	PG
PROGRAM	PR
PROPERTY	PY
REPORTLISTING	RT
SERVER	SV
STORAGESPACE	SC
SYSTEM	SY
TRIGGER	TR
USER	US
UDE type	user-defined
VERIFICATION	VE

Documentation Object Type	Code
VIRTUALMACHINE	VM

With

Syntax

```
With <external type>
```

If the command `DEFAULT` is entered with an external object type, the generation defaults for that object type can be modified. See table below for list of external object types and codes.

External Object Type	Code	Command Word
ADACMP Definition	AC	WAN, CMP
Adabas File	AF	FDT
Adabas Invert Definitions	AI	ADAINV
Vista Table	AT	VISTATAB
Adabas Security Definitions	AS	SCR
Transparency Table for Adabas VSAM Bridge	AV	AVB, ADAVSAM
Assembler Copy Code	BA	BAL, ASSEMBLER
C Include Code	CC	LANG-C
COBOL Copy Code	CO	COBOL
DB2 Database	D2	DB2-DATABASE
Data Definition Module	DD	DDM
User-defined Fields for IMS	UD	UDF
FORTRAN Copy Code	FO	FORTRAN
PL/I Include Code	PL	PLI
Verification Rule	RU	RULE
DB2 Storagegroup	SG	STORAGEGROUP
DB2 Table/View	T2	TABLE
DB2 Tablespace, SQL/DS DBspace	TS	TABLESPACE

With

Syntax

```
With <command word>
```

If the command DEFAULT is entered with one of the command words in the table below, the corresponding function from the Modify Defaults Menu is called, for example GENERAL executes the Modify general defaults function.

Command Word	Code	Function
EXIT		Activate User Exits
GENERAL		Modify General Defaults
COORDINATOR		Modify Coordinator Defaults
PROFILE		Modify DEFAULT Profile
SQL	SQ	Modify Adabas Native SQL Defaults
XREF	XR	Modify List XRef Default Profile

DUPL-LIST ELEMENT

Syntax

```
DUPL-LIST ELEMENT <standard-file-id>
```

Executes special function Maintain standard fields > List duplicate standard fields. This function lists standard fields that have the same ID and identical standard attributes.

Use the parameter Standard file ID to limit the scope of the function. Asterisk notation is possible. If this parameter is not specified, all duplicated fields in all standard files are listed.

DUPL-SELECT ELEMENT

Syntax

```
DUPL-SELECT ELEMENT <standard-file-id>
```

Executes special function Maintain standard fields > Select duplicate standard fields. This function produces a list of standard fields with duplicate IDs and identical standard attributes. This list can be used to place a command in the workplan to resolve a duplication.

Use the parameter Standard file ID to limit the scope of the function. Asterisk notation is possible. If this parameter is not specified, all duplicated fields in all standard files are listed for selection.

GENERATE

Syntax

```
GENERATE <external-type>
```

Calls the Generation Menu for the external object type.

External Object Type	Code	Command Word
Adabas Compression Definition	AC	WAN, CMP
Adabas File	AF	FDT
Adabas Invert Definitions	AI	ADAINV
Vista Table	AT	VISTATAB
Adabas Security Definitions	AS	SCR
Transparency Table for Adabas VSAM Bridge	AV	AVB, ADAVSAM
Assembler Copy Code	BA	BAL, ASSEMBLER
C Include Code	CC	LANG-C
COBOL Copy Code	CO	COBOL
SQL CREATE Statement	CR	SQL-CREATE
DB2 Database	D2	DB2-DATABASE
Data Definition Module	DD	DDM
FORTRAN Copy Code	FO	FORTRAN
OS/400 File Definitions	O4	GENOS4
Oracle Table	OF	ORACLE-TABLE
PL/I Include Code	PL	PLI
Verification Rule	RL	RULE
DB2 Storagegroup	SG	STORAGEGROUP
DB2 Table/View	T2	TABLE
DB2 Tablespace, SQL/DS DBspace	TS	TABLESPACE
User-defined Fields for IMS	UD	UDF

The parameters used depend on the type of external object and are listed in the respective section of *Generation* in the *External Objects in Predict* documentation.

IMPLEMENT

Syntax

```
MPLEMENT [FILE]
```

Displays the File Implementation Menu.

INCORPORATE

Syntax

```
INCORPORATE <external-type>
```

Displays the Incorporation Menu for the external object type.

External Object Type	Code	Command Word
Adabas D Table/View	BF	ESD-TABLE
Adabas Database	AD	ADABAS-DATABASE
Adabas File	AF	FDT
Adabas Table/View	EQ	ESQ
Vista Table	AT	VISTATAB
COBOL Copy Code	CO	COBOL
DB2 Database	D2	DB2-DATABASE
DB2 Storagegroup	SG	STORAGEGROUP
DB2 Tables/Views	T2	TABLE
DB2 Tablespace	TS	TABLESPACE
DDM	DD	DDM
IMS Database	ND	NDB
Ingres Table/View	JF	INGRES-TABLE
Informix Table/View	XF	INFORMIX-TABLE
Natural Security User	NS	SECURITY
Oracle Table/View	OF	ORACLE-TABLE
Super Natural User	SU	SUPER
Sybase Table/View	YF	SYBASE-TABLE

The parameters used depend on the type of external object. For a list see the respective section under *Incorporation* in the *External Objects in Predict* documentation.

LAST

Displays the last executed command in the command line for modification and/or reexecution.

LOGOFF

Executes the Natural LOGOFF command. See also the *Natural System Commands* documentation and the *Natural Security* documentation.

LOGON

Syntax

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

Executes the Natural LOGON command. For further description see the *Natural System Commands* documentation and the *Natural Security* documentation.

MAIN

Displays the Predict Main Menu. Predict session variables will keep their values.

MAINTAIN

Syntax

```
MAINTAIN <object-type>
```

Displays the Maintenance Menu for the given object type.

MENU

Displays the Predict Main Menu. Predict session variables will be reset.

NEWGEN1

Executes the adaptation of the metadata in the FDIC.

NEWEXGEN

Executes the recreation of the XML documents in the SYSEXT library.

NEXT

Executes the next command in the workplan. See Using the Workplan under *Predict User Interface* in the *Introduction to Predict* documentation.

NEXT -

Executes the last command in the workplan.

NPRLOGON

Syntax

```
NPRLOGON <user> <password> <db-ID>
```

Executes the `LOGON` command for Entire System Server. Parameters are described in the *Entire System Server Reference* documentation.

NP2LOGON

Syntax

```
NP2LOGON <user> <password> <passphrase> <db-ID>
```

Executes the `LOGON` command for Entire System Server if passphrase is activated. Parameters are described in the *Entire System Server Reference* documentation.

PLAN

Displays the user's workplan for execution or deletion of commands.

PREPROCESS

Syntax

```
PREPROCESS <language><member><library><user-ID>
```

Processes the source program on workfile 1 with the preprocessor of the specified language. XRef data is created under the member, library and user ID specified.

PROFILE

Syntax

```
PROFILE [<user-ID>| SYSTEM]
```

If entered without a parameter, this command is used to maintain user profiles. For further details see the section *Customizing Predict with Profiles* in the *Introduction to Predict* documentation.

If entered with a user ID, the profile of the given user is activated for the current Predict session.

If entered with SYSTEM, the default system profile is activated.

PUNCH / WRITE

Syntax

```
PUNCH / WRITE <member><library>
```

Punches copy code that was previously generated into a member in a library on the Predict system file to Natural workfile 1.

REASSIGN ELEMENT

Executes special function Reassign standard relationships.

REASSIGN ELEMENT

Syntax

```
REASSIGN ELEMENT <file 1> <field 1> <file 1> <field 2>
```

This function reassigns all references from one standard field to another standard field within the same file.

REASSIGN ELEMENT

Syntax

```
REASSIGN ELEMENT <file 1> <field 1> <file 2> <field 1>
```

References to a standard field are transferred to a standard field with the same ID in a different standard file.

REASSIGN ELEMENT

Syntax

```
REASSIGN ELEMENT <file 1> <field 1> <file 2> <field 2>
```

References to a standard field are transferred to a standard field with a different ID in a different standard file.

RESETHOT

Resets all currently active hot objects.

RETRIEVE

Syntax

```
RETRIEVE <object-type>
```

Displays the Retrieval Menu for the given object type.

SET

Syntax

```
SET <command word>
```

The following SET commands are available:

- SET CATALOG_USER
- SET OUTPUT
- SET RESTRICTION
- SET SCHEMA_OWNER
- SET VIRTUALMACHINE|VM

SET CATALOG_USER

This command is used for function Generate Adabas Table/View and administration functions Purge and Refresh for Adabas SQL Server objects. Use this command to set the ID and password of an Adabas SQL Server user with at least read access to the catalog. This command is required only once per session.

SET OUTPUT

Defines Output options for subsequent retrieval operations. The output options that are applicable depend on the object type, the retrieval type and the output mode. See tables in the corresponding sections of *Predefined Object Types in Predict* or a cover page of a retrieval operation to see which output options are valid.

Note: The `SET OUTPUT` command must be used with keyword parameters. Positional parameters are not permitted.

Keyword	Output Option	
ABSTRACT-CUR	No. comment lines/current	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.
ABSTRACT-REL	No. comment lines/related	
ADA-SIZE	Adabas sizes	
ADA-VER	Adabas version	
ASSOCIATION-ATTR	Association attr.	
ADA-ATTR	Adabas attributes	
ATTR-CUR	Attributes of current object	
ATTR-REL	Attributes of related object	
CHECK-EXP	Check expression	
COMPILER	Compiler	
COMPONENTS	Composed fields	
CON-FORM	Use Con-Form	
COUNTER-LENGTH	Counter length	
COVER	Cover page	
DESC-CUR	Description/current	
DESC-REL	Description/related	
DESCRIPTORS-ONLY	Field descriptors only	
DIS-LENGTH	Display length	
DIS-MOD	Display modifier	
DUMMIES	Dummy/Placeholder/related	
DUMMY-CUR	Dummy/Placeholder/current	
ENTRY-POINTS	Entry points	
EXPRESSION	DV-field expression	
EXTRACT-CUR	Extracts/current	
EXTRACT-REL	Extracts/related	
GENLAY	Generation layout	
KEY-CUR	Keywords	

Keyword	Output Option
KEY-REL	current/related
LANGUAGE	Language
MARK-IMPL	Mark implementation
NAT-OPTIONS	Natural options
SYNC	Alignment/sync.
POS	Position/Offset
OW-CUR	Owners
OW-REL	current/related
PROCEDURE-CODE	Procedure code
PROFILE	Profile
PS	Page size
REPLACE-SYN	Replace with syn.
RULE	Rules
SHOW-IMPL	Show implementation
SORT-EL	Sorted by field
SUBQUERY	Subquery
SYNONYMS	Synonyms
TRIGGER	Trigger
USEREXIT	User exit
US-CUR	Owners/with users/current
US-REL	Owners/with users/related
VERIFICATIONS	Linked verification
3GL-SPEC	3GL specification

SET RESTRICTION

Defines Restrictions for subsequent functions.

Note: The `SET RESTRICTION` command must be used with keyword parameters. Positional parameters are not permitted.

Keyword	Field
EXTRACT	AND included in extract
FROM-DATE	AND from date Enter FROM-DATE in the following format: YYYY-MM-DD<any non-blank char>HH:II. Date and time must be entered together.
KEY	With keyword(s)

Keyword	Field
KEY2	"
KEY3	"
KEY4	"
KEY5	"
KEY-NOT	BUT NOT with keyword
KEY-OP	combined by (AND/OR)
OWNER	AND with owner
PROFILE	Profile
SCAN-VALUE	And containing the string
SCAN-ABSTRACTS	Scan options/ Abstract **Note:** `SCAN-COMMENTS` can be used as a synonym for `SCAN-ABSTRACT`, so that batch jobs created for earlier versions are still valid.
SCAN-DESC	Scan options/ Descr.
SCAN-RULES	Scan options/ Rules
SCAN-IGNORE-CASE	Scan options /Ignore case
SCAN-OBJ-ID	Scan options/ Object ID
SCAN-ABSOLUTE	Scan options/ Absolute

SET SCHEMA_OWNER

For function Generate Adabas Table/View if you are generating an object which uses the schema of another user. This command sets the ID and password of the schema owner.

SET VIRTUALMACHINE|VM

Defines the Current virtual machine. This value can also be set using the Maintenance Options of the function *Modify User Defaults*.

Keyword	Field	Position
ID	Current virtual machine	01

SPECIAL

Syntax

```
SPECIAL <command-word>
```

Executes functions in the DDA Services/Special Functions Menu. If entered without command word, the DDA Services/Special Functions Menu appears.

If entered with command word, the corresponding Special Functions screen appears. Example: The command SPECIAL DEVICE calls the menu Maintenance Adabas Device Types.

Additional Parameters of SPECIAL Commands

The SPECIAL commands marked with X in the table below can perform complete tasks both online and in batch mode and are described in more detail below.

The other commands merely invoke the corresponding special function submenu.

Function	Code	Predict Command	
Adabas device types	A	SPECIAL DEVICE	
Delete old sets	D	**SPECIAL SET**	X
Maintain Predict help texts	H	SPECIAL HELP	
Reposition Implementation Data	I	**SPECIAL IMPLEMENTATION**	X
Adapt references to copy codes and data areas	C	**SPECIAL ADAPT**	X
Mass Grant	M	SPECIAL MASS	
Maintain NSC Definitions	N	SPECIAL NSC	
Recover	R	**SPECIAL RECOVER**	X
Security Adabas Online Services	S	SPECIAL SECURITY	
Consistency of Predict	P	**SPECIAL CONSISTENCY**	X
Maintain Active References	X	**SPECIAL XREF**	X
Maintain standard fields	Z	SPECIAL STANDARD	
Refresh Coordinator FDIC	F	**SPECIAL REFRESH**	X
Mass delete of report listings	L	SPECIAL DELETE	

SPECIAL CONSISTENCY

Checks the consistency of the Predict data. Special data correction may be performed. Update option must be set to Y, if Predict data is to be modified. If set to N, inconsistencies will only be listed.

When operating in batch mode, the corresponding function code from the menu Check Consistency of Predict Data must appear in the line below the command. The keywords that are applicable depend on the function.

The following parameters may be specified in positional or keyword form in the line below:

Function	Code	Keyword	Field	Position
Check database records	B	UPDATE	With update (Y) only listing (N)	1
		OBJ-ID	Object ID	2
Check the extended description	D	UPDATE	With update (Y) only listing (N)	1
Conversion of the EDIT MASKS in field entries	E	UPDATE	With update (Y) only listing (N)	1
		OBJ-ID	Object ID	2
Check consistency of files and fields	F	UPDATE	With update (Y) only listing (N)	1
		OBJ-ID	Object ID	2
Compress help texts after INPL	H	UPDATE	With update (Y) only listing (N)	1
		OBJ-ID	Object ID	2
Check consistency of keywords	K	UPDATE	With update (Y) only listing (N)	1
Check entries for programs	P	UPDATE	With update (Y) only listing (N)	1
Conversion of free and automatic rules	R	UPDATE	With update (Y) only listing (N)	1
		OBJ-ID	Object ID	2
Check verifications	V	UPDATE	With update (Y) only listing (N)	1
Check naming conventions	N	TYPE	Object type	1
Check XRef data	X	UPDATE	With update (Y) only listing (N)	1
		DBNR	Library/ Database number	2
		FNR	Library/ File number	3
		LIBRARY	Library/ Library name	4
		PROGRAM	Program name	5
		USER	User ID used to create set	6

SPECIAL IMPLEMENTATION

This command is used to adapt database/file numbers and implementation pointers to a new environment when the database/file number of a user system file is changed by an Adabas utility. See the section Reposition Implementation Data under *Special Functions* in the *Predict Administration* documentation.

In batch mode, the following parameters can be entered in keyword or positional form in the line following the command:

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	Program 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

SPECIAL ADAPT

This command is used to adapt references to copy codes and data areas to a new environment. See the section Adapt References to Copy Codes and Data Areas under *Special Functions* in the *Predict Administration* documentation.

In batch mode, the following parameters can be entered in keyword or positional form in the line following the command:

Keyword	Field	Position
OLD-DBNR	Database number (source)	01
OLD-FNR	File number (source)	02
OLD-LIB	Library (source)	03
OLD-MEM	Member (source)	04
OLD-TYPE	Member of type (source)	05
NEW-DBNR	Database number	06
NEW-FNR	File number	07
NEW-LIB	Library	08
USING-DBNR	Database number	09

Keyword	Field	Position
USING-FNR	File number	10
USING-LIB	Library	11
UPDATE	Update	12

SPECIAL RECOVER

Unlocks a file description. Some functions issue an intermediate ET to prevent a hold queue overflow. Until the function finishes successfully, the file description is inconsistent and is therefore locked by Predict. See the section *Recover* under *Special Functions* in the *Predict Administration* documentation.

In batch mode, the following parameter can be entered with or without keyword in the line following the command:

Keyword	Field	Position
FILE-ID	File ID	01

SPECIAL REFRESH

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

See the section *Refresh Coordinator FDIC Special Functions* in the *Predict Administration* documentation.

Keyword	Field	Position
DBNR	Number of the database containing the Coordinator FDIC	1
FNR	File number of the Coordinator FDIC	2
PASSWORD	These parameters are only required if the Coordinator FDIC is protected by password and cipher	3
CIPHER		4

SPECIAL SET

Deletes all sets in every library that were created before a certain date.

In batch mode, the following parameter can be entered with or without keyword in the line following the command:

Keyword	Field	Position	
TO-DATE	To date		1

SPECIAL XREF

Deletes the following types of XRef data.

- Preprocessor abend data
- 3GL data
- Natural data.

See Maintain Active References in the section *Special Functions* in the *Predict Administration* documentation.

Keyword	Field	Position
FUNCTION	Function	1
PR	Program	2
LIB	Library	3
FNR	User system file number	4
DBNR	User system database nr	5

Example: to delete a 3GL program with name “TEST” enter

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

or

```
SPECIAL XREF
G,TEST
```

4 Standard Predict Commands

- Standard Maintenance Commands 36
- Standard Retrieval Commands 37
- Standard Administration Commands 38

Commands described in this section apply to all or most predefined and user-defined object types. The following rules apply to this description:

- For details of individual functions see the sections **Maintenance**, **Retrieval** and **Active Retrieval** in this documentation and Administration of External Objects in the *External Objects in Predict* documentation.
- Type-specific features or parameters of standard commands are described in the respective sections under ***Type-specific Commands***.

Active retrieval commands differ greatly for the different object types. For this reason they are described with the individual object types in this section and in the section ***Active Retrieval***.

Standard Maintenance Commands

The following rules apply to all of standard maintenance commands:

- In the following command descriptions, the predefined Predict object type or the user-defined object to be processed is represented by <object-type>.
- All type-specific features of standard maintenance commands are described in detail in the section ***Type-Specific Commands***.
- If no valid <object-ID> is specified with any of the following commands, the maintenance menu for the respective object type is displayed.

Command Syntax	Description
ADD <object-type><object-ID>	If valid <object-ID> is entered or present in hot object table, this is taken as ID for the new object and one or more Add object screens appear. See the respective sections of Predefined Object Types in Predict for a detailed description of the attributes and corresponding input fields for the predefined object types.
COPY <object-type><object-ID><copy-ID>	If both <object-ID> and copy-ID are entered and are valid, the Copy object screen appears. Specify additional attributes to copy <object-ID> to <copy-ID>. Special rules apply when copying fields.
MODIFY <object-type><object-ID>	If valid <object-ID> is entered or present in hot object table, the Modify Object screen for this object appears.
RENAME <object-type><object-ID>	If valid <object-ID> is entered or present in hot object table, the Rename Object screen for this object appears. Special rules apply when renaming databases and files.
PURGE <object-type><object-ID>	If valid <object-ID> is entered or present in hot object table, the Purge Object screen for this object appears. Special rules apply when purging certain object types. These are described in the respective sections of Predefined Object Types in Predict.

Command Syntax	Description
EDIT <object-type> DESCRIPTION<object-ID>	If valid <object-ID> is entered or present in hot object table, the object extended description of an object can be edited directly.
EDIT <object-type> OWNER<object-ID>	If valid <object-ID> is entered or present in hot object table, the owner list of the object can be edited directly.
LINK <object-type> <active-association-type> <object-ID>	If both parameters are entered and are valid, the list of children of type <active-association-type> for <object-ID> can be edited directly. See **Links to Child and Parent objects** .

Standard Retrieval Commands

Command Syntax	Description
DISPLAY| LIST| SELECT*<object-type><parameters>*	Executes the retrieval type Objects - Code D with output mode Display, List or Select. Information is retrieved on individual objects or range of object IDs.
SHOW *<object-type><parameters>*	Executes the retrieval type Objects - Code D. If the session variables table contains output mode Display, List or Select, this value is taken. If output mode is not contained in this table or is invalid for this retrieval type, output mode will be List.
EXECMODEL *<object-type><parameters>*	Executes the retrieval type Execute retrieval models - Code E with output mode Structured List or Cross Reference. Displays information on the associations of objects in the form of a structured or cross reference list.
DUMMY *<object-type> <passive-association-type><parameters>*	Executes retrieval type Dummy/Placeholder objects - Code C with output mode Select, List or Display. Output option Dummy/Placeholder determines whether dummies and/or placeholders are displayed.
PARENTS *<object-type> <passive-association-type>*| ALL *<parameters>*	Executes retrieval type Objects with parents - Code B with output mode List or Display. Lists objects with parents of type <passive-association-type>. If no object ID is entered, all objects are listed (hot object logic does not apply). If no parent type is entered, value is either taken from hot object table (if present) or a selection window appears.
FREE *<object-type> <passive-association-type>* | ANY | ALL *<parameters>*	Executes retrieval type Objects with no parent - Code O with output mode Select, List or Display. Lists objects with no parent of <passive-association-type> or with no parents of any type. If no object ID is entered, all objects are listed (hot object logic does not apply). If no parent type is entered, value is either taken from hot object table (if present) or a selection window appears.

Command Syntax	Description
	Note: In owner retrieval this command refers to owners with no user only. See ***Owner Retrieval*** for further information.
CHILDREN `<object-type>` `<active-association-type>` | ALL `<parameters>`	Executes retrieval type Objects with children - Code T with output mode List or Display. Lists objects with children of <active-association-type>. If no object ID is entered, all objects are listed (hot object logic does not apply). If no child type is entered, value is either taken from hot object table (if present) or a selection window appears.
EMPTY `<object-type>` `<active-association-type>`| ANY | ALL `<parameters>`	Executes retrieval type Objects with no child - Code U with output mode Select, List or Display. Lists objects with no children of <active-association-type> or with no children of any type. Object ID and child type are stored in the hot object table. If no child type is entered, value is either taken from hot object table (if present) or a selection window appears. **Note:** In owner retrieval this command refers to objects with no owners only. See ***Owner Retrieval*** for further information.

Standard Administration Commands

Before executing these commands see the section Administration of External Objects in the *External Objects in Predict* documentation.

Command Syntax	Description
DISCONNECT`<external-object-type><object-ID>`	Disconnects documentation objects and external objects by deleting the implementation pointer of the documentation objects.
DISPLAY`<external-object-type><object-ID>`	Displays documentation data, generation options and the generation log for the specified external object. For some objects, only the generation log is displayed. See Administration of External Objects in the *External Objects in Predict* documentation for more information.
PURGE`<external-object-type><object-ID>`	Physically deletes the external object and any dependent objects. Not available for all external object types. See *Administration of External Objects* in the *External Objects in Predict* documentation.

Command Syntax	Description
SELECT<*external-object-type*><*object-ID*>	Used to select an external object for further processing with one of the Administration functions.

5 Predict Coordinator Commands

These commands are available from library SYSDIC and offer a subset of the options available from library SYSDICBE. Full functionality is available using the Predict Coordinator. See also the descriptions of each Coordinator command in the Predict Coordinator documentation.

CHECK	Use this command to check the integrity of data on the Coordinator FDIC.
CLEAR	This command deletes the contents of the current Coordinator FDIC file. Use this function to clear the Coordinator FDIC for another import/load operation.
CONTINUE	Use this command to restart an import/load operation that was interrupted.
IMPORT	This command imports all objects from the transfer medium (workfile 1) into the Coordinator FDIC file. The Replace option is set to Yes, that is, objects which exist on the Coordinator FDIC file will be overwritten. After execution of the command IMPORT, a window appears which indicates how many objects were imported and how many objects could not be imported. This command corresponds to the Coordinator command `IMPORT OBJECTTYPE ALL`.

6 Type-Specific Predict Commands

- Organization of this Section 45

Type-specific commands and type-specific parameters used with standard functions are described in this section.

Note: The standard maintenance, retrieval and active retrieval functions and the type-specific functions are described in more detail in the sections ***Maintenance***, ***Retrieval*** and ***Active Retrieval*** and in the respective sections of the *Predefined Object Types in Predict* documentation.

Type-specific commands are available for the following object types:

- **Database**
- **Dataspace**
- **Extract**
- **Field**
- **File**
- **File Relation**
- **Interface**
- **Keyword**
- **Library Structures**
- **Member**
- **Method**
- **Network**
- **Node**
- **Owner**
- **Packagelist**
- **Program**
- **Property**
- **Report Listing**
- **Server**
- **Storagespace**
- **System**
- **Trigger**
- **User**
- **Verification**
- **Virtual Machine**
- **UDE - User-Defined**

Organization of this Section

The section is primarily organized by object type. The organization applying to each section on an object type is shown below. The same organization applies to all object types. Function groups that do not apply are omitted.

Object Type

Maintenance

- Overview of functions and commands
- Parameters
- Type-specific maintenance commands (if applicable)

Retrieval

- Overview of functions and commands
- Parameters
- Type-specific retrieval commands (if applicable)

Active Retrieval - if applicable

- Overview of active retrieval functions and commands
- Parameters
- Type-specific commands (if applicable)

Administration - if applicable

- Functions and commands
- Parameters
- Type-Specific administration commands (if applicable)

7 Database

- Database Maintenance 48
- Database Retrieval 49
- Administration Implemented Database 50

Database Maintenance

The Database Maintenance Menu is invoked with `MAINTAIN DATABASE`.

Overview of Database Maintenance Functions and Commands

Function	Code	Command
Add a database	A	ADD DATABASE
Copy database	C	COPY DATABASE
Modify database	M	MODIFY DATABASE
Rename database	N	RENAME DATABASE
Purge database	P	PURGE DATABASE
Display database	D	DISPLAY DATABASE
Link children	L	LINK DATABASE <active-association-type>
Edit owner of a database	O	EDIT OWNER DATABASE
Select database from a list	S	SELECT DATABASE
Edit description of a database	W	EDIT DATABASE DESCRIPTION
Modify Vista elements	K	MODIFY VISTA-DA

Database Maintenance Command Parameters

Keyword	Field	Position
ID	Database ID	1
<>	Copy ID	2

Database-Specific Maintenance Commands

See also *Standard Maintenance Commands*.

RENAME DATABASE<database-ID>

The following database attributes can be modified in one transaction:

- Database ID
- Database type
- Belongs to VM
- Run mode
- Physical database number

- Vista access only.

Note: These values cannot be changed with the function Modify Database.

Database Retrieval

The Database Retrieval Menu is called with *RETRIEVE DATABASE*.

Overview of Database Retrieval Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Databases	D	D	DISPLAY DATABASE
"	D	L	LIST DATABASE
"	D	S	SELECT DATABASE
"	D	D	SHOW DATABASE
Execute retrieval models	E	T	EXECMODEL DATABASE
Explode IMS database	I	X	EXPLODE DATABASE
Dummy/Placeholder databases	C	L	DUMMY DATABASE <passive-association-type>
Databases with parents	B	D	PARENTS DATABASE <passive-association-type>
Databases with no parent	O	L	FREE DATABASE ANY | ALL | <passive-association-type>
Databases with children	T	D	CHILDREN DATABASE <active-association-type>
Databases with no child	U	L	EMPTY DATABASE ANY | ALL | <active-association-type>

Database Retrieval Command Parameters

Keyword	Field	Position
ID	Database ID	1
PARENT	Belongs to VM	2
RES	Restrictions	3
OUT	Output options	4
TYPE	Database of type	5
DBNR	Database number	6
MODEL	Model	7
MODE	Output mode	8

Database-Specific Retrieval Commands

See also **Standard Retrieval Commands**.

EXPLODE DATABASE	Shows the hierarchical structure of an IMS/DL/I database. Additional search criteria are ignored if a unique database ID is entered. The level number before the file ID shows the level of the IMS/DL/I segment in the hierarchy. This command is only available for databases of type I.

Administration Implemented Database

The Administration Implemented Database Menu is called with ADMINISTRATE DATABASE.

Overview of Database Administration Functions and Commands

Function	Code	Command
Disconnect implementation	C	DISCONNECT DATABASE
Display implementation	D	DISPLAY ADABAS-DATABASE
"		DISPLAY DB2-DATABASE
"		DISPLAY NDB
Purge implementation	P	PURGE DB2-DATABASE
Select implementation	D	SELECT ADABAS-DATABASE
		SELECT DB2-DATABASE
		SELECT NDB

Database Administration Command Parameters

Keyword	Field	Position
ID	Database ID	1
PARENT	Belongs to VM	2
DBNR	Database number	3
TYPE	Database of type	4
MODIFIED	Only modified	5
RES	Restrictions	6

Database-Specific Administration Commands

Objects of this type are processed using ***Standard Administration Commands***.

8 Dataspace

- Dataspace Maintenance 54
- Dataspace Retrieval 54
- Dataspace Administration 56

Dataspace Maintenance

The Dataspace Maintenance Menu is called with `MAINTAIN DATASPACE`.

Overview of Dataspace Maintenance Functions and Commands

Function	Code	Command
Add a dataspace	A	ADD DATASPACE
Copy dataspace	C	COPY DATASPACE
Modify dataspace	M	MODIFY DATASPACE
Rename dataspace	N	RENAME DATASPACE
Purge dataspace	P	PURGE DATASPACE
Display dataspace	D	DISPLAY DATASPACE
Link children	L	LINK DATASPACE <active-association-type>
Edit owners of a dataspace	O	EDIT OWNER DATASPACE
Select dataspace from a list	S	SELECT DATASPACE
Edit description of a dataspace	W	EDIT DATASPACE DESCRIPTION

Dataspace Maintenance Parameters

Keyword	Field	Position
ID	Dataspace ID	1
<>	Copy ID	2

Dataspace-Specific Maintenance Commands

Objects of this type are processed using **Standard Maintenance Commands**.

Dataspace Retrieval

The Dataspace Retrieval Menu is called with *RETRIEVE DATASPACE*.

Overview of Dataspace Retrieval Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Dataspaces	D	D	DISPLAY DATASPACE
"		L	LIST DATASPACE
"		S	SELECT DATASPACE
"		D	SHOW DATASPACE
Execute retrieval models	E	T	EXECMODEL DATASPACE
Dummy/Placeholder dataspaces	C	L	DUMMY DATASPACE <passive-association-type>
Dataspaces with parents	B	D	PARENTS DATASPACE <passive-association-type>
Dataspaces with no parent	O	L	FREE DATASPACE ANY | ALL | <passive-association-type>
Dataspaces with children	T	D	CHILDREN DATASPACE <active-association-type>
Dataspaces with no child	U	L	EMPTY DATASPACE ANY | ALL | <active-association-type>

Dataspace Retrieval Parameters

Keyword	Field	Position
ID	Dataspace ID	1
PARENT	Located in DA	2
RES	Restrictions	3
OUT	Output options	4
MODEL	Model	5
MODE	Output mode	6

Dataspace-Specific Retrieval Commands

Objects of this type are processed using *Standard Retrieval Commands*.

Dataspace Administration

The Administration Implemented Dataspace Menu is called with `ADMINISTRATE DATASPACE`.

Overview of Dataspace Administration Functions and Commands

Function	Code	Command
Disconnect implementation	C	DISCONNECT DATASPACE
Display implementation	D	DISPLAY DATASPACE
Purge implementation	P	PURGE DATASPACE
Select implementation	D	SELECT DATASPACE

Dataspace Administration Command Parameters

Keyword	Field	Position
ID	Dataspace ID	1
PARENT	Located in DA	2
MEM	Member	3
LIB	Library	4
RES	Restrictions	5

Dataspace-Specific Administration Commands

Objects of this type are processed using *Standard Administration Commands*.

9 Extract

- Extract Maintenance 58
- Extract Retrieval 59

Extract Maintenance

The Extract Maintenance Menu is called with the command `MAINTAIN EXTRACT`.

Overview of Functions and Commands

Function	Code	Command
Add an Extract	A	ADD EXTRACT
Copy Extract	C	COPY EXTRACT
Modify Extract	M	MODIFY EXTRACT
Rename Extract	N	RENAME EXTRACT
Purge Extract	P	PURGE EXTRACT
Operate on Extracts	T	OPERATE EXTRACT
Export an Extract	U	
Display Extract	D	DISPLAY EXTRACT
Link children	L	LINK EXTRACT <active-association-type>
Edit owners of a Extract	O	EDIT EXTRACT OWNER
Select Extract from a list	S	DISPLAY EXTRACT
Edit description	W	EDIT EXTRACT DESCRIPTION
Edit/link objects	E	EDIT EXTRACT OBJECTS
Build/extend an Extract	B	BUILD EXTRACT

Parameters

Keyword	Field	Position
ID	Extract ID	1
<>	Copy ID	2
DROP-OBJECTS	Drop existing objects	3
LIST-OBJECTS	List objects	4

Extract-specific Maintenance Commands

See also *Standard Maintenance Commands*.

OPERATE EXTRACT <Extract-ID>	Executes the function Operate on Extracts. See the section *Extract* in the Predefined Object Types in Predict documentation.
EDIT EXTRACT OBJECTS <Extract-ID>	With this command you skip the Modify Extract screen and access the object list directly. The system behavior depends on the editor you are using. ■ If you are using the Software AG Editor, the Object List Editor is called. ■ If you are not using the Software AG Editor, the function Link Keyword to Object is called.
BUILD EXTRACT <Extract-ID>	This command is only used *online*. It calls the function Build/extend an Extract for the specified extract.
BUILD EXTRACT <Extract-ID> <Retrieval Command>	This command is only used in *batch mode*. If you execute this command with object type all, the only keyword that is valid is ID.

Extract Retrieval

The Extract Retrieval Menu is called with command `RETRIEVE EXTRACT`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Extracts	D	D	DISPLAY EXTRACT
"		L	LIST EXTRACT
"		S	SELECT EXTRACT
"		D	SHOW EXTRACT
Execute retrieval models	E	T	EXECMODEL EXTRACT
Dummy Extracts	C	L	DUMMY EXTRACT<passive-association-type>
Extracts related to no object	Y	L	UNUSED EXTRACT
Extracts related to objects	X	X	XREF EXTRACT
Extracts with parents	B	D	PARENTS EXTRACT <passive-association-type>
Extracts with no parent	O	L	FREE EXTRACT ANY| ALL| <passive-association-type>
Extracts with children	T	D	CHILDREN EXTRACT <active-association-type>

Retrieval Type	Code	Default Output Mode	Command
Extracts with no child	U	L	EMPTY EXTRACT ANY| ALL| <active-association-type>

Parameters

Keyword	Field	Position
ID	Extract ID	1
RES	Restrictions	2
OUT	Output options	3
MODEL	Model	4
MODE	Output mode	5

Extract-specific Retrieval Commands

See also **Standard Retrieval Commands**.

UNUSED EXTRACT	This function lists extracts that have no objects assigned to them.
XREF EXTRACT	Lists all objects contained in the specified extract or - if asterisk notation is used - in the specified extracts.

10 Field

- Field Maintenance 62
- Field Retrieval 64
- Field Active Retrieval 65
- Field-specific Special Functions 66

Field Maintenance

The Field Maintenance Menu is called with `MAINTAIN ELEMENT`.

Overview of Field Maintenance Functions and Commands

Function	Code	Command
Add a field	A	ADD ELEMENT
Copy field	C	COPY ELEMENT
Display field	D	DISPLAY ELEMENT
Modify field	M	MODIFY ELEMENT
Rename field	N	RENAME ELEMENT
Purge field	P	PURGE ELEMENT
Redefine field	R	REDEFINE ELEMENT
Browse through fields of a file	B	BROWSE ELEMENT
Move field within a file	H	MOVE ELEMENT
Link children	L	LINK ELEMENT <active-association-type>
Edit owners of a field	O	EDIT ELEMENT OWNER
Select field from a list	S	SELECT ELEMENT
Edit description of a field	W	EDIT ELEMENT DESCRIPTION
Edit field expression	Y	EDIT ELEMENT EXPRESSION

Field Maintenance Parameters

Keyword	Field	Position
ID	Field ID	1
<>	in file	2
<>	Copy field ID	3
<>	Copy file ID	4

Field-Specific Maintenance Commands

<table>
<tr><th>Maintenance Command</th><th>Description</th></tr>
<tr><td>ADD ELEMENT
<field-ID><file-ID><copy-field-ID></td><td>When applied to fields, the standard maintenance function Add offers additional options. A field can be added:
<table>
<tr><td>Field ID</td><td>field_1</td><td>field_1</td><td>field_1</td></tr>
<tr><td>in file</td><td>file_1</td><td>file_1</td><td>file_1</td></tr>
<tr><td>Copy field ID</td><td></td><td>*</td><td>field_2</td></tr>
<tr><td>Copy file ID</td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td>COPY ELEMENT
<field-ID><file-ID><copy-field-ID><copy-file-ID></td><td>When applied to fields, the standard maintenance function Copy offers additional options. A field can be copied:
<table>
<tr><td>Field ID</td><td>field_1</td><td>field_1</td><td>field_1</td></tr>
<tr><td>in file</td><td>file_1</td><td>file_1</td><td>file_1</td></tr>
<tr><td>Copy field ID</td><td>field_2</td><td>field_2</td><td></td></tr>
<tr><td>Copy file ID</td><td></td><td>file_2</td><td>file_2</td></tr>
</table></td></tr>
<tr><td>REDEFINE ELEMENT<field-ID><file-ID></td><td>Calls the editor for the function Redefine field.</td></tr>
<tr><td>BROWSE ELEMENT<field-ID><file-ID></td><td>Invokes the Modify field function for each field in the field list of a file, starting with the field specified. If no field ID is given, the file is browsed from the first field in the list.
This function is especially useful when applying general changes to all fields in a file starting with the field ID specified.</td></tr>
<tr><td>MOVE ELEMENT<field-ID><file-ID>
<copy-field-ID></td><td>This command is used to change the order of fields in a file. A field can be moved:
<table>
<tr><td>Field ID</td><td>field_1</td><td>field_1</td></tr>
<tr><td>in file</td><td>file_1</td><td>file_1</td></tr>
<tr><td>Copy field ID</td><td>*</td><td>field_2</td></tr>
<tr><td>Copy file ID</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>EDIT ELEMENT EXPRESSION<field-ID><file-ID></td><td>Calls function Edit field expression. Applicable only to fields of type DV in files of the following types:
<table>
<tr><td>B</td><td>Adabas SQL views</td></tr>
<tr><td>E</td><td>DB2 view</td></tr>
<tr><td>YV</td><td>Sybase view</td></tr>
</table></td></tr>
</table>

Field Retrieval

The Field Retrieval Menu is called with `RETRIEVE ELEMENT`.

Overview of Field Retrieval Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Fields	D	D	DISPLAY ELEMENT
		L	LIST ELEMENT
		S	SELECT ELEMENT
		D	SHOW ELEMENT
Executive retrieval models	E	T	EXECMODEL ELEMENT
		T	MODEL=IM IMPLODE ELEMENT This command will be dropped in a future version.
		X	MODEL=XR XREF ELEMENT This command will be dropped in a future version.
Non-standard fields	N	L	NONSTANDARD ELEMENT
Fields and related views	R	L	RELATED ELEMENT
Fields with parent	B	D	PARENTS FIELDS <passive-association-type>
Fields with no parent	O	L	FREE FIELDS | ANY | ALL | <passive-association-type>
Fields with children	T	D	CHILDREN ELEMENT <active-association-type>
Fields with no children	U	L	EMPTY ELEMENT | ANY | ALL | <active-association-type>
Fields related to a Z-file	Z	L	STANDARD ELEMENT

Field Retrieval Parameters

Keyword	Field	Position
ID	Field ID/Synonym	1
FILE	in file	2
RES	Restrictions	3
OUT	Output options	4
LANGUAGE	Synonym of language	5
TYPE	in files of type	6
MODEL	Model	7

Keyword	Field	Position
MODE	Output mode	8

Field-Specific Retrieval Commands

See also *Standard Retrieval Commands*.

Retrieval Command	Description
NONSTANDARD ELEMENT	Lists all fields which have not been pulled forward from a standard file or have been marked subsequently as non-standard. If both a field name and file name are entered without an asterisk, the field name is ignored and all fields of the specified file which satisfy the other criteria are listed.
RELATED ELEMENT	Lists fields in a userview together with the fields in the master file(s) of the userview from which they are derived. If both a field name and file name are entered without an asterisk, the field name is ignored and all fields of the specified file which satisfy the other criteria are listed.
STANDARD ELEMENT	Lists the fields of standard files with their related fields in other files. If both a field name and file name are entered without an asterisk, the field name is ignored and all fields of the specified file which satisfy the other criteria are listed.

Field Active Retrieval

The Field Active Retrieval Menu is called with ACTIVE ELEMENT.

Overview of Field Active Retrieval Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Fields			see *Field Retrieval*
Fields ref. by members	R	L	REFERENCED ELEMENT
		C	COUNT ELEMENT
Fields inconsistently used	U	L	UNUSED ELEMENT
Fields not documented	O	L	UNDEFINED ELEMENT

Field Active Retrieval Parameters

Keyword	Field	Position
ID	Field ID/Synonym	1
FILE	in file	2
RES	Restrictions	3
OUT	Output options	4
LANGUAGE	Synonym language	5
USE	Usage	6
MASTER	For master fields	7
SET	Save set	8
MODE	Output mode	not applicable

Field-Specific Active Retrieval Commands

Active Retrieval Command	Description
REFERENCED ELEMENT	Lists for every referenced field all members using the field.
COUNT ELEMENT	Shows for every referenced field a summary list which indicates for each library and each field usage (for example read or update) the number of members that use the field.
UNUSED ELEMENT	Lists fields that are not used by any member or which may be used incorrectly. If parameter Usage is specified, only fields *not* used in the specified manner are shown.
UNDEFINED ELEMENT	Lists fields in implemented files that are not documented in Predict.

Field-specific Special Functions

Function	Description
DUPL-LIST ELEMENT <file-ID>	Executes the function List duplicate standard fields. This function lists standard fields with duplicate IDs.
DUPL-SEL ELEMENT <file-ID>	Executes the function Select duplicate standard fields. This function selects standard fields with duplicate IDs.

Function	Description			
SHIFT ELEMENT *<file 1> <field 1> <file 2>*	Executes the function Move field to other standard file. This function assigns a standard field to another standard file. The field specified with parameter From field ID is deleted and added as To field ID. All references to the old standard field are transferred to the new standard field.			
	-	From File ID	1	
	-	From Field ID	2	
	-	To File ID	3	
REASSIGN ELEMENT	Executes the function Reassign standard relationships. References to a standard field are merged with references to another standard field either in the same or in a different standard file.			
	From File ID	file1	file1	file1
	From Field ID	field1	field1	field1
	To File ID	file1	file2	file2
	To Field ID	field2	field1	field2

11 File

- File Maintenance 70
- File Retrieval 72
- File Active Retrieval 74
- Administration Implemented File 75

File Maintenance

The File Maintenance Menu is called with the command `MAINTAIN FILE`.

Overview of File Maintenance Functions and Commands

Function	Code	Command
Add a file	A	ADD FILE
Copy file	C	COPY FILE
Display file	D	DISPLAY FILE
Modify file	M	MODIFY FILE
Rename/Renumber file	N	RENAME FILE
Purge file	P	PURGE FILE
Edit description of a file	W	EDIT FILE DESCRIPTION
Edit elements of a file (see Note:)	E	EDIT FILE ELEMENT
Push backward	B	PUSH FILE
Link children	L	LINK FILE <active-association-type>
Edit owners of a file	O	EDIT FILE OWNER
Select file from a list	S	SELECT FILE
Modify Adabas attributes	J	MODIFY ADA-ATTR
Force standard	F	FORCE FILE
Modify Vista elements	K	MODIFY VISTA-FI
Edit Subquery of a File	Y	EDIT FILE SUBQUERY

Note: The function Edit elements of a file does not appear on the File Maintenance Menu, but can still be called with function code E.

The command `FORCE FILE` is one of the few maintenance commands that can be executed in batch mode.

File Maintenance Parameters

Keyword	Field	Position
ID	File ID	1
<>	Copy ID	2

File-Specific Maintenance Commands

Note: See also **Standard Maintenance Commands** .

Command	Description
RENAME FILE<file-ID>	This command is used to modify the following file attributes in one transaction: ■ file ID ■ file number ■ file type ■ master file ■ logical distribution type ■ Adabas SQL usage **Note:** These values cannot be changed with the Modify Database function.
PUSH FILE<file-ID><copy-file-ID>	The field definitions of the specified file are copied into the standard file according to the following rules: ■ If the field is already present in the standard file and a link exists between the specified file and the standard file, no action is taken ■ If the field is already present in the standard file but no link exists, a link is established. ■ If the field is not found in the standard file, it is copied and a link is established. The standard file is specified in the field Copy file ID. The coupling of field definitions on different levels to ensure consistent data definitions is described in the section *Rippling* under *File* of the Predefined Object Types in Predict documentation.
FORCE FILE<file-ID>	Compares the coupled attributes of all fields defined in the specified standard file with the attributes of the coupled fields in other files. If attributes of coupled fields are different (and these fields are not marked as non-standard), change them to match the standard file.
MODIFY ADA-ATTR <file-ID>	Modifies the physical Adabas attributes of Adabas files.
MODIFY VISTA-FI <file-ID>	Modifies Vista attributes.

Command	Description
EDIT FILE SUBQUERY <file-ID>	Invokes the Subquery Editor, a modified Natural Editor which is used to edit the following: ■ Subquery clause of SQL view ■ Derived field expression
EDIT FILE ELEMENT <file-ID>	Reads the field list of the specified file into the source area and activates the full screen editor which supports many edit and line commands.

File Retrieval

The File Retrieval Menu is called with RETRIEVE FILE.

Overview of File Retrieval Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Files	D	D	DISPLAY FILE
		L	LIST FILE
		S	SELECT FILE
		D	SHOW FILE
Execute retrieval models	E	T	EXECMODEL FILE
Dummy/Placeholder files	C	L	DUMMY FILE <passive-association-type>
Difference of files	A		DIFFERENCE FILE
Files with parents	B	D	PARENTS FILE <passive-association-type>
Files with no parent	O	L	FREE FILE ANY | ALL | <passive-association-type>
Files with children	T	D	CHILDREN FILE <active-association-type>
Files with no child	U	L	EMPTY FILE ANY | ALL | <active-association-type>
Files related to a file	R	L	RELATED FILE

File Retrieval Parameters

Keyword	Field	Position
ID	File ID	1
PARENT	Contained in DA	2
EXTERNAL	External name	3
RES	Restrictions	4
OUT	Output options	5

Keyword	Field	Position
TYPE	Files of type	6
FNR	File number	7
MODEL	Model	8
MODE	Output mode	9

File-Specific Retrieval Commands

See also **Standard Retrieval Commands**.

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>DIFFERENCE FILE</td><td>Compares files and displays the differences. The file attributes, the fields and the field attributes can be compared. The fields are compared using the field ID.
If a userview is compared with its master file, however, the fields are compared by two-character Short name. The userview is always taken as first file, irrespective of which file is entered under FIRST-FILE. See the section File in the Predefined Object Types in Predict documentation for more information.</td></tr>
<tr><td>Using DIFFERENCE FILE in Batch Mode</td><td>The parameters FIRST-FILE, SECOND-FILE, LIST and FILE-ATTR are mandatory. Each field attribute to be compared must be specified.
<table>
<tr><th>Keyword</th><th>Field</th><th>Position</th></tr>
<tr><td>FIRST-FILE;</td><td>First file ID</td><td>1</td></tr>
<tr><td>SECOND-FILE;</td><td>Second file ID</td><td>2</td></tr>
<tr><td>LIST</td><td>List fields</td><td>3</td></tr>
<tr><td>FILE-ATTR</td><td>Compare file attributes</td><td>4</td></tr>
<tr><td>ORDER;</td><td>the order</td><td>5</td></tr>
<tr><td>EXISTENCE</td><td>the existence</td><td>6</td></tr>
<tr><td>COMMENT</td><td>abstract</td><td>7</td></tr>
<tr><td>OW;</td><td>owner IDs</td><td>8</td></tr>
<tr><td>KEY</td><td>keywords</td><td>9</td></tr>
<tr><td>DESC</td><td>description</td><td>10</td></tr>
<tr><td>SYNONYM</td><td>field name synonyms</td><td>11</td></tr>
<tr><td>STANDARD</td><td>standard file</td><td>12</td></tr>
<tr><td>VERIFICATION</td><td>verification</td><td>13</td></tr>
<tr><td>ADABAS-SEC</td><td>Adabas attributes</td><td>14</td></tr>
<tr><td>FIELD-DEF</td><td>field definition</td><td>15</td></tr>
<tr><td>NATURAL</td><td>Natural attributes</td><td>16</td></tr>
<tr><td>3GL-SPECIF</td><td>specification for 3GL</td><td>17</td></tr>
<tr><td>VSAM</td><td>VSAM attributes</td><td>18</td></tr>
</table></td></tr>
</table>

Command	Description

	Example: Compare file PDA-7 with file PDA-13 and report the differences of the field's Adabas and Natural attributes:
	
	`DIFFERENCE FILE`
	
	`PDA-7,PDA-13,LIST=D,N,ADABAS-SEC=X,,X`
	
	Note: For the field attributes to be compared, the corresponding keyword must be specified in the statement DIFFERENCE FILE, but the value assigned to the keyword is of no importance in this case. If a field attribute is not to be compared, the corresponding keyword must be omitted.

RELATED FILE<*file-ID*>	Lists the following types of objects: ■ master files with their userviews ■ userviews with their master files ■ other userviews of these master files For physical VSAM files the related logical VSAM files are also listed; for IMS segments the IMS segment layouts are also listed.

File Active Retrieval

The File Active Retrieval Menu is called with `ACTIVE FILE`.

Overview of File Active Retrieval Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Files			see *File Retrieval*
Files ref. by members	R	L	REFERENCED FILE
		C	COUNT FILE
Files inconsistently used	U	L	UNUSED FILE
Files not documented	O	L	UNDEFINED FILE
Files accessed via dyn. SQL	V	L	DYNAMIC FILE

File Active Retrieval Parameters

Keyword	Field	Position
ID	File ID	1
PARENT	Contained in DA	2
RES	Restrictions	3
OUT	Output options	4
TYPE	Files of type	5
FNR	File number	6
USE	Usage	7
MASTER	For master files	8
SET	Save set	9
MODE	Output mode	not applicable

File-Specific Active Retrieval Commands

Command	Description
REFERENCED FILE	Lists for every referenced file all members using the file.
COUNT FILE	Shows for every referenced file a summary list which indicates for each library how many members use the file and the file usage, for example read or update. If members of more than one library use the file, the total usages are also given.
UNUSED FILE	Lists files that are not used by any member or which may be used incorrectly. Example: a file that is never updated or never read. If the parameter Usage is specified, only those files which are *not* used in this manner are shown.
UNDEFINED FILE	Lists files that are referenced by members but are not documented in Predict and have no DDM.
DYNAMIC FILE	Lists DB2 tables and views (file types D or E) that are used by Natural members but not via static SQL.

Administration Implemented File

The Administration Implemented File menu is called with ADMINISTRATE FILE.

Overview of File Administration Functions and Commands

Function	Code	Command
Disconnect implementation	C	DISCONNECT FDT
		DISCONNECT DDM
		DISCONNECT NDB
		DISCONNECT TABLE
		DISCONNECT ESQ
		DISCONNECT SN-FILE
		DISCONNECT NSC-FILE
Display implementation	D	DISPLAY ADACMP
		DISPLAY ADAINV
		DISPLAY ADASCR
		DISPLAY ADAVSAM
		DISPLAY BAL
		DISPLAY COBOL
		DISPLAY DDM
		DISPLAY ESQ
		DISPLAY FDT
		DISPLAY FORTRAN
		DISPLAY LANG-C
		DISPLAY PLI
		DISPLAY RULE
		DISPLAY TABLE
Rename implementation	N	RENAME ADACMP
		RENAME ADAINV
		RENAME ADASCR
		RENAME ADAVSAM
		RENAME BAL
		RENAME COBOL
		RENAME FORTRAN
		RENAME LANG-C
		RENAME PLI
		RENAME TABLE
Purge implementation	P	PURGE ADACMP
		PURGE ADAINV
		PURGE ADASCR

Function	Code	Command
		PURGE ADAVSAM
		PURGE BAL
		PURGE COBOL
		PURGE ESQ
		PURGE FDT
		PURGE FORTRAN
		PURGE LANG-C
		PURGE NDB
		PURGE NSC-FILE
		PURGE PLI
		PURGE SN-FILE
		PURGE TABLE
Refresh file	R	REFRESH FDT
		REFRESH TABLE
Select implementation	S	SELECT ADACMP
		SELECT ADAINV
		SELECT ADASCR
		SELECT ADAVSAM
		SELECT BAL
		SELECT COBOL
		SELECT DDM
		SELECT ESQ
		SELECT FDT
		SELECT FORTRAN
		SELECT LANG-C
		SELECT NDB
		SELECT NSC-FILE
		SELECT PLI
		SELECT RULE
		SELECT SN-FILE
		SELECT TABLE

File Administration Parameters

Keyword	Field	Position
ID	File ID	1
PARENT	Contained in DA	2
EXTERNAL	External name	3
MEM	Member	4
LIB	Library	5
MODIFIED	Only modified	6
RES	Restrictions	7
TYPE	Files of type	8
FNR	File number	9
PFNR	Phys Fnr	10
PDBNR	Phys DBnr	11

File-specific Administration Commands

See **Standard Administration Commands**.

Command	Description
REFRESH FILE	Deletes all records stored in Adabas files or DB2 tables/views. The data structure remains intact.
PURGE VISTATAB	Deletes entries in Vista translation tables. This function cannot be executed in batch mode.

12 File Relation

- File Relation Maintenance 80
- File Relation Retrieval 80

File Relation Maintenance

The File Relation Maintenance Menu is called with command `MAINTAIN FILERELATION`.

Overview of Functions and Commands

Function	Code	Command
Add a file relation	A	ADD FILERELATION
Copy file relation	C	COPY FILERELATION
Modify file relation	M	MODIFY FILERELATION
Rename file relation	N	RENAME FILERELATION
Purge file relation	P	PURGE FILERELATION
Display file relation	D	DISPLAY FILERELATION
Link children	L	LINK FILERELATION <active-association-type>
Edit owners of rel.	O	EDIT FILERELATION OWNER
Select file relation	S	SELECT FILERELATION
Edit description of a file relation	W	EDIT FILERELATION DESCRIPTION

Parameters

Keyword	Field	Position
ID	File relation ID	1
<>	Copy ID	2

File Relation-specific Maintenance Commands

File relation objects are processed using ***Standard Maintenance Commands***.

File Relation Retrieval

The File Relation Retrieval Menu is called with command *RETRIEVE FILERELATION*.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
File relations	D	D	DISPLAY FILERELATION
		L	LIST FILERELATION
		S	SELECT FILERELATION
		D	SHOW FILERELATION
Execute retrieval models	E	T	EXECMODEL FILERELATION
Dummy/Placeholder file relations	C	L	DUMMY FILERELATION <passive-association-type>
File relations with parents	B	D	PARENTS FILERELATION <passive-association-type>
File relations with no parent	O	L	FREE FILERELATION ANY | ALL | <passive-association-type>
File relations with children	T	D	CHILDREN FILERELATION <active-association-type>
File relations with no child	U	L	EMPTY FILERELATION ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	File relation ID	1
FILE	using file	2
RES	Restrictions	3
OUT	Output options	4
MODEL	Model	5
MODE	Output mode	6

File Relation Specific Retrieval Commands

File relation objects are processed with *Standard Retrieval Commands*.

13 Interface

- Interface Maintenance 84
- Interface Retrieval 84

Interface Maintenance

The Interface Maintenance Menu is called with the command `MAINTAIN INTERFACE`.

Overview of Functions and Commands

Function	Code	Command
Add an interface	A	ADD INTERFACE
Copy interface	C	COPY INTERFACE
Modify interface	M	MODIFY INTERFACE
Rename interface	N	RENAME INTERFACE
Purge interface	P	PURGE INTERFACE
Display interface	D	DISPLAY INTERFACE
Link children	L	LINK INTERFACE <active-association-type>
Edit owners of an interface	O	EDIT INTERFACE OWNER
Select interface from a list	S	SELECT INTERFACE
Edit description of an interface	W	EDIT INTERFACE DESCRIPTION

Parameters

Interface	Field	Position
ID	Interface ID	1
<>	Copy ID	2

Interface-specific Maintenance Commands

Interface objects are processed using *Standard Maintenance Commands*.

Interface Retrieval

The Interface Retrieval Menu is called with the command `RETRIEVE INTERFACE`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Interfaces	D	D	DISPLAY INTERFACE
		L	LIST INTERFACE
		S	SELECT INTERFACE
		D	SHOW INTERFACE
Execute retrieval model	E	T	EXECMODEL INTERFACE
Dummy/Placeholder interfaces	C	L	DUMMY INTERFACE <passive-association-type>
Interfaces with parents	B	D	PARENTS INTERFACE <passive-association-type>
Interfaces with no parent	O	L	FREE INTERFACE ANY | ALL | <passive-association-type>
Interfaces with children	T	D	CHILDREN INTERFACE <active-association-type>
Interfaces with no child	U	L	EMPTY INTERFACE ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	Interface ID	1
RES	Restrictions	2
OUT	Output options	3
MODEL	Model	4
MODE	Output mode	5

Interface-specific Retrieval Commands

See also ***Standard Retrieval Commands***.

14 Keyword

- Keyword Maintenance 88
- Keyword Retrieval 88

Keyword Maintenance

The Keyword Maintenance Menu is called with the command `MAINTAIN KEYWORD`.

Overview of Functions and Commands

Function	Code	Command
Add a keyword	A	ADD KEYWORD
Copy keyword	C	COPY KEYWORD
Modify keyword	M	MODIFY KEYWORD
Rename keyword	N	RENAME KEYWORD
Purge keyword	P	PURGE KEYWORD
Display keyword	D	DISPLAY KEYWORD
Link children	L	LINK KEYWORD <active-association-type>
Edit owners of a keyword	O	EDIT KEYWORD OWNER
Select keyword from a list	S	SELECT KEYWORD
Edit description of a keyword	W	EDIT KEYWORD DESCRIPTION
Link/Unlink objects	E	EDIT KEYWORD OBJECTS

Parameters

Keyword	Field	Position
ID	Keyword ID	1
<>	Copy ID	2

Keyword-specific Maintenance Commands

Keyword objects are processed using *Standard Maintenance Commands*.

Keyword Retrieval

The Keyword Retrieval Menu is called with the command `RETRIEVE KEYWORD`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Keywords	D	D	DISPLAY KEYWORD
		L	LIST KEYWORD
		S	SELECT KEYWORD
		D	SHOW KEYWORD
Execute retrieval model	E	T	EXECMODEL KEYWORD
Dummy/Placeholder keywords	C	L	DUMMY KEYWORD <passive-association-type>
Keywords with parents	B	D	PARENTS KEYWORD <passive-association-type>
Keywords with no parent	O	L	FREE KEYWORD ANY| ALL| <passive-association-type>
Keywords with children	T	D	CHILDREN KEYWORD <active-association-type>
Keywords with no child	U	L	EMPTY KEYWORD ANY| ALL| <active-association-type>
Keywords related to no object	Y	L	UNUSED KEYWORD
Cross reference keyword	X	X	XREF KEYWORD

Parameters

Keyword	Field	Position
ID	Keyword	1
RES	Restrictions	2
OUT	Output options	3
MODEL	Model	4
MODE	Output mode	5

Keyword-specific Retrieval Commands

See also **Standard Retrieval Commands** .

Command	Description
UNUSED KEYWORD	Lists keywords that are not linked to any object. A message is given if all keywords that meet the selection criteria are related to other objects.
XREF KEYWORD	Displays the following information: ■ Keywords ■ objects that are linked to a keyword ■ other keywords linked to these objects.

15 Library Structure

- Library Structure Maintenance 92
- Library Structure Retrieval 92

Library Structure Maintenance

The Library structure Maintenance Menu is called with the command `MAINTAIN LIBRARYSTRUCTURE`.

Overview of Functions and Commands

Function	Code	Command
Add a Library structure	A	ADD LIBRARYSTRUCTURE
Copy Library structure	C	COPY LIBRARYSTRUCTURE
Modify Library structure	M	MODIFY LIBRARYSTRUCTURE
Rename Library structure	N	RENAME LIBRARYSTRUCTURE
Purge Library structure	P	PURGE LIBRARYSTRUCTURE
Display Library structure	D	DISPLAY LIBRARYSTRUCTURE
Link children	L	LINK LIBRARYSTRUCTURE <active-association-type>
Edit owners of a Library structure	O	EDIT LIBRARYSTRUCTURE OWNER
Select Library structure from a list	S	SELECT LIBRARYSTRUCTURE
Edit description	W	EDIT LIBRARYSTRUCTURE DESCRIPTION

Parameters

Keyword	Field	Position
ID	Library structure ID	1
<>	Copy ID	2

Library Structure Specific Maintenance Commands

Library structure objects are processed using *Standard Maintenance Commands*.

Library Structure Retrieval

The Library structure Retrieval Menu is called with the command `RETRIEVE LIBRARYSTRUCTURE`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Library structures	D	D	DISPLAY LIBRARYSTRUCTURE
"		L	LIST LIBRARYSTRUCTURE
"		S	SELECT LIBRARYSTRUCTURE
"		D	SHOW LIBRARYSTRUCTURE
Execute retrieval models	E	T	EXECMODEL LIBRARYSTRUCTURE
Dummy/Placeholder Library str.	C	L	DUMMY LIBRARYSTRUCTURE <passive-association-type>
Library structure with parents	B	D	PARENTS LIBRARYSTRUCTURE <passive-association-type>
Library structure with no parent	O	L	FREE LIBRARYSTRUCTURE ANY | ALL | <passive-association-type>
Library structure with children	T	D	CHILDREN LIBRARYSTRUCTURE <active-association-type>
Library structure with no child	U	L	EMPTY LIBRARYSTRUCTURE ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	Library structure ID	1
RES	Restrictions	2
OUT	Output options	3
MODEL	Model	4
MODE	Output mode	5

Library Structure-specific Retrieval Commands

Library structure objects are processed with ***Standard Retrieval Commands***.

16 Member

- Member Active Retrieval 96

Member Active Retrieval

The Member Active Retrieval Menu is called with the command `ACTIVE MEMBER`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Members	I	L	LIST MEMBER
		S	SELECT MEMBER
Entries ref. by members	R	L	REFERENCED ENTRY
	R	C	COUNT ENTRY
Entries not referenced	U	L	UNUSED ENTRY
Members not documented	O	L	UNDEFINED MEMBER
Members with used files	T	L	ELEMENTS MEMBER
Members using dyn. SQL	V	L	DYNAMIC MEMBER
Members ref by members	M	L	REFERENCED MEMBER

Parameters

Keyword	Field	Position
MEM	Member	1
LIB	Library	2
FROM-DATE	From catalog date	3
TO-DATE	To catalog date	4
ENTRY	Entry	5
OUT	Output options	6
FNR	User system Fnr	7
DBNR	User system DBnr	8
TYPE	Member type	9
SET	Save set	10
MODE	Output mode	not possible

Member-specific Active Retrieval Commands

Command	Description
LIST MEMBER	Displays the following information: ■ type, language, file number, database number of the member ■ user ID, terminal ID, date and time of cataloging ■ documented program together with implementation pointer.
SELECT MEMBER	Displays the following information in a selection list: ■ type, language, file number, database number of the member ■ program in which the member is documented or the screen text >>>Member not documented<<< Members can be selected from the list for further processing or commands can be placed in the workplan.
REFERENCED ENTRY	Lists members that refer to the specified external entry point.
COUNT ENTRY	Lists the libraries which use the specified external entry point and indicates the number of references per library.
UNUSED ENTRY	Lists entry points which are not referenced by any other member.
UNDEFINED MEMBER	Lists members which have been cataloged or precompiled with XRef data but are not documented in Predict.
ELEMENTS MEMBER	Lists members and the files that they use.
DYNAMIC MEMBER	Lists Natural members that use files of type DB2 table or DB2 view (file type D or E) but not via Static SQL.
REFERENCED MEMBER	Lists members that refer to the specified member. In difference to LIST XREF functions, no library structure is required, since the whole XRef data is searched

17 Method

- Method Maintenance 100
- Method Retrieval 100

Method Maintenance

The Method Maintenance Menu is called with the command `MAINTAIN METHOD`.

Overview of Functions and Commands

Function	Code	Command
Add a method	A	ADD METHOD
Copy method	C	COPY METHOD
Modify method	M	MODIFY METHOD
Rename method	N	RENAME METHOD
Purge method	P	PURGE METHOD
Display method	D	DISPLAY METHOD
Link children	L	LINK METHOD <active-association-type>
Edit owners of a method	O	EDIT METHOD OWNER
Select method from a list	S	SELECT METHOD
Edit description of a method	W	EDIT METHOD DESCRIPTION

Parameters

Interface	Field	Position
ID	Method ID	1
<>	Copy ID	2

Method-specific Maintenance Commands

Method objects are processed using *Standard Maintenance Commands*.

Method Retrieval

The Method Retrieval Menu is called with the command `RETRIEVE METHOD`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Methods	D	D	DISPLAY METHOD
		L	LIST METHOD
		S	SELECT METHOD
		D	SHOW METHOD
Execute retrieval model	E	T	EXECMODEL METHOD
Dummy/Placeholder methods	C	L	DUMMY METHOD <passive-association-type>
Methods with parents	B	D	PARENTS METHOD <passive-association-type>
Methods with no parent	O	L	FREE METHOD ANY | ALL | <passive-association-type>
Methods with children	T	D	CHILDREN METHOD <active-association-type>
Methods with no child	U	L	EMPTY METHOD ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	Method	1
PARENT	Belongs to IE	2
RES	Restrictions	3
OUT	Output options	4
MODEL	Model	5
MODE	Output mode	6

Method-specific Retrieval Commands

See *Standard Retrieval Commands*.

18 Network

- Network Maintenance 104
- Network Retrieval 104

Network Maintenance

The Network Maintenance Menu is called with the command `MAINTAIN NETWORK`.

Overview of Functions and Commands

Function	Code	Command
Add a network	A	ADD NETWORK
Copy network	C	COPY NETWORK
Modify network	M	MODIFY NETWORK
Rename network	N	RENAME NETWORK
Purge network	P	PURGE NETWORK
Display network	D	see **Retrieval**
Link children	L	LINK NETWORK <active-association-type>
Edit owners of a network	O	EDIT NETWORK OWNER
Select network from a list	S	see **Retrieval**
Edit description	W	EDIT NETWORK DESCRIPTION

Parameters

Keyword	Field	Position
ID	Network ID	1
<>	Copy ID	2

Network-specific Maintenance Commands

Network objects are processed using ***Standard Maintenance Commands***.

Network Retrieval

The Network Retrieval Menu is called with the command `RETRIEVE NETWORK`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Networks	D	D	DISPLAY NETWORK
		L	LIST NETWORK
		S	SELECT NETWORK
		D	SHOW NETWORK
Execute retrieval models	E	T	EXECMODEL NETWORK
Dummy/Placeholder network	C	L	DUMMY NETWORK <passive-association-type>
Network with parents	B	D	PARENTS NETWORK <passive-association-type>
Network with no parent	O	L	FREE NETWORK ANY | ALL | <passive-association-type>
Network with children	T	D	CHILDREN NETWORK <active-association-type>
Network with no child	U	L	EMPTY NETWORK ANY | ALL | <active-association-type>
Vista numbers	N	L	LIST VISTATAB

Parameters

Keyword	Field	Position
ID	Network ID	1
RES	Restrictions	2
OUT	Output options	3
MODEL	Model	4
MODE	Output mode	5

Network-specific Retrieval Command

See ***Standard Retrieval Commands***.

Command	Description
LIST VISTATAB	Displays information on the use of Vista numbers in list form. For more information see *Network* in the Predefined Object Types in Predict documentation.

19 Node

- Node Maintenance 108
- Node Retrieval 108

Node Maintenance

The Node Maintenance Menu is called with the command `MAINTAIN NODE`.

Overview of Functions and Commands

Function	Code	Command
Add a node	A	ADD NODE
Copy node	C	COPY NODE
Modify node	M	MODIFY NODE
Rename node	N	RENAME NODE
Purge node	P	PURGE NODE
Display node	D	DISPLAY NODE
Link children	L	LINK NODE <active-association-type>
Edit owners of a node	O	EDIT NODE OWNER
Select node from a list	S	SELECT NODE
Edit description	W	EDIT NODE DESCRIPTION

Parameters

Keyword	Field	Position
ID	Node ID	1
<>	Copy ID	2

Node-specific Maintenance Commands

Node objects are processed using *Standard Maintenance Commands*.

Node Retrieval

The Node Retrieval Menu is called with the command `RETRIEVE NODE`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Nodes	D	D	DISPLAY NODE
		L	LIST NODE
		S	SELECT NODE
		D	SHOW NODE
Execute retrieval models	E	T	EXECMODEL NODE
Dummy/Placeholder node	C	L	DUMMY NODE <passive-association-type>
Node with parents	B	D	PARENTS NODE <passive-association-type>
Node with no parent	O	L	FREE NODE ANY | ALL | <passive-association-type>
Node with children	T	D	CHILDREN NODE <active-association-type>
Node with no child	U	L	EMPTY NODE ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	Node ID	1
PARENT	Contained in NW	2
RES	Restrictions	3
OUT	Output options	4
MODEL	Model	5
MODE	Output mode	6

Node-specific Retrieval Commands

Node objects are processed with *Standard Retrieval Commands*.

20 Owner

- Owner Maintenance 112
- Owner Retrieval 112

Owner Maintenance

These functions are called from the User Maintenance Menu. This menu is called with the command `MAINTAIN USER`.

Overview of Functions and Commands

Function	Code	Command
Rename/Merge owner	R	RENAME OWNER
Purge owner	E	PURGE OWNER

Parameters

Keyword	Field	Position
OWNER	Owner	1

Owner-Specific Maintenance Command

Command	Description
RENAME OWNER	Renames an owner, or - if the new owner name already exists - assigns the objects of the old owner to the new owner ID. In both cases the old owner no longer exists after this function has been executed.
PURGE OWNER	The owner will be deleted from the owner list of all objects. **Note:** An owner entry will not be deleted, if this would result in the last owner being deleted from an object which has the option OWNER=Force included in its object type definition.

Owner Retrieval

The Owner Retrieval Menu is called with the command `RETRIEVE OWNER`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Owners	D	D	DISPLAY OWNER
		L	LIST OWNER
		S	SELECT OWNER
		D	SHOW OWNER
Owner with no user	O	L	FREE OWNER
Objects with no owners	U	L	EMPTY OWNER ALL | <object-type>
Cross reference owners	X	X	XREF OWNER

Parameters

Keyword	Field	Position
ID	Owner ID	1
OUT	Output options	2
MODE	Output mode	3

Owner-Specific Retrieval Commands

Owner objects are processed with *Standard Retrieval Commands*.

21 Packagelist

- Packagelist Maintenance 116
- Packagelist Retrieval 116

Packagelist Maintenance

The Packagelist Maintenance Menu is called with the command `MAINTAIN PACKAGELIST`.

Overview of Functions and Commands

Function	Code	Command
Add a packagelist	A	ADD PACKAGELIST
Copy packagelist	C	COPY PACKAGELIST
Modify packagelist	M	MODIFY PACKAGELIST
Rename packagelist	N	RENAME PACKAGELIST
Purge packagelist	P	PURGE PACKAGELIST
Display packagelist	D	DISPLAY PACKAGELIST
Link children	L	LINK PACKAGELIST <active-association-type>
Edit owners of a packagelist	O	EDIT PACKAGELIST OWNER
Select packagelist from a list	S	SELECT PACKAGELIST
Edit description	W	EDIT PACKAGELIST DESCRIPTION

Parameters

Keyword	Field	Position
ID	Packagelist ID	1
<>	Copy ID	2

Packagelist-specific Maintenance Commands

Packagelist objects are processed using *Standard Maintenance Commands*.

Packagelist Retrieval

The Packagelist Retrieval Menu is called with the command `RETRIEVE PACKAGELIST`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Packagelists	D	D	DISPLAY PACKAGELIST
		L	LIST PACKAGELIST
		S	SELECT PACKAGELIST
		D	SHOW PACKAGELIST
Dummy/Placeholder packagelist	C	L	DUMMY PACKAGELIST <passive-association-type>
Execute retrieval models	E	T	EXECMODEL PACKAGELIST
Packagelist with parents	B	D	PARENTS PACKAGELIST <passive-association-type>
Packagelist with no parents	O	L	FREE PACKAGELIST ANY | ALL | <passive-association-type>
Packagelist with children	T	D	CHILDREN PACKAGELIST <active-association-type>
Packagelist with no children	U	L	EMPTY PACKAGELIST ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	Packagelist ID	1
PARENT	Contained in SY	2
RES	Restrictions	3
OUT	Output options	4
TYPE	Packagelist type	5
MODEL	Model	6
MODE	Output mode	7

Packagelist-specific Retrieval Commands

Packagelist objects are processed with *Standard Retrieval Commands*.

22 Program

- Program Maintenance 120
- Program Retrieval 122
- Program Active Retrieval 123

Program Maintenance

The Program Maintenance Menu is called with the command `MAINTAIN PROGRAM`.

Overview of Functions and Commands

Function	Code	Command
Add a program	A	ADD PROGRAM
Copy program	C	COPY PROGRAM
Modify program	M	MODIFY PROGRAM
Rename program	N	RENAME PROGRAM
Purge program	P	PURGE PROGRAM
Edit description	W	EDIT PROGRAM DESCRIPTION
Display program	D	DISPLAY PROGRAM
Link children	L	LINK PROGRAM <active-association-type>
Edit owners of a program	O	EDIT PROGRAM OWNER
Edit entry-points	R	EDIT PROGRAM ENTRY
Select program	S	SELECT PROGRAM
Redocument program	X	REDOCUMENT PROGRAM
Edit procedure code of a program	Y	EDIT PROGRAM PROCEDURE

Parameters

Keyword	Field	Position
ID	Program ID	1
<>	Copy ID	2

Program-specific Maintenance Commands

See also *Standard Maintenance Commands*.

Command	Description
EDIT PROGRAM ENTRY<Program-ID>	Use this command to edit entry points of a program directly.
EDIT PROGRAM PROCEDURE<Program-ID>	Use this command to edit the procedure code of a program directly.

Command	Description		
REDOCUMENT PROGRAM<Member><Library>	MEM	Member	1
	LIB	Library	2

Redocumenting Programs in Batch Mode

This is one of the few maintenance commands that can be used in batch mode. The following parameters are only available in batch mode and cannot be used online.

Keyword	Field	Position
MEMBER	Member	1
LIBRARY	Library	2
LANGUAGE	Language	3
SOURCE	Source/XRef	4
MEMBER-TYPE	Member types	5
DBNR	Database number	6
FNR	File number	7
PSW	Password	8
CIPHER	Cipher	9
PROCESS-OPT	Processing option	10
SYSTEM	Link to system	11
PREFIX	Program ID prefix	12
LIBRARY-PREFIX	Lib.name as sec.prefix	13
COMMENTS	Abstract	14
DESCRIPTION	Description	15
DESC-OPT	Replace/append descript.	16
PROGRAM-LIST	Program list	17
FILE-LIST	File list	18
DEFAULT-OWNER	Default owner	19
DEFAULT-KEY1	First default keyword	20
DEFAULT-KEY2	Second default keyword	21
Implementation Pointer		
POINTER-LIB	Library	22
POINTER-FNR	Fnr	23
POINTER-DBNR	DBnr	24
ABSTRACT	Abstract	25
STRUCTURE	Library structure	26

Example

Batch command to document all programs in library TESTLIB in Adabas file 8, database 1:

REDOCUMENT PROGRAM LIBRARY=TESTLIB,FNR=8,DBNR=1,PROCESS-OPT=A

or

REDOCUMENT PROGRAM ,TESTLIB,8,1,,A

This example uses the Natural parameters IA==, ID=,

Program Retrieval

The Program Retrieval Menu is called with the command `RETRIEVE PROGRAM`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Programs	D	D	DISPLAY PROGRAM
		L	LIST PROGRAM
		S	SELECT PROGRAM
		D	SHOW PROGRAM
Execute retrieval models	E	T	EXECMODEL PROGRAM
Dummy/Placeholder program	C	L	DUMMY PROGRAM <passive-association-type>
Programs with parents	B	D	PARENTS PROGRAM <passive-association-type>
Programs with no parents	O	L	FREE PROGRAM ANY | ALL | <passive-association-type>
Programs with children	T	D	CHILDREN PROGRAM <active-association-type>
Programs with no child	U	L	EMPTY PROGRAM ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	Program ID	1
PARENT	Contained in SY	2
MEM	Member	3
LIB	Library	4
RES	Restrictions	5
OUT	Output options	6
TYPE	Program of type	7
LANGUAGE	Language	8
FNR	User system Fnr	9
DBNR	User system DBnr	10
MODEL	Model	11
MODE	Output mode	12

Program-specific Retrieval Commands

Program objects are processed with *Standard Retrieval Commands*.

Program Active Retrieval

The Program Active Retrieval Menu is called with the command `ACTIVE PROGRAM`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Programs			see *Retrieval*
Entries ref. by members	R	L	REFERENCED PROGRAM
		C	COUNT PROGRAM
Entries not referenced	U	L	UNUSED PROGRAM
Programs not implemented	N	L	Note:XIST PROGRAM
Programs using programs	P	L	USING PROGRAM
Programs using files	F	L	USING FILE

Parameters

Keyword	Field	Position
ID	Program ID	1
PARENT	Contained in SY	2
MEM	Member	3
LIB	Library	4
ENTRY	Entry	5
RES	Restrictions	6
OUT	Output options	7
TYPE	Programs of type	8
LANGUAGE	Language	9
FNR	User system Fnr	10
DBNR	User system DBnr	11
SET	Save set	12
LIB-STRUCTURE	Library structure	13
MODE	Output mode	not possible

Program-specific Active Retrieval Commands

Retrieval Command	Description
REFERENCED PROGRAM	Lists the members that reference the specified entry points of the documented programs.
COUNT PROGRAM	Lists the libraries which use the specified entries and indicates the number of references per library and, if the entry is used in more than one library, the total number for all libraries.
UNUSED PROGRAM	Lists entry points not referenced by any member.
Note:XIST PROGRAM	Lists programs that are defined in Predict but do not correspond to members actually implemented.
USING FILE	Lists programs with Files which are linked or used according to XRef data.
USING PROGRAM	Lists programs with Programs which are linked or used according to XRef data.

23 Property

- Property Maintenance 126
- Property Retrieval 126

Property Maintenance

The Property Maintenance Menu is called with the command `MAINTAIN PROPERTY`.

Overview of Functions and Commands

Function	Code	Command
Add a property	A	ADD PROPERTY
Copy property	C	COPY PROPERTY
Modify property	M	MODIFY PROPERTY
Rename property	N	RENAME PROPERTY
Purge property	P	PURGE PROPERTY
Display property	D	DISPLAY PROPERTY
Link children	L	LINK PROPERTY <active-association-type>
Edit owners of a property	O	EDIT PROPERTY OWNER
Select property from a list	S	SELECT PROPERTY
Edit description of a property	W	EDIT PROPERTY DESCRIPTION

Parameters

Interface	Field	Position
ID	Property ID	1
<>	Copy ID	2

Property-specific Maintenance Commands

Property objects are processed using *Standard Maintenance Commands*.

Property Retrieval

The Property Retrieval Menu is called with the command `RETRIEVE PROPERTY`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Properties	D	D	DISPLAY PROPERTY
		L	LIST PROPERTY
		S	SELECT PROPERTY
		D	SHOW PROPERTY
Execute retrieval model	E	T	EXECMODEL PROPERTY
Dummy/Placeholder properties	C	L	DUMMY PROPERTY <passive-association-type>
Properties with parents	B	D	PARENTS PROPERTY <passive-association-type>
Properties with no parent	O	L	FREE PROPERTY ANY | ALL | <passive-association-type>
Properties with children	T	D	CHILDREN PROPERTY <active-association-type>
Properties with no child	U	L	EMPTY PROPERTY ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	Method	1
PARENT	Defined in IE	2
RES	Restrictions	3
OUT	Output options	4
MODEL	Model	5
MODE	Output mode	6

Property-specific Retrieval Commands

See also *Standard Retrieval Commands*.

24 Report Listing

- Report Listing Maintenance 130
- Report Listing Retrieval 131

Report Listing Maintenance

The Report Listing Maintenance Menu is called with the command `MAINTAIN REPORTLISTING`.

Overview of Functions and Commands

Function	Code	Command
Modify Report listing	M	MODIFY REPORTLISTING
Rename Report listing	N	RENAME REPORTLISTING
Purge Report listing	P	PURGE REPORTLISTING
Display Report listing	D	DISPLAY REPORTLISTING
Link children	L	LINK REPORTLISTING <active-association-type>
Edit owners of a Report listing	O	EDIT REPORTLISTING OWNER
Select Report listing from a list	S	SELECT REPORTLISTING

Parameters

Keyword	Field	Position
ID	Report listing ID	1
<>	Copy ID	2

Report Listing-specific Maintenance Commands

Report listing objects are processed using ***Standard Maintenance Commands***.

Note: Report listings are added automatically by Coordinator and Conversion functions. It is not possible to add report listings manually. For this reason, commands ADD and COPY are not available.

See also *Mass Delete of Report Listings* in *Special Functions* in the *Predict Administration documentation*. This function deletes multiple report listings in one operation.

Report Listing Retrieval

The Report Listing Retrieval Menu is called with the command `RETRIEVE REPORTLISTING`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Report listings	D	D	DISPLAY REPORTLISTING
		L	LIST REPORTLISTING
		S	SELECT REPORTLISTING
		D	SHOW
Execute retrieval models	E	T	EXECMODEL REPORTLISTING
Dummy/Placeholder Report listing	C	L	DUMMY REPORTLISTING <passive-association-type>
Report listing with parents	B	D	PARENTS REPORTLISTING <passive-association-type>
Report listing with no parent	O	L	FREE REPORTLISTING ANY | ALL | <passive-association-type>
Report listing with children	T	D	CHILDREN REPORTLISTING <active-association-type>
Report listing with no child	U	L	EMPTY REPORTLISTING ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	Report listing ID	1
RES	Restrictions	2
OUT	Output options	3
MODEL	Model	4
MODE	Output mode	5

Report Listing-specific Retrieval Commands

Report listing objects are processed with ***Standard Retrieval Commands***.

25 Server

- Server Maintenance 134
- Server Retrieval 134

Server Maintenance

The Server Maintenance Menu is called with the command `MAINTAIN SERVER`.

Overview of Functions and Commands

Function	Code	Command
Add a Server	A	ADD SERVER
Copy Server	C	COPY SERVER
Modify Server	M	MODIFY SERVER
Rename Server	N	RENAME SERVER
Purge Server	P	PURGE SERVER
Display Server	D	DISPLAY SERVER
Link children	L	LINK SERVER <active-association-type>
Edit owners of a Server	O	EDIT SERVER OWNER
Select Server from a list	S	SELECT SERVER
Edit description	W	EDIT SERVER DESCRIPTION

Parameters

Keyword	Field	Position
ID	Server ID	1
<>	Copy ID	2

Server-specific Maintenance Commands

Server objects are processed using *Standard Maintenance Commands*.

Server Retrieval

The Server Retrieval Menu is called with the command `RETRIEVE SERVER`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Servers	D	D	DISPLAY SERVER
		L	LIST SERVER
		S	SELECT SERVER
		D	SHOW SERVER
Execute retrieval models	E	T	EXECMODEL SERVER
Dummy/Placeholder Server	C	L	DUMMY SERVER <passive-association-type>
Server with parents	B	D	PARENTS SERVER <passive-association-type>
Server with no parent	O	L	FREE SERVER ANY | ALL | <passive-association-type>
Server with children	T	D	CHILDREN SERVER <active-association-type>
Server with no child	U	L	EMPTY SERVER ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	Server ID	1
PARENT	Contained in NO	2
RES	Restrictions	3
OUT	Output options	4
MODEL	Model	5
MODE	Output mode	6

Server-specific Retrieval Commands

Server objects are processed with *Standard Retrieval Commands*.

26 Storagespace

- Storagespace Maintenance 138
- Storagespace Retrieval 138
- Storagespace Administration 140

Storagespace Maintenance

The Storagespace Maintenance Menu is called with the command `MAINTAIN STORAGESPACE`.

Overview of Functions and Commands

Function	Code	Command
Add a storagespace	A	ADD STORAGESPACE
Copy storagespace	C	COPY STORAGESPACE
Modify storagespace	M	MODIFY STORAGESPACE
Rename storagespace	N	RENAME STORAGESPACE
Purge storagespace	P	PURGE STORAGESPACE
Display storagespace	D	DISPLAY STORAGESPACE
Link children	L	LINK STORAGESPACE <active-association-type>
Edit owners of a storagespace	O	EDIT STORAGESPACE OWNER
Select storagespace from a list	S	SELECT STORAGESPACE
Edit description of a storagespace	W	EDIT STORAGESPACE DESCRIPTION

Parameters

Keyword	Field	Position
ID	Storagespace ID	1
<>	Copy ID	2

Storagespace-specific Maintenance Commands

Storagespace objects are processed using *Standard Maintenance Commands*.

Storagespace Retrieval

The Storagespace Retrieval Menu is called with the command `RETRIEVE STORAGESPACE`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Storagespaces	D	D	DISPLAY STORAGESPACE
		L	LIST STORAGESPACE
		S	SELECT STORAGESPACE
		D	SHOW STORAGESPACE
Execute retrieval models	E	T	EXECMODEL STORAGESPACE
Dummy/Placeholder storagespaces	C	L	DUMMY STORAGESPACE <passive-association-type>
Unused storagespaces	N	L	UNUSED STORAGESPACE
Storagespaces with parents	B	D	PARENTS STORAGESPACE <passive-association-type>
Storagespaces with no parent	O	L	FREE STORAGESPACE ANY | ALL | <passive-association-type>
Storagespaces with children	T	D	CHILDREN STORAGESPACE <active-association-type>
Storagespaces with no child	U	L	EMPTY STORAGESPACE ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	Storagespace ID	1
RES	Restrictions	2
OUT	Output options	3
MODEL	Model	4
MODE	Output mode	5

Storagespace-specific Retrieval Command

Storagespace objects are also processed with *Standard Retrieval Commands*.

Command	Description
UNUSED STORAGESPACE	Lists unused Storagespaces. A Storagespace is regarded as unused if it is not referenced in a Dataspace or Field object.

Storagespace Administration

The Administration Implemented Storagespace Menu is called with the command `ADMINISTRATE STORAGESPACE`.

Overview of Functions and Commands

Function	Code	Command
Disconnect implementation	C	DISCONNECT STORAGESPACE
Display implementation	D	DISPLAY STORAGESPACE
Purge implementation	P	PURGE STORAGESPACE
Select implementation	D	SELECT STORAGESPACE

Parameters

Keyword	Field	Position
ID	Storagespace ID	1
MEM	Member	2
LIB	Library	3
RES	Restrictions	4

Storagespace-specific Administration Commands

Storagespace objects are processed using *Standard Administration Commands*.

27 System

- System Maintenance 142
- System Retrieval 142
- System Active Retrieval 144

System Maintenance

The System Maintenance Menu is called with the command `MAINTAIN SYSTEM`.

Overview of Functions and Commands

Function	Code	Command
Add a system	A	ADD SYSTEM
Copy system	C	COPY SYSTEM
Modify system	M	MODIFY SYSTEM
Rename system	N	RENAME SYSTEM
Purge system	P	PURGE SYSTEM
Display system	D	DISPLAY SYSTEM
Link children	L	LINK SYSTEM <active-association-type>
Edit owners of a system	O	EDIT SYSTEM OWNER
Select system from a list	S	SELECT SYSTEM
Edit description of a system	W	EDIT SYSTEM DESCRIPTION

Parameters

Keyword	Field	Position
ID	System ID	1
<>	Copy ID	2

System-specific Maintenance Commands

System objects are processed using *Standard Maintenance Commands*.

System Retrieval

The System Retrieval Menu is called with the command `RETRIEVE SYSTEM`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Systems	D	D	DISPLAY SYSTEM
		L	LIST SYSTEM
		S	SELECT SYSTEM
		D	SHOW SYSTEM
Execute retrieval models	E	T	EXECMODEL SYSTEM
Dummy/Placeholder systems	C	L	DUMMY SYSTEM <passive-association-type>
Systems with parents	B	D	PARENTS SYSTEM <passive-association-type>
Systems with no parent	O	L	FREE SYSTEM ANY | ALL | <passive-association-type>
Systems with children	T	D	CHILDREN SYSTEM <active-association-type>
Systems with no child	U	L	EMPTY SYSTEM ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	System ID	1
TYPE	System of type	2
RES	Restrictions	3
MODEL	Model	4
OUT	Output options	5
LIB	Library	6
FNR	User system Fnr	7
DBNR	User system DBnr	8
MODE	Output mode	9

System-specific Retrieval Commands

System objects are processed with *Standard Retrieval Commands*.

System Active Retrieval

The System Active Retrieval Menu is called with the command `ACTIVE SYSTEM`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Systems	D		see *Retrieval*
Systems containing programs	P	L	CONTAINING PROGRAM
Libraries	I	L	LIST LIBRARY
		S	SELECT LIBRARY
Systems not implemented	N	L	Note:XIST SYSTEM
Libraries not documented	O	L	UNDEFINED LIBRARY
Libraries with members	T	L	ELEMENTS LIBRARY
Systems containing programs	P	L	CONTAINING PROGRAM

Parameters

Keyword	Field	Position
ID	System ID	1
LIB	Library	2
RES	Restrictions	3
OUT	Output options	4
TYPE	System of type	5
FNR	User system Fnr	6
DBNR	User system DBnr	7
LIB-STRUCTURE	Library structure	8
MODE	Output mode	not possible

System-specific Active Retrieval Commands

Command	Description
LIST LIBRARY	Lists the implemented libraries. If the library is documented, the corresponding system is also indicated. If not, the following text appears: >>>library not documented<<<.
SELECT LIBRARY	Produces the same list as LIST LIBRARY. A library can be selected for further processing from this list.
Note:XIST SYSTEM	Lists systems that are documented but do not correspond to an implemented library.
UNDEFINED LIBRARY	Lists libraries that are implemented but have no corresponding system object in Predict.
ELEMENTS LIBRARY	Lists libraries with members. If a member is documented, the corresponding program is also indicated. If not, the member is marked >>>>not documented<<<<.
CONTAINING PROGRAM	Lists systems with programs which are linked or for which XRef data exist in the documented library. Predict 3.2 XRef data is evaluated.

28 Trigger

- Trigger Maintenance 148
- Trigger Retrieval 149

Trigger Maintenance

The Trigger Maintenance Menu is called with the command `MAINTAIN TRIGGER`.

Overview of Functions and Commands

Function	Code	Command
Add a trigger	A	ADD TRIGGER
Copy trigger	C	COPY TRIGGER
Modify trigger	M	MODIFY TRIGGER
Rename trigger	N	RENAME TRIGGER
Purge trigger	P	PURGE TRIGGER
Display trigger	D	DISPLAY TRIGGER
Link children	L	LINK TRIGGER <active-association-type>
Edit owners of a trigger	O	EDIT TRIGGER OWNER
Select trigger from a list	S	SELECT TRIGGER
Edit description	W	EDIT TRIGGER DESCRIPTION
Edit Trigger code	Y	EDIT TRIGGER TRIGGER

Parameters

Keyword	Field	Position
ID	Trigger ID	1
<>	Copy ID	2

Trigger-specific Maintenance Commands

EDIT TRIGGER TRIGGER<*trigger-ID*>	Invokes the Trigger code Editor, a modified Natural Editor used to edit the Trigger code.

Trigger Retrieval

The Trigger Retrieval Menu is called with the command `RETRIEVE TRIGGER`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Triggers	D	D	DISPLAY TRIGGER
		L	LIST TRIGGER
		S	SELECT TRIGGER
		D	SHOW TRIGGER
Execute retrieval models	E	T	EXECMODEL TRIGGER
Dummy/Placeholder trigger	C	L	DUMMY TRIGGER <passive-association-type>
Trigger with parents	B	D	PARENTS TRIGGER *<passive-association-type>*
Trigger with no parent	O	L	FREE TRIGGER ANY | ALL | *<passive-association-type>*
Trigger with children	T	D	CHILDREN TRIGGER *<active-association-type>*
Trigger with no child	U	L	EMPTY TRIGGER ANY | ALL | *<active-association-type>*

Parameters

Keyword	Field	Position
ID	Trigger ID	1
PARENT	Triggers FI	2
RES	Restrictions	3
OUT	Output options	4
MODEL	Model	5
MODE	Output mode	6

Trigger-specific Retrieval Commands

Trigger objects are processed with ***Standard Retrieval Commands***.

29 User

- User Maintenance 152
- User Retrieval 153

User Maintenance

The User Maintenance Menu is called with the command `MAINTAIN USER`.

Overview of Functions and Commands

Function	Code	Command
Add a user	A	ADD USER
Copy user	C	COPY USER
Modify user	M	MODIFY USER
Rename user	N	RENAME USER
Purge user	P	PURGE USER
Display user	D	DISPLAY USER
Link children	L	LINK USER <active-association-type>
Edit owners of a user	O	EDIT USER OWNER
Select user from a list	S	SELECT USER
Edit description of a user	W	EDIT USER DESCRIPTION
Rename/Move owner	R	See *Owner Maintenance*.
Purge owner	E	See *Owner Maintenance*.

Parameters

Keyword	Field	Position
ID	User ID	1
<>	Copy ID	2
NAME	User name	3

User-specific Maintenance Commands

User objects are processed using *Standard Maintenance Commands*.

User Retrieval

The User Retrieval Menu is called with the command `RETRIEVE USER`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Users	D	D	DISPLAY USER
		L	LIST USER
		S	SELECT USER
		D	SHOW USER
Execute retrieval models	E	T	EXECMODEL USER
Dummy/Placeholder users	C	L	DUMMY USER *<passive-association-type>*
Users related to no object	Y	L	UNUSED USER
Users related to objects	X	X	XREF USER
Users with parents	B	D	PARENTS USER *<passive-association-type>*
Users with no parent	O	L	FREE USER ANY | ALL | *<passive-association-type>*
Users with children	T	D	CHILDREN USER *<active-association-type>*
Users with no child	U	L	EMPTY USER ANY | ALL | *<active-association-type>*

Parameters

Keyword	Field	Position
ID	User ID	1
NAME	User name	2
RES	Restrictions	3
OUT	Output options	4
MODEL	Model	5
MODE	Output mode	6

User-specific Retrieval Commands

See also *Standard Retrieval Commands*.

Command	Description
UNUSED USER	Lists users that are not linked to an object via an owner. A message is given if all users that meet the selection criteria are linked to an object.
XREF USER	Lists users and all objects that are linked to these users via owners.

30 Verification

- Verification Maintenance 156
- Verification Retrieval 157
- Verification Active Retrieval 158

Verification Maintenance

The Verification Maintenance Menu is called with the command `MAINTAIN VERIFICATION`.

Overview of Functions and Commands

Function	Code	Command
Add a verification	A	ADD VERIFICATION
Copy verification	C	COPY VERIFICATION
Modify verification	M	MODIFY VERIFICATION
Rename/change status verific.	N	RENAME VERIFICATION
Purge verification	P	PURGE VERIFICATION
Edit description	W	EDIT VERIFICATION DESCRIPTION
Display verification	D	DISPLAY VERIFICATION
Link children	L	LINK VERIFICATION <active-association-type>
Edit owners of a verification	O	EDIT VERIFICATION OWNER
Edit rule of a verification	R	EDIT VERIFICATION RULE
Select verification from a list	S	SELECT VERIFICATION

Parameter

Keyword	Field	Position
ID	Verification ID	1
<>	Copy ID	2

Verification-specific Maintenance Commands

See also *Standard Maintenance Commands*.

Command	Description
EDIT VERIFICATION RULE	Calls the Predict Rule Editor to edit the processing rule of a verification directly.

Verification Retrieval

The Verification Retrieval Menu is called with the command `RETRIEVE VERIFICATION`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Verifications	D	D	DISPLAY VERIFICATION
		L	LIST VERIFICATION
		S	SELECT VERIFICATION
		D	SHOW VERIFICATION
Execute retrieval models	E	T	EXECMODEL VERIFICATION
Verifications to regenerate	K	L	REGENERATE VERIFICATION
Dummy/Placeholder verifications	C	L	DUMMY VERIFICATION *<passive-association-type>*
Verifications with parents	B	D	PARENTS VERIFICATION *<passive-association-type>*
Verifications with no parent	O	L	FREE VERIFICATION ANY | ALL | *<passive-association-type>*
Verifications with children	T	D	CHILDREN VERIFICATION *<active-association-type>*
Verifications with no child	U	L	EMPTY VERIFICATION ANY | ALL | *<active-association-type>*

Parameters

Keyword	Field	Position
ID	Verification ID	1
RES	Restrictions	2
OUT	Output options	3
STATUS	Verif. of status	4
FORMAT	Format	5
MODEL	Model	6
MODE	Output mode	7

Verification-specific Retrieval Commands

See also **Standard Retrieval Commands** .

Command	Description
REGENERATE VERIFICATION	Lists verifications of type automatic that have been modified since generation of a DDM which contains a field that uses one of the verifications.

Verification Active Retrieval

The Verification Active Retrieval Menu is called with the command `ACTIVE VERIFICATION`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Verifications ref. by members	R	L	REFERENCED VERIFICATION
		C	COUNT VERIFICATION
Verifications not referenced	U	L	UNUSED VERIFICATION

Parameters

Keyword	Field	Position
ID	Verification ID	1
STATUS	Status	2
FILE	in file	3
FORMAT	Format	4
RES	Restrictions	5
OUT	Output options	6
SET	Save set	7
MODE	Output mode	not possible

Verification-specific Active Retrieval Commands

Command	Description
REFERENCED VERIFICATION	Lists all members that use the specified verifications, together with with some main attributes. If an automatic rule is used as a free rule, a corresponding text appears on screen.
COUNT VERIFICATION	Lists all members that use the specified verifications and indicates the number of references per library. If the verification is used in more than one library, the total number of references for all libraries is also given.
UNUSED VERIFICATION	Lists verifications that are not used by any member.

31 Virtual Machine

- Virtual Machine Maintenance 162
- Virtual Machine Retrieval 162

Virtual Machine Maintenance

The Virtual Machine Maintenance Menu is called with the command `MAINTAIN VIRTUALMACHINE`.

Overview of Functions and Commands

Function	Code	Command
Add a virtual machine	A	ADD VIRTUALMACHINE
Copy virtual machine	C	COPY VIRTUALMACHINE
Modify virtual machine	M	MODIFY VIRTUALMACHINE
Rename virtual machine	N	RENAME VIRTUALMACHINE
Purge virtual machine	P	PURGE VIRTUALMACHINE
Display virtual machine	D	See **Retrieval**.
Link children	L	LINK VIRTUALMACHINE `<active-association-type>`
Edit owners of a virtual machine	O	EDIT VIRTUALMACHINE OWNER
Select virtual machine	S	See ***Retrieval***.
Edit description	W	EDIT VIRTUALMACHINE DESCRIPTION

Parameters

Keyword	Field	Position
ID	Virtual machine	1
<>	Copy ID	2

Virtual Machine-specific Maintenance Commands

Virtual machine objects are processed using ***Standard Maintenance Commands***.

Virtual Machine Retrieval

The Virtual Machine Retrieval Menu is called with the command RETRIEVE VIRTUALMACHINE.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
Virtual machines	D	D	DISPLAY VIRTUALMACHINE
		L	LIST VIRTUALMACHINE
		S	SELECT VIRTUALMACHINE
		D	SHOW VIRTUALMACHINE
Execute retrieval models	E	T	EXECMODEL VIRTUALMACHINE
Dummy/Placeholder virt. machines	C	L	DUMMY VIRTUALMACHINE *<passive-association-type>*
Virtual machines with parents	B	D	PARENTS VIRTUALMACHINE *<passive-association-type>*
Virtual machines with no parents	O	L	FREE VIRTUALMACHINE ANY | ALL | *<passive-association-type>*
Virtual machines with children	T	D	CHILDREN VIRTUALMACHINE *<active-association-type>*
Virtual machine with no child	U	L	EMPTY VIRTUALMACHINE ANY | ALL | *<active-association-type>*

Parameters

Keyword	Field	Position
ID	Virtual machine ID	1
PARENT	Belongs to VM	2
RES	Restrictions	3
OUT	Output options	4
MODEL	Model	5
MODE	Output mode	6

Virtual Machine Specific Retrieval Commands

Virtual machine objects are processed with ***Standard Retrieval Commands***.

32 UDE - User-Defined

- UDE Maintenance 166
- UDE Retrieval 166

UDE Maintenance

The Maintenance Menu for a user-defined object type is called with the command `MAINTAIN UDE-TYPE`.

Overview of Functions and Commands

Function	Code	Command
Add a UDE	A	ADD UDE-TYPE
Copy UDE	C	COPY UDE-TYPE
Modify UDE	M	MODIFY UDE-TYPE
Rename	N	RENAME UDE-TYPE
Purge UDE	P	PURGE UDE-TYPE
Display UDE	D	DISPLAY UDE-TYPE
Link children	L	LINK UDE-TYPE <active-association-type>
Edit owners of a UDE	O	EDIT UDE-TYPE OWNER
Select UDE from a list	S	SELECT UDE-TYPE
Edit description	W	EDIT UDE-TYPE DESCRIPTION

Parameters

Keyword	Field	Position
ID	UDE ID	1
<>	Copy ID	2

UDE-specific Maintenance Commands

UDE objects are processed using *Standard Maintenance Commands*.

UDE Retrieval

The UDE Retrieval Menu is called with the command `RETRIEVE UDE-TYPE`.

Overview of Functions and Commands

Retrieval Type	Code	Default Output Mode	Command
UDEs	D	D	DISPLAY UDE-TYPE
		L	LIST UDE-TYPE
		S	SELECT UDE-TYPE
		D	SHOW UDE-TYPE
Execute retrieval models	E	T	EXECMODEL UDE-TYPE
Dummy/Placeholder UDEs	C	L	DUMMY UDE-TYPE <passive-association-type>
UDEs with parents	B	D	PARENTS UDE-TYPE <passive-association-type>
UDEs with no parent	O	L	FREE UDE-TYPE ANY | ALL | <passive-association-type>
UDEs with children	T	D	CHILDREN UDE-TYPE <active-association-type>
UDEs with no child	U	L	EMPTY UDE-TYPE ANY | ALL | <active-association-type>

Parameters

Keyword	Field	Position
ID	UDE ID	1
RES	Restrictions	2
OUT	Output options	3
MODEL	Model	4
MODE	Output mode	5

UDE-specific Retrieval Commands

UDE objects are processed with *Standard Retrieval Commands*.

II Maintenance in Predict

33 Maintenance in Predict

- Overview of Maintenance Functions 172
- Creating Predict Objects 176
- Selecting and Displaying Objects 178
- Purging Objects 179
- Renaming Objects 179
- Support of Dummy and Placeholder Objects 180
- Overview of Standard Attributes 181
- Abstract and Extended Description 182
- Creating and Modifying Type-Specific Attributes 189

Overview of Maintenance Functions

In the following sections, first the standard maintenance functions and then the type-specific functions are described briefly. References are given where the functions are described in more detail.

Calling Maintenance Functions

Predict maintenance functions can be called via menu or via command by either

- entering a menu-oriented command of the form MAINTAIN object-type,
- or by selecting the function code M and the object code in a Predict main menu.

Overview of Standard Maintenance Functions

Note: Some standard maintenance functions work differently for different types of objects, for example the Select object from a list functions evaluate type-specific parameters to restrict the scope of the selection. All type-specific characteristics of a function are described in the respective sections of Predefined Object Types in Predict documentation.

Function	Code	Command	Description
Add an object	A	ADD	One or more input screens with input fields for the object attributes are displayed. A detailed description of the attributes and the corresponding input fields of all predefined object types is contained in *Predefined Object Types in Predict*. See also ***Creating Predict Objects***.
Copy object	C	COPY	The ID of the object to be created is specified in the field Copy ID.
Display object	D	DISPLAY	Note: that output options do not apply to a maintenance Display function: the maximum amount of information is displayed.
Modify object	M	MODIFY	The functions Modify and Add use the same input screens. An additional initial screen may however be used for the Add function.
Rename object	N	RENAME	General rules for renaming objects are described in the section ***Renaming Objects***. Type-specific rules are described in the respective section of the Predefined Object Types in Predict documentation.
Purge object	P	PURGE	See ***Purging Predict Objects*** for general rules. Type-specific rules are described in respective section of the Predefined Object Types in Predict documentation.
Select object from a list	S	SELECT	See ***Selecting and Displaying Objects***.
Link children	L	LINK	Invokes the List Editor to edit a child list. An association must be specified. The Link Editor is described in the section ***Editors in Predict***.

Function	Code	Command	Description
Physical Attributes	J		One or more input screens with input fields for the object attributes are displayed. A detailed description of the attributes and the corresponding input fields of all predefined object types is contained in *Predefined Object Types in Predict*. See also ***Creating Predict Objects***.

The following functions are available for all object types from a submenu in the corresponding Add/Copy/Modify screens:

Function	Code	Command	Description
Desc.	W	EDIT DESCRIPTION	The description editor will be invoked to modify the extended description of an object. See ***Editors in Predict***.
Owner	O	EDIT OWNER	The function is used to edit the owner list of an object.

See also *Additional attributes / Associations* in the Predefined Object Types in Predict documentation.

Overview of Type-Specific Maintenance Functions

Database-Specific Maintenance Functions

Note: See also *Database Maintenance* in the section *Database* in the *Predefined Object Types in Predict documentation*.

Function	Code	Command	Description
Modify the Vista element	K	MODIFY VISTA-DA DATABASE	Invokes the Vista element maintenance for a database. Only applicable to Adabas databases with run mode "Vista".

Extract-Specific Maintenance Functions

Note: See also *Extract Maintenance* in the section *Extract* in the *Predefined Object Types in Predict documentation*.

Function	Code	Command	Description
Link/unlink objects	E	EDIT EXTRACT OBJECTS	This function links/unlinks an object to/from an extract
Build/extend	B	BUILD EXTRACT	This function creates or extends the object list of an extract.
Operate on Extract	T	OPERATE EXTRACT	This function adds the result of a set operation to the object list of an extract.

Field-Specific Maintenance Functions

Note: See also Field Maintenance in the section Field in the *Predefined Object Types in Predict documentation*.

Function	Code	Command	Description
Browse through fields of a file	B	BROWSE	The Modify field function is invoked for each field of a file.
Move field within a file	H	MOVE	Changes the order of fields in a file.
Redefine Field	R	REDEFINE ELEMENT	Invokes the list editor for defining a redefinition.
Edit Field Expression	Y	EDIT ELEMENT EXPRESSION	Invokes the Predict editor for editing a field expression.

File-Specific Maintenance Functions

Note: See also File Maintenance in the section File in the *Predefined Object Types in Predict documentation*.

Function	Code	Command	Description
Edit list of fields	E	EDIT FILE ELEMENT / LINK FILE ELEMENT / EDIT ELEMENT	This function can also be called by setting the option Field List in the Additional attributes / Associations line of the Add, Copy and Modify screens.
Force standard	F	FORCE FILE	Compares the coupled attributes of all fields defined in the specified standard file with the attributes of the coupled fields in other files. If attributes of coupled fields are different (and these fields are not marked as non-standard), change them to match the standard file.
Push backward	B	PUSH FILE	Connects fields of a file to fields in a standard file.
Modify the Adabas attributes of an Adabas file object	J	MODIFY ADA-ATTR	This function can also be called by setting the option Database link attr. in Add, Copy and Modify screens to Y.
Modify the Vista elements of an Adabas file object	K	MODIFY VISTA-FI	This function can also be called by setting the option Additional attributes in Add, Copy and Modify screens to Y.
Edit Subquery of a File	Y	EDIT FILE SUBQUERY	Invokes the Expression Editor. Only applicable to SQL views.

Keyword-Specific Maintenance Functions

Note: See also *Keyword Maintenance* in the section *Keyword* in the *Predefined Object Types in Predict documentation*.

Function	Code	Command	Description
Link/unlink objects	E	EDIT KEYWORD OBJECTS	This function links/unlinks a keyword to/from an object.

Program-Specific Maintenance Functions

Note: See also *Program Maintenance* in the section *Program* in the *Predefined Object Types in Predict documentation*.

Function	Code	Command	Description
Edit a list of entry points	R	EDIT PROGRAM ENTRY	Invokes the list editor for the entry point list.
Redocument program	X	REDOCUMENT PROGRAM	Creates a Predict program object from an implemented program.
Edit procedure code of a program	Y	EDIT PROGRAM PROCEDURE	Invokes the Procedure Code Editor.

Trigger-Specific Maintenance Functions

Note: See also *Trigger Maintenance* in the section *Trigger* in the *Predefined Object Types in Predict documentation*.

Function	Code	Command	Description
Edit Trigger code	Y	EDIT TRIGGER TRIGGER	Invokes the Trigger Code Editor.

User-Specific Maintenance Functions

Note: See also *User Maintenance* in the section *User* in the *Predefined Object Types in Predict documentation*.

Function	Code	Command	Description
Rename or merge an owner	R	RENAME OWNER	The owner will be renamed in the owner list of all objects.
Purge an owner	E	PURGE OWNER	The owner will be deleted from the owner list of all objects.

Verification-Specific Maintenance Functions

Note: See also *Verification Maintenance* in the section *Verification* in the *Predefined Object Types in Predict documentation*.

Function	Code	Command	Description
Edit rule of a verification	R	EDIT VERIFICATION RULE	Invokes the editor for the rule of a verification.

Creating Predict Objects

There are several ways to create Predict objects:

- Objects can be created (and maintained) manually using the Add, Copy and Modify functions. The functions Add and Modify can also be called from the list editor when editing the object list of an object. Copying an object and changing its attributes is an easy way to create a new object.
- Predict objects for implemented external objects can be created with incorporation functions and the Redocument program function. See the section *Incorporation* in the *External Objects in Predict documentation* and *Redocument Program* in the section *Program* in the *Predefined Object Types in Predict documentation*.
- Objects can be loaded using the Predict Coordinator. See the Predict Coordinator documentation.

Naming Objects

Each object in the data dictionary is identified by its ID. Except with field objects, the ID of a data dictionary object must be unique among objects of that type. Field objects can have the same ID if they belong to different files.

IDs are assigned when first creating an object with the Add, Copy or an incorporate function. To change the ID of an object, use the function Rename. See the section ***Renaming Objects***.

See also *Naming Conventions* in the section *General Information* in the *Predefined Object Types in Predict documentation*.

Displaying Date and User for Creation/Modification

The parameter Store user ID of modifier in the Maintenance options of the General Defaults function determines whether the ID of the user who created or modified the object is stored at all. This parameter is usually set by your data dictionary administrator.

If Store user ID of modifier is set to Y, each user may determine with the parameter Display modifier whether the user who created or last modified the object to be maintained is then displayed. The parameter Display modifier is set in the Output options of the Modify User Defaults function.

For All Maintenance Functions Except Display

- If Display modifier is set to Y, the following information is given for all maintenance functions except Display:

  ```
  Added YYYY-MM-DD at HH:MM
     by USER-ID
  ```

  or

  ```
  Modified YYYY-MM-DD at HH:MM
        by USER-ID
  ```

- If Display modifier is set to N, the following information is given for all maintenance functions except Display:

  ```
        Added YYYY-MM-DD at HH:MM
     Modified YYYY-MM-DD at HH:MM
  ```

For file objects the following information is also given:

```
    Fields modified YYYY-MM-DD at HH:MM
```

For Maintenance Function Display

Because for this function the maximum amount of information is always output, the output option Display modifier has no effect. The following information is given:

```
   Added YYYY-MM-DD at HH:MM by USER-ID
Modified YYYY-MM-DD at HH:MM by USER-ID
```

For file objects the following information is also given:

```
Fields modified YYYY-MM-DD at HH:MM by USER-ID
```

Note: If the general default Store user ID of modifier has been set to N by your data dictionary administrator, the user ID will not be displayed.

Selecting and Displaying Objects

The Select function displays a list of objects for selection. Short information on each object is given by displaying the values of some attributes of the object.

Asterisk notation can be used to specify a range of objects. For example, all files whose IDs begin with AB can be listed by entering AB* in the object ID field. Additional restrictions can be specified to restrict the number of objects listed.

Processing Objects from Selection Lists

A single object can then be selected for immediate processing or objects can be put into the workplan for later processing.

- Select a single object for immediate processing: enter S, X or a slash (/) in the Cmd column or position the cursor on the line containing the object and press ENTER. Selection by cursor position is possible if no command code is specified in the Cmd column and the field in the lower left corner is blank.
- Put object(s) into the workplan for later processing: mark object(s) with a command code in the column Cmd. Enter an asterisk to display list of possible codes for the particular object. See *Using the Workplan* in the section *Predict User Interface* in the *Introduction to Predict documentation*.

Displaying Objects

The Display function displays all attributes of an object. This function is similar to the retrieval type Objects with output mode D. However, some differences exist:

- Only one object can be displayed in one function call. Internal ID must be entered (asterisk notation is not permitted)
- Output options will not be evaluated. The maintenance function Display always displays a maximum amount of information.

Purging Objects

The Purge function deletes objects from the data dictionary.

For most object types, the Purge function is confirmed with `DELETE`. The following objects are deleted:

- the object itself
- links from the main object to child objects
- links from parent objects to the main object

For the following object types, an additional option `SCRATCH` is available:

- Database
- System
- User

The effects of this option are described in the corresponding section of the Predefined Object Types in Predict documentation.

Renaming Objects

The ID of an object or the subtype or number of an existing file or database object can be changed with the Rename/retype/renumber object function - Code N. The name of the object that has been changed will be changed accordingly in all objects which are linked to the renamed object.

Type-specific rules are given in the respective sections of the Predefined Object Types in Predict documentation.

Support of Dummy and Placeholder Objects

Dummy Objects

A dummy object can be created in one of two ways:

- When a link is added from an existing object to an object that has not yet been created in Predict, a dummy object is created. As of this version of Predict, a record is also stored physically in the Predict file for this dummy object.
- When an object that is linked to another object is imported/loaded with the Predict Coordinator *without* internal ID, and the referenced object is not imported/loaded and does not exist in the target environment, a dummy is added in the target environment for the referenced object. (If the object is loaded/imported *with* internal ID, a placeholder is created. See ***Placeholders***.)

See the Predict Coordinator documentation for details of importing/loading with and without internal ID.

Dummies are marked with *?* in the Link Editor.

Maintenance Functions for Dummy Objects

Function	Description
Add	When an new object is added and the ID is already used for a dummy object, a message is given indicating that a dummy object is to be converted into a "normal" object.
Copy	Dummy object is allowed as target object (Copy ID) for a Copy function. The dummy object is converted into a “normal” object. **Note:** The values for Created on / by and Changed on / by are reset to the time and date at which a dummy is converted to a normal object with the Add or Copy function.
Purge	With this version of Predict, you can delete dummies as you would a normal object. Also, the record stored when a dummy is created is not deleted automatically when the entry in the link list is deleted. These dummies that are not referenced by any other object can be deleted with the Purge function.

Placeholders

When an object that is linked to another object is imported/loaded with the Predict Coordinator *with* internal ID, and the referenced object is not imported/loaded and does not exist in the target environment, a Placeholder is added in the target environment for the referenced objects. (If the object is imported/loaded *without* internal ID, a dummy is created).

The job of this placeholder is to reserve the object ID of the referenced object in the target environment so that the link in the old environment can be recreated in the new environment at a later time.

The following rules apply:

- Placeholders cannot be modified. The only maintenance function available is Purge.
- Placeholders are marked with § in the Link Editor.
- The status of a placeholder can only be changed by importing/loading the “real” object.

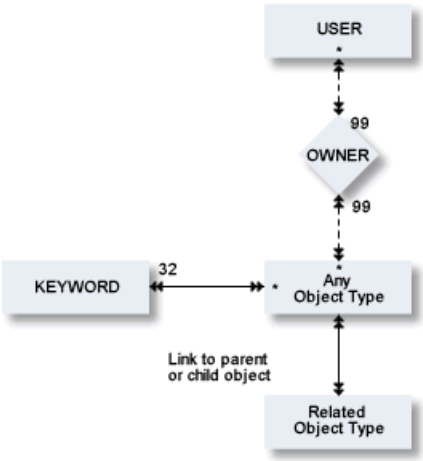
Overview of Standard Attributes

The following sections describe how standard attributes are modified. For a description of object type-specific attributes see the respective sections of the Predefined Object Types in Predict documentation.

The properties of objects are documented with attributes of Predict objects. The following types of attributes can be distinguished:

- Abstracts and extended descriptions
- Links to parent/child objects
- Links to keywords and owners
- Type-specific attributes

The diagram gives an overview.



Abstract and Extended Description

Each object can have an abstract and an extended description which provide information about the object.

Abstracts and/or extended descriptions containing the same text string can be selected together. For a description of this option, see *Relating Objects Logically* in the section *Predict Functionality* in the *Introduction to Predict documentation*, and **Using Restrictions**.

Abstracts and extended descriptions can also be used as online help text in Natural help screens. See ***SYSHELP***.

Format of Abstracts and Extended Descriptions

The following rules apply to the format of abstracts and extended descriptions.

- For abstracts, up to 16 lines of up to 30 characters can be specified.
- When using the Natural Editor, the maximum size of extended descriptions depends on the Natural ESIZE parameter. See the *Natural Parameter Reference* documentation for more information.
- When using the Software AG Editor there is no software restriction of the maximum size of extended descriptions.
- Abstracts and extended descriptions can contain upper and lower-case letters. If the parameter Upper/Lower case in the Miscellaneous section of the General Defaults is set to U, all alphabetic characters in abstracts and extended descriptions are converted to upper-case.
- Extended description can contain Con-form instructions that will be interpreted when displaying the text if the parameter Use Con-form of the session profile is set to Y. See **Using Con-form in Extended Descriptions**.

Adding an Abstract to an Object

Abstracts can be added, removed or modified whenever the Add, Copy or Modify function is used to maintain an object.

The first few lines of the abstract are displayed in the initial Modify screen. Enter "Y" in the Zoom field to display the maximum of 16 lines.

```
14:25:44             *****  P R E D I C T  *****              2007-05-31
                            - Modify Dataspace -
Dataspace ID .... HNO-DC                   +------------ Abstract ------------+
Type ............ DB2 mainframe            !  AN ABSTRACT CAN CONSIST OF UP   !
Located in DA ... HEB-DB22                 !  TO 16 LINES EACH CONTAINING     !
Keys ..                                    !  UP TO 30 CHARACTERS.            !
                                           !  ENTER Y IN THE ZOOM FIELD TO    !
Dataspace attributes                       !  DISPLAY THE FULL ABSTRACT.      !
  Tablespace name .. TEST                  !                                  !
Physical attributes in <Default Server>    !                                  !
  Nr of partitions .                       !                                  !
  Buffer pool .....*                       !                                  !
  Locksize ........*   (none)              !                                  !
  Close option ..... N (Y,N)               !                                  !
  Lockmax .......... 999                   !                                  !
  Lockpart .........   (Y,N)               !                                  !
  Maxrows .......... 99                    !                                  !
  CCSID ...........*   (none)              !                                  !
  Member cluster ...   (Y,N)               !                                  !
Abstract   * Zoom: Y                       +----------------------------------+
  AN ABSTRACT CAN CONSIST OF UP
  TO 16 LINES EACH CONTAINING
  Additional attributes ..* N          Associations ..* N
```

The following line commands are available for abstracts:

Line Command	Description
.c	Copy one line.
.d	Delete one line.
.i	Insert three lines.
.j	Join line with next line.
.s	Split line at cursor position.

Note: Editor line commands are introduced by the character defined as the escape character in the Natural parameter module.

Modifying Extended Descriptions

To edit an extended description, invoke the description editor in one of the following ways:

- Select the option Description (code W) of the Additional attributes window in any Add, Copy or Modify screen
- call the function Edit description in any maintenance menu.
- enter the command EDIT<object-type>DESCRIPTION<object ID>

For a complete description of the description editor see ***The Description Editor***.

Note: An extended description can use Long Lines of up to 250 characters. However, if you intend to use extended descriptions with the SYSHELP utility to generate your own online help system, you should limit the line length of your extended descriptions to 72 characters. Further information on the SYSHELP utility is given in the section ***SYSHELP***.

Disallowing or Forcing Extended Descriptions

By setting the metadata administration parameter Edit description to Allow, Disallow or Force, the DDA can make the adding of an extended description optional, prohibited or mandatory. This parameter can be specified for each object type. If Edit description is set to Allow, any user can specify a default setting for the Description parameter in the additional attributes window of Add/Copy/Modify screens. The default is specified with the parameters Edit description, Edit owner, Edit default child in the Maintenance Options of the Modify User Defaults screen.

See also *Maintenance Options* in the section *Predict User Interface* in the *Introduction to Predict documentation*.

Default Extended Descriptions

A skeleton for extended descriptions can be defined for each subtype of each object type. Extended description skeletons will appear when the extended description of an object is edited the first time.

See *Extended Description Skeleton* in the section *Defaults* in the *Predict Administration documentation*.

Protecting Parts of Extended Descriptions

Text in an extended description skeleton can be protected by enclosing it in a pair of special characters. The characters are defined in the Maintenance Options screen of the Modify User Defaults function with the parameters Start/End character protect extended desc. See *Maintenance Options* in the section *Predict User Interface* in the *Introduction to Predict documentation*.

Using Con-form in Extended Descriptions

If Software AG's Con-form text formatting facility is installed and the Use Con-form parameters in the current session profile and output options are set to Y, Con-form instructions will be interpreted in the following cases:

- when the command `DISPLAY` is entered while executing the function Edit description, or
- with display-oriented retrieval functions (both online and when printed)

Con-form instructions will *not* be interpreted with the Display function called from a maintenance menu.

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.

Available formatting instructions are detailed in the Con-form Reference documentation and summarized in the Con-nect Quick Reference Guide. The size of the Con-nect buffer area is determined by the Natural parameter CSIZE. For details of this parameter see the *Natural Parameter Reference* documentation.

Keywords and Owners

Predict objects can have lists of keywords and owners, attributes which usually describe the business context of the object. Keywords and owners help to find and process all objects that belong to a given context. See *Users/Owners* in the section *General Information on Predict Functions* in the *Introduction to Predict documentation*.

Keywords

A keyword usually refers to a particular business purpose for which certain data processing objects are used: for example, a business area or a project.

- A keyword must exist as a Predict object before it can be assigned to an object. See the section *Keyword* in the *Predefined Object Types in Predict documentation*.
- Objects of type keyword cannot have owners as attributes.
- Up to 32 keywords can be assigned to any Predict object.

Maintaining Keyword Lists of an Object

The following rules apply when assigning keywords:

- Keyword objects are added or removed with the Add, Copy, Modify or Rename function.
- Up to 32 keywords separated by the current input delimiter character can be specified in an input line near the top of the initial screen (behind the Keys parameter). The input delimiter character is defined by the Natural GLOBALS command ID parameter.
- An asterisk before the Zoom field indicates that more keywords have been specified than can be displayed on one line. In this case enter Y here to modify existing keywords or add new keywords.
- To display a range of keyword objects for selection, enter a keyword ID ending with an asterisk. Mark a keyword in the list with any character or by positioning the cursor on the desired keyword.

Owners

The attribute Owner can be used to document who is responsible for an object or who uses it.

An owner is not a Predict object type, it is an attribute that all Predict object types can have.

Creating Owners

The following rules apply when assigning owners:

- An owner is created by adding its ID to at least one owner list of a Predict object of type user.
- Any user can belong to several owners.
- The owner list of dictionary object can contain up to 99 owners.

Maintaining the Owner List of an Object

These lists can be edited using a full-screen editor which is specially provided for this purpose. See ***Editors in Predict***. The editor is invoked in one of the following ways:

- By entering Y in the field Additional attributes in the bottom line of any Add, Copy or Modify screen and then selecting Owner. See also **Using the Additional attributes / Associations Line**.
- With function Edit owners of an object - Code O in a maintenance menu.
- With command `EDIT<object-type>OWNER<object ID>`.

Disallowing or Forcing Owner Entries

The data dictionary administrator can make the adding of owners optional, prohibited or mandatory by setting the metadata administration parameter Edit owner to Allow, Disallow or Force. This parameter can be specified for each object type. If the Edit owner parameter is set to Allow, any

user can specify a default to be displayed in the Owner parameter in the additional attributes window of Add/Copy/Modify screens.

See *Maintenance Options* in the section *Predict User Interface* in the *Introduction to Predict documentation*.

Links to Child and Parent Objects

Predict objects can be linked to other objects in a parent-child relationship if a corresponding association has been defined. Associations are either

- *predefined*, for example systems can have children of type program, or
- *user-defined*: the DDA can define associations for user-defined object types using metadata administration functions. Also, a default child type can be defined for each user-defined object type.

A link is established by entering the ID of an object into the child list of another object. For example, systems have a program list, databases can have a file list and a dataspace list.

Maintaining the Child List of an Object

Lists of child objects of a Predict object are edited with the object list editor. See ***Editors in Predict***.

The object list editor can be invoked in one of the following ways:

- By calling the function Link children and specifying an association in a maintenance menu.
- With the command `LINK <object type><active-association-type><object ID>`
- By entering Y in the field Association in the bottom line of an Add, Copy or Modify screen. This option is restricted to certain associations, depending on the type of object. See also ***Using the Additional attributes / Associations Line***.

The retrieval function Objects with children reports on associations of objects.

The basic attributes of fields that apply to most types of data storage systems are defined in the input fields of the screen shown below. This screen is displayed for the Add/Copy/Modify field function.

Using the Additional attributes / Associations Line

Add, Copy, Modify screens and screens to enter type-specific attributes can contain an Additional attributes / Associations line at the bottom of the screen. Options in this line invoke subsequent input screens or editors for maintaining attributes. The Additional attributes / Associations line of the Modify database screen is shown below:

```
* Additional attributes ..* N        * Associations ..* S
```

Options of the Additional attributes / Associations line

Option		Description		
*		An asterisk before any option in the bottom line indicates that attributes of the respective type exists.		
Associations	Y	If associations are to be defined. Select one or more associations from the upcoming screen.		
Additional attributes	Y	If additional attributes are to be defined. The following options are valid for all object types:		
		Owner	O	If the owner list of an object is to be edited. The Predict Link editor is invoked. See *Editors in Predict* .
		Description	W	If the extended description is to be edited. One of the following editors is called depending on the environment in which you are working and your current profile settings. ■ the Natural-based Predict Description Editor ■ the Software AG Editor ■ Word for Windows See *Editors in Predict* .

Note: All type-specific options in the Additional attributes / Associations line (options other than editing the description or owner of a Predict object type) are described in the respective section of the Predefined Object Types in Predict documentation.

Creating and Modifying Type-Specific Attributes

In addition to general Predict attributes, objects have a variety of type-specific attributes. For example, programs have the attribute language and fields have the attribute length. The following rules apply:

- Type-specific attribute values can be added, removed or modified with the Add, Copy or Modify functions.
- For many type-specific attributes, input fields are provided in the first input screen of these functions.
- For objects of some types, additional type-specific attributes can be added by entering Y in the Additional attributes field in the bottom line of the screen.
- Some type-specific attributes are in the form of lists. These lists can be edited with the object list editor. See ***Editors in Predict***.

All type-specific attributes are explained in detail in the respective section of the *Predefined Object Types in Predict documentation*.

Required Attributes for User-Defined Object Types

For user-defined object types it is now possible to define attributes as required.

- If an attribute is defined as required in the metadata administration, a value must be entered and this value must conform to the range or table of values defined.

The following exceptions to this rule apply, however, and the attribute does not have to be entered if:

- the screen containing the attribute is not called
- the attribute was defined as required after the object type was added
- the user is not given the opportunity to specify the value, for example when working with the link editor.

If an attribute is *not* defined as required, the attribute may be left blank.

III Editors in Predict

This section describes the editors that are used to create or modify attributes of objects and output from various Predict functions. Predict provides full text editors and list editors.

The Predict text and list editors are designed for their specific tasks.

For example: the Rule Editor for the processing rule of a verification includes line sequence numbers automatically.

General Information

Natural Editor

Software AG Editor

Task-Specific Editors

Transferring Text

34 General Information

- Defining Editor Defaults 194
- Calling an Editor in Predict 195
- Functional Scope of the Editors available in Predict 196

Defining Editor Defaults

Specify your preferred editor in the Profile > Handling screen shown below. Since not every editor can be used for every task in Predict, and not every editor is supported in every environment, the editor that is actually called will depend on the function and your current environment.

For example:

- If you have the Software AG Editor as first choice and the Natural Editor as second choice editor, the **Link Editor** (a modified Natural Editor) will be called for the function Link children because the Software AG Editor cannot be used to edit a link list.
- The Software AG Editor will always be called to process the result of a retrieval operation, irrespective of the preferred editors specified in the screen below.

```
13:01:26             *****  P R E D I C T  *****              2007-05-31
                                - Handling -

Profile ......... HNO

Display first screen (logo screen) ....... Y  (Y,N)
Menu type ..............................* F  Function Main Menu
Use as editor (first choice) ...........* S  Software AG Editor
              (second choice) ..........* C  Natural Editor
              (third choice) ...........*    (none)
  Additional parameters for Word for Windows
    Download PC part ..................... N
    Entire Connection task name ..........
    Predict Word for Windows directory
    (Directory must end with backslash) ..                                  -
```

Note: It is recommended to set the editor preferences as shown in the screen above.

Limited Checks with Word for Windows as First Choice Editor

If Word for Windows is specified as first choice editor, Predict only checks whether the connection from host to PC is active. This means that the Natural parameter PC is set to ON, either by

- specifying PC=ON when starting Natural or
- entering %+ in the command line of your Predict session.

If Word for Windows cannot be called for any other reason (for example due to an invalid task in Entire Connection or because workfiles are incorrectly defined), the function Edit description or

Edit extended description skeleton will be terminated. Your second or third choice editors will not be invoked.

Effects of Editor Defaults

The preferences specified in the Profile > Handling screen affect the following maintenance functions in Predict:

All object types	Edit description Edit default extended description
Extract	Edit/link objects **Note:** For extract maintenance function Edit/link objects: If your first-choice editor is SAG or Word for Windows, the Software AG Editor is called as link editor. If your first-choice editor is Natural, the function Build/Extend an Extract is called.
Field	Edit field expression
File	Edit subquery of a file
Program	Edit procedure code of a program
Verification	Edit rule of a verification

Calling an Editor in Predict

The Natural Editor and the Software AG Editor must be installed in order to use Predict.

Natural Editor

The **Natural Editor** can be called using one of the following methods. The editor must have been given a higher priority in the Profile > Handling **screen**.

- Enter Y in the Additional attributes field in the bottom line of any Add, Copy or Modify screen and select, for example, Description to edit the extended description of the current object.
- Call an Edit function in a maintenance menu, for example Edit subquery of a file in the File maintenance menu to edit the subquery expression of a file.
- Enter an `EDIT` command, for example `EDIT FILE DESCRIPTION <file-id>` to edit the extended description of the specified file.

Software AG Editor

The **Software AG Editor** is called automatically for the following functional areas.

- Retrieval
- Active retrieval
- File implementation: Display implementation plan
- Function Display for file, database, dataspace and storagespace administration
- Online help

The Editor is called irrespective of which editor is defined as preferred editor in the Profile > Handling screen.

Word for Windows

Word for Windows is called automatically for functions Add / Modify extended description, Add / Modify extended description skeleton. The editor must have been given a higher priority in the Profile > Handling screen and the other prerequisites must be met. See *Maintaining Extended Descriptions with Word for Windows*.

Functional Scope of the Editors available in Predict

Attribute	Natural-Based Editors				Software AG Editor	Word for Windows
	Description Editor	Subquery Editor	Rule Editor	Link Editor		
Extended Description	Y				Y	Y
Extended Description Skeleton	Y				Y	Y
Check Expression of SQL Table	Y				Y	
Trigger code	Y				Y	
Subquery Clause of SQL View		Y			Y	
Derived field expression		Y			Y	
Rule of a Verification			Y		Y	
Link List (child objects, Owner list and Entry Points of a Program)				Y		
Function						
All retrieval and active retrieval functions					Y	
File Implementation function Display implementation plan					Y	

Function Display for File, Database, Dataspace and Storagespace administration					Y	
Extract Maintenance function Edit/link objects					Y (see Note)	
Special function Maintain help texts > Modify help text	Y					
Special function Maintain help texts > Display help text Online help					Y	

Note: For extract maintenance function Edit/link objects: If your first-choice editor is SAG or Word for Windows, the Software AG Editor is called as link editor.If your first-choice editor is Natural, the function Build/Extend an Extract is called.

35 Natural Editor

- Functional Scope 200
- Prerequisites 200
- Presettings 200
- Profile 200
- How Editor Commands are Called 201
- Basic Functions 202

Functional Scope

"Natural Editor" is a general term used to describe the Predict editors listed below. These editors can be used to modify all editable attributes of Predict objects. Many main and line commands are common to all editors. All these editors share a common profile. Editor-specific differences are described under ***Task-Specific Editors***.

Prerequisites

No additional prerequisites need to be met when working with the Natural Editor.

Presettings

The following commands can be used to customize the Natural Editor:

- `PROFILE [name]`, see below.
- `SET ESC` **char**.

Profile

Common Natural and Predict Profile

The editor profile you define in Predict and your Natural profile are identical. Changes you make to your Predict profile may not be suitable for working in Natural and vice-versa.

System Profile

If a user does not have a profile defined, the profile SYSTEM is taken as default. The SYSTEM profile can be modified and saved under a given name.

The SYSTEM profile can be modified in module USR0070P in library SYSLIBS.

Upper and Lower Case

A combination of Predict default and Natural parameter settings determine whether text in an extended description is converted to upper case. See *table of possible values and recommendations*.

How Editor Commands are Called

Editor functions are executed with one of the following types of commands:

- **Main**
  commands are typed in the command field of the editor (top left corner).
- **Line**
  commands are typed at the beginning of a line and start with an escape character, normally a period (.).

Note: If Con-form instructions are to be used, another escape character must be assigned in the session profile and with the **SET ESC command** of the editor.

Note: The special line commands *X, *Y and *CURSOR that allow block-oriented text editing must be assigned to a PF key.

Command Field	Indicator Field

```
>                      > + FI: JCA-FI1                          L: 1    S: 3
 All  ....+....1....+.... Extended description ...+....5....+....6....+....7..

      This is a sample extended description
      This is the second line of the description.
```

Undoing Changes

Changes to text that have not been confirmed with ENTER can be undone with

- the main command `LET`
- the line command `.L`

Indicator Field

The value in the indicator field (plus or minus) determines the direction of certain main and line commands.

Examples:

- The main command `ADD` adds blank lines at the beginning (-) or end (+) of the text.
- The line command .C copies text above (-) or below (+) the current line.
- SCAN commands search forward (+) or backward (-)

The value can be changed by overwriting it.

Basic Functions

The basic functions of the Natural Editor below are described in greater detail in the Natural Utilities documentation. The section here gives an overview of the basic functionality of the Natural Editor and describes any differences of the individual functions when used in Predict. This section covers the following topics:

- General Functions
- Scrolling
- Marking Text
- Copying Text
- Inserting Lines
- Deleting Lines
- Moving Lines
- Modifying Lines
- Searching and Replacing Strings

General Functions

The parts of editor commands and parameters that can be omitted are enclosed in square brackets ([]).

Editor Command	Description
*=	Repeat the previous editor command.
*	Display the previous editor command.
CAT, SA[VE][R[ETURN]]	Save and quit the editing session. If RETURN is specified: Save and stay in the editing session. **Note:** Additional syntax checks are performed when cataloging or saving subquery clauses or rules.

Editor Command	Description
E[ND], Q[UIT], .	End the editing session and leave the editor (without saving the text).
HELP, ?	Display a summary of commands for the editor.
HELP nnnn, ? nnnn	Display the specified Natural error message in the Natural help system.
HELP DICnnnn, ? DICnnnn	Display the specified Predict error message.
LET	Undo changes to the text that have not yet been confirmed with ENTER.
PROFILE	Display the current Natural Editor profile settings, for example PF keys and escape character assignments. This profile can be modified and saved under a given name. The default is the Natural Editor profile "SYSTEM". **Note:** Predict and Natural use the same **profile**. Changes made to the profile here are also in effect for your Natural session.
PROFILE name	Load the specified profile for the current session.
SET ESC char	Change the escape character with which editor line commands begin to char (example: SET ESC ;). **Note:** If Con-form is used, you cannot use a period (.) as escape character.

Scrolling

Some of the following commands are normally assigned to PF keys.

Editor Commands	Description
+|- n	Move forward|backward n lines.
+|- H	Move forward|backward half a page (10 lines). These commands are normally assigned to PF5 (move forward) and PF4 (move backward).
+|- [P]	Move forward|backward one page (20 lines). -P is normally assigned to PF1.
SET STAY	No scrolling is done when ENTER is pressed and the contents of the screen have not been altered.
SET STAY OFF	The screen display is moved forward one page when ENTER is pressed and the contents of the screen have not been altered.
B[OT], ++	Move to the bottom of the text. This command is normally assigned to PF3.
T[OP], -	Move to the top of the text. This command is normally assigned to the key PF2.
POINT	Move the line where the .N line command was given to the top of the screen (see description of .N command below).
X|Y	Move to the line marked with the line command X or Y (see description of line commands below).

Line Command	Description
.P	Position the current line to the top of the screen.

Marking Text

A section of text can marked by marking its first line with X and its last line with Y. Copy, delete, move, modify and scan operations can then be restricted to a range of text.

Note: If the PF key commands *X and *Y are used, the marked section can include partial lines.

Editor Command	Description
RES[ET]	Delete X and Y marks and reset values specified with the SCAN command.

Line Commands	Description
.X, .Y	Mark the current line with X or Y. These marks are used by the following commands: - Main commands DX, DX-Y, DY, EX, EX-Y, EY, X,Y - Line commands .CX, .CX-Y, .CY, .MX, .MX-Y, .MY.
*X, *Y	Mark the line and column of current cursor position with X or Y. If marks have been set with *X and *Y, the following commands process the text section starting in the line and column marked with *X and ending in the line and column marked with *Y. - Main commands DX-Y and EX-Y - Line commands .CX-Y and .MX-Y. **Note:** The commands *X, *Y must be assigned to PF keys. See Customizing Predict with Profiles in the section *Predict User Interface* in the *Introduction to Predict documentation*.
.N	Note the current line (see description of the POINT command in the section **Scrolling**.)

Copying Text

Main Commands	Description
EXP[ORT]	See *Exporting Text to an External Target*
IMP[ORT]	See *Importing Text from an External Source*
SEL[ECT]	See *Selecting Text from another Source within Predict*

Note: The commands above are not applicable to the Link Editor.

The following commands are used for copying text within an object (for example an extended description).

Line Commands	Description
.C[(n)]	Copy the current line n times (default: 1 copy).
.CX[(n)]	Put n copies (default: 1 copy) of the line marked with X either above (-) or below (+) the current line. If a character is marked with X, do not copy characters before it in the line.
.CX-Y[(n)]	Put n copies (default: 1 copy) of the text between the X and Y marks (inclusive) either above (-) or below (+) the current line.
.CY[(n)]	Put n copies (default: 1 copy) of the line marked with Y either above (-) or below (+) the current line. If a character is marked with Y, do not copy characters after it in the line.
.H	Identifies the target line of an `IMPORT` or `SELECT` command. Depending on the value in the **indicator field**, the text is added above (-) or below (+) the current line.

Inserting Lines

Editor Command	Description
ADD	Insert 10 blank lines at the beginning (-) or end (+) of the text and scroll accordingly. New lines that are still blank will be deleted when the ENTER key is pressed.

Line Commands	Description
.I[(n)]	Insert n (default: 10) blank lines either above (-) or below (+) the current line.
.W(n)]	Inserts n (default: 10) blank lines either above (-) or below (+) the current line. New lines that are still blank will be deleted when the ENTER key is pressed.

Deleting Lines

Editor Commands	Description
CLEAR	Deletes all lines. Note: that Predict reads the default extended description skeleton whenever an empty extended description is opened with the description editor.
DX	Delete the character marked with X and characters after it in the same line. If a whole line is marked with X, delete it.
DX-Y	Delete the text between the X and Y marks inclusive.
DY	Delete the character marked with Y and characters before it in the same line. If a whole line is marked with Y, delete it.
EX	Delete all text before the X mark (exclusive).
EX-Y	Delete all text before the X mark and all text after the Y mark (exclusive).
EY	Delete all text after the Y mark (exclusive).

Line Command	Description
.D[(n)]	Delete the current line and the following n-1 lines.

Moving Lines

Line Commands	Description
.MX	Move the line marked with X to either above (-) or below (+) the current line. If a character is marked with X, do not move characters before it in the line.
.MX-Y	Move the text between the X and Y marks (inclusive) to either above (-) or below (+) the current line.
.MY	Move the line marked with Y to either above (-) or below (+) the current line. If a character is marked with Y, do not move characters after it in the line.

Modifying Lines

Note: With the exception of the command .L, the commands for modifying lines are not available in the Link Editor.

Editor Commands	Description
COM[PRESS]	Concatenate text between the X and Y marks (inclusive) into as few lines as possible. If no X mark is set, concatenate from the beginning of the text. If no Y mark is set, concatenate up to the end of the text.
SHIFT +n	Shift the lines between the X and Y marks (inclusive) n columns to the right.
SHIFT -n	Shift the lines between the X and Y marks (inclusive) n columns to the left.

Line Commands	Description
.J	Join the next line to the end of the current line.
.L	Restore the current line to the state it had immediately after the ENTER key was last pressed.
.S	Split the current line at the cursor position.
*CURSOR	`*CURSOR` performs two operations: The current line is split at the cursor position and marked text or blank lines are inserted between the two split parts. The line command (.C, .M, .I or .W) entered in the line of the cursor position when *CURSOR is executed determines whether text or blank lines are entered.

Searching and Replacing Strings

The plus (+) or minus (-) sign displayed in the indicator field at the top of the editor screen determines whether search commands are executed forward (towards the end of the text) or backward (towards the beginning). Change the +/- sign by overwriting it.

The `SCAN` and `CHANGE` commands contain a special character that delimits strings of text. In the table below, the apostrophe (') is used as delimiter but another special character can be substituted.

The way `SCAN` and `CHANGE` work can be changed with the `SET` command. See *Setting SCAN Options* below.

Editor Commands	Description
SCAN 'string'	Search either forward (+) from the top of the current screen display to the end of the text or backward (-) from the bottom of the current screen display to the beginning of the text for lines containing the specified string, mark these lines with S and move to the first of them. See the `SET ABS`, `SET RANGE` and `SET NULL` commands.
SCAN, SC	Display a menu offering all the SCAN/CHANGE options mentioned above. This command is normally assigned to the key PA2.
SCAN=, SC=	Repeat the previous `SCAN` command. This command is normally assigned to PF6.
CHANGE 's1's2'	Scan for lines containing s1, replace each s1 with s2 and mark the line with R.

Setting SCAN Options

Option	Description
SET ABS [ON]	Absolute scan. `SCAN` and `CHANGE` find all occurrences of the string no matter what comes before or follows them.
SET ABS OFF	Cancel the above setting. A string will be found only if it is preceded and followed by a space, new line or special character.
SET RANGE [ON]	Limit the range of `SCAN` or `CHANGE` to the lines between the X and Y marks inclusive.
SET RANGE OFF	Cancel the `SET RANGE` command.
SET NULL [ON]	Delete all strings that satisfy the next `SCAN` command.
SET NULL OFF	Cancel the `SET NULL` command.

36 Software AG Editor

- General Information 210
- Common Functions of the Software AG Editor 212
- Troubleshooting 213

Each object in Predict has attributes which can be modified with the Software AG Editor. In addition, many Predict functions produce output which can be processed by the Software AG Editor. The functions of the Software AG Editor are described in detail in the *Natural Editors* documentation.

Where to Find More Information:

Installing the Software AG Editor is described in the *Natural Installation* documentation. The *Natural Utilities* documentation provides information on the editor buffer pool services utility SYSEDT.

General Information

- Functional Scope
- Prerequisites
- Software AG Editor Profiles
- Software AG Editor Screen
- Long Lines

Functional Scope

Editing Attributes

The Software AG Editor can be used to edit the following attributes of Predict objects:

- Extended descriptions and extended description skeletons
- Derived field expressions/subquery clauses of SQL views
- Rules of a verification
- Extract link list, see the section Extract in the Predefined Object Types in Predict documentation.

Editing Output from Predict Functions

Software AG Editor is called as standard for further processing of output from the following functional areas:

- Retrieval
- Active Retrieval
- File Implementation: Display implementation plan
- Function Display for file, database, dataspace and storagespace administration
- Online help

The Software AG Editor is called even if the Natural Editor is defined as first choice in the Profile > Handling screen.

Note: With the parameter Defaults > General Defaults > Miscellaneous > Use Software AG Editor for output it is possible for your administrator to deactivate the Software AG Editor for the above functions. If this parameter is set to N, retrieval output appears as in Predict version 3.2. See the section Defaults in the Predict Administration documentation.

Prerequisites

The Software AG Editor is available as standard in the Natural version required for Predict: No additional prerequisites need to be met.

Software AG Editor Profiles

To a certain extent the functions of the Software AG Editor can be adapted using the parameters of the editor profile. All available options are described in the Software AG Editor documentation.

The following rules apply when using the Software AG Editor in Predict:

- Enter command SET to define an editor profile.
  - If the user does not have his/her own profile, the SYSTEM profile is displayed. This can be modified as required and saved under a new name.
  - If the user has already defined a profile with his/her ID, this is displayed for modification instead of the SYSTEM profile.
  - If another user-defined profile is active, this is displayed for modification.
- The SYSTEM profile can be modified by any user.
- Different profiles can be stored under different names. If a user ID is used as a profile name, the profile can only be modified by this user.
- Editor profiles are activated with the command USE profile_name.

Software AG Editor Screen

The Software AG Editor screen as used in Predict is divided into four parts:

Header Lines

This area provides the following information:

- ID of current object
- type of attribute being edited, for example Extended Description
- date and time.

Line Command Area

This area contains line numbers. It is used for entering line commands and can also contain information pertaining to the current line. Line commands are described in detail in the Software AG Editor Reference documentation.

Data Area

This area contains the text of the attribute.

Footer Lines

This area is used for entering main commands, and indicates the active PF keys and the active SCROLL value. Main commands are described in detail in the Software AG Editor Reference documentation.

Long Lines

The Software AG Editor enables you to edit and save lines of up to 250 characters. Lines exceeding 72 characters are marked =long.

Common Functions of the Software AG Editor

Standard Commands

For a detailed description of main and line commands, refer to the Software AG Editor Reference documentation.

Main Standard Commands

Main commands are issued from the command line. All Software AG Editor commands are available in Predict except the **deactivated main commands**.

Line Commands

You can enter a line command on any data line by typing over the prefix information on the left of your edit screen. You can also enter a line command in the first column of the object itself if you precede it with the Editor escape character. The escape character is generally a period (.) or comma (,). A line command always applies to the line in which you enter it (or to a block of lines marked by multiple line commands).

Line commands can also be entered in command line at the top of the Editor screen. In this case, the command must be preceded by a colon (:) and applies to the line marked by the cursor.

Additional Editor Functionality in Predict

The following commands have been implemented in the Software AG Editor especially for use in Predict.

Main Commands

NUM[BER]	Activate line numbering.
NUM[BER] OFF	Deactivate line numbering.
SET	Define an editor profile. See *Software AG Editor Profiles*.
USE	Activate an editor profile. See *Software AG Editor Profiles*.
KEYS	Define PF keys. A window appears for defining PF keys and descriptive text. Different PF key definition settings can be stored under different names. If a user ID is used as a name, the PF key definition can only be changed by that user.

Deactivated Main Commands

The following Software AG Editor commands have been deactivated because the corresponding functions are not applicable in Predict.

COPY, CREATE, CREATEALL, ENTER, FILE, MSO, NEXT, NEXTFILE, PREV, PREVFILE, PUTATTR, QUEUE, REPLACE, SEQUENCE, SORT, SUBMIT, TYPE, UNREN.

Troubleshooting

This section lists messages that may appear concerning the Software AG Editor buffer pool. Related messages are grouped together, and corrective action is suggested.

Installation Errors

Message	Explanation/Action
SSIZE too small	Modify your Natural parameter module using SSIZE=64. This message is only applicable to mainframe environments.
Checkpoint file not active -caution- profile changed to "recovery off" (from "recovery on") because checkpoint file not active (in Editor session)	The Editor workfile was not formatted correctly, no space was reserved for the checkpoint file. Check your job that allocates the workfile.
Failure in open Buffers	The allocation of the Editor area failed or was not sufficient (SSIZE buffer). Check the SSIZE parameter. If the parameter has a value of at least 64, you can either increase the MSIZE parameter, or decrease the

Message	Explanation/Action
	size value of other buffers you do not need, for example CSIZE, FSIZE or TSIZE (see the *Natural Parameter Reference* documentation).
BP not active	The buffer pool was not initialized correctly. See the section *Installing the Software AG Editor* in the *Natural Installation* documentation.

Editor Workfile and Timeout Parameters

Message	Explanation/Action
Checkpoint of file failed	Mainframe: These messages indicate that the Editor workfile is too small or the timeout parameters are too large, causing data to be kept in the buffer pool too long. You must increase the workfile size or decrease the timeout parameters. Open Systems: `FIN`ish your current Natural session. Change the number of buffer pool blocks by increasing the value for `EDTBPSIZE` in your Natural parameter module. The next Natural session you start with uses the increased number of buffer pool blocks.
No space for data block	
No space for index block	
Cannot allocate BP file -caution- profile changed to “recovery off” (from “recovery on”) because checkpoint file is full (in Editor session) -caution- profile changed to “log off” (from “log on”) because log file is full (undo command not active) (in Editor session)	
Cannot allocate BP block	
Cannot allocate checkpoint block	
Suspending of a session failed	
Activation of a session failed	The session has been deleted due to a buffer pool timeout. Restart the session.

Internal Software AG Editor Problems

The following messages are not usually displayed. They appear in the rare case of some internal problem.

Message	Explanation/Action
Some data may be ignored	Contact your Software AG support representative.
Session does not exist	
Cannot free BP block	
Log error	

I/O Problems on Editor Workfile or Volume

Message	Explanation/Action
I/O ERROR	Check the workfile. If the error persists, reallocate the file on another volume.
Cannot read index block	
Write to BP failed	
Read from BP failed	
Read failed	
Error reading log block	
Cannot read checkpoint block	
Cannot write checkpoint block	
Recovery of file failed	

Other Inconsistencies

Message	Explanation/Action
Error in getting lines	The Natural source area is corrupted, for example if the Predict editor session is aborted by %% and the source area was modified.
Getting text failed	The Editor message table is inconsistent.

37 Task-Specific Editors

- Description Editor 218
- Subquery Editor 220
- Rule Editor 221
- Link Editor 223

Description Editor

Functional Scope

The description editor is used for editing

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

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

Description Editor Specific Commands

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

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

Upper and Lower Case in Descriptions when using the Natural Editor

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

- The Predict profile parameter Upper/lower case > Description is specified in function Defaults > General Defaults > Miscellaneous. This setting is made by the Dictionary Administrator and applies to all Predict users.
- The Natural profile parameters Editing in lower case and Dynamic conversion of lower case can be defined by each user individually.

- The Predict setting has priority over the Natural parameter settings: If Predict default Upper/lower case > Description is set to Upper case, descriptions are stored in upper case irrespective of the settings in your Natural profile.
- If Predict default Upper/lower case > Description is set to Lower case, descriptions are stored in upper or lower case depending on the settings in your Natural profile.

Overview

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

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

Recommended Natural Editor Settings

Recommend settings for editing extended descriptions with the Natural editor:

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

Upper and Lower Case Settings for the Software AG Editor

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

- The Predict profile parameter Upper/lower case > Description is specified in function Defaults > General Defaults > Miscellaneous. This setting is made by the Dictionary Administrator and applies to all Predict users.
- The SAG profile parameter CAPS can be defined by each user individually.

- The Predict setting has priority over the Software AG Editor setting: If Predict default Upper/lower case > Description is set to Upper case, descriptions and other attributes are stored in upper case irrespective of the settings in your Natural profile.
- If Predict default Upper/lower case > Description is set to Lower case, descriptions are stored in upper or lower case depending on the settings in your Software AG Editor profile.

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

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

Recommended Software AG Editor Settings

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

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

Subquery Editor

Functional Scope

The Subquery Editor is used to edit the following:

- subquery clause of an SQL view
- derived field expression

Subquery Editor Specific Commands

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

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

Rule Editor

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

Rule Editor-Specific Commands

Editor Commands	Description
CAT [[FREE] R[ETURN]], SA[VE] [[FREE] R[ETURN]]	Save a text as a rule of a verification and leave the editor. **Note:** Note: that the SAVE or CAT command does not perform a syntax check. The syntax is checked however, when cataloging a map that uses a rule.
Option FREE	Catalog/save the edited rule as a free rule. This command is only available when creating new rules and when editing conceptual rules.
Option RETURN	Save a text as a rule of a verification and stay in editor.
C[HECK]	Check whether the edited rule's Natural syntax is valid and report errors.
GEN[ERATE]	Generate a rule from the values defined in the verification and add it to the end of the Natural source in the rule editor. This command is not available for verifications of type U. See **Verification types of the GENERATE command** for the Natural statements that are generated for the different types of verifications.
GEN[ERATE] N	Generates a rule for Natural Construct from a verification of status documented (D). The status of the verification will be changed to N.
GEN[ERATE] S	Generates an SQL clause for all verification types except user routine. When the rule is saved, the status of the verification is changed to S.
GLOBALS SM=OFF	Switch to the reporting mode of Natural.
GLOBALS SM=ON	Switch to the structured mode of Natural.
RENUM[BER], N	Renumber the source lines in steps of 10 and renumber references to them accordingly.

Editor Commands	Description
RUN, CHECK	See **Rule commands**.

Verification types of the GENERATE command

Generate a rule from the values defined in the verification and add it to the end of the Natural source in the rule editor. This command is not available for verifications of type U. The table below shows which Natural statements are generated for the different types of verifications:

Type	Description			
E	Equal to	1	IF NOT (&= value)	& = value
		0 or 1	IF NOT & /* for format logical	& = value
G	Greater than	1	IF & LE value	& > value
L	Less than	1	IF & GE value	& < value
N	Not equal to	n	IF (&= value1 OR= value2 ...)	& ^= value 1
		0 or 1	IF & /* for format logical	& ^= value 2 ...
R	Range of values	2	IF NOT (&= value1 THRU value2)	& between value1 and value2
T	Table of values	n	IF NOT (&= value1 OR= Value2 ...)	& in (value1, value2...)
U	User routine			
B	Range, but not	3	IF NOT (&= value1 THRU value2 BUT NOT value3)	& between value1 and value2 and & ^=value3
		4	IF NOT (&= value1 THRU value2 BUT NOT Value3 THRU value4)	& between value1 and value2 and & not between value3 and value4
I	Not in range	2	IF (&= value1 THRU value2)	&not between value1 and value2
blank	(none) - no rule defined			

Rule commands

Check the edited rule. If no errors are found, a map is produced with which the user can test the rule by entering input values. The following rules apply:

A	A	66
B	B	33
D	D	
L	L	1
N	N	27

1. RUN tests a rule of format K (function key) without input data.
2. For a rule of format L (logical), a blank space means false and any other input value means true.

3. The stack must not be changed.
4. The contents of the source area must not be changed.

Note: All variables used except the ampersand (&) must be defined within the code.

5. The variable names SYSDIC-C1 and SYSDIC-C2 are used for internal purposes and must not be used within the rule.
6. The source will be renumbered.

Link Editor

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

- Functional Scope
- Calling the Link Editor
- Link Editor Commands for all Object Types
- Type-specific List Editor Functions

Functional Scope

The Link Editor is used to edit the following lists:

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

The Link Editor displays a formatted list of related objects.

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

The example below shows the field list of a file.

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

Calling the Link Editor

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

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

Link Editor Commands for all Object Types

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

Editor Commands	Description
C[HECK]	Performs various validation checks, for example that no IDs in the list are duplicated.
SEL	Displays a list of objects for selection. A screen appears in which type-specific criteria can be entered to limit the scope of the selection. Objects can then be inserted in the current list either at the end (mark with +) or at the beginning (mark with -). To insert an object at a given position, enter the line command .H simultaneously. Objects will then be inserted after the line marked with .H. This command allows you to include fields from another file in the field list currently being edited. It can be used to pull forward fields from a standard file to a master file or from a master file to a related userview.

Editor Commands	Description
SORT	Sorts the list of object IDs alphabetically. When sorting field lists containing structures or redefined fields, the following applies: A CHECK command must be executed before a field list containing a structure/redefinition that has been changed is sorted. Fields not on level 1 are not sorted, so group structures are not changed. This command does not apply to verification lists.

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

Type-specific List Editor Functions

Database

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

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

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

File - all File types

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

Editor Commands	Description
ADA	Generate field short names for fields that do not already have a short name.
NU[LL]	Only applicable for new fields: If the suppression option of a field is blank, it is reset as follows: ■ For SQL tables: the suppression option is set to U (null allowed). ■ For other files, the suppression options is set to N (null value suppression).

Editor Commands	Description
READA	Delete any existing field short names and generate new ones for all fields. Only applicable to fields with attribute Field short name.
SORT ADA	Sort the field alphabetically by field short name. Fields not on level 1 are not sorted, so group structures are not changed. Only applicable to fields with attribute Field short name.
SORT	Sort the fields alphabetically by field ID. Fields not on level 1 are not sorted, so group structures are not changed.
SET ADA [ON]	Apply future SCAN commands to field short names instead of field IDs.
SET ADA OFF	Cancel the command SET ADA [ON].

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

File - SQL File Types

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

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

```
Ty L Field ID                          from Field ID
*- - -------------------------------- --------------------------------
```

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

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

FLIP Return to normal editor input:

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

Owner

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

Program

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

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

38 Transferring Text

- Selecting Text From Another Source Within Predict 230
- Calling the Select Function 231
- Importing Text from an External Source 233
- Exporting Text to an External Target 242
- Maintaining Extended Descriptions with Word for Windows 247

Selecting Text From Another Source Within Predict

Note: The options you have for positioning the text from another Predict source depends on the editor you are using.

Sources

Text can be selected from the following Predict sources:

- extended description
- default extended description
- abstract
- rule of a verification
- SQL subquery expression
- SQL procedure code
- derived field expression
- trigger code

Targets

Text can be copied to the following attributes of Predict objects:

- extended description of any Predict object
- subquery of a file
- rule of a verification
- field expression
- trigger code

Note: With the `IMPORT` command you can also import text from a variety of external sources. See *Importing Text from an External Source*.

Calling the Select Function

Selecting text from another Predict source is a process of up to 5 steps.

Step 1: Enter SEL Command

Enter "SEL" in the command line of your editor. The following screen is displayed.

```
13:07:29             *****  P R E D I C T  *****              2007-05-31
Plan   0                      - Select Text -

   Type of source                                Type of source

   E  Extended description                       V  SQL subquery expression
   D  Default extended description               C  SQL procedure code
   A  Abstract                                   F  Derived field expression
   P  Processing rule                            G  Trigger code


Type of source .. E

Object type ....*                                   Type of SQL code .*
Object ID .......
Belongs to FI ...


Restriction ....*   Profile USR,used


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

Step 2: Specify Type of Source

Enter one of the codes from the menu to select text from the corresponding source. The selection criteria that can/must be specified depend on the type of source.

Step 3: Enter Selection Criteria

Parameter	Description
Object type	This parameter must be specified for source types E, D and A. The other source types apply to one particular object type. This object type is added automatically.
Object ID	The scope of the function can be limited by a fully qualified object ID or ID with asterisk notation. If this field is left blank, all objects that meet the other selection criteria are listed for selection.
Belongs to FI	Only valid for object type field. The scope of the function can be limited by a fully qualified file ID or ID with asterisk notation. If no file ID is specified, fields in all files are listed for selection.
Type of SQL code	Only valid for source types V and C. Possible values for source type SQL subquery expression: L : SQL view subquery T1 : SQL code for table check constraint Possible value for source type SQL procedure code: T2 : SQL code for procedure
Restrictions	Additional restrictions can be specified to further limit the selection. See Restrictions in the section *Predict User Interface* in the *Introduction to Predict* documentation.

A list of all Predict objects that meet the selection criteria is displayed for selection. See example below.

```
13:05:44             *****  P R E D I C T  *****       2007-05-31
Plan   0               - Select Description Of File -

Cmd  File ID                             Type   Fnr    DDM Impl Other

 __  ABC-BT                               BT      0
 __  ABC-FI-S-1                           S       0
 __  ABC-FI1                              A     123
 __  ABC-OT                               OT      0
 __  ABC-OV                               OV      0
```

Step 4: Mark objects to be selected

You have two options:

- Enter "A" in the Cmd column to import the entire object.
- Enter "S" in the Cmd column to display the object. Select the lines you wish to import by marking them with a non-blank character

Step 5: Mark lines of object to be selected

Only applicable if you entered "S" in step 4. Mark lines of object to be copied with any non-blank character.

```
    > Mark line to be selected from: File,JCA-BT                          ↩
<
        ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
    _  line 1 of description
    X  line 2 of description
    _  line 3 of description
    X  line 4 of description
    _  line 5 of description
```

Press ENTER to return to the Select Text menu and PF3 to return to your editor session.

Importing Text from an External Source

- **Positioning the imported text**
- **Calling the import function**

Sources

With this version of Predict you can import text from the following *external* sources:

- **Natural source**, **Natural maps**
- **Con-nect document**,
- **User-defined source**,
- **PC ASCII file**,

Note: With the `SELECT` command you can also import text from a variety of sources within Predict. See *Selecting Text from another source within Predict*.

Targets

The text from the above sources can be imported to the following targets:

- extended description of any Predict object
- subquery of a file
- rule of a verification
- field expression
- trigger code

Positioning the Imported Text

The following rules apply to imported text from any of the sources listed above. Depending on the editor you are using, you can position the imported text as follows:

Natural Editor

If you are using a Natural-based editor, the position of the imported text depends on the value in the indicator field highlighted below.

```
>                     > + FI: USR-FI1                     L: 1    S: 12 All
  ....+....1....+.... Extended description ...+....5....+....6....+....7
```

Plus (+) Text is added to the bottom of the current object

Minus(-) Text is added to the top of the current object

Software AG Editor

If you are using the Software AG Editor, the text is added to the end of the current object as standard. You can also mark any line with H in the prefix area of the editor to import the text to a specific position. The text is imported after the marked line.

```
13:04:43                      - File : USR-FI1 -              2007-05-31
      --------------- < Extended description > ---------------
***** ******************* top of data ************************
00001           t===================================
H0002           =e==================================
00003           ==s=================================
00004           ===t================================
```

Note: It is possible that your Software AG Editor environment has been set up differently. If in doubt, contact your system administrator.

Calling the Import Function

To import the text from one of the sources listed on above, enter command `IMPORT` in the command line of the respective editor. The following screen appears.

The prerequisites and options for the various external environments are described below.

```
> import              > + FI: USR-FI1                          L: 1    S: 5
 All  ....+....1....+.... Extended description ...+....5....+....6....+....7..
                 ====================================
                    File additional description
                 ====================================
      This is the exte +------ Import Text -------+ I1.
      This is the seco !                          !
                       !     Type of source       !
                       !                          !
                       !   N Natural source       !
                       !   C Connect document     !
                       !   U User defined source  !
                       !   P PC ascii file        !
                       !                          !
                       !    Type of source .. N   !
                       +--------------------------+




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

Natural Source

Calling the Function

Enter one of the following in the Editor command line

- Command `IMPORT` to display the Import Text selection window, then source type "N"
- or IMPORT N

The following screen appears:

```
> imp                 > + FI: USR-FI1                          L: 1    S: 5
 All  ....+....1....+.... Extended description ...+....5....+....6....+....7..
                 ====================================
                    File additional description
                 ====================================
      This is the +- Import Source Into PREDICT Description -+
      This is the !                                          !
                  !                        source .. Natural !
                  !      Member ...........                 !
                  !      Library ..........                 !
                  !      User system Fnr ... 54             !
                  !      User system DBnr .. 180            !
                  !      Cipher ...........                 !
                  !      Password .........                 !
                  !                                         !
```

The following parameters are available:

Parameter	Description
Member, Library	These parameters identify the member(s) to be imported. Leave blank or use asterisk notation to display a list of members for selection.
User system Fnr/DBnr	The database and file number of the file containing the member(s) to be imported. These values must be specified.
Cipher, Password	Only required if the file containing the members to be imported is protected by password or cipher.

A list of members that meet the parameters specified is then displayed for selection. See example below.

```
13:03:24             *****  P R E D I C T  *****             2007-05-31
                            Import Natural Source

Cmd    T Member       Library         Program

 __    M:DEMAP        ABC
 __    P:FUNCT1       ABC
 __    P:FUNCT3       ABC
 __    P:MENU         ABC             AB-DYN
 __    P:FUNCT2       ABCSTEP
 __
```

You have two options:

- Enter "A" in the Cmd column to import the entire member.

- Enter "S" in the Cmd column to display the member. Select the lines you wish to import by marking them with a non-blank character

Either all of the member or selected lines will be appended to the current object.

Note: For Linux users: Only programs and subprograms can be imported.

Importing Natural Maps

It is possible to import Natural maps into the extended description of a Predict object. When a member of type Map is selected for import, an additional window appears in which you must define the filler characters to be used in the Predict extended description for modifiable and read-only fields.

```
13:06:20             *****  P R E D I C T  *****             2007-05-31
                            Import Natural Source
                                                  +All--Possible functions------
Cmd    T Member       Library         Program    ! _ A   Insert All
                                                 ! _ S   Select
 __    P:ZBOE1        AAM                        ! _ ' ' No command
 __     +-------------------------------------------------+
 __     !     Please enter filler characters setting !    !
 __     !                                                 !
 __     !   Filler char. for INP/MOD fields ...           !
 __     !   Filler char. for OUTPUT fields ....           !
 __     !                                                 !
 __     !                                                 !
 __     +-------------------------------------------------+
 X_    M:ARHM2        ARH                        !
 __    P:ARH1         ARH             ARH-PR-OT  !
```

Specify parameters Filler char. for INP/MOD fields and Filler char. for OUTPUT fields as required. The map is displayed in the extended description fully formatted in long lines of 250 characters.

```
13:24:55                      - File : USR-FI1 -                    2007-05-31
          --------------- < Extended description > ---------------
    00001           ====================================
    00002              File additional description
    00003           ====================================
    00004
    00005 This is a sample map imported from a Natural member. Filler character for
    00006 output is +, input filler character is &.
    00007
    =long  ++++++++          ++++++++++++++++++++++++++++++++++++++++            +
    =long                    ++++++++++++++++++++++++++++++++++++++++
    =long                                                        ++++++++++++++++++++
    =long  Profile ......... ++++++++                            ++++++++++++++++++++
    =long
    =long  File for select ........................... &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
    =long  Current virtual machine ................* &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

Con-nect Document

Prerequisites

Con-nect must be installed.

Calling the Function

Enter one of the following in the Editor command line

- Command `IMPORT` to display the Import Text selection window, then source type "C"
- or IMPORT C

The following screen appears:

```
> imp                 > + FI: USR-FI1                          L: 1    S: 13
All  ....+....1....+.... Extended description ...+....5....+....6....+....7..
                ====================================
                   File additional description
                ====================================
     This  +---- Import Document into Predict Description -----+
     This  !                                                   !
     DEFIN !                                 source .. Connect !
     END-D !                                                   !
     DECID ! Specification of Connect document                 !
         F !                                                   !
         F ! Cabinet ....                                      !
         F ! Password ...                                      !
         I ! Folder .....
     END   ! Document ...                                      !
           !                                                   !
           +---------------------------------------------------+
```

The following parameters are available:

Parameter	Description
Cabinet	Cabinet of the user from which the text is to be imported. If you are importing text from the cabinet of another user, this user must have granted you permission to his/her cabinet in Con-nect.
Password	Only required if you are importing from the cabinet of another user.
Folder	Enter the name of a folder that exists in the cabinet specified or leave blank. If a valid folder is specified, all documents contained in this folder are listed for selection; parameter Document is not evaluated.
Document	If Folder is left blank, you can enter a value for document. For example: If you you enter “test” for Document, all documents in the specified cabinet starting from “test” are displayed for selection.

A list of documents that meet the selection criteria above is displayed for selection. See example below.

```
13:28:27            *****  P R E D I C T  *****            2007-05-31
                          Import Connect Document

Cmd   Document name                   Date       Description

 __   test                            21-04-2002 test
 __   to_pc                           01-02-2002
 __   todo                            20-02-2002
 __   tran                            19-03-2002
 __   vmsgl                           23-04-2002 VMS-GLOSSARY
 __   who's who                       04-03-2002 (Who is who / Feb. 2002)
 __   xxx                             11-08-2002
```

You have two options:

- Enter "A" in the Cmd column to import the entire Con-nect document.
- Enter "S" in the Cmd column to display the Con-nect document. Select the lines you wish to import by marking them with a non-blank character.

Either all of the member or selected lines will be appended to the current object.

User-defined Source

With this version of Predict you can import text from the sources listed on to any user-defined subsystem.

Prerequisites

User exits U-IMPSC and U-IMPSRP which are delivered in source form with this version of Predict must still exist and must have been adapted to meet your site requirements.

This functionality is described on the documentation file delivered with Predict. See description of the following Program objects for more information.

- SAG-PRD-U-IMPSRC
- SAG-PRD-U-IMPSRP

Modify this program to suit your site-specific requirements.

Calling the Function

Enter one of the following in the Editor command line

- Command `IMPORT` to display the Import Text selection window, then source type "U"
- or IMPORT U

PC ASCII File

Prerequisites

- Your Predict session must be running under terminal emulation in a PC environment.
- Entire Connection for Windows must be installed (Version 4.3.1 or above).
- Workfile 7 must be defined as PC workfile on the mainframe.
- The Natural parameter PC must be set to ON, either by
  - specifying PC=ON when starting Natural or
  - entering %+ in the command line of your Predict session.

Note: In Windows and Linux environments, Workfile 7 must be set to type “Transfer”.

Calling the Function

- Enter one of the following in the Editor command line to call the Upload Data dialog box
  - Command `IMPORT` to display the Import Text selection window, then source type "P"
  - or IMPORT P
- Specify source file in the Upload Data dialog box. The file must be in ASCII format.
- Click the OK button in this window to import the text Predict.

Exporting Text to an External Target

Targets

With this version of Predict you can export text to the following target environments:

- Natural source,
- Con-nect document,
- User-defined target,
- PC ASCII file,

Sources

The text can come from one of the following sources in Predict

- extended description of any Predict object
- subquery of a file
- rule of a verification
- field expression
- trigger code
- the output of a display-oriented function from one of the following functional groups:
  - retrieval
  - active retrieval
  - administration (of external objects)
  - Display function from a maintenance menu.

Calling the EXPORT Function

To export the text, enter command `EXPORT` in the command line of the respective editor or output screen. The following screen appears. The prerequisites and options for the various external environments are described below.

```
> export              > + FI: USR-FI1                          L: 1    S: 4
 All  ....+....1....+.... Extended description ...+....5....+....6....+....7..
                 ====================================
                    File additional description
                 ====================================
      This is the exte +------- Export Data ------+ I1.
                       !                          !
                       !     Type of target       !
                       !                          !
                       !   N Natural source       !
                       !   C Connect document     !
                       !   U User defined target  !
                       !   P PC ascii file        !
                       !                          !
                       !    Type of target .. N   !
                       +--------------------------+




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

Natural Source

Calling the Function

Enter one of the following in the Editor command line

- Command `EXPORT` to display the Export Text selection window, then target type "N"
- or EXPORT N

The following screen appears:

```
> export              > + FI: USR-FI1                          L: 1    S: 4
  All  ....+....1....+.... Extended description ...+....5....+....6....+....7..
                 ====================================
                    File additional description
                 ====================================
       This is the +-------------- Export Data ---------------+
                   !                                          !
                   !                        target .. Natural !
                   !      Member ............                 !
                   !      Library ...........                 !
                   !      User system Fnr ... 54              !
                   !      User system DBnr .. 180             !
                   !      Cipher ............                 !
                   !      Password ..........                 !
                   !                                          !
                   !      Replace object .... N (Y/N)         !
                   !                                          !
                   +------------------------------------------+
```

The following parameters are available

Parameter	Description
Member	Name of the member to which the extended description is to be exported.
Library	Name of the library containing the member.
User system Fnr/DBnr	File and database number of user system file.
Replace object	Y: If the member already exists in the specified library it will be overwritten. N: Text will not be exported and an error message is given.

Con-nect Document

Prerequisites

Con-nect must be installed.

Calling the Function

Enter one of the following in the Editor command line

- Command `EXPORT` to display the Export Text selection window, then target type "C"
- or EXPORT C

The following screen appears:

```
> exp                 > + FI: USR-FI1                          L: 1    S: 5
 All  ....+....1....+.... Extended description ...+....5....+....6....+....7..
                 ====================================
                    File additional description
                 ====================================
      This  +------------------ Export Data --------------------+
      This  !                                                    !
            !                                 Target .. Connect !
            !                                 Folder .. Work    !
            !                                                    !
            ! Specification of Connect document                  !
            !                                                    !
            ! Cabinet .... usr                                   !
            ! Password ...                                       !
            ! Document ...                                       !
            ! Replace .... N  (Y/N)                              !
            +----------------------------------------------------+
```

The following parameters are available

Parameter	Description
Folder	Text is always transferred to folder Work in Con-nect.
Cabinet	Cabinet of the user to which the text is to be exported. If you are exporting text to the cabinet of another user, this user must have granted you permission to his/her cabinet in Con-nect.
Password	Only required if you are exporting to the cabinet of another user.
Document	Enter name to be assigned to document in Con-nect. Documents exported from Predict automatically receive the description Document added via Predict.
Replace	Y: If the document name already exists in Con-nect it will be replaced. N: If the document name already exists in Con-nect, the text will not be transferred and an error message is given.

User-defined Target

With this version of Predict you can export text from the sources listed on to any user-defined subsystem.

This functionality is described on the documentation file delivered with Predict. See description of the following Program objects for more information.

- SAG-PRD-U-EXPSRC
- SAG-PRD-U-EXPSRP

Modify this program to suit your site-specific requirements.

Prerequisites

User exits U-EXPSC and U-EXPSRP which are delivered in source form with this version of Predict must still exist and must have been adapted to meet your site requirements.

Calling the Function

Enter one of the following in the Editor command line

- Command `EXPORT` to display the Export Text selection window, then target type "U"
- or EXPORT U

PC ASCII File

Prerequisites

- Your Predict session must be running under terminal emulation in a PC environment.
- Entire Connection for Windows must be installed (Version 4.3.1 or above).
- Workfile 5 must be defined as PC workfile on the mainframe.
- The Natural parameter PC must be set to ON, either by
  - specifying PC=ON when starting Natural or
  - entering %+ in the command line of your Predict session.

Note: In Windows and Linux environments, Workfile 5 must be set to type “Transfer”.

Steps

1. Enter one of the following in the Editor command line to call the Download Data dialog box: - Command `EXPORT` to display the Export Text selection window, then target type "P" - or EXPORT P
2. Specify target file in the Download Data dialog box. If the file exists in the target environment, a window will appear to warn you.
3. If the text to be exported contains accented characters such as “ä” or “à” click on the OEM transformation button. These characters will then be converted correctly in your PC environment.
4. Click on the OK button in this window to export the text from Predict to the PC file.

Maintaining Extended Descriptions with Word for Windows

This section covers the following topics:

- Prerequisites
- Installing the Predict - Word for Windows Interface
- Functional Scope
- Add / Modify Extended Description Skeleton

Extended descriptions of Predict objects can be edited with Word for Windows (Version 6, 7 or 97). Storing this Word for Windows document together with the dictionary object means that all Predict users have access to the fully formatted PC text.

If an extended description is edited with Word for Windows, the PC document containing full formatting instructions is stored in *binary* format together with the object on the Predict file. An *ASCII* version of the description containing only text is also stored together with the object.

Prerequisites

Predict

- Your Predict session must be running under terminal emulation in a PC environment.
- Predict - Word for Windows Link must have been established for the PC on which you are currently working. See ***Installing the Predict - Word for Windows Interface***

Natural

- Workfiles 6 and 7 must be defined as PC workfiles.
- Natural parameter PC must be set to ON, either by
  - specifying PC=ON when starting Natural or
  - entering %+ in the command line of your Predict session.

Entire Connection for Windows

- Entire Connection for Windows must be installed (Version 2.1.1 or above).
- Patch level 25 must be installed.

PC Environment

- Terminal emulation for Predict session.
- Word for Windows Version 6, 7 or 97 must be installed.
- The installation drive must not be write-protected (the drive specified under Predict profile parameter Predict Word for Windows directory).
- The following items are created automatically when the Predict - Word for Windows Interface is installed. These items may not be deleted or renamed: - installation directory, - document template PRD.DOT in the installation directory.

Installing the Predict - Word for Windows Interface

Step 1 - Define User Task in Entire Connection to Start Word for Windows

See your *Entire Connection documentation* for detailed information on how to define tasks.

Step 2 - Specify Predict Settings

Now perform the following in your Predict environment (which must be running under terminal emulation on your PC).

- If Entire Connection is not active, activate it by entering %+.
- Enter command PROFILE in a Predict command line; enter function code H to display the Handling screen.
- Enter W (Word for Windows) as first choice editor. This value will be rejected if you are not working in a PC environment.
- Set Download PC part to Y.
- Enter the Entire Connection task name defined in **Step 1**. This task will call Word for Windows. Note: that this name is case-sensitive with Entire Connection version 3.1 and above.

- Specify a target drive and directory for the Predict document and template file. This entry *must* end with a backslash!
- Press ENTER.

The document template PRD.DOT is downloaded to the specified drive and directory on your PC. The Predict-Word for Windows Interface is now installed.

Functional Scope

The following functions are available when the Predict Interface for Word for Windows has been installed and the prerequisites on are met:

- Add Extended Description, see below
- Modify Extended Description
- Add / Modify Extended Description Skeleton

Add Extended Description

The system behavior depends on how your Predict / PC environment is set up: Software AG or Natural Editor as First Choice

If Word for Windows is not defined as first-choice editor (see ***Step 2 Specify Predict Settings***), this function behaves as in Predict version 3.2.

Word for Windows as First Choice Editor

If Word for Windows is defined as first-choice editor, Predict only checks whether the connection from host to PC is active (see ***Natural Prerequisites***).

- If the connection is not active, no PC session is called and the description is edited with the Software AG or Natural editor.
- If Word for Windows cannot be called for any other reason, for example the user task defined for Entire Connection for Windows is not executable, or workfiles 6 and 7 are not defined correctly, then the function is terminated. No second or third choice editor is called.

If all the prerequisites are met, a Word for Windows session is started and document PRD.DOC is opened for editing.

Uploading the description to Predict

- From the Predict menu, choose “Upload to Predict”, or
- Close your document or exit Word. A window appears in which you are asked whether you wish to upload the document to Predict.

If you confirm, the following files are uploaded to Predict:

- PRDULD.DOC. This file contains full formatting information. It is stored as a Word for Windows document together with the corresponding object on the Predict file.
- PRDULD.TXT. This is a text version without formatting information.

These files are deleted on the PC after they have been successfully transferred to Predict.

Modify Extended Description

The system behavior depends on how your Predict/PC environment is set up: Software AG or Natural Editor as First Choice

If Word for Windows is not defined as first-choice editor (see *Step 2 Specify Predict Settings*), the system checks whether a Word for Windows document exists in Predict for the object to be processed. If a Word for Windows document exists, the system behavior depends on the setting of the following parameter in the Defaults > General Defaults > Protection screen. This setting is checked when the Editor is called.

PC text modif. on Mainframe	Y	A warning is given and the Word for Windows document is deleted in Predict. Description is edited with either the Software AG or Natural Editor
	N	The description cannot be edited.

Word for Windows as First Choice Editor

If Word for Windows is defined as first-choice editor, Predict only checks whether the connection from host to PC is active (see *Natural Prerequisites*):

- If the connection is not active, no PC session is called and the description is edited with the Software AG or Natural editor.
- If Word for Windows cannot be called for any other reason, for example the Entire Connection user task is not executable, or workfiles 6 and 7 are not defined correctly, then the function is terminated. No second or third choice editor is called.

If all prerequisites are met, the Word document file is downloaded from Predict; a Word for Windows session is started on your PC and the downloaded document is opened for editing.

Uploading the description back to Predict is described above.

Add / Modify Extended Description Skeleton

The system behavior depends on how your Predict/PC environment is set up and whether the skeleton exists additionally as a Word for Windows document. Software AG or Natural Editor as First Choice

If Word for Windows is not defined as first-choice editor (see **Step 2 Specify Predict Settings**), this function behaves as in Predict version 3.2. Word for Windows as First Choice Editor

If Word for Windows is defined as first-choice editor, Predict only checks whether the connection from host to PC is active (see ***Natural Prerequisites***):

- If the connection from Host to PC is not active, no PC session is called and the description skeleton is edited with the Software AG or Natural editor.
- If Word for Windows cannot be called for any other reason, for example the Entire Connection user task is not executable or workfile 7 is not defined correctly, then the function is terminated. No second or third choice editor is called.

If all prerequisites are met, the system checks whether the skeleton exists additionally as a Word for Windows document:

- If the extended description skeleton exists additionally as Word for Windows document, this document is downloaded to PC.
- If no Word for Windows document exists for the extended description skeleton, the mainframe version is downloaded to PC.
- A Word for Windows session is started and the downloaded skeleton is opened for editing.

Uploading the description skeleton to Predict is described under ***Uploading the description to Predict***.

IV Predict Retrieval

Concepts of Retrieval
Using Retrieval Options
Layout of Reports
Standard Retrieval Types
Type-specific Retrieval

39 Concepts of Retrieval

- Types of Information you can Retrieve 256
- Printing the Output of Retrieval Operations 257
- Executing Retrieval Operations in Batch Mode 258
- Using Retrieval Operations to Select Objects for Further Processing 258
- Indicating Dummy and Placeholder Objects 258

The complexity of information processing systems results both from the complexity of their individual components and from the complex relationships between these individual components.

The Predict dictionary system helps to manage this complexity by providing functions that retrieve information on the internal structure of information processing systems and properties of their components.

Because it is rarely possible to keep an overview of very complex systems without reducing the amount of information, Predict offers a variety of options to limit the scope of reports to exactly what is needed.

Predict offers a great variety of operations to retrieve information from the dictionary.

To execute a retrieval operation, define selection criteria and choose a retrieval type and a retrieval mode. In addition, you can specify output options to determine how much of the information retrieved from the dictionary is actually displayed.

All retrieval operations display reports on the screen unless you specify that the report is to be printed.

This chapter describes the concepts of retrieval and the use of all options provided to retrieve information.

Types of Information you can Retrieve

Using Predict retrieval you can evaluate the following basic types of information:

Information on Attributes of Individual Objects

Most retrieval types report on the attributes of individual objects. Information on the attributes of several individual objects can be retrieved with a single retrieval operation.

With the retrieval types Objects with children/parents and Objects with no child/parent associations of objects are used to determine which objects are to be included in a report.

Information on the Associations of Objects

Retrieving information on the associations of objects means retrieving information on the structure of the information processing system. Information of this type is primarily collected with the retrieval type Execute retrieval model in combination with the output modes Structured List and Cross reference.

The retrieval type Execute retrieval model evaluates the metadata model starting from an individual object.

Information on the Implementation of Objects

See the sections ***Active Retrieval*** and ***Displaying Information on the Implementation***.

Printing the Output of Retrieval Operations

With the Software AG Editor

If the default parameter General defaults > Miscellaneous > Software AG Editor for output is set to Y as standard, retrieval output is written to an Software AG Editor session.

Printing the results of a retrieval operation is a three-step process:

1. Specify a page size in the **output options**.
2. Execute the retrieval operation.
3. Enter PRINT in the command line.

The report is printed completely.

If you only want to print a range of lines, enter `PRINT x y`

where *x* is the number of the first line to be printed and *y* is the number of the last line to be printed.

Retrieval ouput is created as report 0 and sent to the hardcopy device (*HARDCOPY).

With the Natural Editor

If you set default parameter General defaults > Miscellaneous > Software AG Editor for output to N, retrieval output is displayed as in earlier versions of Predict.

Enter "%H" in the bottom line of an output screen or press the respective PF key.

Specify the printer in the additional screen that appears.

In this case it is not possible to specify a range of lines. The report is printed completely.

Executing Retrieval Operations in Batch Mode

Retrieval operations can be executed in batch. See the section *Predict in Batch Mode* in the *Predict Administration documentation*.

Using Retrieval Operations to Select Objects for Further Processing

The output mode Select can be used to select single objects for immediate processing or to select several objects for later processing using the workplan.

The workplan is described in the section Predict User Interface under Using the Workplan in the Introduction to Predict documentation.

Indicating Dummy and Placeholder Objects

Dummies

An association to an object that has not yet been created in the data dictionary is called a dummy. Dummies are included in the report and marked with

>>> object not documented <<<

if the parameter Dummy/Placeholder (for related objects) in the output options is set to Y or to D.

Placeholders

When an object that is linked to another object is imported with the Predict Coordinator together with its Internal ID, and the referenced object is not imported and does not already exist in the target environment, a Placeholder is added in the target environment for this referenced object. The job of this placeholder is to reserve the object ID of the referenced object in the target environment so that the link in the old environment can be recreated in the new environment at a later time.

Placeholders are included in the report and marked with

>>> object is placeholder <<<

if the parameter Dummy/Placeholder (for related objects) in the output options is set to Y or P.

40 Using Retrieval Options

- Overview of Retrieval Options 262
- Selection Criteria 262
- Restrictions 264
- Retrieval Type 267
- Output Mode 269
- Output Options 272
- Indicating which Objects are Implemented 280
- Displaying Information on the Implementation of Programs 280
- Using Restrictions and Output Options from Profiles 282

Overview of Retrieval Options

The following retrieval options determine precisely which objects are to be included in a report and which information on these objects and on the associations of objects is to be output.

Retrieval Options	
Selection Criteria	Retrieval operations can be executed for single objects or for groups of objects. Selection criteria are used to specify the objects to be included in a report.
Retrieval Type	Determines the type of information to be retrieved from the dictionary. The **standard retrieval types** mainly differ with respect to the amount of information that is displayed. Standard retrieval types, for example Objects with no parent, can be applied to all object types. **Type-specific retrieval types**, for example Difference of files, can be applied to specific object types.
Output Mode	In addition to the retrieval type, an output mode must be specified. The output mode determines how information retrieved from the dictionary is output and whether objects can be selected for further processing. Retrieval type and output mode together determine the type of Retrieval operation, which is indicated in the header of reports. See ***Layout of Reports***. Not all output modes are available for all retrieval types.
Output Options	In addition to the output mode, Output options can be set to determine which information, collected according to the retrieval type and selection criteria, is finally output. Output options can be specified for two types of object: ■ Current objects. Included in a report because they meet the specified selection criteria. ■ Related objects. Included in a report because they are associated to a current object.
Retrieval Models	Retrieval models can be used to specify exactly which part of the structure is to be evaluated.

Selection Criteria

Retrieval operations can be executed for single objects, for a subset of objects and for all objects in the dictionary.

Any retrieval operation applies to all objects which meet given selection criteria. Selecting objects has up to three steps:

1. The first step when selecting objects is to fill the field Object ID (for example File ID) or to leave this field intentionally blank.
   - To select a *single object*, an Object ID must be specified without asterisk notation. Additional selection criteria are then ignored.

- To select a *subset of all objects*, asterisk notation can be used (for example, by specifying ABRA*, objects with IDs as ABRAHAM, ABRAKADABRA, etc. are included in the selection).
- To select *all* objects, the field Object ID can be left blank (instead of entering an asterisk).

2. The selection can then be limited further by specifying object-specific additional selection criteria. See **Type-Dependent Selection Criteria** below. A general option for additional selection criteria is the parameter `default passive association`, for example `Belongs to VM` or `Contained in DA`. Only objects that are associated as a child to the given object are included in a report (asterisk notation is possible).
3. In the last step the selection can again be limited by specifying **Restrictions**.

Type-Dependent Selection Criteria

Depending on the type of object additional selection criteria can be specified.

Parameter	Applicable to
object of type	Applicable to different object types. Only objects with the given subtype are included in the selection. For example: a file of type A (Adabas file).
default passive association	Applicable to different object types. Only objects linked as child objects to the given parent object are included in the report.
Database number	Database
External name	File
File number	File
Synonyms	Field
Language	Program
Implementation pointer ■ Library ■ Member ■ User system Fnr ■ User system DBnr	Program
Implementation pointer ■ Library ■ User system Fnr ■ User system DBnr	System
Using file	File relation
User name	User

Parameter	Applicable to
Verif. of status	Verification
Format	Verification

The meaning of the above selection parameters is described in detail in the retrieval sections of the documentation Predefined Object Types in Predict.

Restrictions

Restrictions can either be specified temporarily by entering code T in the field Restrictions or they can be taken from a profile. See **Using Restrictions and Output Options from Profiles** . Temporary restrictions are lost when a session is terminated.

Restrictions are evaluated only for current objects.

Objects	Restrictions
Keywords	Up to five keywords can be specified. See *Relating objects logically* in the *Introduction to Predict documentation* and section *Keyword* in the *Predefined Object Types in Predict documentation*.
Owner	You can restrict the retrieval operation to objects that are assigned to a particular owner. See *Relating objects logically* in the *Introduction to Predict documentation* and section *User/Owner* in the *Predefined Object Types in Predict documentation*.
Extract	You can restrict the retrieval operation to objects that are contained in a specified extract. See the section *Extract* in the *Predefined Object Types in Predict documentation*.
String	You can restrict the retrieval operation to objects whose abstract, extended description, rules or ID contains the specified string.
Date	Retrieval operations can also be restricted by the parameter AND from date: only objects that were added or modified after a given date are evaluated. See the parameters described below.

```
13:50:03             *****  P R E D I C T  *****              2007-05-31
Plan   0                 - (FI) Fil +Top-----------Restrictions-------------+
                                    ! With keyword(s)                        !
    Retrieval Type                  !                                        !
                                    !                                        !
    D  Files                        !                                        !
    E  Execute retrieval models     !                                        !
    C  Dummy/Placeholder files      !                                        !
    A  Difference of files          ! combined by (AND/OR) OR                !
    N  File number retrieval        ! AND with owner                         !
                                    !                                        !
                                    ! BUT NOT with keyword                    !
Retrieval type ...                  !                                        !
Output mode .....* D Display        ! AND included in extract                !
File ID .......... HNO-FI           !                                        !
Contained in database ......        ! AND containing the string              !
External name ....                  !                                        !
Restrictions ....* T Temporary profi ! Scan options:                         !
Output options ..*   Profile HNO    !   Abstract N (Y,N) Rules ..... N (Y,N) !
                                    !   Descr.   N (Y,N) Object-ID . N (Y,N) !
Command ===>                        !   Absolute Y (Y,N) Ignore case Y (Y,N) !
                                    !Command ==> +___                        !
                                   +More-----------------------------------+
```

```
13:50:03             *****  P R E D I C T  *****              2007-05-31
Plan   0                 - (FI) Fil +More----------Restrictions-------------+
                                    ! AND from date (YYYY-MM-DD HH:II)       !
    Retrieval Type                  ! 0000-00-00 00:00                       !
                                    !                                        !
    D  Files                        !                                        !
    E  Execute retrieval models     !                                        !
```

Parameters	
With keyword(s)	Only objects with the given keywords are included in the selection. Up to five keywords can be specified. Asterisk notation is possible.
combined by (AND/OR)	The keywords are combined in a boolean expression with AND or OR.
AND with owner	Only objects with the given owner are included in the selection. Asterisk notation is possible.
BUT NOT with keyword	Objects with the given keyword are excluded. Asterisk notation is possible.
AND included in extract	Only objects contained in the specified extract are included in the selection. With asterisk notation you can specify a range of extracts.

Parameters			
AND containing the string		Only objects containing the given string in one or more of the following are included in the selection. - Abstract - Rules (only for verifications) - Extended description - Object ID **Note:** Retrieval operations may be significantly slower when using a text string as selection criterion. For this reason it is not possible to define a default value for this parameter.	
Scan options			
	Abstract	Y	Abstract is to be scanned for the search string.
	Rules	Y	Rules are to be scanned for the search string. This parameters applies exclusively to verifications. With objects of other types, this input field is write protected.
	Descr.	Y	Extended descriptions are to be scanned for the search string.
	Object ID	Y	Objects will only be selected if the string occurs in the object ID.
	Absolute	Y	Embedded text strings are found (for example ring in strings).
		N	Only occurrences of the search string that are delimited by a blank or special character are found.
	Ignore case	Y	The scan function will ignore upper and lower case (with ring specified, RiNg is found).
AND from date (YYYY-MM-DD HH:II)		Only objects that were added or modified on or after this date will be selected. If a MODIFY date exists, this will be evaluated. If no MODIFY date exists for an object, the ADD date is evaluated.	

Saving Restrictions for Later Use

Restrictions that have been specified for temporary use can be stored in your own profile (no matter which profile is currently active) with the command SAVE, STOW or CAT in the Command==> field of the Additional Restrictions window. Text strings specified with the AND containing the string parameter will not be stored.

A SAVE, STOW or CAT command issued by a user not defined in Predict with an object of type User is rejected.

How the Use of Restrictions is Indicated

Once activated, restrictions apply to all subsequent retrieval operations until they are *explicitly deactivated*. The word empty behind the field Restrictions indicates that no restrictions are active.

Deactivating Restrictions

Restrictions can be deactivated with code E (Empty restrictions) in the field Restrictions.

Whenever the result of a retrieval operation is smaller than expected, it is possible that restrictions of a previous retrieval operation are active.

Using Restrictions from Profiles

See ***Using Restrictions and Output Options from Profiles***.

Retrieval Type

Retrieval type and output mode together determine the type of retrieval operation that is indicated in the header of reports. See ***Layout of Reports***.

The retrieval type determines the type of information to be retrieved:

- Some retrieval types can be applied to most object types (including user-defined object types). These are called Standard retrieval types. Standard retrieval types mainly differ with respect to the amount of information that is displayed on the associations of objects. See ***Standard Retrieval Types*** for a detailed description of all type-independent retrieval types.
- Other retrieval types are type-specific, such as the retrieval type Difference of files contained in the File Retrieval menu below. Type-specific retrieval types are described in the respective sections of documentation Predefined Object Types in Predict.

```
13:13:33             *****  P R E D I C T  *****             2007-05-31
Plan   3                  - (FI) File Retrieval -                  Profile HNO

 Retrieval Type                            Retrieval Type

 D  Files                                  B  Files with parents
 E  Execute retrieval models               O  Files with no parent
 C  Dummy/Placeholder files                T  Files with children
 A  Difference of files                    U  Files with no child
                                           R  Files related to a file

Retrieval type ...
Output mode .....*

File ID ..........                                 Files of type ....*
Contained in DA ..                                 File number .......
External name ....
Restrictions ....*   Profile HNO,used             Model ............*
Output options ..*   Profile HNO                   Association ......*

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
```

Restricting the Information on Related Objects

With retrieval types that report on the associations of objects, the parameter Association can be used to determine which types of related objects are to be included in the report. For example: If Programs with children is executed with FI (`Uses FI concept.`) specified as Association, only programs which have children of type file via that association are listed.

The parameter Association applies to the following retrieval types:

T	Objects with children
B	Objects with parents
U	Objects with no child
O	Objects with no parent
C	Dummy/Placeholder objects

Note: For functions Objects with no child and Objects with no parent, the association can be an association code (FI, DA etc.), blank (all), or - for these two functions only - the value AY (any). See table ***Association for Objects with no child and Objects with no parent*** for the effect of these related types on the retrieval operation.

Output Mode

An output mode must be specified in addition to the retrieval type. The following output modes are given:

- Output Modes Display, List, Select and Count
- Output Mode Structured List - Code T
- Output Mode Cross Reference - Code X

Output Modes Display, List, Select and Count

The output modes Display, List, Select and Count are used to retrieve information on individual objects.

Display - Code D	Displays extensive information on individual objects. The **output options** are used to determine what exactly is displayed.
List - Code L	Displays the following information in list form: ■ one line of information per object ■ Abstracts The output option No. abstract lines determines how many abstract lines are displayed per object. ■ Asterisk marking implemented objects (optional) The output option Mark implementation determines whether implemented objects are marked.
Select - Code S	The output mode Select can be used to select single objects for immediate processing or to select several objects for later processing using the workplan. See also *Using the Workplan* in the *Introduction to Predict documentation*. The following information is displayed in list form: ■ one line of information per object ■ Asterisk marking implemented objects (optional) The output option Mark implementation determines whether implemented objects are marked.
Count - Code C	Used for Active Retrieval operations to count how many objects meet given selection criteria.

Output Mode Structured List - Code T

The output mode Structured list provides information on the structure of an application. The following rules apply:

- All objects directly or indirectly connected to an object along a predetermined search path are displayed in the form of a structured list.
- The connection can be either of the following:
  - **Standard link**
    A Standard link (predefined or user-defined association)
  - **Special link**
    A Special link, for example Master Files and Userviews.
- The hierarchy of associations is displayed.
- Objects can be displayed more than once.
- The attributes of an association can be displayed.

See description of parameter Link in the section *Retrieval Model Administration* in the *Predict Administration documentation* .

See also ***Execute Retrieval Model***.

Sample Output with Structured list

The example below was created with the predefined retrieval model EX for a system.

```
13:56:24             *****  P R E D I C T  *****              2007-05-31
                       - Structured List for System -

System ID .......... DEMO-FLIGHT-BOOKING-SYSTEM
-------------------------------------------------------------------------------
01  SY Uses PR concept. .......... * DEMO-FB-MENU
02  :   PR uses PR concept. .......... * DEMO-MAP
02  :   PR uses PR concept. .......... DEMO-FLIGHT-BOOKING
03  :   :   PR uses PR concept. .......... DEMO-FB-DATA
03  :   :   PR uses PR concept. .......... DEMO-FB-CHECK
04  :   :   :   PR uses FI concept. ............. * DEMO-FB-FLIGHT
05  :   :   :   :   FI Has direct views .... * DEMO-FLIGHT
05  :   :   :   :   FI Contained in DA ......... DEMO-FLIGHT-BOOKING
03  :   :   PR uses PR concept. .......... DEMO-FB-STORE-DATA
04  :   :   :   PR uses PR concept. .......... DEMO-FB-UPD-FLIGHT
    :   :   :   >>> object not documented <<<
03  :   :   PR uses PR concept. .......... DEMO-FB-PRINT-TICKET
04  :   :   :   PR uses FI concept. ............. * DEMO-FB-FLIGHT-BOOKING
05  :   :   :   :   FI Contained in DA ......... DEMO-FLIGHT-BOOKING
04  :   :   :   PR uses FI concept. ............. * DEMO-FB-FLIGHT
Command ===>                                                Scroll ==> CSR
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Quit        RFind Flip  -     +           Left  Right
```

Output Mode Cross Reference - Code X

The following rules apply:

- All objects that are directly or indirectly connected to an object are listed.
- The connection can be either of the following:
  - **Standard link**
    A Standard link (predefined or user-defined association)
  - **Special link**
    A Special link, for example master files and userviews.
- The hierarchy of associations is *not* displayed.
- Each object is shown only once.

See description of parameter Link in the section *Retrieval Model Administration* in the *Predict Administration documentation*.

See also ***Execute Retrieval Model***.

Sample output with Cross Reference

The example below was created with the predefined retrieval model EX for a system.

```
13:55:15             *****  P R E D I C T  *****              2007-05-31
                        - Cross Reference for System -

 System ID .......... DEMO-FLIGHT-BOOKING-SYSTEM
 ------------------------------------------------------------------------------
 Database ID ........ DEMO-FLIGHT-BOOKING
 File ID ............ * DEMO-FB-FLIGHT
                      * DEMO-FB-FLIGHT-BOOKING
                      * DEMO-FLIGHT
 Program ID ......... DEMO-FB-CANCEL
                      DEMO-FB-CHANGE
                      DEMO-FB-CHECK
                      DEMO-FB-DATA
                      * DEMO-FB-MENU
                      DEMO-FB-PRINT-MASK
                      DEMO-FB-PRINT-TICKET
                      DEMO-FB-STORE-DATA
                      DEMO-FB-UPD-FLIGHT
                      >>> Object not documented <<<
                      DEMO-FI-MENU
 Command ===>                                                  Scroll ==> CSR
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Quit        RFind Flip  -     +           Left  Right
```

Output Options

The amount of information contained in a report can be determined with Output options and parameters of a Retrieval model.

- The contents of a report created with the output mode Display are determined with output options. A small subset of output options also applies to the output modes List, Select and Cross Reference.
- The contents of a report created with retrieval type Execute retrieval model (output mode Structured List or Cross Reference) are basically determined with parameters of the retrieval model. On the basis of these settings, the contents of such a report can then be restricted with the output options described below.

Output options can be read from profiles (see *Using Restrictions and Output Options from Profiles*) or specified in a window (see screen below) for temporary use.

```
13:13:29             *****  P R E D I C T  *****             2007-05-31
Plan   3                 - (SY) System Retrieval -                 Profile HNO
                                          +All-------Output options-----------+
 Retrieval Type                           !                  current   related!
                                          ! Description ....... Y  (Y,N)  Y   !
 D  Systems                               ! No. abstract lines  0  (0-16) 0   !
 E  Execute retrieval models              ! Keywords .......... 20 (0-32) 20  !
 C  Dummy/Placeholder systems             ! Extracts .......... 32 (0-32) 32  !
                                          ! Owners ............ 35 (0-99) 39  !
                                          !   with users .....* Y         Y   !
Retrieval type ...                        ! Attributes ........ Y  (Y,N)  N   !
Output mode .....*                        ! Dummy/Placeholder * D         Y   !
                                          !                                   !
System ID ........                        ! Association attr. .....* Y        !
Library ..........                        ! Mark implementation .... Y (Y,N)  !
                                          ! Show implementation .... Y (Y,N)  !
                                          ! Display modifier ....... Y (Y,N)  !
Restrictions ....*   Profile HNO,used     ! Use Conform ............ N (Y,N)  !
Output options ..* T Profile HNO          ! Connecting character ... ?        !
                                          ! Cover page ............. N (Y,N)  !
Command ===>                              ! Page size ............. 25       !
                                          !Command ==> ____                   !
                                          +-----------------------------------+
```

Not all output options apply to all retrieval types. Output option Attributes, for example, does not apply to the retrieval type Dummy/Placeholder objects.

Output options are object type dependent. The cover page of a retrieval report indicates which output options are active for that retrieval operation (see parameter Cover page). You will also find an overview of the valid output options for the individual retrieval operations at the end of each section in the documentation Predefined Object Types in Predict.

Saving Output Options for Later Use

```
13:13:45             *****  P R E D I C T  *****             2007-05-31
Plan   3                 - (SY) System Retrieval -                 Profile HNO
                                                +All----Output options--------+
 Retrieval Type                          Retri ! _ U   Your profile          !
                                                ! _ A   Active profile        !
 D  Systems                              B  Sy ! _ D   Profile Default       !
 E  Execute retrieval models             O  Sy ! _ T   Temporary profile     !
 C  Dummy/Placeholder systems            T  Sy ! _ ' ' Confirm current       !
                                         U  Sy !       profile and execute   !
                                                !       function              !
Retrieval type ...                              !                             !
Output mode .....*                              !                             !
                                                !                             !
System ID ........                              !                             !
Library ..........                              !                             !
                                                !                             !
                                                !                             !
Restrictions ....*   Profile HNO,used           !                             !
Output options ..* * Profile HNO                !                             !
                                                !                             !
Command ===>                                    !                             !
                                                !Command ==> _______          !
                                                +-----------------------------+
```

Output options that have been specified for temporary use can be stored in profiles with the command SAVE, STOW or CAT in the Command==> field of the Output options window. Output options are stored in your own profile, irrespective of which profile is currently active.

A SAVE, STOW or CAT command issued by a user not defined in Predict with an object of type User is rejected.

General Rules Applying to Output Options

- Many output options can be applied both to objects that meet the specified selection criteria (the current objects) and to objects that are included in a report as children or parents of these objects (the related objects).
- Output options for related objects are applicable to retrieval types that report on attributes of related objects:

B	Objects with parents
T	Objects with children
C	Dummy/Placeholder objects (output mode Display)
E	Execute retrieval model
X	Cross reference objects (applicable to extracts, keywords and owners)

- Changing an output option has no effect if a retrieval operation does not retrieve the information to which the option refers.

For example, if you use Execute retrieval model with a retrieval model that does not report on keywords, changing the parameter Keywords in the output option has no effect.

Output Options Valid For All Objects

Output Options valid for all Objects		
Description	Y	Display extended description of an object.
No. abstract lines	Number of abstract lines to be displayed.	
Keywords	Number of keywords to be displayed.	
Extracts	Number of extracts to be displayed.	
Owners	Number of owners to be displayed.	
with users	Determines, if users assigned to the owners are displayed.	
	Y	All users assigned to an owner are displayed.
	S	If only one user is assigned to the owner, the user is displayed. If more than one user is assigned, >>> multiple <<< is displayed.
	N	Users assigned to an owner will not be displayed.
Attributes	Y	Display attributes of objects. **Note:** The display of Adabas attributes of Files and Adabas sizes of Databases is controlled using type-specific output options. For further information see **Type-specific Output Options** .
Dummy/Placeholder, related objects	Applicable to the following functions ■ Objects with children (code T) with output mode List ■ Execute retrieval model (code E) with output mode Structured List	
	N	Only documented objects are displayed.
	Y	Dummies/placeholders and documented objects are displayed.
	D	Only dummies are displayed.

Output Options valid for all Objects		
	P	Only placeholders are displayed.
Dummy/Placeholder, current objects	Applicable to function Dummy/Placeholder objects (code C):	
	D	Only dummies are displayed.
	P	Only placeholders are displayed.
Association attr.	Y	All attributes of associations between objects are displayed.
	S	Short. Only attributes that appear in the respective link list are displayed.
	N	No attributes of associations are displayed.
Mark implementation	Y	Implemented objects are to be marked with an asterisk (*). Applicable to files, databases, systems, storagespaces, programs, verifications, dataspaces. See **Indicating which Objects are Implemented** for a description of when an object is regarded as implemented.
Show implementation	Y	Display the implementation. Applicable to files, databases, storagespaces, systems, programs, dataspaces.
Display modifier	If not deactivated in the general defaults, Predict remembers who created or changed an object.	
	Y	The last user who changed an object is displayed when retrieving information on objects or maintaining objects. See the description of the parameter Modification Log in the section *Maintenance Options* of section *Defaults* in the *Predict Administration documentation*.
Connecting character	Character that is used to connect output lines on the same level when creating a report with retrieval type Execute retrieval model and output mode Structured list. The Connecting character is also used in the display of Owners/Users and condition names/values.	
Cover page	Y	Introduce reports with a cover page. The cover page contains a list of all output options and selection criteria that were applied when creating the report. If a retrieval model was used the model is displayed.
Page size	This parameter defines the maximum number of lines to be displayed on each logical page. If set to zero, the system default value for the user's current online terminal is substituted.	

Output Options valid for all Objects			
	Note: Not applicable in batch mode. **Note:** Page size is only applicable when printing or if general default parameter "Use SAG Editor for output" is set to Y.		
User Exit	N		Deactivates the user exit U-DSP that can be called after retrieval operations. The parameter is only displayed if Retrieval / after display in the Activate user exits screen of the function Defaults is set to O. See the section *Defaults* in the *Predict Administration documentation*.
Use Conform	Y		Con-form formatting instructions are interpreted in extended descriptions. This parameter can only be set to Y if the Con-form text formatting facility is installed. **Note:** In an IBM environment, the size of the Con-form buffer area is determined by the Natural parameter CSIZE. For details of this parameter, see the *Natural Parameter Reference* documentation. **Note:** The following parameters only apply if Con-form is installed and parameter Use Con-form is set to Y. They apply whenever the extended description of an object is output by either the DISPLAY command of the description editor or the Display function of a retrieval menu.
	Predict header	Y	The usual Predict header, including the current date,time, and page number, appears at the top of every page.
		N	The Predict header is omitted from the top of every page. However, a Con-form page title command with a number sign (#) can still be used to number the pages. See also Extended Description Skeleton in the section *Defaults* in the *Predict Administration documentation*.
	Use FPROFILE	Y	The Con-form formatting profile (FPROFILE) is inserted whenever an extended description is output. Default values for a page header, page size, line size and Con-form macros can be stored in FPROFILE.
	List error messages	Y	Con-form error messages are displayed with the description.
Web Service usage	Y		Objects that are used by or in web service components of an application are displayed with

Output Options valid for all Objects		
		an additional comment. The comment contains the name of the web service and where it is located.

Type-specific Output Options

The following type-dependent output options can be specified in subsequent screens after Y has been entered in the field MORE Type dependent options and an object type has been entered in the selection window that is then displayed.

Additional Output Options for File		
Adabas attributes	Y	Adabas attributes are shown in detail.
	S	A summary of Adabas attributes is given.
	N	No Adababas attributes are shown.
Vista elements	Y	Entries for files in Vista tables are displayed.
Subquery	Y	Display subqueries.
Trigger	Y	Display triggers linked to files via *Triggers FI*.
Check expression	Y	Display check expressions.
Linked verification	Y	Verifications linked to fields via *Verifies EL* are displayed. Triggers linked to fields via *Triggers EL* are displayed. Applicable to the retrieval type Files with children, association EL or blank (' ') with output mode Display.
Generation layout	Y	The generation layout of the file is displayed. Applicable when executing the retrieval type Files with children (code T) for association has fields. The output options Adabas version, Language, Alignment/sync.:, Position/Offset, Counter length, Compiler and Replace with synonym specify in detail how a record layout is displayed.
Adabas version	When listing a record buffer structure of an Adabas file or userview, the Adabas version for which the buffers are applicable.	
Language	CO	COBOL
	BA	Assembler (BAL)
	PL	PL/I
	FO	FORTRAN
	CC	Language C
Alignment/Sync	N	Fields will not be aligned
	Y	All appropriate fields are aligned.
	S	Only fields with the attribute “Synchronized” in the corresponding Predict entries are aligned.

Additional Output Options for File		
Position/Offset	P	Display absolute position of field in record.
	O	Display offset of the field.
Counter length	Applicable to Assembler copy code: the length of additional counter fields. Possible values: 1,2.	
Compiler	I	IBM
	S	SIEMENS
	7	FORTRAN 77 Standard
	For language COBOL:	
	7	COBOL 74
	8	COBOL 85
	W	WANG COBOL
Replace with syn.	Y	If a synonym name exists for the generated language, the field name will be replaced by the synonym name.
	N	The field name will not be replaced by a synonym name.

Additional Output Options for Field		
Composed fields	Y	Display names of fields hyper/super/subfields are derived from when fields of these types are displayed.
3GL specification	Y	The following 3GL-specific attributes of fields are displayed: Gr.structur, Justify, Synchronized, Init. value, Indexed by, Depending on, Condition name and Condition value.
Descriptors only	Y	Only fields with descriptor option not set to blank, N or E are shown.
DV-field expression	Y	Derived field expressions are displayed.
Natural options	Y	Up to three headers and the definition of the Natural edit mask are displayed.
Sorted by field	Y	Field list is sorted alphabetically by field.
	N	Fields are sorted by file. Fields appear in the order they are defined in the file. For a detailed description of this parameter see *Sorting Fields and Files* in the section *Field* in the *Predefined Object Types in Predict documentation*. **Note:** This output option is also available for verifications.
Synonyms	Display language-specific synonyms of field IDs.	
	#	Display synonyms for all languages.
	*	A selection window for language code appears.

Additional Output Options for Field		
Display length	Format in which the length of fields is displayed:	
	N	Natural Format
	P	Physical Format

Additional Output Option for Database		
Adabas sizes	Y	Display physical size assigned to a database.

Additional Output Options for Programs		
Entry points	Y	Entry points are displayed.
Procedure code	Y	Display procedure code.

Additional Output Option for Verification		
Rules	Y	Display rules of a verification.
Sorted by field	Y	Field list is sorted alphabetically by field.
	N	Fields are sorted by file. Fields appear in the order they are defined in the file.
	Applicable to the retrieval type Verifications with parents, association EL or blank (' '). **Note:** The default value for this option is specified as type-specific option for field.	

Indicating which Objects are Implemented

If the output option Mark implementation is set to Y, objects that are implemented are marked with an asterisk.

What is meant by implemented depends on the object type. For details refer to When is an Object regarded as implemented? in the section Retrieving Information from Predict in the Introduction to Predict documentation.

Displaying Information on the Implementation of Programs

If the Show implementation parameter is set to Y, information on the implementation of an object contained in the XRef data is displayed: If several implemented objects exist in different libraries, these are listed.

Documentation data and corresponding XRef data are identified with the implementation pointer.

The following screen contains implementation information on a program.

```
13:47:03             *****  P R E D I C T  *****              2007-05-31
                            - Display Program -

Program ID ......... HNO-PR
Type ............... Program                   Added 2007-05-31 at 08:30 by HNO
                                            Modified 2007-05-31 at 09:32 by HNO
--------------------------------------------------------------------------------

Program attributes
  Language .. Natural
  Mode ...... (none)

Implementation pointer
  Member .... ARH1      Library .. ARH2      Fnr ..      DBnr ..
  NAT-Func ..
  Load Lib ..

                --------------- Implementation ---------------
Member    Library  Fnr   DBnr  Type  Cat-user   Cat-tid   Cat-date     Cat-time
ARH1      ARH2      54   180   P     ARH        DAEDC438  2007-05-31   09:08:32
Command ===>                                                   Scroll ==> CSR
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Quit        RFind Flip  -     +           Left  Right
                                             line 1 of 18 (line 8 on screen)
```

Using Restrictions and Output Options from Profiles

```
13:49:58             *****  P R E D I C T  *****              2007-05-31
Plan   3                  - (FI) File Retrieval -                  Profile HNO
                                                +All-----Restrictions---------+
 Retrieval Type                           Retri ! _ U   Your profile          !
                                                ! _ A   Active profile        !
 D  Files                                 B  Fi ! _ D   Profile Default       !
 E  Execute retrieval models              O  Fi ! _ T   Temporary profile     !
 C  Dummy/Placeholder files               T  Fi ! _ E   Empty restrictions    !
 A  Difference of files                   U  Fi ! _ ' ' Confirm current       !
                                          R  Fi !       profile and execute   !
                                                !       function              !
Retrieval type ...                              !                             !
Output mode .....* D Display                    !                             !
                                                !                             !
File ID ..........                              !                             !
Contained in DA ..                              !                             !
External name ....                              !                             !
Restrictions ....* * Profile HNO,used           !                             !
Output options ..*   Profile HNO                !                             !
                                                !                             !
Command ===>                                    !                             !
                                                !Command ==> _______          !
                                                +-----------------------------+
```

Restrictions and output options can be stored in profiles. To use restrictions/output options of a profile, the profile does not necessarily have to be active as the current session profile. The following options are possible:

Your profile	Use restrictions/output options in the profile of the current user.
Active profile	Use the restrictions/output options of the profile that is currently active. The active profile is indicated in the upper right corner the screen (in the field Profile).
Profile default	Use the restrictions/output options of the Predict default profile with code D (Profile default).
Temporary profile	Use temporary restrictions/output options. The Restrictions window is then displayed containing the values of the current profile. The values can be overwritten for use in the current session.
Empty restrictions	Deactivate restrictions.

To use the restriction profile of another user, the profile of that user has to be activated with the command PROFILE user-ID.

41 Layout of Reports

- Header Information 284
- Cover Page 285
- Body of Report 286
- Footer 287

The layout of reports varies with the information that has to be displayed.

Header Information

Header information is separated from the rest of the retrieval report by a horizontal line.

```
13:57:22            *****  P R E D I C T  *****             2007-05-31
                             - List Program -

-------------------------------------------------------------------------------
```

Time, Date

The time on the left indicates the time at which the objects currently displayed were included in the report. This time changes accordingly when you page backwards and forwards.

Function

The current function is displayed.

Page number only for Printed Output

Retrieval output is now line-oriented and not page-oriented as in earlier versions of Predict. When a retrieval report is printed, however, the page number appears in the top right hand corner. The page length is taken from the parameter Page size, which is only relevant for printed output.

Object ID is permanently displayed

For display-oriented functions and for functions with parents or with children, each new main object starts on a new screen page. Object ID and subtype (if applicable) are displayed on the left. Date and time the object was added or last modified appear on the right.

```
13:26:29            *****  P R E D I C T  *****               2007-05-31
                             - Display Program -

Program ID ......... HEB-E
Type ............... External program                Added 2007-05-31 at 08:59
                                                  Modified 2007-05-31 at 13:19
-------------------------------------------------------------------------------
```

Cover Page

If the parameter Cover page of the **output options** is set to Y, a cover page will introduce reports.

```
13:54:43             *****  P R E D I C T  *****              2007-05-31
                            - Display Program -

-------------------------------------------------------------------------------

                   --------------- Cover page ---------------
Program ID ......... HNO-PR

Output options
  Type independent      curr.          Type independent      curr.
  Description ........    Yes          Mark implementation     Yes
  No. abstract lines..      0          Show implementation     Yes
  Keywords ...........     20          Display modifier ...    Yes
  Extracts ...........     32          Use Conform ........     No
  Owners .............     35          Page size ..........     25
    with users .......    Yes
  Attributes .........    Yes

  Type dependent        curr.          Type dependent        curr.
  Entry points .......    Yes          Procedure code .....    Yes
```

The cover page contains a listing of the selection criteria, the output options and the retrieval model (if any) used for the creation of the report.

A cover page is especially useful when keeping a printed report for later use.

```
   13:57:22             *****  P R E D I C T  *****              2007-05-31
                                - List Program -

   ------------------------------------------------------------------------------

Cnt

  Program ID                          Type Lang Member   Library  Fnr DBnr

      1 ARH-DIALOG                         T    N
      2 ARH-DYN                            I    N   MENU
      3 ARH-DYN2                           I    N   XXX
      4 ARH-DYN3                           I    N   ARH1
      5 ARH-DYN3-1                         I    N
      6 ARH-MENU-ARH1-64-180               P    N   MENU     ARH1      64  180
      7 ARH-MENU-54-180                    P    N   MENU     ARH       54  180
      8 ARH-MENU-64                        P    N   MENU     ARH1      64  180
      9 ARH-MO-99                                                      64  180
     10 ARH-PR                             P        LSTXXR   NEWDIC    64  180
     11 ARH-PR-ARH1-54                     P    N   XXX      ARH1      54  180
     12 * ARH-PR-ARH1-54-180               P    N   ARH1     ARH2      54  180
                                   Implementation: ARH1     ARH2      54  180
     13 ARH-PR-ARH1-64                     P    N   XXX      ARH1      64  180
    Command ===>                                                  Scroll ==> CSR
   Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                     Quit        RFind Flip  -     +           Left  Right
                                                 line 13 of ... (line 5 on screen)
```

Body of Report

The structure of the body of the report depends on the following factors:

- Output mode See the description of the various output modes in this section for output examples.
- **Output options**
- Object type Sample output for each object type with output mode List is given in the respective section of the Predefined Object Types in Predict documentation.

Cnt

For some retrieval functions, the column Cnt contains the sequential number of the main object in the retrieval list.

Note: This column displays a maximum of four digits, so if the list contains more than 9999 objects, the numbering restarts at 0.

Footer

Standard PF Key Assignment

PF3 Terminate the function

PF5 Repeat last search command

PF6 Flip PF key display (Switch from keys 1-12 to keys 13-24)

PF7 Scroll backwards by value in field Scroll ==>

PF8 Scroll forwards by value in field Scroll ==>

PF10 Scroll to the left

PF11 Scroll to the right

PF17 Find string under cursor

PF18 Flip PF key display (Switch from keys 13-24 to keys 1-12)

PF19 Scroll to top of report

PF20 Scroll to bottom of report

Note: These PF key assignments cannot be modified.

Line nn of ... (line n on screen)

When the report exceeds a screen page, Line nn of ... appears in the Natural message line. These dots are replaced with the total number of lines in the report when the end of the report is reached.

Line n on screen refers to the first line of "real" data in the current screen. In the example below, the first six lines contain header information.

```
 Command ===>                                                    Scroll ==> CSR
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                  Quit        RFind Flip  -     +           Left  Right
                                                line 12 of ... (line 7 on screen)
```

Message End of Report

The end of reports is marked with *** End of report ***.

42 Standard Retrieval Types

- Objects - Code D 290
- Objects with Parents - Code B, Objects with Children - Code T 290
- Objects with No Parent - Code O , Objects with No Child - Code U 291
- Dummy/Placeholder Objects - Code C 292
- Execute Retrieval Model - Code E 293

Standard retrieval types can be applied to all types of Predict objects.

The effect of different output modes on standard retrieval types is described in the section ***Output Mode***.

Objects - Code D

This type retrieves information on all individual objects that meet the specified selection criteria.

Note: The Display objects and Select objects operation can also be invoked from all maintenance menus. If called from a maintenance menu, however, output options have no effect and the maximum amount of information is always displayed.

Commands:	DISPLAY, LIST, SELECT (depending on the output mode)
Valid output modes:	Display, List, Select.
Limit scope of report:	Output options

Objects with Parents - Code B, Objects with Children - Code T

These types retrieve information on objects and on the parent or child objects of the objects. Only those objects linked to another object by a passive/active association are included in the report.

The retrieval types will also report on the attributes of objects included in the report because they are related to the objects that meet the specified selection criteria. The amount of information displayed on related objects can be restricted with output options (in the column related).

When retrieving information on an Adabas database and the file objects contained in it, Predict checks if the physical attributes of all files in a database correspond to the respective physical definitions in the database.

Commands:	Commands: PARENTS, CHILDREN
Valid output modes:	Display, List.
Limit scope of report:	The parameter Association and output options can be used to restrict the scope of a report on both the current objects and the related objects. All is valid for Association.

Objects with No Parent - Code O , Objects with No Child - Code U

These retrieval types report on objects that have no parent or child for a given association or for any association.

Commands:	FREE, EMPTY
Valid output modes:	Display, List, Select.
Limit scope of report:	The parameter Association and output options can be used to restrict the scope of the report. Any and All are valid for Association.

Indicating which Associations are Valid

If for a valid passive/active association no objects are linked, the respective association is displayed in a column no parent or no child. The parameter Association can be used to specify which objects are included in a report (see below).

With the output mode Select, either one association type or **of some types** is displayed (because Select displays only one line of information per object).

Association for Objects with no child and Objects with no parent

An Association must be entered for these functions. Valid values:

Association	Result
association code (FI, DA ...)	All objects without a passive/active association of the specified type are included in the report.
blank (all)	Objects that do not have a passive/active association to at least one object of each and every other type are included in the report. For example: The object type database can be linked via an active association to objects of type file, dataspace, user. If it is linked to a file object and a user object but not to a dataspace object, it is included in the report.

Association	Result
AY (any)	All objects without a passive/active association to any other object are included in the report. In other words: Objects having at least one passive/active association to another object (regardless of type) are not included in the report. For example: The object type database can be linked via an active association to objects of type file, dataspace, user. It is linked to a file object and therefore not included in the report. **Note:** AY (any) will only be contained in the selection window of Association when code U (Objects with no child) or O (Object with no parent) have been entered as the retrieval type.

Dummy/Placeholder Objects - Code C

The retrieval type Dummy/Placeholder objects together with the output option Dummy/Placeholder for current objects is used to find all dummies/placeholders.

If a dummy/placeholder is contained in an extract, the ID of this extract is also displayed (with output mode Display).

Objects which are associated with a dummy/placeholder are also included in the report. The scope of the report can be limited by specifying an Association. Only parent objects of a certain type will then be included in the report.

Command:	DUMMY
Valid output modes:	List, Display, Select.
Limit scope of report:	The parameter Association can be used to restrict the scope of the report.

Finding Objects with Dummy Children

To list objects that have links to child objects that do not yet exist (dummies), use the retrieval type T and set the output option Dummy/Placeholder (related) to D.

Finding Objects with Placeholder Children

When an object that is linked to another object is imported with the Predict Coordinator with internal ID, and the referenced object is not imported and does not exist in the target environment, a Placeholder is added in the target environment for the referenced object.

The job of this placeholder is to reserve the object ID of the referenced object in the target environment so that the link in the old environment can be recreated in the new environment at a later time.

To list placeholder objects, use the retrieval type T and set the output option Dummy/Placeholder (related) to P.

Execute Retrieval Model - Code E

Execute Retrieval Model displays information on the associations of objects in the form of a structured list. A retrieval model must be specified in the input field model.

Command:	EXECMODEL
Valid output modes:	Structured list, Cross reference.
Limit scope of report:	The scope of a report is determined by the retrieval model. In addition to the settings in a retrieval model, output options can be used to restrict the scope of a report further.

The following rules apply to this retrieval type:

- The retrieval type Execute Retrieval Model does not apply to owners. Associations of owners can be evaluated with the retrieval type Cross reference. See *Type-Specific Retrieval*.
- Retrieval models determine
  - which objects related to an object are included in a report
  - which information is displayed for each individual object. This setting can be restricted with output options.
  - the layout of the report (only with output mode Structured list).
- The properties of a retrieval model are displayed in the cover page of reports if the parameter Cover page of the output options is set to Y.

43 Type-Specific Retrieval Types

- Database-Specific Retrieval Types 296
- Extract-Specific Retrieval Types 296
- Field-Specific Retrieval Types 297
- File-Specific Retrieval Types 298
- Keyword-Specific Retrieval Types 299
- Network-Specific Retrieval Types 299
- Owner-Specific Retrieval Types 300
- Storagespace-Specific Retrieval Type 301
- User-Specific Retrieval Type 301
- Verification-Specific Retrieval Types 301

Predict provides a variety of retrieval types that are designed for specific object types.

Note: Type-specific retrieval types are described in more detail in the respective sections of *Predefined Object Types in Predict* documentation.

Database-Specific Retrieval Types

Explode IMS Databases - Code I

Shows the hierarchical structure of an IMS/DL/I database. The level number before the file ID shows the level of the IMS/DL/I segment in the hierarchy. This retrieval type is only available for databases of type I.

Command:	EXPLODE DATABASE
Valid output mode:	Structured list
Limit scope of report:	Output options

Extract-Specific Retrieval Types

Extracts Related to no Object - Code Y

Lists extracts which are not assigned to any objects.

Command:	UNUSED EXTRACT.
Valid output modes:	Display, List, Select
Limit scope of report:	Output options

Extracts related to Objects - Code X

Lists objects that are related to the specified extracts.

Command:	RELATED EXTRACT
Valid output mode:	Cross reference
Limit scope of report:	Output options

Field-Specific Retrieval Types

Non-standard Fields - Code N

Reports on fields which are not derived from standard files, and on fields derived from standard files but subsequently changed to non-standard fields.

Command:	NONSTANDARD ELEMENT
Valid output modes:	Display, List, Select
Limit scope of report:	Output options

Fields and related views - Code R

Shows the fields that are used in views to reference a field.

Command:	RELATED ELEMENT
Valid output mode:	List
Limit scope of report:	Output options

Fields related to a Z-File - Code Z

Reports on fields which are derived from standard files.

Command:	STANDARD ELEMENT
Valid output modes:	List, Display
Limit scope of report:	Output options

File-Specific Retrieval Types

Difference of Files - Code A

Displays a screen to enter the IDs of the two files and select the attributes to be compared. See also the help screens of this retrieval type.

This type compares files and displays differences of files. The file attributes, the fields and the field attributes can be compared. The fields are compared by field ID, except where a userview is compared with its master file. In this case the fields are compared by field short name.

Command:	DIFFERENCE FILE
Valid output mode:	List
Limit scope of report:	Output options

Files Related to a File - Code R

Displays the relationships of master files and userviews. The following relationships are displayed:

- master files with their userviews
- userviews, their master files and other userviews of these master files.

For physical VSAM files, the related logical VSAM files are also listed; for IMS segments, the IMS segment layouts are also listed.

Command:	RELATED FILE
Valid output mode:	List
Limit scope of report:	Output options

Keyword-Specific Retrieval Types

Keywords Related to no Object - Code Y

Lists keywords which are not assigned to any objects.

Command:	UNUSED KEYWORD
Valid output modes:	Display, List, Select
Limit scope of report:	Output options

Cross reference Keywords - Code X

Lists objects, that are related to the specified keywords.

Command:	XREF KEYWORD
Valid output mode:	Cross reference
Limit scope of report:	Output options

Network-Specific Retrieval Types

Vista numbers - Code N

Displays information on the use of Vista numbers in list form.

Command:	LIST VISTATAB
Valid output mode:	List

Owner-Specific Retrieval Types

Owners with no user - Code O

Lists owners which are not assigned to any user.

Command:	FREE OWNER
Valid output modes:	List, Select

Objects with no owner - Code U

Reports on objects that have no owner.

Command:	EMPTY OWNER <OBJECT-TYPE>
Valid output mode:	List, Select It is not possible to select objects for immediate processing from lists produced with the output mode Select. Objects can, however, be selected for later processing from the workplan.
Limit scope of report:	Related type, Output options

Cross reference owners - Code X

Reports on all objects whose owner list contains specified owners.

Command:	XREF OWNER
Valid output mode:	Cross reference
Limit scope of report:	Output options

Storagespace-Specific Retrieval Type

Unused Storagespaces - Code N

List storagespaces that are unused (not referenced in a dataspace or field object).

Command:	UNUSED STORAGESPACE
Valid output modes:	List, Select
Limit scope of report:	Output options

User-Specific Retrieval Type

Users Related to no Object - Code Y

Lists users which are not related to any other objects in Predict. The association between a user and any other object (except keywords) is always established indirectly through an owner, by associating the owner with the user and with the object.

Command:	UNUSED USER
Valid output modes:	Display, List, Select
Limit scope of report:	Output options

Verification-Specific Retrieval Types

List Verifications to Regenerate - Code K

Lists verifications whose definitions have been modified since a DDM was generated containing a field that uses one of the verifications.

Command:	REGENERATE VERIFICATION
Valid output modes:	List, Select
Limit scope of report:	Output options

V Active Retrieval

Predict active retrieval functions evaluate XRef data and Predict documentation data to detect the differences between the documentation of an application and what has actually been implemented.

This section covers the following topics:

Concepts of Active Retrieval Functions

Executing Active Retrieval Operations

Saving the Result of Active Retrieval Operations in Sets

Member Active Retrieval

Field Active Retrieval

File Active Retrieval

Program Active Retrieval

System Active Retrieval

Verification Active Retrieval

44 Concepts of Active Retrieval Functions

- How XRef Data is Created 306
- What is an Implemented Member 306
- Where to Find Comprehensive Information on XRef Data 307

In addition to documentation data (Predict objects), XRef data is written for implemented objects to document what has actually been implemented for an application. Active retrieval functions evaluate both documentation data and XRef data to determine

- if objects documented in the dictionary are not yet implemented,
- if implemented members are not yet documented or
- if entry points of implemented members are not yet documented or
- if documentation data differs from the implementation.

Inconsistent use of field definitions can be found, along with answers to question such as "Which programs refer to file ABC?".

Active retrieval functions evaluate XRef data for objects in all libraries.

How XRef Data is Created

XRef data can be created in any of the following cases:

- By cataloging a Natural object with the parameter with XREF set to Y.
- By processing a 3GL program with Adabas Native SQL or the Predict Preprocessor.
- By generating a DB2 request module with the Natural for DB2 function Create DBRM.
- By creating a Predict system program object of type E (external object) and language Z (system program) documenting a program that is only available as object code. The implementation pointer for a system program must be specified explicitly. One entry point (with the ID of the program object) is created by Predict, additional entry points have to be specified manually.

What is an Implemented Member

An implemented member is a program object for which XRef data has been created with any of the options listed above.

Where to Find Comprehensive Information on XRef Data

For a general description of how XRef data is created and used, see the section Overview of Predict in the Introduction to Predict documentation.

See the sections **LIST XREF for Natural** and **LIST XREF for Third Generation Languages** for a more detailed description of the structure and contents of XRef data, and functions to retrieve information from XRef data.

45 Executing Active Retrieval Operations

- Selecting Objects for Active Retrieval Operations 310
- Using Implementation Pointers to Establish a Connection between Documentation and XRef Data 310

The active retrieval menu for a particular object type is called with code A and the object code in a Predict main menu or the command `ACTIVE <object-type>`. The table below shows the valid object types and the respective codes:

Object Type	Code
Field	EL
File	FI
Member	ME
Program	PR
System	SY
Verification	VE

Calling active retrieval functions with commands is described under the corresponding object type.

Selecting Objects for Active Retrieval Operations

Active retrieval operations are applied to either Predict objects or XRef data for implemented objects that are selected with selection criteria. All parameters used to specify selection criteria are described in the respective *Limiting the Scope of Object Type Active Retrieval* sections.

With the exception of Active Retrieval Member operations, all Active Retrieval operations are applied to documentation objects and evaluate XRef data accordingly. Active Retrieval Member operations are applied to XRef data and evaluate documentation data accordingly.

Using Implementation Pointers to Establish a Connection between Documentation and XRef Data

Active retrieval functions require that Predict knows which documentation objects document which members. The connection of documentation data and XRef data is established with the implementation pointer of documentation objects (member name, library name, user system file number and user system database number).

If the same member exists in several libraries, avoid multiple documentation of this member by omitting the library name, file number and database number. Predict then finds out for itself all the libraries in which this member exists.

With some functions, all implemented members are displayed. In others, the “best hit” principle is applied: the member with the most detailed implementation pointer is displayed. Example: if

one member is identified by member and library, and another by member, library and file number, the latter will be displayed.

46 Saving the Result of Active Retrieval Operations in Sets

- How and Where Sets are Saved 314
- Rules for Using Sets 314

The list of members generated by several active retrieval functions can be saved in sets for further processing with the List XRef function.

How and Where Sets are Saved

Sets are created by setting the SAVE SET flag in a retrieval menu to Y or with the function Create new set via selection in the XRef menu. See *LIST XREF for Natural* for more details.

There is no difference in the way sets are saved for Natural and 3GL.

Rules for Using Sets

The following rules apply when working with sets:

- Sets can be displayed, purged, sent to another user, merged, subtracted or intersected. When a set is sent, a short comment can be included that will appear when the set is displayed at the terminal of the recipient.
- Sets are saved separately for each user and each library. Any user defined in Predict can create and use up to ninety-nine sets in any library.
- Natural objects contained in sets can be cataloged or stowed, their contents can be edited or listed directly from the set.

Note: The use of sets applies to all active retrieval functions for all object types. See the sections *LIST XREF for Natural* and *LIST XREF for Third Generation Languages*.

47 Member Active Retrieval

- Displaying the Member Active Retrieval Screen 316
- Retrieval Type, Output Mode and Save Set Option 316
- Limiting the Scope of Member Active Retrieval 317
- Retrieval Types 318

Displaying the Member Active Retrieval Screen

The Member Active Retrieval screen is displayed with function code A and object code ME in a main menu or the command ACTIVE MEMBER.

```
13:44:44             *****  P R E D I C T  *****              2007-05-31
Plan   3             - (ME) Member Active Retrieval -              Profile HNO

 Retrieval type                              Retrieval type

 I  Members                                  O  Members not documented
 R  Entries referenced by members            T  Members with used files
 U  Entries not referenced                   V  Members using dynamic SQL
                                             M  Members referenced by members

Retrieval type .....
Output mode .......*
Member .............                                User system Fnr ..
Library ............                                User system DBnr .
From catalog date .. 0000-00-00 (YYYY-MM-DD)        Member of type ..*
To catalog date .... 2007-05-31 (YYYY-MM-DD)
Entry ..............


Output options ....*   Profile HNO                  Save set ......... N (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
```

Member Active Retrieval functions show where and how entry points, members and files are referenced by implemented members.

Retrieval Type, Output Mode and Save Set Option

An active retrieval operation consists of a retrieval type and an output mode. The retrieval types are described in more detail below. The output mode determines how the results are presented. Not all output modes are valid for all retrieval types. See table below.

Retrieval Type	Code	select	list	disp	count	Save Set
Members	I	X	X			X
Entries ref. by members	R	X	X		X	X
Entries not referenced	U	X	X			
Members not documented	O	X	X			
Members with used files	T		X			
Members using dynamic SQL	V	X	X			X
Members ref. by members	M		X			X

The results of retrieval types I, R, M and V can be saved in sets and processed with the Natural command LIST XREF, the Natural Object Handler, the Natural utility SYSMAIN or with Natural ISPF.

Limiting the Scope of Member Active Retrieval

Not all parameters are valid for all retrieval types. See table below.

Parameter	Retrieval Type						
	I	R	U	O	T	V	M
Member	X	X	X	X	X	X	X
User system Fnr	X	X	X	X	X	X	X
Library	X	X	X	X	X	X	X
User system DBnr	X	X	X	X	X	X	X
Member of type	X	X		X	X	X	X
From catalog date	X			X	X	X	X
Entry		X	X				
Output options		X	X				

Parameter	Description
Member	Restricts operation to member with ID specified. Use asterisk to specify range.
User system Fnr Library User system DBnr	These three parameters restrict the operation to members with the implementation pointer specified. Enter asterisk to specify range.
From catalog date	Restricts the operation to members that have been cataloged after the given date. Default=0000-00-00.
Member of type	Identifies the type of member to be retrieved. Enter asterisk for valid values.
To catalog date	Restricts the operation to members that have been cataloged before the given date. Default value is the current date.

Parameter	Description
Entry	Point of reference into an external program.
Output options	Output options determine the amount of information contained in a report. Output options can be specified temporarily with code T in the field Output options or taken from a profile. Enter asterisk for valid values or see *Output Options* for more information.

Retrieval Types

Members - Code I

Provides information on implemented members, their entry points and corresponding program descriptions in the data dictionary. Output mode L or S must be specified:

Output Mode	Description
L	Shows for each member: Type, language, file number, database number of the member; User-ID, terminal-ID, date and time of cataloging, entry points; Documented program together with implementation pointer. If the program is documented: Entry points of the member which are not yet documented are marked <<< not documented. Command: LIST MEMBER
S	Produces a list of files with the following information: Type, language, file number, database number of the member; Program in which the member is documented or the screen text >>>Member not documented<<<. List can be used to select members for further processing or add commands to the workplan. Enter an asterisk in field Cmd for list of possible functions. Command: SELECT MEMBER.

Entries referenced by members - Code R

Provides information on members which refer to the specified external entry. Output mode L, S or C must be specified:

Output Mode	Description
L	Lists members which refer to the specified external entry. Command: `REFERENCED ENTRY`
S	Shows a list of members which refer to the specified external entry. Members can be selected from list.

Output Mode	Description
C	Lists the libraries which use the specified external entry and indicates the number of references per library. Command: `COUNT ENTRY`

Entries not referenced - Code U

Provides information on entry points which are not referenced by any other member. Output mode L or S must be specified:

Output Mode	Description
L	Lists for each member the entry points that are not referenced by any other member sorted by documented program. Command: `UNUSED ENTRY`
S	Produces selection list as above from which entry points can be selected for further processing.

Members not documented - Code O

Provides information on members which have been cataloged or precompiled but are not documented in Predict. Output mode L or S must be specified:

Output Mode	Description
L	Lists undocumented members. The following information is also given: Type, language, file number, database number of the member; User-ID, terminal-ID, date and time of cataloging. Command: UNDEFINED MEMBER
S	Produces selection list of undocumented members. The following information is also given: Type, language, file number, database number of the member; ID of user who cataloged the member. Members can be selected from this list for further processing.

Members with used files - Code T

Provides information on files referred to by the specified member.

Command: ELEMENTS MEMBER

Output mode L must be specified:

Output Mode	Description
L	Lists members and the files that they use. Members that do not refer to any file are marked >>>>no file used<<<<. Files documented in Predict are marked with D.

Members using dynamic SQL - Code V

Provides information on Natural members that use DB2 tables and views (file types D or E) but not via Static SQL.

This function does not apply to 3GL members. Output mode L or S must be specified:

Output Mode	Description
L	Lists members. Some main attributes as well as user ID, terminal ID, date and time of cataloging are also given. Command: `DYNAMIC MEMBER`
S	Produces list of members, from which members can be selected for further processing or commands added to the workplan.

Members referenced by members - Code M

Provides information on members that refer to the specified member.

This function corresponds to the LIST XREF functions

- “Program referenced in programs” in the “Invoked programs” subsystem
- “Data area referenced in programs” in the “Data Area and Variables” subsystem
- “Copycode referenced in programs” in the “Copycode” subsystem

But in difference to LIST XREF, this function does not consider any steplibs structures. This means that all references to a member are shown, which are part of the Xref data.

Output mode L must be specified:

Output Mode	Description
L	Lists members that refer to the specified member. Command: `REFERENCED MEMBER`

48 Field Active Retrieval

- Retrieval Type, Output Mode and Save Set Option 322
- Limiting the Scope of Field Active Retrieval 323
- Retrieval Types 325

The Field Active Retrieval screen is displayed with function code A and object code EL in a main menu or the command `ACTIVE ELEMENT`.

```
13:55:18            *****  P R E D I C T  *****             2007-05-31
Plan   3             - (EL) Field Active Retrieval -             Profile HNO


                   Retrieval type

                   D  Fields
                   R  Fields referenced by members
                   U  Fields inconsistently used
                   O  Fields not documented

Retrieval type ...
Output mode .....*

Field ID/Synonym .                                  Synonym language .*
in file ..........                                  Usage ............*

Restrictions ....*   Profile HNO,used              For master fields . N (Y/N)
Output options ..*   Profile HNO                   Save set .......... N (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
```

Field Active Retrieval functions retrieve information from Predict field objects and from XRef data on the use of fields by implemented members. XRef data on the use of fields can only be evaluated for fields contained in implemented files. A file is regarded as implemented if

- a DDM has been generated for the file, or
- at least one 3GL program using the file has been processed by the Predict Preprocessor, Adabas Native SQL or the Adabas SQL Precompiler.

Retrieval Type, Output Mode and Save Set Option

An active retrieval operation consists of a retrieval type and an output mode. The retrieval types are described in more detail below. The output mode determines how the results are presented. Not all output modes are valid for all retrieval types. See table below:

Retrieval Type	Code	Output Mode				Save Set
		select	list	disp	count	
Fields	D	X	X	X		
Fields referenced by members	R	X	X		X	X
Fields inconsistently used	U	X	X			
Fields not documented	O		X			

Limiting the Scope of Field Active Retrieval

Not all parameters are valid for all retrieval types. See table below:

Parameter	Retrieval Type			
	D	R	U	O
Field ID/Synonym	X	X	X	
File	X	X	X	X
Language	X	X	X	
Usage		X	X	
Master		X	X	
Restrictions	X	X	X	
Output options	X	X	X	

Parameter	Description
Field ID/Synonym	Name of field or field name synonym. Synonyms are defined with the field attribute Field name synonym. Synonyms are used as field names when generating external objects from files.
Synonym language	Language of field name synonym. Enter asterisk for valid values. See table below for result of retrieval operations when synonym language is entered.

Field ID/synonym	Synonym language	Result
ABC*	C (COBOL)	- Fields with a COBOL synonym (for example) starting with ABC. - Fields with name starting with ABC and having no COBOL synonym.
ABC*	all	- Fields with any language synonym starting with ABC. - Fields with name starting with ABC without a synonym in at least one language.

Parameter	Description		
	ABC*	blank	All fields with name starting with ABC.
in file	Limits operation to fields in implemented file with specified ID. Enter asterisk to specify range.		
Usage	How the field is used.		
	A	Defined in area. The field is defined in a view of a Natural data area.	
	C	Counter. The C* (counter) variable of a MU or PE is used:	
	S	Search. The field is used in a search criterion:	
	R	Read. The field is used in a FIND or READ statement.	
	O	Read only. The field is read but not modified.	
	M	Modify. The field is used in an UPDATE or STORE statement.	
	N	Store. The field is used in a STORE statement.	
	U	Update. The field is used in an UPDATE statement:	
	blank	All.	
Restrictions	The scope of the function can be limited by **Restrictions** . Restrictions can be specified temporarily with code T in this field or taken from a profile. See **Using Restrictions and Output Options from Profiles**.		
For master fields	Retrieval types R and U can be restricted to show only the references to master fields. The related master fields are selected for each user view matching the other selection criteria.		
Output options	Output options determine the amount of information contained in a report. Output options can be specified temporarily with code T in the field Output options or taken from a profile. Enter asterisk for valid values or see **Output Options** for more information. Only output options Cover page, Page size and Sorted by field apply to field active retrieval operations.		

Retrieval Types

Fields - Code D

Provides information on fields by invoking the retrieval type of same name. Output mode D, L or S must be specified:

Output Mode	Description
D	Shows for each field the attributes that are specified in the output options.
L	Shows a single-line list of fields with some main attributes.
S	Shows list as above, from which fields can be selected for further processing or commands added to the workplan. Enter an asterisk in field Cmd for list of possible functions.

Fields referenced by members - Code R

Provides information on members using fields. Output mode L, C or S must be specified:

Output Mode	Description
L	Shows for every referenced field the list of all members using the field. Member type/language/library, file number, database number and field usage are also indicated. Command: REFERENCED ELEMENT
C	Shows for every referenced field a summary list which indicates for each library and each field usage (for example read or update) the number of members that use the field. Command: COUNT ELEMENT
S	Shows a list of referenced fields indicating how many members use each field. Fields can be selected for further processing or commands added to the workplan. Enter an asterisk in field Cmd for list of possible functions.

Fields inconsistently used - Code U

Shows a list of fields that are not used by any member or which may be used inconsistently, for example a field that is never updated or a descriptor that is never used for a search.

If the parameter Usage is specified, only those fields which are *not* used as specified are shown. For example: If Usage=U, only fields which are not updated are shown.

Output mode L or S must be specified:

Output Mode	Description
L	Lists fields inconsistently used. Some main attributes are also given, together with a brief description of the possible inconsistency. Examples: - used, but not updated - not used for search. Command: `UNUSED ELEMENT`
S	Shows list of fields used inconsistently with file ID and usage code. Fields can be selected from the list for further processing or commands added to the workplan. Enter an asterisk in field Cmd for list of possible functions.

Fields not documented - Code O

Shows a list of fields referenced by members but not documented in Predict. Output mode L must be specified:

Output Mode	Description
L	Lists undocumented referenced fields. Member type/language/library, file number, database number and field usage are also indicated. Command: `UNDEFINED ELEMENT`

49 File Active Retrieval

- Retrieval Type, Output Mode and Save Set Option 328
- Limiting the Scope of File Active Retrieval 329
- Retrieval Types 330

The File Active Retrieval screen is displayed with function code A and object code FI in a main menu or the command ACTIVE FILE.

```
13:58:15            *****  P R E D I C T  *****             2007-05-31
Plan   3              - (FI) File Active Retrieval -             Profile HNO

                    Retrieval type

                    D  Files
                    R  Files referenced by members
                    U  Files inconsistently used
                    O  Files not documented
                    V  Files accessed via dynamic SQL

Retrieval type ...
Output mode .....*

File ID ..........                                  Files of type ...*
in database ......                                  File number ......
                                                    Usage ...........*
Restrictions ....*   Profile HNO,used               For master files . N (Y/N)
Output options ..*   Profile HNO                    Save set ......... N (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
```

File Active Retrieval functions retrieve information from Predict file objects and XRef data on the use of implemented files. A file is regarded as implemented

- if a DDM has been generated for the file, or
- if at least one 3GL program using the field has been processed by the Predict Preprocessor, Adabas Native SQL or the Adabas SQL Precompiler

Retrieval types R, O and V consider only implemented files.

Retrieval Type, Output Mode and Save Set Option

An active retrieval operation consists of a retrieval type and an output mode. The retrieval types are described in more detail below. The output mode determines how the results are presented. Not all output modes are valid for all retrieval types. See table below.

Retrieval Type	Code	Output Mode				Save Set
		select	list	disp	count	
Files	D	X	X	X		
Files referenced by members	R	X	X		X	X
Files inconsistently used	U	X	X			
Files not documented	O		X			X
Files accessed via dynamic SQL	V	X	X			X

The results of retrieval types R, O and V can be saved in sets and processed with the Natural command LIST XREF. See **Using Sets** .

Limiting the Scope of File Active Retrieval

Not all parameters are valid for all retrieval types. See table below.

Parameter	Retrieval Type				
	D	R	U	O	V
File ID	Y	Y	Y	Y	Y
Files of type	Y	Y	Y		Y
in database	Y	Y	Y		Y
File number	Y	Y	Y		Y
Usage		Y	Y		Y
Restrictions	Y	Y	Y		Y
For master files		Y	Y		
Output options	Y	Y	Y		Y

Parameter	Description	
File ID	ID of Predict files to be processed. Use asterisk to specify range.	
in database	Restricts operation to file(s) in database specified. Use asterisk to specify range of databases.	
Files of type	Restricts operation to files of type specified. Enter asterisk for valid values. Enter blank for all file types.	
File number	If a file number is entered, the operation is restricted to this file number. If no number is entered, all files are considered. It is not possible to enter a range of values.	
Usage	This parameter applies to retrieval types R, U and V. Valid values: A, S, R, O, M, N, U, D, blank.	
	A	Defined in area. A view of the file is defined in a Natural data area.
	S	Search.

Parameter	Description	
	R	Read. The file is used in a FIND or READ statement.
	O	Read only. The file is read but not modified.
	M	Modify. The file is used in an UPDATE, STORE or DELETE statement.
	N	Store. The file is used in a STORE statement.
	U	Update. The file is used in an UPDATE statement.
	D	Delete. The file is used in a DELETE statement.
	blank	All.
Restrictions	The scope of the function can be limited by **Restrictions**. Restrictions can be specified temporarily with code T in this field or taken from a profile. See ***Using Restrictions and Output Options from Profiles***.	
For master files	Retrieval types R and U can be restricted to show only the references to master files. For each master file the members using the master file either directly or via a userview are listed.	
Output options	Output options determine the amount of information contained in a report. Output options can be specified temporarily with code T in the field Output options or taken from a profile. Enter asterisk for valid values or **Output Options** for more information.	

Retrieval Types

Files - Code D

Provides information on files by invoking the retrieval type of same name. Output mode D, L or S must be specified:

Output Mode	Description
D	Shows for each file the attributes that are specified in the output options. Command: `DISPLAY FILE`
L	Shows a single-line list of files with some main attributes. Command: `LIST FILE`
S	Shows a list as above, from which files can be selected for further processing or commands added to the workplan. Enter an asterisk in field Cmd for list of possible functions. Command: `SELECT FILE`

Files referenced by members - Code R

Provides information on members using the specified files. Output mode L, S or C must be specified:

Output Mode	Description
L	Shows for every referenced file the list of all members using the file. Member type/language/library, file number, database number and file usage are also indicated. Command: `REFERENCED FILE`.
S	Shows a list of referenced files for selection indicating how many members use each file. Files can be selected for further processing or commands added to the workplan. Enter an asterisk in field Cmd for list of possible functions.
C	Shows for every referenced file a summary list which indicates for each library how many members use the file and the file usage, for example read or update. If members of more than one library use the file, the total usages are also given. Command: `COUNT FILE`.

Files inconsistently used - Code U

Shows a list of files that are not used by any member or which may be used incorrectly.

Example: a file that is never updated or never deleted

If the parameter Usage is specified, only those files which are *not* used as specified are shown.

Example: If Usage=U, only files which are *not* updated are shown. Only implemented files are considered with this retrieval type.

Output mode L or S must be specified:

Output Mode	Description
L	Lists files not used in the manner specified. Some main attributes are also given, together with a brief description of the possible inconsistency. Examples: - used, but not updated - not used for search. Command: `UNUSED FILE`
S	Produces list as above, from which files can be selected for further processing or commands added to the workplan. Enter an asterisk in field Cmd for list of possible functions.

Files not documented - Code O

Shows a list of files that are referenced by members but are not documented in Predict and have no DDM. To find out which DDMs are not documented in Predict, use the function Incorporate DDM.

Command: UNDEFINED FILE

Output mode L must be specified:

Output Mode	Description
L	Lists undocumented referenced files. Member type/language/library, file number, database number and file usage are also indicated.

Files accessed via dynamic SQL - Code V

Lists DB2 tables and views (file types D or E) that are used by Natural members but not via static SQL.

Output mode L or S must be specified:

Output Mode	Description
L	Lists the specified files. Member type/language/library, file number, database number and file usage are also indicated. Command: `DYNAMIC FILE`
S	Shows a list as above, from which files can be selected for further processing or commands added to the workplan. Enter an asterisk in field Cmd for list of possible functions.

50 Program Active Retrieval

- Displaying the Program Active Retrieval Screen 334
- Retrieval Type, Output Mode and Save Set Option 335
- Limiting the Scope of Program Active Retrieval 335
- Retrieval Types 336
- Applying the Function to Different Program Types 339

Displaying the Program Active Retrieval Screen

The Program Active Retrieval screen is displayed with function code A and object code PR in a main menu or the command ACTIVE PROGRAM.

```
13:06:33             *****  P R E D I C T  *****              2007-05-31
Plan   3             - (PR) Program Active Retrieval -             Profile HNO

 Retrieval type                            Retrieval type

 D  Programs                               N  Programs not implemented
 R  Entries referenced by members          P  Programs using programs
 U  Entries not referenced                 F  Programs using files

Retrieval type ....
Output mode ......*
Program ID ........                                  Programs of type .*
in system .........                                  Language .........*
Member ............                                  User system Fnr ...
Library ...........                                  User system DBnr ..
Library structure *
Entry .............
Restrictions .....*   Profile HNO,used
Output options ...*   Profile HNO                 Save set .......... N (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
```

Program Active Retrieval functions retrieve information from Predict program objects and from XRef data for the following types of objects:

- Natural objects cataloged with with XREF set to Y.
- 3GL programs processed by the Predict Preprocessor or Adabas Native SQL
- DB2 database request modules generated with the Natural for DB2 function Create DBRM with the active reference option switched on.

Retrieval types Entries referenced by members and Entries not referenced only apply to 3GL programs.

Retrieval Type, Output Mode and Save Set Option

An active retrieval operation consists of a retrieval type and an output mode. The retrieval types are described in more detail below. The output mode determines how the results are presented. Not all output modes are valid for all retrieval types. See table below.

Retrieval Type	Code	Output Mode				Save Set
		select	list	disp	count	
Programs	D	X	X	X		
Entries referenced by members	R	X	X		X	X
Entries not referenced	U	X	X			
Programs not implemented	N	X	X			
Programs using programs	P		X	X		
Programs using files	F		X	X		

The results of retrieval type R can be saved in sets and processed with the Natural command LIST XREF.

Limiting the Scope of Program Active Retrieval

Parameter	Description
Program ID	Restricts operation to program with ID specified. Use asterisk notation to specify a range.
Programs of type	Restricts operation to programs of type specified. Enter asterisk for possible values. Enter blank for all program types.
in system	Restricts operation to programs in system specified. Use asterisk notation to specify a range.
Language	Restricts the operation to programs with the language specified. Enter blank for all program languages. This parameter specifies the language of the caller program.
Member, Library, User system Fnr/DBnr	Limits the function to Programs with the implementation pointer specified. For Member and Library you can enter a range of values using asterisk notation.
Library structure	This parameter is only evaluated for the function **Programs using programs**. Only Programs which have the same implementation pointer as one of the Systems in the Library Structure and have linked children of type Program are evaluated. If you do not enter a Library Structure, the programs are evaluated as in earlier versions of Predict without steplib logic.

Parameter	Description
	If you do enter a Library Structure, it must be fully qualified. If you use asterisk notation and only one Library Structure falls within the range specified, this is taken. Otherwise a window appears in which you can select a Library Structure from a list. The implementation pointer of the first System in the Library Structure - the main library - must be complete (parameters Library, Fnr, DBnr). If this is not the case, a window appears in which you must specify the missing parameters. See the section Steplib Support in the Predict and Other Systems documentation.
Entry	Restricts the operation to programs with the entry points specified. Not applicable to active retrieval types D and N.
Restrictions	The scope of the function can be limited by Restrictions. Restrictions can be specified temporarily with code T in this field or taken from a profile. See **Using Restrictions and Output Options from Profiles** Restrictions apply only to Predict objects. They cannot be applied to libraries, for example.
Output options	Output options determine the amount of information contained in a report. Output options can be specified temporarily with code T in the field Output options or taken from a profile. Enter asterisk for valid values or see **Output Options** for more information. Not applicable to active retrieval types D and N.

Retrieval Types

Programs - Code D

Provides information on programs by invoking the retrieval type of same name. Output mode D, L or S must be specified:

Output Mode	Description
D	Displays programs with attributes, implementation pointers, entry points and user exits. Command: `DISPLAY PROGRAM`
L	Shows a single-line list of programs with some main attributes. Command: `LIST PROGRAM`
S	Displays list of programs from which entries can be selected for further processing or commands added to the workplan. Enter an asterisk in field Cmd for list of possible functions. Command: `SELECT PROGRAM`.

Entries referenced by members - Code R

Provides information on members which refer to the specified entries. Output mode L, S or C must be specified:

Output Mode	Description
L	Lists the members which refer to the specified entries. Command: `REFERENCED PROGRAM`
S	Produces list of programs with their entries from which programs can be selected for further processing or commands added to the workplan.
C	Lists the libraries which use the specified entries and indicates the number of references per library and, if the entry is used in more than one library, the total number for all libraries. Command: `COUNT PROGRAM`

Entries not referenced - Code U

Provides for each member (implemented program) the entry points that are not referenced by any other member. Output mode L or S must be specified:

Output Mode	Description
L	Lists entry points not referenced by any other member. Command: `UNUSED PROGRAM`
S	Produces a list as above from which entries can be selected for further processing.

Programs not implemented - Code N

Provides information on programs that are defined in Predict but do not correspond to members actually implemented. Output mode L or S must be specified:

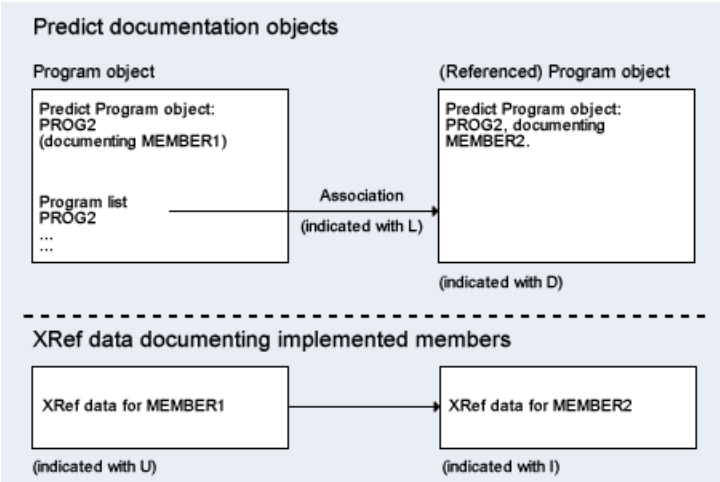
Output Mode	Description
L	Lists programs that are defined in Predict but have no corresponding implemented member. Command: Note: `XIST PROGRAM`
S	Produces a list as above from which programs can be selected for further processing or commands be added to the workplan.

Programs using programs - Code P

Lists programs with referenced programs. For each program all linked programs and all programs which are used according to XRef data are listed. Output mode L or D must be specified

Output Mode	Description
L	Name, type and implementation pointer of the using program and at least one line for each used program is displayed, depending on the output options specified. Command: `USING PROGRAM`
D	Equal to "L" but more information about the using program is displayed.

The diagrams below show the possible types of data used to document programs/members and relationships of programs/members in Predict. The four boxes represent the status information on referenced programs in columns L, D, I and U of the output list of this function. If information in any of the four boxes or information represented by the arrows is missing, this indicates that the implementation or the documentation of an application is incomplete.



Note: In the following description, the term program object is used for a Predict object of type program, whereas the term member identifies an implemented program/subroutine for which XRef data exists.

Applying the Function to Different Program Types

The evaluation of documentation and XRef data is different for related programs of type program and related programs of type subroutine. 3GL programs are treated as subroutines, where Predict scans for the entry points.

```
13:18:23            *****  P R E D I C T  *****            2007-05-31
                      - List Program using Programs -             Page:    1

Program ID ......... * #MAP1___SU-XREF_0094100010
Type ............... Map / Helpmap
-------------------------------------------------------------------------------

Implementation
  Member .. 'MAP1    Library .. SU-XREF  Fnr .. 941   DBnr .. 10

Cnt  Uses PR concept.                Ty Member   Library    Fnr  DBnr  L D I U

   1 #HELP___SU-XREF_0094100010      H  #HELP    SU-XREF    941    10    D I U
   2 #INCLUDESU-XREF_0094100010      C  #INCLUDE SU-XREF    941    10    D I U
   3 #LDA____SU-XREF_0094100010      L  #LDA     SU-XREF    941    10    D I U
```

Description of the Status Information in the Columns L, D, I and U

For a description of all other columns, see the section Layout of Program Lists in the Predefined Object Types in Predict documentation.

Parameter	Description
L (linked)	An association to the related Program ID exists. If L is displayed and D is not, the child object is a dummy (related Program ID is in the program list but not documented with a documentation object). In the example above, the objects YSHLP3 and YSHLP4 are dummies.
D (documented)	A Predict program object for related Program ID exists.
I (implemented)	Related Program ID is implemented. Predict evaluates XRef data to determine whether an object is implemented. It is therefore possible that a related Program ID that is not (yet) documented appears in the list and is indicated as implemented. In the example above, SYSHLP7 and SYSHLP8 are implemented but not documented.
U (used)	The related Program ID object is used by an implemented member. Predict evaluates XRef data and identifies objects as used if a FETCH or PERFORM statement is issued for them. Hence, a member that appears with a related Program ID in the list does not necessarily have to be implemented. In the example above, TSH-SYSHLP5 is used but not implemented.

Evaluating Links to Programs

The table below describes actions to restore consistency between documentation and implementation depending on the current status of the objects/members indicated with related program ID in a List Program with programs list.

Combination of				Recommended Actions to Restore Consistency
L	D	I	U	
L	D	I	U	No action necessary. The related program Program ID is linked, documented, implemented and used.
L	D	I	-	Use the member or purge the link and the program object in the Predict documentation.
L	D	-	-	Implement and use a member as defined in the Predict documentation or purge the association and the program object in the Predict documentation.
L	D	-	U	Implement the related Program ID or purge the link and the program object in the Predict documentation and remove the corresponding FETCH statement from the member.
-	D	I	U	Link the related program object Program ID to the program object in the Predict documentation.
-	D	-	U	Link the related Program ID to the program object in the Predict documentation and implement the member.
-	-	I	U	Document the related Program ID and link it to the program object in the Predict documentation.
-	-	-	U	Implement the related Program ID, document it in Predict and link it to the program object or remove FETCH statement from member.
L	-	-	-	Document, implement and use the program or purge the link in the Predict documentation.

Evaluating Links to Subroutines / 3GL Programs

If the related program is a subroutine or a 3GL program, one of two additional remarks may be displayed if the L/D/I/U combination is not sufficient to express the actual status of the related (sub)program.

Note: For a description of all other possible L D I U combinations refer to the table in the section *Evaluating Links to Programs* above.

Combination of				With Remark	Recommended Actions to Restore Consistency
L	D	I	U		
L	D	I	U	Entry not in Docu (see diagram 1 below)	The related program object Program ID (type subroutine) is correctly linked to the program object, but the XRef data for the corresponding members indicates that - MEMBER1 does not call a function / 3 GL program as documented in the Predict object PROG2 and - MEMBER2 does not contain a function as documented in the Predict object PROG2.

L	D	I	U	Entry not impl (see diagram 2 below)	The related program object Program ID (type subroutine or program with language set to a 3GL language) is correctly linked to the program object, but the XRef data for the corresponding members indicates that - MEMBER2 does *not* contain a function / 3GL entry point that is documented in the Predict program object PROG2, but - MEMBER1 calls a function / entry point as documented.

Diagram 1

The diagram below illustrates the status indicated with L D I U Entry ... not in Docu.

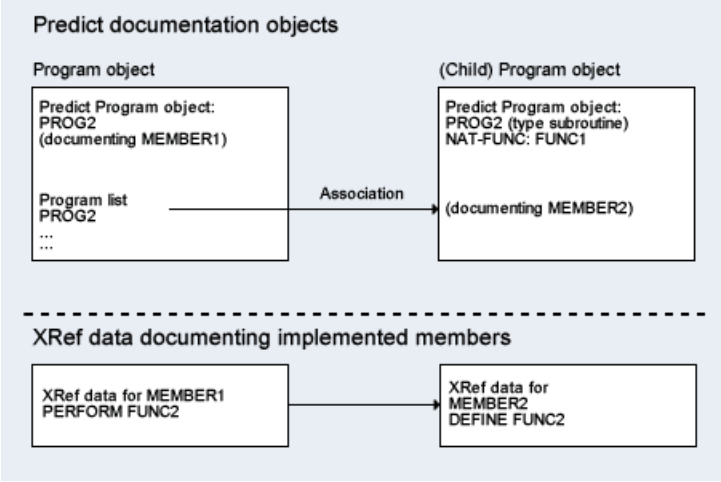
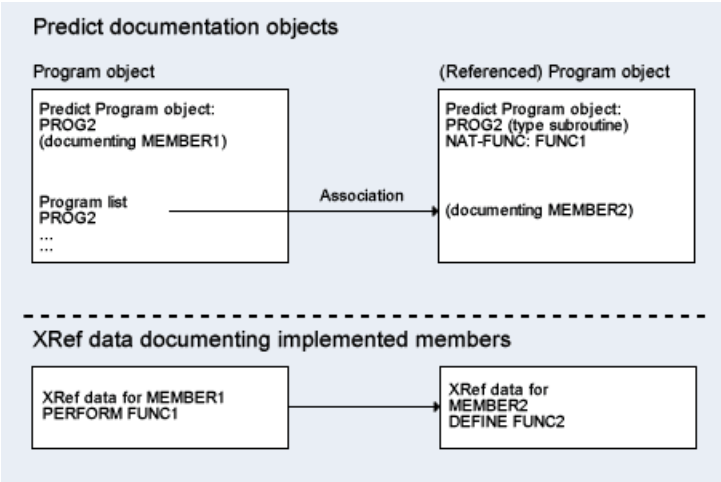


Diagram 2

The diagram below illustrates the status indicated with L D I U Entry ... not in Impl.



Programs using files - Code F

Lists programs and files which are linked as children or which, according to XRef data, are used by the programs. Information is displayed for each file about whether it is linked, documented, implemented or used. Output mode L or D must be specified.

Output Mode	Description
L	For each program, the name, type, implementation pointer and some more attributes are given. At least one line for each file is displayed, depending on the output options specified. Command: `USING FILE`
D	As above, but more detailed information about the program is given.

51 System Active Retrieval

- Displaying the System Active Retrieval Screen 344
- Retrieval Type and Output Mode 345
- Limiting the Scope of System Active Retrieval 345
- Retrieval Types 346

Displaying the System Active Retrieval Screen

The System Active Retrieval screen is displayed with function code A and object code SY in a main menu or the command `ACTIVE SYSTEM`.

```
13:34:23             *****  P R E D I C T  *****              2007-05-31
Plan   3              - (SY) System Active Retrieval -             Profile HNO

  Retrieval type                           Retrieval type

  D  Systems                               I  Libraries
  P  Systems containing programs           O  Libraries not documented
  N  Systems not implemented               T  Libraries with members

Retrieval type .....
Output mode .......* L  List

System ID ..........                                  System of type ..*
Library ............                                  User system Fnr ..
Library structure .*                                  User system DBnr .


Restrictions ......*   Profile HNO,used
Output options ....*   Profile HNO

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
```

System Active Retrieval functions retrieve information from Predict system objects that are regarded to be implemented and XRef data for objects contained in the respective libraries (systems). A Predict system object is regarded to be implemented if

- an implementation pointer exists and points to a Natural library that contains objects cataloged with the active reference option switched on
- DB2 application plan exists, indicating the system is physically implemented in DB2.

Retrieval Type and Output Mode

An active retrieval operation consists of a retrieval type and an output mode. The retrieval types are described in more detail below. The output mode determines how the results are presented. Not all output modes are valid for all retrieval types. See table below.

Retrieval Type	Code	Output Mode		
		select	list	disp
Systems	D	Y	Y	Y
Libraries	I	Y	Y	
Systems not implemented	N	Y	Y	
Libraries not documented	O	Y	Y	
Systems containing programs	P		Y	Y
Libraries with members	T		Y	

Limiting the Scope of System Active Retrieval

Not all parameters are valid for all retrieval types. See table below.

Parameter	Retrieval Type					
	D	I	N	O	P	T
System ID	X		X		X	
System of type	X		X		X	
Library	X	X		X	X	X
User system Fnr	X	X		X	X	X
User system DBnr	X	X		X	X	X
Library structure					X	
Restrictions	X		X		X	
Output options	X	X	X	X	X	X

Parameter	Description
System ID	Restricts operation to system with ID specified. Use asterisk notation to specify a range.
Systems of type	Restricts operation to systems of type specified. Enter asterisk for possible values. Enter blank for all program types.
Library, User system Fnr, DBnr	These three parameters restrict the operation to programs with the implementation pointer specified. Enter asterisk to specify range.
Library structure	Only systems with an implementation pointer that can be found in the steplib structure defined in this Library Structure and with children of type Program are evaluated. See **Systems containing programs**. The full implications of this parameter are described in the section Steplib Support #n the Predict and Other Systems documentation.
Restrictions	The scope of the function can be limited by Restrictions. **Restrictions** can be specified temporarily with code T in this field or taken from a profile. See ***Using Restrictions and Output Options from Profiles***. Restrictions apply only to Predict objects. They cannot be applied to libraries, for example.
Output options	Output options determine the amount of information contained in a report. Output options can be specified temporarily with code T in the field Output options or taken from a profile. Enter asterisk for valid values or see ***Output Options*** for more information.

Retrieval Types

Systems - Code D

Provides information on systems by invoking the retrieval type of same name. Output mode D, L or S must be specified:

Output Mode	Description
D	Shows for each system the attributes that are specified in the output options. Command: `DISPLAY SYSTEM`
L	Lists systems with their main attributes. Command: `LIST SYSTEM`
S	Produces a list as above from which systems can be selected for further processing or commands added to the workplan. Command: `SELECT SYSTEM`

Libraries - Code I

Provides information on applications that use Natural members along with any Predict system definitions. Output mode L or S must be specified:

Output Mode	Description
L	Lists implemented libraries. If the library is defined in Predict, the corresponding system is also indicated. If not, libraries are marked >>>>library not documented<<<<. Command: `LIST LIBRARY`
S	Produces list as above, from which libraries can be selected for further processing. Command: `SELECT LIBRARY`

Systems not implemented - Code N

Provides information on systems which are documented in Predict but do not correspond to applications actually in use. Output mode L or S must be specified:

Output Mode	Description
L	Lists systems not implemented. System type, library, file number and database number are also given. If no implementation data is available, the message >>>no pointer<<< appears. For DB2: plan name is given instead of library, file and database number. If plan name does not exist, the message >>>not in catalog<<<< appears. Command: Note: `XIST SYSTEM`
S	Produces list as above, from which systems can be selected for further processing or commands added to the workplan. Enter an asterisk in field Cmd for list of possible functions.

Libraries not documented - Code O

Provides information on libraries that use Natural members but have no corresponding Predict definitions. Output mode L or S must be specified:

Output Mode	Description
L	Lists libraries not defined in Predict. File number and database number are also indicated. List is sorted by database number. Command: `UNDEFINED LIBRARY`
S	Produces list as above, from which libraries can be selected for further processing.

Libraries with members - Code T

Provides information on applications which use Natural members and data areas, along with any members and any corresponding Predict program definitions that exist.

Command: `ELEMENTS LIBRARY`

Output mode L must be specified:

Output Mode	Description
L	Lists libraries with members. If a member is defined in Predict, the corresponding program is also indicated. If not, the member is marked >>>not documented<<<.

Systems containing programs - Code P

Lists systems with programs contained in them. For each system all linked programs and all programs for which XRef data exist in the documented library are listed. Output mode L or D must be specified.

Output Mode	Description
L	Name, type and implementation pointer of the system and at least one line for each program contained in the system is displayed, depending on the output options specified. Command: `CONTAINING PROGRAM`
D	Produces list as above, but more information about the system is displayed.

For more information on executing this function with parameter Library structure, see the section Steplib Support in the Predict and Other Systems documentation.

52 Verification Active Retrieval

- Displaying the Verification Active Retrieval Screen 350
- Retrieval Type, Output Mode and Save Set Option 350
- Retrieval Types 352

Displaying the Verification Active Retrieval Screen

The Verification Active Retrieval screen is displayed with function code A and object code VE in a main menu or the command ACTIVE VERIFICATION.

```
13:52:32              *****  P R E D I C T  *****              2007-05-31
Plan   3            - (VE) Verification Active Retrieval -          Profile HNO



                    Retrieval type

                    D  Verifications
                    R  Verifications referenced by members
                    U  Verifications not referenced

Retrieval type ...
Output mode .....*

Verification ID ..                                      Status ........*
in file ..........                                      Format ........*

Restrictions ....*   Profile HNO,used
Output options ..*   Profile HNO                      Save set ....... N (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
```

Verification Active Retrieval functions provide information on where and how verifications are referenced by cataloged Natural objects.

Retrieval Type, Output Mode and Save Set Option

An active retrieval operation consists of a retrieval type and an output mode. The retrieval types are described in more detail below. The output mode determines how the results are presented. Not all output modes are valid for all retrieval types. See table below.

Retrieval Type	Code	Output Mode				Save Set
		select	list	disp	count	
Verifications	D	X	X	X		
Verifications ref. by members	R	X	X		X	X
Verifications not referenced	U	X	X			

The results of retrieval types R can be saved in sets and processed with the Natural command LIST XREF.

Limiting the Scope of Verification Active Retrieval

Not all parameters are valid for all retrieval types. See table below.

Parameter	Retrieval Type		
	D	R	U
Verification ID	X	X	X
Status	X	X	X
in file		X	X
Format	X	X	X
Restrictions	X	X	X
Output options	X	X	X

Parameter	Description
Verification ID	Restricts operation to verification with specified ID. Use asterisk to specify a range.
Status	Restricts operation to verifications with the specified status. Enter asterisk for valid values. Enter blank for all.
in file	Restricts operation to verifications attached to fields in the specified file. Enter asterisk to specify a range.
Format	Restricts operation to verifications with the specified format. Enter asterisk for valid values. Enter blank for all.
Restrictions	The scope of the function can be limited by Restrictions. ***Restrictions*** can be specified temporarily with code T in this field or taken from a profile. See ***Using Restrictions and Output Options from Profiles***.
Output options	Output options determine the amount of information contained in a report. Output options can be specified temporarily with code T in the field Output options or taken from a profile. Enter asterisk for valid values or see **Output Options** for more information.

Retrieval Types

Verifications - Code D

Provides information on verifications by invoking the retrieval type of same name. Output mode D, L or S must be specified:

Output Mode	Description
D	Displays specified verification with attributes, verification values and processing rule.
L	Lists verifications with attributes status, format, compatible formats, type. Verification values are also listed, if applicable.
S	Produces a list as above, from which verifications can be selected for further processing or commands added to the workplan. Enter an asterisk in field Cmd for list of possible functions. List of verification values may be truncated.

Verifications referenced by members - Code R

Provides information on all members using the verifications. Output mode L, C or S must be specified:

Output Mode	Description
L	Lists all members that use the specified verifications with some main attributes. If an automatic rule is used as a free rule, a message is displayed. Command: `REFERENCED VERIFICATION`
C	Lists the members which use the specified verifications and indicates the number of references per library and, if the verification is used in more than one library, the total number for all libraries. Command: `COUNT VERIFICATION`
S	Produces list of verifications and indicates by how many members each verification is used. Verifications can be selected for further processing or commands added to the workplan.

Verifications not referenced - Code U

Provides information on verifications that are not used by any member. Output mode L or S must be specified:

Output Mode	Description
L	Lists verifications not used by any member with following attributes: status, formats, compatible format, type. Command: `UNUSED VERIFICATION`
S	Produces list as above from which verifications can be selected for further processing or commands added to the workplan. Enter an asterisk in field Cmd for list of possible functions.

VI SYSHELP - Active Help System for Your Application

53 SYSHELP - Active Help System for Your Application

- Active Help System of Predict - Overview 358
- Overview of Activities when Using SYSHELP 360
- Using Field Help 361
- Using Screen Help 363
- Interface to the SYSHELP Subprogram 363

SYSHELP, the Active Help System provided by Predict for your application, consists of the subprogram SYSHELP and the help routines SYSHLP1 to SYSHLP10.

With the rules coded in the help routines, SYSHELP reads information from Predict objects and makes this information available as screen or field online help.

Note: An extended description can use Long Lines of up to 250 characters. However, if you intend to use extended descriptions with the SYSHELP utility to generate your own online help system, you should limit the line length of your extended descriptions to 72 characters.

Active Help System of Predict - Overview

Predict provides routines which enable you to use information contained in the abstract or extended description of a Predict object as online help text in your application for screens (referred to in this section as screen help, and for fields (referred to here as field help).

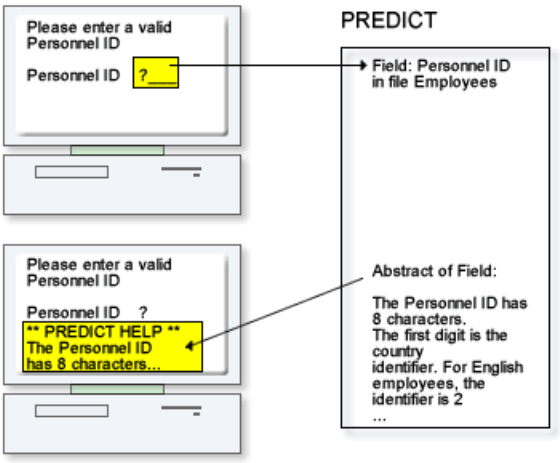
The following information can be displayed as online help:

- Abstracts and extended descriptions of databases, files, file relations, fields, verifications, systems and programs of type M (Map).
- Permitted values of verifications linked to a field. Permitted values can be displayed for selection.

Abstracts and verification values are displayed in a window, extended descriptions are displayed full screen.

Online help read from Predict objects can be displayed in two forms:

- Help on individual input fields (field help) can be displayed by entering a question mark (?) in the field a help routine it is assigned to. See ***Using Field Help***.
- Help on screens (screen help) can be displayed by pressing PF1 in a map. The abstract and extended description of a Predict Program objects can be used as **screen help**.



The help screens need not use all of the abstracts or extended descriptions: display can start at any line.

How the Help Facility Works

The Predict help facility consists of the subprogram SYSHELP and the help routines SYSHLP1 to SYSHLP10. SYSHELP retrieves information from Predict objects according to rules coded in the help routines and displays this information as screen help or as field help.

Overview of Help Routines

Help Routine	Use
SYSHLP1	Abstract for a field and for linked Verifications
SYSHLP2	Permitted values for verifications. Allows one value to be selected by marking it with the cursor
SYSHLP3	Extended description for verifications
SYSHLP4	Extended description for fields
SYSHLP5	Abstract for programs of type M (map)
SYSHLP6	Extended description for programs of type M (map)
SYSHLP7	Abstract for databases, files, systems
SYSHLP8	Extended description for databases, files, systems
SYSHLP9	Abstract for file relations
SYSHLP10	Extended description for file relations

The help routines delivered with Predict are examples.

Additional help routines must be created when displaying Verification values for several fields with different length. See *Using SYSHLP2 to Display Verification Values for Selection*.

The parameters contained in the parameter data area SYSHELPA are passed when SYSHELP is called. These parameters determine which information is to be retrieved and how help routines work and are described under ***Interface to the SYSHELP Subprogram***.

How the Help Facility is Delivered

SYSHELP and the routines SYSHLP1 to SYSHLP10 are contained in the library SYSDICH. The following applies to the delivery of the help facility:

- SYSHELP is delivered as a load module to guarantee an upward compatible interface to Predict. This ensures that your help system will continue to work with the same source code when a new version of Predict is installed.
- The help routines SYSHLP1 to SYSHLP10 are delivered as source code. These routines are working examples to show the functionality and scope of the Predict help facility. The routines can be modified as desired.
- The program EXAPGM demonstrates the help facility. Handling of this program is self-explanatory.

Overview of Activities when Using SYSHELP

The following activities are required when using SYSHELP:

1. The help routine to be used has to be assigned to the map (for screen help) or to the field of a map (for field help). Different help routines are used to deliver different types of information. See **Overview of Help Routines**.
2. The Predict object delivering the help text has to be specified. How this is done depends on the type of Predict object. See the respective sections *How to specify the field/database/file relation/system/verification object* below.
3. The help routine to be used can be changed to meet specific demands. One option is to move values other than the default settings to variables in the parameter data area SYSHELPA. See ***Interface to the SYSHELP Subprogram***. Help routines may also be rewritten.
4. When using SYSHLP2 to display selection values, the definition of the variable MAP-VARIABLE has to be changed to the format and length of the input field in the map.
5. Help routine and map have to be cataloged.

Using Field Help

This section describes the use of text from Predict objects as field help. For a description of using Predict text as screen help, see **Using Screen Help** .

Assigning the Help Routines

Help routines are assigned to a field of a map with the parameter HE of the Extended Field Editing screen of the Natural map editor.

```
HE= 'SYSHLPn',=
```

SYSHLP*n* is the help routine to be called where *n* is a number in the range from 1 to 10.

Note: See the section *Natural Session Parameters* of the *Natural Parameter Reference* documentation and section *Natural Help Routines* of the *Natural Programming Guide* for a detailed description of the HE parameter.

Help from Field Objects

Information from Predict field objects is retrieved with the help routines SYSHLP1 and SYSHLP4.

How to Specify the Field Object

File ID and field ID (separated by a period) of the Predict field object to be used must be specified in the input field Field of the Extended Field Editing screen of the map editor. For example: CUSTOMER.NAME for the field NAME in the file CUSTOMER.

SYSHELP moves the values to the variables FILENAME (A32) and FIELDNAME (A32) in the parameter data area SYSHELPA. If only a field and no file is specified, a default file ID in the help routine is used. When SYSHLP1 and SYSHLP4 are delivered, the default file ID is MISCELLANEOUS.

Using the default file ID is especially useful when providing online help information for fields that are not implemented in a file (for example fields that are used for navigating in an application) and the field and file ID cannot be selected from the DDM.

Help from Database, File Relation, System Objects

Information from these Predict objects is retrieved with the help routines SYSHLP7, SYSHLP8, SYSHLP9 and SYSHLP10.

How to Specify a Database, File Relation or System Object

The database, file relation, or system ID of the Predict object to be used must be specified in the input field Field of the Extended Field Editing screen of the map editor.

SYSHELP moves the value to the variable FILENAME (A32) in the parameter data area SYSHELPA.

Help from Verification Objects

Information from Predict Verification objects is retrieved with the help routines SYSHLP1, SYSHLP2, and SYSHLP3.

How to Specify the Verification Object

File ID and field ID (separated by a period) of the Predict field object must be specified in the input field Field of the Extended Field Editing screen of the map editor. If a field is specified without a file, a default file name in the help routine is used. See the respective description for field objects above.

Using SYSHLP2 to Display Verification Values for Selection

SYSHELP will display the abstract/description of all verifications of type T (table) linked to the Field.

The following rules apply when using verification values as online information:

- Verification values specified in the documentation are displayed. If values in a rule generated from a verification have been changed manually (using the rule editor) this has no impact on the online help derived from the verification.
- When selecting a verification value for a numeric field, the value must be converted from alpha-numeric to numeric. This can be done with the following statement:

```
IF HELP-TEXT1 (#HELP) IS (N27)
 COMPUTE MAP-VARIABLE=VAL(HELP-TEXT1(#HELP))
END-IF
```

- If Verification values for several Fields of different length are to be displayed, additional help routines have to be created. Copy SYSHLP2 (for example into a new member SYSHLP11) and assign the new help routine to the parameter HE.

- If Verification values are displayed for selection, line comments can be displayed as descriptive text for the values. These line comments are specified in the field Value of the Add/Modify Verification function and are preceded by /* (a slash and an asterisk). When you select a value, the line comments are not moved to the input field.

Using Screen Help

To use the abstract or extended description of a Predict program object for online screen help, specify SYSHLP5 (for abstract) or SYSHLP6 (for the description) in the field Help of the Define Map setting for MAP screen (the input screen for defining a map profile).

SYSHELP identifies by the implementation pointer the Predict object to be used. The program object with the map name and the library name of the map in the implementation pointer is used.

If no Predict object is found, the STEPLIB name of the map is used as search criterion instead of the library name of the map. If this again fails, the first Predict object with the same member name as the map name is used.

Interface to the SYSHELP Subprogram

The following parameters are passed when SYSHELP is called. These parameters are stored in the parameter data area SYSHELPA.

Parameter	Format	Output Input	Description
FIELDNAME	A32	I	Predict field object identifying the extended description/abstract/verification values to be read. Response code H'FF' will be received if the field does not exist in Predict. Synonyms of fields can be specified. For example: field name synonyms beginning with # can be used for customer-specific naming conventions.
FILENAME	A32	I	ID of the Predict object to provide the help information. If the field for which help information is required is not a field of a DDM (userview), it can, for example, be described together with other such fields in a special file, whose name can, for example, be the name of the application. If the file name is the name of a dataview that is defined in a data area for a userview, SYSHELP will search using XRef data for the name of the userview and will substitute that name.
VENAME	A32	O	The name of a verification that is linked to the field. If no verification is found, this parameter is blank. Information-type C must be retrieved first so that Verification-name contains either the name of the first verification or blanks. If information-type T, V or W are retrieved,

Parameter	Format	Output Input	Description
			response-code H'03' indicates that the verification-name contains either the name of the next verification or blanks.
VETYPE	A1	O	Only verifications of type T (table) will be evaluated.
INFTYPE	A1	I	The type of information to be read from Predict. The following values can be specified: C — Abstract of a database C — Abstract of a field of a file D — Extended description of a database E — Extended description of a field of a file F — Abstract of a file G — Extended description of a file P — Abstract of a program of type M (Map) R — Extended description of a file relation L — Abstract of a file relation Q — Extended description of a program of type M (Map) S — Abstract of a system T — Possible values of a verification V — Abstract of a verification of status automatic W — Extended desc. of a verification of status automatic Y — Extended description of a system
STARTLN-OLD	N4	I O	The first line of help information to be displayed. If no value is specified, display will start with the first line. After the call has been executed, this parameter contains the number of the next line of help information to be displayed. If information-type C or E is retrieved, response-code H'03' indicates that starting-line contains the value 1. If information-type V is retrieved and starting-line contains the value 0, SYSHELP searches for the next verification.
RSP	B1	O	Response code: H'00' — O.K. No error H'01' — A wrong type was passed to SYSHELP H'03' — No more information available for object H'04' — No more information of this type is available H'05' — SYSHELP was called without using parameter STARTLN but the description exceeds 9999 lines H'FF' — This object is not described in the dictionary.

Parameter	Format	Output Input	Description
HELPINF	A240	O	The help information taken from a Predict object. If information-type C is retrieved, this parameter contains up to 8 lines of abstract and/or verification values. If information-type E is retrieved, this parameter contains up to three lines of extended description. The help information must be translated into a help screen by the user's help routine which calls SYSHELP.
VE-INDEX			Verification indicator. For internal use only.
DESC-UNIQUE-ID			Internal ID of description. For internal use only.
STARTLN	N8	I O	Same as STARTLIN-OLD but overall number of lines in the description can be up to 99999999. This parameter is optional.

Support for the Structured Mode of Natural

Within a data area, file names can be used which are different from the names of the corresponding DDMs. SYSHELP searches for the correct file name using XRef data.

VII LIST XREF For Natural

In addition to documentation data, the Predict system file contains information on which members are actually implemented in an application and how these members are used (how they are actively referenced). Information on active references to members is stored in Predict XRef data. XRef data mirrors the 'real world': an application in its current status.

XRef data for Natural objects can be evaluated with functions of the Natural `LIST XREF` command. This 'command' is in fact a comprehensive utility offering a wide range of retrieval functions. The Natural `LIST XREF` utility is described in this section.

Other ways of evaluating XRef data are described under

- **Active Retrieval**
- **LIST XREF for Third Generation Languages**

See also the *XRef Evaluation documentation* for further information.

For a general description of how XRef data is created and used, see the section *XRef Data* in the *Introduction to Predict documentation*.

This section covers the following topics:

Creation of XRef Data for Natural

Maintaining XRef Data

Contents of XRef Data

Types of LIST XREF Functions

Steplib Support

Multilanguage Support

Calling LIST XREF Functions

Retrieving Information for Individual Members

Verify Application Integrity

Using Sets

Maintaining a LIST XREF Profile

LIST XREF Commands

54 Creation of XRef Data For Natural

XRef data for Natural is generated in the following cases:

- Natural writes XRef data for Natural programs and data areas when these are cataloged (provided that the `XREF` parameter has been set to either `ON` or `FORCE`, see below). This does not affect the performance of the program; only the cataloging time is insignificantly extended.
- Natural Security writes XRef data for programs that are used as Startup, Restart or Error-Transaction in an application or special link if the `XREF` parameter is set to `ON` or `FORCE` in the application's Natural Security definition and a user system file is defined for the application.
- The Natural Command Processor (NCP) writes XRef data for programs that are used as Startup, Restart or Error-Transaction in an application or special link if the `XREF` parameter is set to `ON` or `FORCE` in the application's NCP definition and a user system file is defined for the application.
- If Natural for DB2 is installed, the function `CREATE DBRM` of Natural for DB2 creates XRef data for Natural programs that use static SQL.

The Natural `XREF` parameter determines whether XRef data is created. Valid values:

`ON`	XRef data is created when a Natural object is cataloged.
`OFF`	XRef data is not created when a Natural object is cataloged.
`FORCE`	A Natural object can only be cataloged if a Predict documentation object exists for it. XRef data is then created when cataloging an object.
`DOC`	A Natural object can only be cataloged if a Predict documentation object exists for it. No XRef data is created.

The parameter XREF also determines how XRef data is treated when processing Natural members with the **SYSMAIN, NATUNLD or INPL utilities**.

There are different ways to set the Natural XREF parameter:

- In the Natural parameter module.

- As a dynamic parameter when starting a Natural session.
- In Natural Security. If Natural Security has been used to set the `XREF` parameter, the `XREF` command may only be used to enforce this setting (from `ON` to `FORCE`, from `OFF` to `ON` or `FORCE`).
- With the Natural `XREF` command. If Natural Security is not installed, the `XREF` parameter is usually set with the Natural `XREF` command. The Natural command `XREF ?` displays the current setting of the `XREF` parameter.

55 Maintaining XRef Data

- Using the Natural Utilities SYSOBJH (Object Handler), SYSMAIN, NATLOAD, NATUNLD or INPL 372
- Using the Natural Commands SCRATCH, UNCAT , DELETE, PURGE and RENAME 372
- Using Predict Special Functions 373
- Using the Predict Coordinator 373

XRef data is only meaningful in conjunction with the program from which it is derived. Therefore applications and commands used to copy, rename, move or delete members also permit the processing of XRef data along with the members.

The different methods of maintaining XRef data are listed below.

Using the Natural Utilities SYSOBJH (Object Handler), SYSMAIN, NATLOAD, NATUNLD or INPL

Using the Natural utilities SYSOBJH (Object Handler), SYSMAIN, NATLOAD, NATUNLD or INPL XRef data can be copied, renamed, moved or deleted along with Natural members. The parameter XREF then determines how XRef data is treated:

N	Process the program alone
Y	Process both the program and its XRef data only if XRef data exists for the program in the source library
F	Process both the program and its XRef data only if XRef data for the program exists in the source library, and then only if a data dictionary object for the program exists in the target library
S	Process any XRef data present

See the respective Natural documentation for more information.

If a program is replaced by another program, the XRef data for the old program is deleted.

Using the Natural Commands SCRATCH, UNCAT , DELETE, PURGE and RENAME

- If a Natural member is deleted with the `SCRATCH` (`UNCAT`, `DELETE` or `PURGE`) command, its XRef data is deleted along with the *object code*.
- If a Natural object code is renamed with the `RENAME` command, its XRef data is renamed along with the *object code*.

Using Predict Special Functions

XRef data documenting members that have been deleted can be deleted using functions from the Predict Maintain Active References menu, which is called from the Special Functions menu.

Using the Predict Coordinator

XRef data can be transferred using the Predict Coordinator. See the *Predict Coordinator* documentation for more information.

56 Contents of XRef Data

The following information is stored for each Natural program cataloged:

- References to programs that are invoked, how they are invoked (for example `CALLNAT`, `FETCH`, `FETCH RETURN`, `PERFORM`, `RUN`, `STACK`, `SEND EVENT`, `SEND METHOD`) and the type of program (such as: main program, subroutine, subprogram, help routine, map or data area);
- References to data areas and variables, with additional information on usage of variables: use, modification or use for dynamic source code creation (described in the Natural Statements documentation in relation to the `RUN` statement). Additionally, library, DBnr and Fnr of the referenced data areas are stored.
- References to views and data area views, with additional information about their usage (for example definition in a data area, delete, read, or update);
- References to fields of views, with additional information about their usage: definition in a data area, count, read, search or update. For 3GL: precompiled by Adabas SQL Server, Adabas Native SQL;
- References to files used in the program and the type of file use (deleting, reading or updating);
- References to Natural copy code (source code), library, DBnr and Fnr of the copy codes used are stored.
- References to maps and help routines used in the program;
- References to SQL stored procedures via the Natural statement `CALLDBPROC`;
- References to Natural error numbers used in the program and the types of error messages (system-wide or application-specific);
- References to Natural printers;
- References to Natural workfiles used in the program and the type of use (whether read from, written to or close);
- References to Natural classes;
- References to Natural methods;

- References to Natural interfaces;
- References to External resources;
- References to external programs invoked from a program;
- References to entry points or functions defined in the program which can be invoked by other programs;
- References to retained sets used in the program and the type of use (build, use, or release a retained set);
- References to processing rules used in a map and their type of use (automatic, free, or inline processing rules);
- Reference to a DBRM (DB2 request module) for programs using static SQL;
- Use of EXPERT Models via the Natural statement `INVESTIGATE`;
- Use of the Natural command processor via the statement `PROCESS COMMAND`;
- Statistical information:
  - Date and time the program was cataloged/precompiled;
  - ID of the user who cataloged/precompiled the program;
  - ID of the terminal from which the program was cataloged/precompiled (in batch mode: the job name);
- Language.

Note: In Predict, the term "External resource" refers to the Natural object type "Resource". Further information on this object type can be found in the section *Ressource* of *Objects for Natural Application Management* in the *Natural Programming Guide*.

57 Types of LIST XREF Functions

XRef data is retrieved with functions of the Natural `LIST XREF` command.

Functions provided by the Natural `LIST XREF` command can be grouped into three groups:

- Functions retrieving information on specific types of members (for example programs or data areas). All functions from Invoked programs through Report programs with xref data in the XRef Menu are of this type. See ***Retrieving Information for Individual Members***.
- Functions retrieving information on the consistency of an application as a whole. The functions are called from a submenu that is called with code A (Verify application). See ***Verifying the Integrity of an Application***.
- Functions that are used to manage sets. Sets allow efficient use of information retrieved from XRef data. See ***Using Sets***.

58 Steplib Support

- LIST XREF Main Menu 380
- Using a Structure Documented in Predict 383
- Using the Runtime Structure 383
- Working Without any Structure 384
- Displaying the Current Environment 385
- Steplib Support in Batch Mode 386
- Defining a Default Structure 386

Predict offers you three possible methods of evaluating XRef data listed below. Select the method you require in the `LIST XREF` menu before you call a function. This option is valid for the duration of your session or until you select another option in the `LIST XREF` menu.

- **The Library Structure Documented in Predict**
  Enter a valid library structure for parameter Structure. See ***Using a Structure Documented in Predict*** The structure to be used is taken from the link list Library structure > System.
- **The Runtime Structure**
  Enter *R for parameter Structure.
- **Without any Structure**
  The `LIST XREF` functions evaluate XRef data as before. Only objects in the current library are displayed. See **Working without any Structure**.

Notes:

1. If XRef data of programs contain library information about used data areas or copy codes, this information is evaluated regardless of the value specified for parameter Structure.
2. For invoked objects only the steplib information can be used to identify objects that reside on other libraries.
3. For passive requests the Active retrieval function **Members referenced by members** can be used to find all objects referencing an object in a central steplib regardless of any steplibs structures.

LIST XREF Main Menu

The LIST XREF Main Menu is invoked by entering `LIST XREF` after the Natural NEXT prompt. The command can be abbreviated to `L X` (separated by a blank). Alternatively, the XRef menu can be invoked in the Natural Development Facilities menu.

```
15:56:04            *****  P R E D I C T *****                      2009-09-02
Library: SYSDIC              -   Xref Menu   -          DBnr:   180 Fnr:    64


Code  Object                              Code  Object

  I   Invoked programs                      S   Retained sets
  D   Data areas and variables              R   Processing rules
  V   Views and fields                      F   External resources
  C   Copycode                              X   Report programs with xref data
  E   Error numbers                         A   Verify application
  P   Printers                              N   Create new sets via selection
  W   Work files                            O   Operate on sets


      Code .........:   (  ? Help  . Terminate )
      Structure ...*:
      System .......:


Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
     Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Code

Enter one of the function codes offered in the menu. Before a function is called, a structure can be entered.

Structure

The structure to be used to evaluate the XRef data. Enter the ID of a library structure, or

blank	No library structure. Only objects in the current library will be evaluated
***R**	The runtime structure will be used.
*	The following selection window appears:

```
  E D I C T  *****              2007-05-31
    XRef Menu   -          DBnr:   180 Fnr:    54

          +-------------------------------------+
          !     *** without structure ***        !
  ------  !     *** run time structure ***       !
          !     LS-NEWDICLX                      !
          !     LS-PDLX                          !
          !     PD-COB                           !
          !                                      !
```

In this window you can specify that

- no library structure is to be used for evaluating XRef data
- the **runtime structure** is to be used.
- a library structure defined in Predict is to be used. A list of all library structures in Predict where the system which documents the current library is contained in the link list Library Structure > System is displayed.

See Library Structure in the *Predefined Object Types in Predict* documentation.

Notes:

1. The first two options are always available. In order to select a library structure, at least two objects must be present:
   - a system object that documents the current library
   - a library structure with a link list that contains this system object.
2. If you set the option Save set to Y, the objects from different libraries will be saved in different sets.
3. If XRef data of programs contain library information about data areas or copy codes used, this information is evaluated regardless of the value specified for parameter Structure.

See examples given in **Effects of Steplib Support**.

System

This field shows the ID of the system object in Predict that documents the library. This value is taken from the implementation pointer of the system object. The system with the most detailed implementation pointer is used, see table.

	LIBRARY	FNR	DBNR
1	x	x	x
2	x	x	
3	x		

Example: An implementation pointer consisting of Library/Fnr/DBnr is used before a pointer consisting of Library/Fnr only.

Using a Structure Documented in Predict

The link list Library structure-System of the library structure specified must contain the system displayed in field **System**.

The link list Library structure-System is evaluated. Each system in the list is checked as follows:

- If no information is present in the implementation pointer of the system, the system is ignored
- If the implementation pointer is incomplete, the system searches for possible XRef data. This XRef data is used to supply the missing Fnr and DBnr information in the implementation pointer.
- If no XRef data is found, the values of the current FUSER file are used to supply the missing DBnr and Fnr information in the implementation pointer.
- If the current library is a Natural library, the structure is appended with:

  --> *STEPLIB <--.

  SYSTEM FUSER (only if different to *STEPLIB)

  SYSTEM FNAT

Note: A maximum of 20 steplibs is supported.

See the section *Library Structure* in the *Predefined Object Types in Predict* documentation.

Using the Runtime Structure

The following screen appears if you enter *R for parameter **Structure**.

```
 11:20:01            *****  P R E D I C T *****                      2007-05-31
 Library: V461             - Library Structure -         DBnr:   180 Fnr:    64
 Struct.: *** run time structure ***

 Structure ...........: *** run time structure ***


    Nr System Id                          Library  DBnr  Fnr
    -- -------------------------------- -------- ----- -----
-->  1                                   V461       180    64
     2                                   V452LX     180    64
     3                                   V452       180    64
     4 V451LX                            V451LX     180    64
     5 V451                              V451       180    64
     6                                   V342EN     180    64
     7                                   SYSTEM     180   104
     8                                   SYSLIBS    180   104
     9 --> *STEPLIB <--                  SYSTEM     180    64
    10 -->  SYSTEM  FUSER  <--           SYSTEM     180    54

Store setting as Library Structure ...: N (Y/N)
Use Library Structure in reverse order ..: N (Y/N)
```

Working Without any Structure

If you do not specify any library structure or runtime structure, only objects in the current library will be evaluated.

```
    13:25:27            *****  P R E D I C T  *****              2007-05-31
    Library: SYSDIC              -   XRef Menu   -          DBnr:   180 Fnr:    64


    Structure ...........:


                System Id                          Library   DBnr   Fnr
                --------------------------------  --------  -----  -----
                                                  SYSDIC    180    64
```

Displaying the Current Environment

If there is any Structure used (not *** without structure ***), this structure is shown on any LIST XREF menu in the header as shown below.

If the structure is reverted, the header contains a remark indicating this.

```
10:59:37            *****  P R E D I C T  *****                      2007-05-31
Library: SYSDICLX          - Invoked Programs -         DBnr:   180 Fnr:   104
Struct.: *** run time structure ***           reverse order                    ↩
```

Using the command `INFO` you can display at any time during your LIST XREF session all libraries that will be evaluated by LIST XREF functions. The current library is marked with an arrow. In addition, you can save a runtime stucture and can then toggle between reverse and notreverse order.

See example below.

```
13:28:05            *****  P R E D I C T  *****              2007-05-31
Library: SYSDIC              -   XRef Menu   -          DBnr:   180 Fnr:    64


Structure ...........: SYSDIC


          System Id                          Library   DBnr   Fnr
          --------------------------------   --------  -----  -----
     -->                                     SYSDIC    180    64
          V4210BE                            V4210BE   180    64
          --> *STEPLIB <--                   SYSTEM    180    64



Store setting as Library Structure ......: N  (Y/N)
Use Library Structure in reverse order ..: Y  (Y/N)
```

Steplib Support in Batch Mode

In batch mode, the library can be set via various methods:

- If Natural Security is active, the logon user ID is used to identify the LIST XREF User Profile.
- If no LIST XREF User Profile for the logon user ID exists or Natural Security is not active, the LIST XREF Default Profile gets active.

These Profiles may contain a default library structure, which is set if it is valid for the current logon library. This is reported via a appropriate message.

In addition, the command STRUCTURE is used to specify which library structure is to be used. Also the command `PROFILE <user>` activates a library structure specified in the LIST XREF User Profile, if it is valid for the actual logon library.

Defining a Default Structure

A default library structure can either be defined in the LIST XREF User Profile for individual users or in the LIST XREF Default Profile for users without own profile and batch mode users in non-security environments. See ***Maintaining a LIST XREF Profile*** in the section LIST XREF for Natural in the *Predict Reference* documentation and LIST XREF Default Profile in the section Predict Defaults in the *Predict Administration* documentation.

59 Multilanguage Support

Predict also supports objects which contain an ampersand (&) to identify the language code. For example, MENU&, where & is replaced with the language code for *LANGUAGE.

The position of the ampersand character is determined by the parameter Position of '&' with the function General defaults > Miscellaneous.

60 Calling List XRef Functions

- Processing Several Objects in One Call 390
- Help Information 390
- Using the LIST XREF Profile 390

Many functions of the Natural `LIST XREF` command can be called from a menu or with a command. See the syntax and parameters of ***LIST XREF direct commands***.

The XRef Menu (shown above) is invoked by entering `LIST XREF` after the Natural NEXT prompt. The command can be abbreviated to `L X` (separated by a blank). Alternatively, the XRef menu can be invoked in the Natural Development Facilities menu.

All `LIST XREF` menus include a command input field, and the system displays the next appropriate menu after successful execution of a direct command; thus it is always possible to enter `LIST XREF` commands in either way. Any Natural command can be entered in the command input field; its use is not limited to `LIST XREF` commands.

Processing Several Objects in One Call

With the functions that retrieve information on individual objects, single objects or several objects can be processed in one call. If several objects are to be processed, selection criteria can be specified to determine the range of objects.

Help Information

The Natural `LIST XREF` command offers different forms of online help:

- Help on the meaning of a function. This form of help is displayed by entering a question mark (?) in the Code field of any menu. The XREF help menu is then displayed. Marking one of the topics and pressing ENTER causes Natural to display information on the respective function.
- Help information about direct commands. This form of help is displayed by entering a question mark (?) in the Command===> field. The XREF command help menu is then displayed. Marking one of the topics and pressing ENTER causes Natural to display information on the respective direct command.

Using the LIST XREF Profile

The way functions work can be changed with parameters set in the `LIST XREF` session profile. For example: Functions that display menus of the Natural `LIST XREF` command can be assigned to some of the program function (PF) keys. See **Maintaining a LIST XREF Profile** .

61 Retrieving Information For Individual Members

- Two Ways to Retrieve Information on Individual Members 392
- Information about Indirect Calls 393
- Which Invocation Methods are Evaluated 393
- Invoked Programs 395
- Data Areas and Variables 401
- Views and Their Fields 404
- Copy Code 407
- Error Numbers 409
- Printers 411
- Workfiles 412
- Retained Sets 414
- Processing Rules 415
- External Resources 417
- Report Programs with XRef Data 419

Two Ways to Retrieve Information on Individual Members

Information on individual Natural members is either retrieved top-down or bottom-up. See also diagrams below. In both cases the retrieval starts with an individual member (program, data area, etc.):

- **Top-down Retrieval**
  The functions look for objects that are used by a given member.

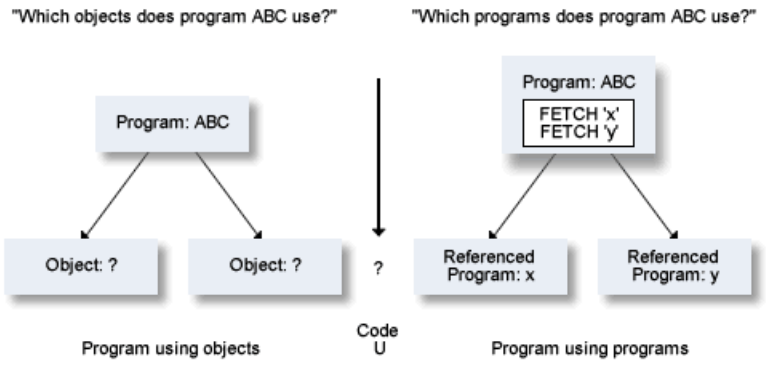
For example: The function Program using programs. A program is specified and all programs that are used by this program are shown.

- **Bottom-up Retrieval**
  The functions look for objects that use a given member. The notation referenced in is always used in retrieval functions of this type. For example: The function Program referenced in programs. A program is specified and all programs that use this program are listed.

Note: In the screens Data Areas and Variables and Copy Code only starting objects in the current library are taken into account.

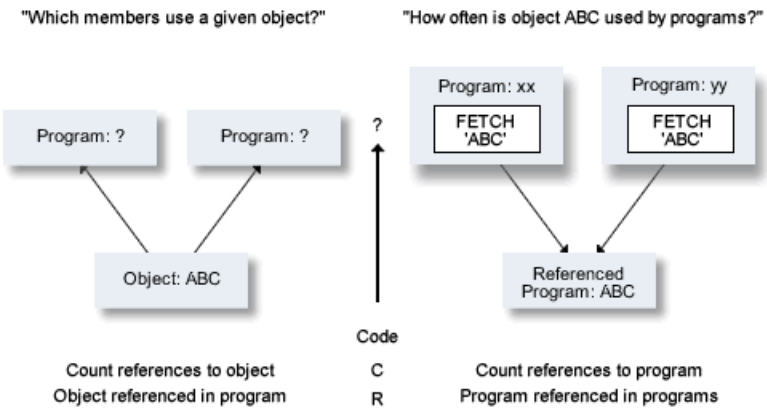
Top-Down: Finding Objects that a Given Member is Using

Top-down retrieval provides answers to question such as these:



Bottom-Up: Finding Members that Reference a Given Object

Bottom-up retrieval provides answers to question such as these:



Information about Indirect Calls

Indirect references can be evaluated, for example, “Which programs refer to programs which refer to the program ABC?” Recursion can be extended up to seven levels to show the invocation structure of an application.

Which Invocation Methods are Evaluated

When Natural finds in a program a reference to another program, it stores not only the type and name of the program that is used but also how that program is invoked. The following invocation methods are reported by LIST XREF functions:

- CALL statements to 3GL programs;
- CALLNAT statements;
- FETCH statements;
- FETCH RETURN statements;
- Assignment of a program to a function key (using a SET KEY PFn= statement);
- INPUT USING MAP statements;
- CALLDBPROC statements to SQL stored procedures;
- Function call to Natural Functions;

- REINPUT USING HELP statements or an assignment via session parameter HE in an INPUT statement or within a map;
- PERFORM statements;
- RUN statements;
- STACK statements;
- CREATE OBJECT statements;
- SEND METHOD statements;
- Open dialog;
- SEND EVENT statements;
- Using property;
- Method implementation;
- Calling an entry in a DBRM (database request module) - only for Natural programs that use static SQL;
- Assignment of a program as an error transaction, a startup transaction or a restart transaction:
- either by assignment in a Natural program (using a *ERROR-TA= or *STARTUP= statement)
- or by definition in Natural Security for a Natural library or a special link to this library, provided that XREF for the application is set to either Y or F in Natural Security and that a user system file has been assigned to the library. These transactions are considered to be invoked by a dummy member *NSC (a program of type P) in the library.
- INVESTIGATE statements (invocation of a Natural Expert model).

Invoked Programs

```
 15:59:18            *****  P R E D I C T *****                        2009-09-02
 Library: SYSDIC            - Invoked Programs -         DBnr:   180 Fnr:    64


                    Code Function

                      D  Count references in program
                      C  Count references to program
                      U  Program using programs
                      R  Program referenced in programs
                      E  Program using programs recursively
                      I  Program referenced in programs recursively
                      X  Xref program using programs
                      Y  Xref program referenced in programs
                Code: ?             Save set: N (Y,N)
       Using program:           Program type:   (?)
  Referenced program:           Program type:   (?) via:                (?)
Subroutine/Entry/Class/Method/Property/Event:
          Rec. depth: 7  (1-7)
          Set number:    (?)
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

The functions display active reference information about programs that are invoked from Natural programs. The input screen is displayed with code I in the XRef menu.

Code	Function	Answer to the question	Comments
D	Count references in program	How many programs does the program ABC refer to?	The total number of referenced programs is displayed and is listed for each invocation method
C	Count references to program	How many times is the program ABC referred to by programs	The total number of references is listed for each invocation method.
U	Program using programs	Which programs does the program ABC refer to	The invocation method is indicated. See also *Effects of Steplib Support*.
R	Program referenced in programs	Which programs refer to the program ABC	The invocation method is indicated. See also *Effects of Steplib Support*.
E	Program using programs recursively	Which programs are referred to by the program ABC, or which programs are referred to by programs which are referred to by the program ABC, ...	The output is structured to show the direct references to programs on the next level. See also *Termination of Listing*.

Code	Function	Answer to the question	Comments
I	Program referenced in programs recursively	Which programs refer to the program ABC, or which programs refer to programs which refer to the program ABC, ...	The output is structured to show the direct references to programs on the next level. See also *Termination of Listing*.
X	XREF program using programs	Which programs are referred to by the program ABC, or which programs are referred to by programs which are referred to by the program ABC, ...	The total number of programs at each level is reported as well as the type and ID of each program. This function summarizes the output of the function with code E but omitsthe structural information.
Y	XREF program referenced in programs	Which programs refer to the program ABC, or which programs refer to programs which refer to the program ABC, ...	The total number of programs at each level is reported as well as the type and ID of each program. This function summarizes the output of the function with code I but omits the structural information.

Termination of Listing

In certain circumstances, the listing of a LIST XREF function is terminated and marked as follows:

- Recursion is marked with *rec* and the output is terminated.
- The output can be truncated from a level (up to 7) specified by the user. Truncated output is marked with trunc.
- The user can specify other parameters that limit the scope of the output, for example programs of a certain type or those with IDs that start with a particular character string. Output terminated in such a manner or due to a reference to a 3GL program is marked with stop.
- If a program that has already been displayed in the invocation hierarchy reappears at a lower recursion level, the hierarchy is not displayed again. Instead the program is marked with suppr. (for suppressed).
- If the called program is contained in another library, the output is stopped and the program is marked with steplib.

Save set	Any user defined in Predict can save the types and names of programs listed by functions U, R, X or Y in a set. See **Using Sets**. **Note:** Sets only contain the resulting objects of the called function.
Using program	The meaning of this parameter depends on the function: ■ For the functions D, U, E and X, the ID of the program for which the function is to be executed. ■ For the functions C, R, I and Y, the program or programs that use the program for which the function is to be executed.

	Asterisk notation is possible. **Note:** Enter *NSC to list members that are assigned in Natural Security as startup, restart, end or error transactions.	
Program type	The type of the Natural program specified as the Using program. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	Main program
	S	subroutine
	T	dialog
	Y	Natural Expert model
	O	command processor
	4	class
Referenced program	The meaning of this parameter depends on the function: ■ For the functions U, E and X: the ID of the programs that are used by the program for which the function is executed. ■ For the functions C, R, I and Y: the ID of the program for which the function is to be executed. If asterisk notation is used, the function is applied to all programs which meet the given specification. **Note:** Enter *DYNAMIC to list all programs that invoke other programs via variables. An example: ASSIGN #A = 'SUB1' CALLNAT #A	
Program type	The type of the program specified as the Referenced program. See the list of program types above.	
Via	The method used to invoke programs. Valid values:	
	C	CALL statement
	CN	CALLNAT statement
	CP	Command processor
	CR	CREATE OBJECT
	F	FETCH statement
	FR	FETCH RETURN statement
	H	REINPUT USING HELP statement or HE= assignment
	I	INVESTIGATE statement

	IR	Invoke return (stands for all methods that return control to the invoking program: CALL, CALLNAT, FETCH RETURN, Help, INVESTIGATE, Map, PERFORM and Static SQL).
	K	SET KEY statement
	M	INPUT USING MAP statement
	ME	Method implementation
	O	Open dialog
	P	PERFORM statement
	PR	Property
	R	RUN statement
	S	STACK statement
	SE	Send event
	SM	Send method
	SQ	Using static SQL
	T	Assignment to a Natural Security transaction
	U	Function
Subroutine/Entry/Class/ Method/Property/Event:	One of the following: ■ the function of a Natural subroutine that is performed, or ■ the point of entry into an external (non-Natural) program or DBRM that is called. ■ the class, method or property name. This parameter applies when a program is invoked without using its member name. To identify the (non-Natural) program to which an external entry belongs, Predict uses information from Adabas Native SQL or the Predict Preprocessor. To identify the DBRM to which an external entry belongs, Predict uses information created by the function CREATE DBRM of Natural DB2.	
Rec. depth	The recursion depth for the functions E, I, X and Y. A number from 1 to 7.	
Set number	If a set number is specified, XRef data is only evaluated for those objects defined in the given set. All other search criteria are then ignored. Set numbers can not be used as input for Count functions.	

Effects of Steplib Support

Steplib support affects Invoked Programs functions as follows. A distinction is made between Top-down and Bottom-up functions:

Top-down

Example: Function Program using program

```
09:50:43            *****  P R E D I C T  *****              2007-05-31
Library: PDLX              - Invoked Programs -         DBnr:   180 Fnr:    54
Command: PROG * (*) USING PROG * (*) WITH * VIA *                   Page:    1

     T:Program                   using                          via
-----------------------------------------------------------------------------
   1 P:ZPDFIELD                1 N:N-BUFEDT <<- nfnd            Callnat

   2 P:ZPDP0                   1 P:ZPDP1                        Fetch
                               2 S:ZPDS1  (NEWDICLX,180,54)     Perform
                                 Function: SUB-IN-ZPDS
```

Comments :

Only programs in the current library that call other programs are displayed.

to 1	The note <<- nfnd means that the called program was not found within the structure specified. "Not found" in this context means that no XRef data is present.
to 2	If the called program is not contained in the current library, Library, DBnr, Fnr are displayed in parentheses.

Bottom-up

Example: Function Programs referenced in programs

```
09:53:09            *****  P R E D I C T  *****              2007-05-31
Library: PDLX              - Invoked Programs -         DBnr:   180 Fnr:    54
Command: PROG * (*) REF PROG * (*) WITH * VIA *                    Page:    1

     T:Program                  referenced in                 via
-------------------------------------------------------------------------------
   1 ?:*DYNAMIC
                              1 P:ZPDP3                       Fetch

   2 ?:N-BUFEDT
                              1 P:ZPDFIELD                    Callnat

   3 M:ZPDM1
                              1 P:ZPDP1                       Map
                              2 P:ZPDP2  (NEWDIC,180,54)      Map

   4 P:PGMCO002 (*SYSCOB*,255,255)
      Entry   : PGMCO002
                              1 P:ZPDP1                       Call
                              2 P:ZPDP2                       Call
                              3 S:ZPDS1                       Call
```

Comments:

to 1	*DYNAMIC produces a list of all programs that call up other programs by means of variables: ASSIGN #A = 'SUB1' FETCH #A
to 2	The question mark means that the program N-BUFEDT was not found within the specified structure. "Not found" in this context means that no XRef data was found for the program object. The program is, however, referenced by program P:ZPDFIELD via CALLNAT.
to 3	Member ZPDM1 was found within the current library. If the called program is contained in the current library (here ZPDM1), programs not contained in the current library that call this program are also displayed (here ZPDP2 in Library NEWDIC). Library, DBnr, Fnr are displayed in parentheses.
to 4	The called program (PGMCO002) was found, but in another library within the structure (library *SYSCOB*). In this case only calling programs within the current library are displayed.

Example: Function Program referenced in programs recursively

```
13:44:35            *****  P R E D I C T  *****              2007-05-31
 Library: NEWDICLX          - Invoked Programs -         DBnr:   180 Fnr:    54
 Command: PROG XHMENU10 (*) REF REC * (*) WITH * VIA *               Page:    1
          DEPTH 7
      1 M:XHMENU10
 1 ---------2 ---------3 ---------4 ---------5 ---------6 ---------7 ---------
 P:XPHELP   P:XPCIMPL  P:XPVERI   M:XMCIMP00 P:XPCIMPL  <--- rec
                                  M:XMREFE00 P:XPREFE   P:XPVERI   <--- rec
                                  P:XPCIMPL  <--- rec
                                  P:XPCMDP   M:XMCIMP00 <--- suppr
                                             M:XMCONS00 P:XPVCONS  P:XPVERI
                                                                   <--- rec
                                             M:XMCOPY00 P:XPCOPY   P:XPMENU
                                                                  <--- steplib
```

Comments:

The note <--steplib means that the evaluation was stopped at this point because the called program is contained in another library.

Data Areas and Variables

The functions display active reference information about data areas and variables. The input screen is displayed with code D in the XRef menu.

```
 16:25:15            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC         - Data Area and Variables -     DBnr:   180 Fnr:    64


                   Code Function

                     C  Count references to variable
                     U  Program using variables
                     R  Variable referenced in programs
                     D  Program using data areas
                     E  Data area referenced in programs

               Code: ?              Save set: N  (Y,N)
            Program:            Program type:    (?)
          Data area:          Data area type:    (?)
         Data block:
          Structure:
           Variable:
              Usage:    (?)
         Set number:  1 (?)
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Note: All functions use objects contained in the current library as starting objects. In the reports, on the left side only objects in the current library are listed, regardless of whether or not a library structure was specified.

Code	Function	Answer to the question...
C	Count references to variable	How many times is the variable ABC (contained in the current library) referred to by programs (contained in any library)
U	Program using variables	Which variables (contained in any library) does the program ABC (contained in the current library) refer to
R	Variable referenced in programs	Which programs (contained in any library) refer to the variable ABC (contained in the current library)
D	Program using data areas	Which data areas (contained in any library) does the program ABC (contained in the current library) refer to
E	Data area referenced in programs	Which programs (contained in any library) refer to the data area ABC (contained in the current library)

Effects of Steplib Support

Steplib support affects the above Data Area and Variable functions as follows.

For programs cataloged using Natural 4.1.1 or above on a Windows or Linux platform or Natural 3.1.2 or above on the mainframe, additional XRef information about each referenced data area is stored: the library name, database number and file number. This information is used to determine the referenced data areas/variables. In addition, the specified library structure is used to determine the data areas/variables that will be used when objects will be cataloged a new.

For programs cataloged using an older version of Natural the additional XRef information is not available, therefore the specified library structure is used to determine the referenced data areas/variables.

Parameters Limiting the Scope of the Functions		
Save set	Any user defined in Predict can save the types and names of programs listed by functions R or E in one or more sets (one set per library). See **Using Sets**. Using function D, only types and names of the data areas will be included in the set(s).	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
	*, blank	All
Data area	The data area or data areas that are used by the program selected above. Asterisk notation is possible. **Note:** Enter *DYNAMIC to list all area independent variables (AIV).	
Data area type	Type of data area used by the program. One of the following three values can be specified:	
	G	Global data area
	L	Local data area
	P	Parameter data area
Data block	The data block used by the program.	

Parameters Limiting the Scope of the Functions		
Structure	For fields contained in a group, the group name has to be specified in addition to the field name to ensure uniqueness. For example, a group is defined as follows: 1 FULL-NAME 2 FIRST-NAME 2 LAST-NAME To identify the variable LAST-NAME, the field name LAST-NAME has to be specified in the field Variable and the group name FULL-NAME has to be specified in the field Structure.	
Variable	The variable used by the program. **Note:** A counter field in a data area view has a name beginning with C*. Enter C** to specify all counter fields.	
Usage	Limits the function to variables that are used in a particular way:	
	M	Modified by the program
	U	Used but not modified by the program
	S	Used for dynamic source code creation (see description of the RUN statement in the Natural Statements documentation)
		If nothing is specified, then all variables are shown.
Set number	If a set number is specified, XRef data is only evaluated for those objects defined in the given set. All other search criteria are then ignored. Set numbers can not be used as input for Count functions.	

Views and Their Fields

The functions display active reference information about views and fields. The input screen is displayed with code V in the XRef menu.

```
 16:47:51            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC          -    Views and Fields   -      DBnr:   180 Fnr:    64


                 Code Function

                   C  Count references to view
                   D  Count references to field of view
                   U  Program using views
                   R  View referenced in programs
                   F  Program using fields of views
                   G  Field of view referenced in programs
                   A  Program using data area views
                   B  Data area view referenced in programs
             Code: ?                                   Save set: N (Y,N)
          Program:                                  Program type:   (?)
             View:
   Data area view:                                    View usage:   (?)
            Field:                                   Field usage:   (?)
       Set number:  1 (?)
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Note: In this section only, the term program applies to global and local data areas (types G and L) as well as to programs (types H, M, N, P and S).

Code	Function	Answer to the question...
C	Count references to view	How many times is the view ABC referred to by programs
D	Count references to field of view	How many times is the field ABC referred to by programs
U	Program using views	Which views does the program ABC refer to
R	View referenced in programs	Which programs refer to the view ABC
F	Program using fields of views	Which views does the program ABC refer to, and which fields of these views does it refer to
G	Field of view referenced in programs	Which programs refer to the field ABC
A	Program using data area views	Which data area views does program ABC refer to
B	Data area view referenced in programs	Which programs refer to data area view ABC

Parameters Limiting the Scope of the Functions	
Save set	Any user defined in Predict can save the types and names of programs listed by functions R, G or B in one or more sets (one set per library). See **Using Sets**.
Program	The ID of the program (or data area) for which the function is to be executed. Asterisk notation is possible.
Program type	The type of Natural program (or data area) for which the function is to be executed. One of the following values can be specified:

Parameters Limiting the Scope of the Functions		
	F	function
	G	global data area
	H	help routine
	K	ISPF Macro
	L	local data area
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	*, blank	All
View	The view (DDMs) that are used by the program (or data area). Asterisk notation is possible.	
Data area view	The data area view(s) used by the program or data area. Asterisk notation is possible. **Note:** Specify the string *DDM if direct use of a standard views (not via a data area view) in reporting mode is to be reported.	
View usage	How the view is used by the program (or data area). One of the following values can be specified:	
	A	Defined in a data area
	D	Deleted
	M	Modified (either deleted, updated or stored)
	N	Stored
	O	Only read (neither deleted, updated nor stored)
	P	Database addr.
	R	Read, deleted or updated
	T	Coupled search
	U	Updated
	*, blank	All
	Usage A is valid only for data areas. All other values are valid only for programs.	
Field	The field or fields used by the program (or data area). Asterisk notation is possible.	
Field usage	How the field is used by the program (or data area):	
	A	Defined in a data area
	C	Used as a counter field
	H	Used in the WITH clause of a search criterion as hyperdescriptor search
	L	Used in the WITH clause of a search criterion as non descriptor search
	M	Stored or updated

Parameters Limiting the Scope of the Functions		
	N	Stored
	O	Either used as a counter field or read (but not modified)
	R	Updated, used as a counter field or read
	S	Used in the WITH clause of a search criterion
	T	Used in the COUPLED clause of a search criterion
	U	Updated
	*, blank	All
	Note: Usage A is valid only for data areas. All other values are valid only for programs.	
Set number	If a set number is specified, XRef data is only evaluated for those objects defined in the given set. All other search criteria are then ignored. Set numbers can not be used as input for Count functions.	

Copy Code

The functions display active reference information about Natural copy code (source code). The input screen is displayed with C in the XRef menu.

```
 16:48:49            *****  P R E D I C T *****                        2009-09-02
 Library: SYSDIC                - Copycode -             DBnr:   180 Fnr:    64



                    Code Function

                      C  Count references to copycode
                      U  Program using copycodes
                      R  Copycode referenced in programs

                Code: ?              Save set: N (Y,N)
             Program:            Program type:   (?)
           Copycode:                   Usage:   (?)
          Set number:  1 (?)




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Note: All functions use objects contained in the current library as starting objects. In the reports, on the left side only objects in the current library are listed, regardless of whether or not a library structure was specified.

Code	Function	Answer to the question...
C	Count references to copy code	How many times is the copy code ABC (contained in the current library) referred to by programs (contained in any library)
U	Program using copy codes	Which copy codes (contained in any library) does the program ABC (contained in the current library) refer to
R	Copy code referenced in programs	Which programs (contained in any library) refer to the copy code ABC (contained in the current library)

Effects of Steplib Support

Steplib support affects the above Copy Code functions as follows.

For programs cataloged using Natural 4.1.1 or above on a Windows or Linux platform or Natural 4.1.1 or above on the mainframe, additional XRef information about each referenced copy code is stored: the library name, database number and file number. This information is used to determine the referenced copy codes. In addition, the specified library structure is used to determine the copy codes that will be used when objects will be cataloged a new.

For programs cataloged using an older version of Natural the additional XRef information is not available, therefore the specified library structure is used to determine the referenced copy codes.

Parameters Limiting the Scope of the Functions		
Save set	Any user defined in Predict can save the types and names of programs listed by functions U or R in one or more sets. See **Using Sets**.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
	*, blank	All

Parameters Limiting the Scope of the Functions		
Copy Code	Only copy code with IDs beginning with a particular sequence of characters is included in the report.	
Usage	Usage Only copy code that is used by programs in a particular way is included in the report. The following values can be specified:	
	D	Direct use of copy code
	I	Indirect use of copy code
	*, blank	Both types of use
Set number	If a set number is specified, XRef data is only evaluated for those objects defined in the given set. All other search criteria are then ignored. Set numbers can not be used as input for Count functions.	

Error Numbers

The functions display active reference information about Natural error numbers. The input screen is displayed with code E in the XRef menu.

```
 16:49:37            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC              - Error Numbers -          DBnr:   180 Fnr:    64



                  Code Function

                    C  Count references to error number
                    U  Program using error numbers
                    R  Error number referenced in programs

              Code: ?               Save set: N (Y,N)
           Program:             Program type:   (?)
      Error number:      to:      Error type:   (?)
        Set number:  1 (?)





Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Code	Function	Answer to the question...
C	Count references to error number	How many times is the error number ABC referred to by programs
U	Program using error numbers	Which error numbers does the program ABC refer to
R	Error number referenced in programs	Which programs refer to the error number ABC

Parameters Limiting the Scope of the Functions		
Save set	Any user defined in Predict can save the types and names of programs listed by function R in one or more sets. See **Using Sets**.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
	*, blank	All
Error number	Only error messages in the range of error numbers are included in the report. Finding error messages that are used dynamically: Error number *DYN as input or output indicates an error number that is used dynamically, as in the following example: VAR1(N4) = 1000 REINPUT WITH TEXT *VAR1	
Error type	Only error messages of the given type are included in the report. The following values can be specified:	
	S	Natural system messages
	U	Error messages defined in SYSDIC
	*, blank	Both types
Set number	If a set number is specified, XRef data is only evaluated for those objects defined in the given set. All other search criteria are then ignored. Set numbers can not be used as input for Count functions.	

Printers

The functions display active reference information about printers which are assigned in the OUTPUT statement of a DEFINE PRINTER statement. The input screen is displayed with code P in the XRef menu.

```
 16:50:16            *****  P R E D I C T *****                        2009-09-02
 Library: SYSDIC                -  Printers -            DBnr:   180 Fnr:    64



                     Code Function

                       U  Program using printers
                       R  Printer referenced in programs

                 Code: ?                Save set: N (Y,N)
              Program:              Program type:   (?)
       Printer number:    to:       Printer name:
           Set number:  1 (?)





Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Code	Function	Answer to the question...
U	Program using printers	Which printers does the program ABC refer to
R	Printer referenced in programs	Which programs refer to the printer ABC

Parameters		
Save set	Any user defined in Predict can save the types and names of programs listed by function R in one or more sets. See **Using Sets**.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map

Parameters		
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
	*, blank	All
Printer name	Only printers that meet the given name are included in the report. Asterisk notation is possible.	
Printer number	Only printers with numbers in the specified range are included in the report.	
Set number	If a set number is specified, XRef data is only evaluated for those objects defined in the given set. All other search criteria are then ignored.	

Workfiles

The functions display active reference information about Natural workfiles. The input screen is displayed with code W in the XRef menu.

```
 16:51:39            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC               - Work Files -            DBnr:   180 Fnr:    64



                     Code Function

                       U  Program using work files
                       R  Work file referenced in programs

                Code: ?             Save set: N (Y,N)
             Program:           Program type:   (?)
    Work file number:    to:           Usage:   (?)
          Set number:  1 (?)





Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Code	Function	Answer to the question...
U	Program using workfiles	Which workfiles does the program ABC refer to
R	Workfile referenced in programs	Which programs refer to the workfile ABC

Parameters		
Save set	Any user defined in Predict can save the types and names of programs listed by function R in one or more sets. See **Using Sets**.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
	*, blank	All
Workfile number	Only workfiles with numbers in the specified range are included in the report.	
Usage	Only workfiles used in a particular way are included in the report:	
	R	Read (statement READ WORKFILE)
	W	Write (statement WRITE WORKFILE)
	P	Write PC command (statement WRITE PC FILE n COMMAND)
	C	Close workfile (statement CLOSE WORKFILE)
Set number	If a set number is specified, XRef data is only evaluated for those objects defined in the given set. All other search criteria are then ignored.	

Retained Sets

The functions display active reference information about retained sets (as defined in the RETAIN clause of a Natural FIND statement). The input screen is displayed with code S in the XRef menu.

```
 16:52:27            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC             -  Retained Sets  -         DBnr:   180 Fnr:    64



                  Code Function

                    U  Program using retained sets
                    R  Retained set referenced in programs

              Code: ?               Save set: N (Y,N)
           Program:             Program type:   (?)
      Retained set:
             Usage:    (?)
        Set number:  1 (?)




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Code	Function	Answer to the question...
U	Program using retained sets	Which retained sets does the program ABC refer to
R	Retained set referenced in programs	Which programs refer to the retained set ABC

Parameters		
Save set	Any user defined in Predict can save the types and names of programs listed by function R in one or more sets. See **Using Sets**.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map

Parameters		
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
	*, blank	All
Retained sets	Only retained sets that meet the given name will be included in the report. Asterisk notation is possible. **Note:** Enter *DYNAMIC to list all programs that handle sets via variables. Examples: ■ FIND ... RETAIN AS #SET-NAME ■ RELEASE SET #SET-NAME **Note:** Enter *ALL to list all programs that handle all existing sets. Example: ■ RELEASE SET	
Usage	Only retained sets used in a particular way are included in the report:	
	B	Sets built by FIND ... WITH ... RETAIN AS 'SET' statements.
	R	Sets released by RELEASE SET 'SET' or RELEASE SETS statements.
	U	Sets used by FIND ... WITH 'SET' statements.
	*, blank	All
Set number	If a set number is specified, XRef data is only evaluated for those objects defined in the given set. All other search criteria are then ignored.	

Processing Rules

The functions display active reference information about processing rules. The input screen is displayed with code R in the XRef menu.

```
 16:53:11            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC           -  Processing Rules  -        DBnr:   180 Fnr:    64



                        Code Function

                          C  Count references to rule
                          U  Map using rules
                          R  Rule referenced in maps

                     Code: ?              Save set: N (Y,N)
                      Map:
                     Rule:
               Rule status:    (?)       Rule usage:   (?)
                Set number:  1 (?)




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Code	Function	Answer to the question...
C	Count references to rule	How many times is the rule ABC referred to by cataloged maps
U	Map using rules	Which rule does the cataloged map ABC refer to
R	Rule referenced in maps	Which cataloged maps refer to the rule ABC

Parameters		
Save set	Any user defined in Predict can save the types and names of maps listed by function R in one or more sets. See **Using Sets**.	
Map	The ID of the map for which the function is to be executed. Asterisk notation is possible.	
Rule	The ID of the rule for which the function is to be executed. Asterisk notation is possible.	
Rule status	Only rules of the given type are included in the report:	
	A	Automatic rules
	F	Free rules
	*, blank	All
Usage of rule	Only rules that are used in the given way are included in the report:	
	A	Used as automatic rule
	F	Used as free rule
	I	Used as inline rule
	*, blank	All

Parameters	
Set number	If a set number is specified, XRef data is only evaluated for those objects defined in the given set. All other search criteria are then ignored. Set numbers can not be used as input for Count functions.

External Resources

The functions display active reference information about External resources.

Note: In Predict, the term "External resource" refers to the Natural object type "Resource". Further information on this object type can be found in the section *Ressource* of *Objects for Natural Application Management* in the *Natural Programming Guide*.

The input screen is displayed with code F in the XRef menu.

```
 16:53:50            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC          - External Resources -         DBnr:   180 Fnr:    64


                  Code Function

                    C  Count references to External resources
                    U  Program using External resources
                    R  External resources referenced in programs

             Code: ?                  Save set: N (Y,N)
          Program:                Program type:   (?)
 External resource:




            Usage:    (?)
       Set number:  1 (?)

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Code	Function	Answer to the question...
C	Count references to External resources	How many times is the External resource ABC referred to by programs
U	Program using External resources	Which External resources does the program ABC refer to
R	External resources referenced in programs	Which programs refer to the External resource ABC

Parameters		
Save set	Any user defined in Predict can save the types and names of programs listed by functions U or R in one or more sets. See **Using Sets**.	
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	F	function
	H	help routine
	K	ISPF Macro
	M	map
	N	subprogram
	P	main program
	S	subroutine
	T	dialog
	4	class
	*, blank	All
External resource	Only External resources with IDs beginning with a particular sequence of characters are included in the report.	
Usage	Only External resources which are used in a particular way are included in the report.	
	B	Sets built by FIND ... WITH ... RETAIN AS 'SET' statements.
	R	Sets released by RELEASE SET 'SET' or RELEASE SETS statements.
	U	Sets used by FIND ... WITH 'SET' statements.
	*, blank	All
Set number	If a set number is specified, XRef data is only evaluated for those objects defined in the given set. All other search criteria are then ignored. Set numbers can not be used as input for Count functions.	

Report Programs with XRef Data

The function Report programs with xref data displays summary active reference information about programs. The input screen is displayed with code X in the XRef menu.

```
 16:54:30            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC             -  Report Program -         DBnr:   180 Fnr:    64


         Program: ________ (?) Program type: _ (?)
         User Id: ________      Terminal Id: ________
       from date: __________        to date: 2009-09-02
      short list: N (Y,N)          save set: N (Y,N)
      Set number:  1 (?)

      Report                         Nr     Report                        Nr

      Statistical data                1     Predict description           2
      Natural program list            3     Using/Referenced programs     4
      Views, Da-views and fields      5     Da-areas and variables        6
      Workfiles, printers, errors... 7     format extended description   N

      expand copycodes / rules        N     suppress empty reports        Y


Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↵
```

Up to seven types of information can be retrieved:

Information Reported by Report Program with XRef Data	
Statistical data	The program ID, the program type, the date and time when it was cataloged, the ID of the user who cataloged it, the ID of the terminal where it was cataloged and the information that would be displayed for this program by the Natural command LIST DIR object-ID.
Predict description	The extended description and abstract that are stored in the Predict object for that program. If Format extended description is set to Y, Con-form, Software AG's text formatting facility, is used whenever a description is output. See Extended Description Skeleton in the section *Defaults* in the *Predict Administration documentation*.
Natural program list	A Source code listing of the Natural program itself. If Expand copycodes / rules is set to Y, all copy code used by an INCLUDE statement is expanded in the listing of the program and all processing rules (free and automatic) used in a map are expanded in the listing of the map.

Information Reported by Report Program with XRef Data	
Using/referenced programs	For each program that either invokes that program or is invoked by that program, the program type, the program ID, the method of invoking and - for maps - the processing rules used are included in the report. **Note:** For every method sent by a program, all classes which support this method will be listed.
Views, Da-views and fields	Each view, data area view and field of a view used by that program is included in the report.
Da-areas and variables	Each data area and each variable used by that program is included in the report.
Workfiles, printers, errors	Natural workfiles, retained sets, copy code, printers and error message numbers used by that program are included in the report.

Parameters		
All parameters except Save set limit the scope of the function.		
Program	The ID of the program for which the function is to be executed. Asterisk notation is possible.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	A	parameter data area
	G	global data area
	F	function
	H	help routine
	K	ISPF Macro
	L	local data area
	M	map
	N	subprogram
	O	command processor
	P	main program
	S	subroutine
	T	dialog
	Y	Natural Expert model
	4	class
	*, blank	all
User ID	Only those programs cataloged by the specified user will be processed. Asterisk notation is possible.	
Terminal ID	Only those programs cataloged from the specified terminal are processed. Asterisk notation is possible.	
From date ... to date	Only those programs cataloged within the range of dates specified are processed.	

Parameters		
Short list	Y	Displays the following information for each program selected in a single line: - Program ID - Program type - Information about the existence of source, cataloged object and XRef data - Program mode (report mode or structured mode) - ID of the user who last saved or cataloged the program - Version of Natural - Date and time when the program was last saved or cataloged.
	N	A more extensive report is created, containing up to seven categories of information. Enter digits in the Nr column to determine the sequence of the information displayed. Enter 0 to suppress categories that are not to appear.
Save set	Any user defined in Predict can save the types and names of listed programs in a set. This feature is normally used together with the Short list option. See ***Using Sets***.	
Format extended description	Y	Any Con-form instructions contained in the extended description of any object are executed. See Extended Description Skeleton in the section *Defaults* in the *Predict administration documentation*.
Expand Copy Codes/Rules	Y	All copy code used by an INCLUDE statement is expanded in the listing of the program and all processing rules (free and automatic) used in a map are expanded in the listing of the map.
Suppress empty reports	Y	Suppress reports where no information is found in a particular category.

Example of NaturalX Support

The following sample output of function Report Programs with XRef data shows how features of NaturalX are documented in Predict.

```
*** Report for program references ***

4:CLASS1   class name NATURAL.BEAR
           interface  SOUND
              method    1  SNIFF                              impl. SNIFF-N
            property    1  NATURALBEARPROPERTY
           interface  IMOVE
              method    1  WALK                               impl. WALK-N

*** Report for data areas, blocks and variables ***

4:CLASS1   using data area    1  L:GUID-L
                 variables    1  INTERFACESOUNDGUID                  Used
                              2  NATURALBEARGUID                     Used

                 data area    2  L:BEAR-O
                                 as object data area
                 variables    1  BEAR-PROPERTY-VARIABLE              Used

                 data area    3  P:SNIFF-A

                 data area    4  P:WALK-A
```

62 Verify Application Integrity

- Verify Application Integrity Menu 424
- Verify Consistency 425
- Verify Implementation 427
- Verify References 429
- Handling Objects that are not Referenced 431
- Avoiding Dynamic Invocation 432
- Verifying All Aspects of an Application 432

The Verify Application Integrity screen offers functions verifying different aspects of the integrity of Natural applications. These functions support quality assurance.

- **Consistency**
  The functions find programs or objects used by programs that have been changed after compilation.
- **Completeness**
  The Objects not implemented but referenced functions find programs or objects that are referenced by programs but are not implemented.
- **Correctness**
  The Objects implemented but not referenced functions find implemented objects that are not referenced.

Verify Application Integrity Menu

The Verify Application Integrity menu is displayed with code A in the XRef menu.

```
 16:55:19            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC         - Verify Appl. Integrity -      DBnr:   180 Fnr:    64




                 Code Function

                   X  Consistency of application
                   I  Objects not implemented but referenced
                   R  Objects implemented but not referenced
                   A  Verify all


             Code: ?





Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Code	Function	Answer to the Question...	Comments
X	Consistency of application.	Which programs or objects that are used by programs have been modified since compilation	
I	Objects not implemented but referenced	Which programs or objects referenced by programs are not implemented	
R	Objects implemented but not referenced	Which implemented objects are not referenced	
A	Verify all		A second screen is displayed for marking functions to be executed with X.

The different functions are described in the sections below.

Verify Consistency

If work with XRef data is to be effective, the data must be current:

- Source programs must not have been changed after compile time. This ensures both the integrity of source programs and object codes as the consistency of XRef data with both of them.
- All resources used by the program (such as copy code, data areas and processing rules) must not have been changed or deleted after compilation of a program. For global data areas, this applies only if the Natural parameter RECAT (dynamic recatalog) is set to OFF. This parameter is explained in the *Natural Parameter Reference* documentation.

The Verify Consistency screen is displayed with code X in the Verify Application Integrity menu.

```
 16:56:19            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC         -  Verify  Consistency  -       DBnr:   180 Fnr:    64




                 Code Object

                  X   Timestamp of source, module and xref data
                  R   Timestamp of resources used in programs


            Code: ?
        Save set: N   (Y,N)




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Code	Function	Answer to the Question...	Comments
X	Timestamp of source, module and xref data	Which programs have been changed after compilation (and generation of XRef data)	If the XRef data is older than the object code, simply recatalog the program. If XRef data exists for which there is no program to recatalog, the data dictionary administrator should delete the XRef data. See ***Using Predict Special Functions***.
R	Timestamp of resources used in programs	Which views, data areas, copy code and free or automatic processing rules have been modified since programs that refer to them were cataloged	

Any inconsistencies can usually be corrected by recataloging the whole application (provided that all sources still exist).

Parameter	
Save set	Any user defined in Predict can save the types and names of programs listed by function R in one or more sets. See ***Using Sets***.

When you enter code X or R, the output screen shows the column S/C/X:

Code	Explantation
S	Source
C	Cataloged
X	XRef data

When you enter code X, the following output is possible in column Note:

Note	Explantation
no XRef data	No XRef data exist on current FDIC file.
XRef invalid	Timestamp of XRef data and cataloged object do not match.
recataloged	Object was recataloged. Timestamp of source and cataloged object do not match.
not cataloged	Source is not cataloged.
source updated	Timestamp of source and cataloged object do not match.

Verify Implementation

All objects referred to should be implemented. The functions of the Verify Implementation menu described below find objects that are referred to but not implemented. The Verify Implementation screen is displayed with code I in the Verify Application Integrity menu.

```
 16:57:01            *****  P R E D I C T *****                      2012-06-30
 Library: SYSDIC          - Verify Implementation -      DBnr:   180 Fnr:    64


                    Code Function

                      P  Programs not implemented but referenced
                      E  Externals not implemented but referenced
                      S  Sets not built but referenced
                      N  Error numbers without text
                      I  Programs not impl./ref. starting from one


                Code: ?              Save set: N (Y,N)                          ↩

        Program type:   (?)  Verify Structure: N (Y,N)                          ↩

             Program:
Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit
```

Code	Function	Answer to the question...	Comments
P	Programs not implemented but referenced	Which Natural programs that are referred to by programs are not implemented	Natural programs that are invoked but not cataloged are indicated. The invocation method (see **Which Invocation Methods are Evaluated**) is also indicated. The option `Program type` limits this function to programs of a particular type.
E	Externals not implemented but referenced.	Which external entries are referenced but not implemented	An external entry is considered to be implemented if XRef data exists for the 3GL program containing the entry. See the section *Overview of Predict* in the Introduction to Predict documentation for a description where and when XRef data is created for third generation languages. There are three kinds of external entries: Entries in DBRMs for programs using static SQL - Entries in system programs - Entries in other 3GL programs.
S	Sets not built but referenced	Which retained sets that are referred to by programs (in a FIND ... WITH 'SET' statement or a RELEASE SET 'SET' statement) are not built by any programs (in a FIND ... WITH ... RETAIN AS 'SET' statement)	
N	Error numbers without text	Which error numbers that are referred to by programs (for example, in a REINPUT *<number> statement) have no text assigned in SYSERR	
I	Programs not impl./ref. starting from one	Which objects are referred to, directly or indirectly, by the startup program but are not implemented Which implemented programs are not referenced either directly or indirectly by the startup program	In many applications, all objects that are implemented should be referred to, direct or indirectly, by one particular program: the startup program. The function finds objects that are referred to, directly or indirectly, by the startup program but are not implemented, and all programs that are implemented and not referenced directly or indirectly by the startup program

Parameters		
Save set	Any user defined in Predict can save the types and names of programs listed by function *P, E, S, N* in a set. See ***Using Sets***.	
	All parameters except Save set limit the scope of the function.	
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	H	help routine
	M	map
	N	subprogram
	O	command processor
	P	main program
	S	subroutine
	T	dialog
	4	class
	Y	Natural Expert model
	4	class
	*, blank	All
Program	Start program for the function I (Programs not impl./ref. Starting from one).	
Verify Structure	See **Verify Structure** for details.	

Verify References

All objects implemented in the application (except utility programs) should be referred to in the application. The functions of the Verify References menu find objects that are not used.

The Verify References screen is displayed with code I in the Verify Application Integrity menu.

```
 16:57:37            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC           - Verify References  -        DBnr:   180 Fnr:    64


                    Code Function

                      D  Data areas not referenced
                      V  Variables in data area not referenced
                      U  Programs with unused data areas
                      C  Copycodes not referenced
                      N  Error numbers not referenced
                      P  Programs not referenced
                      I  Programs not impl./ref. starting from one

                Code: ?        Save set: N (Y,N)
           Data area:          Data area type :   (?)
        Program type:   (?)
             Program:


Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Code	Function	Answer to the Question...	Comments
D	Data areas not referenced	Which cataloged data areas stored in the current library are not referred to by any programs	
V	Variables in data area not referenced	Which variables in a data area stored in the current library are not referred to by any programs	
C	Copycodes not referenced	Which copy code stored in the current library is not referred to by any programs	
U	Programs with unused data areas	Which programs use data areas without any reference to variables defined in these data areas.	
N	Error numbers not referenced	Which application error numbers that have texts assigned are not referred to by any programs	
P	Programs not referenced	Which programs are not referred to by any other program	
I	Programs not impl./ref. starting from one.	Which objects are referred to, directly or indirectly, by the startup program but are not implemented Which programs are implemented but not referenced either directly or indirectly by the startup program	In many applications, all objects that are implemented should be referred to, directly or indirectly, by one particular program: the startup program.

Parameters		
Save set	Any user defined in Predict can save the types and names of programs listed by function *D*, *U* or *P* in a set. See **Using Sets**.	
Data area	The data area(s) for which the unused variables are to be checked. Asterisk notation is possible. **Note:** Enter the string *DYNAMIC if all area independent variables (AIV) are to be reported. **Note:** Only data areas in the current library will be checked.	
Data area type	Type of data area used by the program. Valid values:	
	G	global data area
	L	local data area
	P	parameter data area
Program type	The type of Natural program for which the function is to be executed. One of the following values can be specified:	
	H	help routine
	M	map
	N	subprogram
	O	command processor
	P	main program
	S	subroutine
	T	dialog
	Y	Natural Expert model
	4	class
	*, blank	All
Program	Start program for the function I (Programs not impl./ref. Starting from one).	

Handling Objects that are not Referenced

The following sections describe how objects that are not used might be processed.

Data areas	Data areas that are not used can usually be deleted.
Variables in data areas	A variable that is not referenced may be used in a redefinition. This possibility should be considered before removing apparently unused variables from data areas.
Copy code	Copy code that are not used can be deleted.
Programs with unused data areas	The definition of data areas without any reference to variables defined in these data areas can usually be removed. If a parameter data area definition is removed, all referencing objects must be adapted too.

Error numbers	An error number that is referenced dynamically (by assignment to a variable) is always marked not referenced.
Natural programs	A program that is invoked dynamically (by assignment to a variable) is always marked not referenced.

Avoiding Dynamic Invocation

An application is documented more accurately if dynamic invocation is avoided. Dynamic invocation can be replaced by a DECIDE statement, as in the following example:

Replace

```
COMPRESS MNT OBJECT INTO PROG LEAVING NO FETCH PROG
```

by

```
DECIDE ON FIRST VALUE OF OBJECT
  VALUE EMPL
    FETCH 'MNTEMPL'
  VALUE VEH1
    FETCH 'MNTVEH1'
      .
      .
      .
  NONE VALUE
  WRITE INVALID OBJECT
END-DECIDE
```

Verifying All Aspects of an Application

Before an application is put into production, the functions of the Verify Application Integrity Menu should be executed, their results checked and appropriate action taken to remedy any errors.

Verify All can be used to perform several or all functions of the Verify Application Integrity menu in a single run.

```
 16:58:50            *****  P R E D I C T *****                      2012-06-30
 Library: SYSDIC              -  Verify All  -           DBnr:   180 Fnr:    64



    Object consistency        Object not implemented    Object not referenced

  X Source,Module,Xref      X Programs                X Programs
  X Resources used in pgm.  X Externals               X Data areas
                            X Error numbers           X Variables
                            X Sets not built          X Copycodes
                                                      X Error numbers
                                                      X Data areas in Programs


         Save set: X (Y,N)
 Verify Structure: X (Y,N)                                                    ↩




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit
```

Executing Verify All Interactively

If the verify all function is to be used to call all or several verification functions interactively, the Verify all screen can be displayed with code A in the Verify Application Integrity menu (see the screen **above**). The functions that are to be executed then have to be marked with X.

Note: With a large application, Verify all can take a long time. Default parameter settings are taken from the current XREF profile.

Executing Verify All in Batch Mode

Verify all is executed in batch mode with the following command:

```
VERIFY ALL
```

Verify All executes the functions selected in the current XREF profile. See **Maintaining a LIST XREF Profile** for further information. The default profile is used if no user profile is specified with the following command:

```
LIST XREF PROFILE user-ID
```

Saving the Result of Verify All in Sets

A user who is defined in the Predict data dictionary can save the sets of program types and names produced by both Verify consistency functions, four Verify implementation functions and two Verify references functions. A maximum of ninety-nine sets is allowed per user and Natural library.

Verify Structure

The scope of the function Programs not implemented can be set to the programs of the current library or to the programs of all libraries defined in the current library structure when the Verify all function is used with the option Save set Y.

63 Using Sets

- Functional Scope 436
- Examples for the Use of Sets 436
- Using Sets to Share Work among Team Members 437
- Internal Structure of Sets 437
- How are Sets Created? 438
- Who can Work with Sets? 438
- Working With Sets 438
- Creating a New Set via Selection 438
- Operate on Sets 439
- Application Programming Interface 443

Functional Scope

Predict can save the output list of XRef data retrieval in sets for subsequent processing:

- The Natural objects listed in a set can be cataloged, checked or stowed and their contents can be edited or listed directly from the set.
- An existing set of object types and names can be displayed, purged or sent to another user. When a set is sent, a short comment can be included that will appear when the set is displayed at the terminal of the recipient.
- Sets can be merged, subtracted or intersected.
- Sets are identified by the functions that were used to create them.
- Sets are saved separately for each user and each Natural library. Any user who is defined in the Predict data dictionary can create and use up to ninety-nine sets in any library.

Notes:

1. Sets can be used as input for certain `LIST XREF` commands. This is documented where applicable for each command individually in section ***Retrieving Information For Individual Members***. This also applies when issuing these commands in batch mode. Refer to ***Commands for Retrieving Information for Individual Members*** in section *LIST XREF Commands* for details.
2. Sets can also be used in the Natural Object Handler, the SYSMAIN utility, the Natural System Commands `CATALL` and `SCAN` and in Natural ISPF.

All functions used to process sets are called from the Operate on sets screen. See **Operate on Sets**.

Examples for the Use of Sets

- Whenever copy code is changed, all programs that use the copy code may require corresponding changes. The Natural `LIST XREF` function Copy code referenced in programs can be used to find out which programs may be affected. If the types and names of these programs are saved as a set, the programs can be modified, one after another, to match the changed copy code with a single call.
- Whenever a free or automatic processing rule is changed, all maps that use the processing rule must be recataloged. A Natural `LIST XREF` function can be used to find out which programs (maps) are affected and to save their types and names as a set. Later, a single function can be used to recatalog all maps included in the list.

Using Sets to Share Work among Team Members

When a project team is working on a Natural application, one member of the team may create a set that describes objects which require work from other members of the team. Using the send function, either the whole set or selected members can be copied or moved to any other user who is authorized to log on to the same library. A short comment can be sent with the set to tell the recipient why it has been sent. The set that was sent will appear next time that the recipient logs on to the library and displays the Operate on sets screen of the Natural `LIST XREF` command. The short comment will appear when the set of object types and names is next displayed.

Internal Structure of Sets

A set is a list of the types and names of program and/or data areas selected from a Natural library. All objects in the list are members of the same Natural library.

The type and name specification for each Natural object has the format type:name, for example:

S:SUB1

Valid types	
A	Parameter data area
G	Global data area
H	Help routine
L	Local data area
M	Map
N	Subprogram
O	Command processor
P	Main program
S	Subroutine
T	Dialog
Y	Natural Expert model
4	Class
F	Function
8	Adapter
C	Copy code
X	Text
5	Resource

Hence, the example above specifies the Natural subroutine named SUB1.

Sets are saved separately for each user and each Natural library.

How are Sets Created?

There are two ways to create a set:

- Set the Save set parameter to Y before executing a retrieval function for XRef data. A default value can be specified for this parameter in the XREF profile. This value is used when the command `LIST XREF VERIFY ALL` is executed.
- Set the `SAVE` parameter to Y when executing a `LIST XREF` command (online or batch mode).
- Use the function **Creating a New Set via Selection**.

Who can Work with Sets?

Any user defined in Predict can use functions of Predict and the Natural `LIST XREF` command to save sets of Natural object types and names. Up to ninety-nine sets of object types and names can be saved at any time in any Natural application (library) to which that user has access. Attempts to create a hundredth set will produce an error message.

Working With Sets

Two functions in the XREF menu apply to sets:

N	Create new set via selection
O	Operate on sets

Creating a New Set via Selection

The input screen is displayed with N in the XRef menu.

```
 16:59:34            *****  P R E D I C T *****                      2009-09-02
 Library: SYSDIC               - Create  Set -           DBnr:   180 Fnr:    64


      Object name: *_______
   Mark to select: _ all stowed objects
                   _ all objects not yet catalogued
                   _ all catalogued objects without source

   Mark to select: _ Global Data Areas           _ Local Data Areas
                   _ Parameter Data Areas        _ Subroutines
                   _ Subprograms (CALLNAT)       _ Helproutines
                   _ Maps                        _ Programs
                   _ Classes                     _ Dialogs
                   _ Functions                   _ Adapters
                   _ Copy Codes                  _ Texts
                   _ Resources



Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Sets can be created by entering selection criteria in the set creation screen. Asterisk notation is allowed, and one or more selection marks can be entered to restrict the set to Natural objects of certain types. All Natural objects that meet the given selection criteria are included in the set.

Note: Only objects from the current library are written to the set. A library or runtime structure can not be evaluated.

Operate on Sets

Sets can be displayed, deleted, sent, sorted (by name or type), modified, combined (merge, intersect, subtract), cataloged, checked, stowed, listed and edited. To process sets, display the Operate on sets screen with code O from the XRef menu.

```
 19:32:57                 *****  P R E D I C T  *****                 2011-10-25
 Library: SYSDIC              - Operate on sets -         DBnr:   180 Fnr:   106

 O  Nr  created with function                                  Date   C/K/W Count
    __                                (1 of 3)
 _  1   REPORT * (*) BY * ON * SAVE Y                        2011-10-25         1
 _  2   SELECT MENU                                          2011-10-25  C      1
 _  3   ERNIE -> SELECT AB8* (N)                             2011-10-25        74









 O: (I,U,X,Y       Intersection,Union,Difference X minus Y)
    (C,K,W,E,L     Cat,Check,Stow,Edit,List) Keep set after Cat/Check/Stow .. N
    (D,P,S,T,N,M,? Display,Purge,Send,Sort by type,Sort by name,Modify,Help)
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

Explanations

- A set is identified by a number and the text of the command that corresponds to the function with which the set was created (see Set 1 in the screen above).
- Sets created with the function **Creating a New Set via Selection** are identified with the text SELECT (see Set 2 in the screen above).
- C, K or W in the column C/K/W means that all entries in the Set were cataloged with `CAT`, `CHECK` or `STOW` and that errors occurred during checking or cataloging (Set 2). The set contains only members which produced errors (if option Keep set after Cat/Check/Stow is set to N, as in the screen above).
- Sets that were sent by another user are identified with the user ID of the sender and -->.

  Set 3 in the example above was sent by user ERNIE.

Displaying, Deleting, Sending and Sorting Sets

Sets can be displayed (D), purged (P), sent to another user (S) or their members can be sorted by type (T) or by name (N).

- **Sending a Set**
  Using the Send function, either the whole set or selected members can be copied or moved. A short comment can be sent with the set to tell the recipient why it was sent. The set that was sent will appear next time that the recipient logs on to the library and displays the Operate on sets screen of the Natural `LIST XREF` command. The short comment will appear when the set of object types and names is next displayed.
- **Sorting a Set**
  When sorting by name (code N), the Natural objects are sorted in alphabetical order. When sorting by type (code T), the Natural objects are displayed in the following order:

G	Global data area
L	Local data area
A	Parameter data area
S	Subroutine
F	Function
N	Subprogram
H	Help routine
M	Map
8	Adapter
P	Main program
4	Class
T	Dialog
C	Copy code
X	Text
5	Resource

Within one type, the objects are displayed alphabetically.

Modifying Sets

Enter function code M in the column O of the Operate on sets menu to display the following screen for the selected set.

```
13:18:56            *****  P R E D I C T  *****              2009-05-31
Library: SYSDIC             - Operate on sets -         DBnr:   180 Fnr:    54


Set Nr ............: 1
  Header ..........: PROG * (*) USING PROG * (*) WITH * VIA *

  Abstract ........:
```

You can modify the following set attributes:

Set nr.	You may enter a new set number between 1 and 99. The new number must not already exist.
Header	The header of a set indicates how the set was created. This information can be modified if required.
Abstract	Up to 10 abstract lines of up to 50 characters each can be specified.

Combining Sets

New sets can be created by combining existing sets in one of the following ways:

- Merging sets by marking two or more sets with U (union);
- Intersecting sets by marking two or more sets with I;
- Subtracting sets by marking them with X and Y.

Note: Sets can only be merged or intersected if they are sorted with the same sort order.

Predict assigns names to new sets that indicate which sets were combined and in which manner, for example: (2 i 3) denotes a set created by intersecting sets 2 and 3.

Cataloging, Checking, Stowing, Listing and Editing Objects Contained in Sets

Natural objects contained in sets can be processed directly from the set. Objects can be cataloged (C), checked (K) or stowed (W) and the source code of all the objects (if it exists) can be listed (L). The catalog, check, stow and list functions are also available as commands.

The edit (E) function can be used to display a set of object types and names as a menu. Individual Natural objects can then be marked to select their contents for editing (E) or listing (L), or to list directory (I).

If option Keep set after Cat/Check/Stow is set to Y, the set is retained after cataloging, checking or stowing. Otherwise the set is deleted or contains the entries for which errors were found.

Application Programming Interface

Two Application Programming Interfaces (API) are delivered which allow you to retrieve and maintain information about sets. For further information see Using API / Subprogram USR2033N and Using API / Subprogram USR4013N in the section *Application Programming Interface* of the *Predict Administration* documentation.

64 Maintaining a LIST XREF Profile

- Functional Scope 446
- Assigning Functions to PF Keys 447
- Limiting the Scope of the Report Programs with XRef Data Function 448
- Parameter Keep set after cat/stow processing 448
- Parameter Default Library Structure 449
- Parameter Search Usage in Web Services 449
- Parameter Verify Structure 450
- Assigning Defaults to the Verify All Function 450

Functional Scope

LIST XREF parameters can be changed using a specific LIST XREF profile. The following settings can be defined:

- PF key assignment, **Assigning Functions to PF Keys**.
- whether a set is retained after CAT or STOW, **Parameter Keep set after cat/stow processing**
- which library structure is to be used by default, **Parameter Default Library Structure**
- which reports are created with the Report program function and in which sequence these reports are displayed, **Limiting the Scope of the Report Programs with XRef Data Function**.
- which functions are executed by the Verify all function, **Assigning Defaults to the Verify All Function**.

The following rules apply for the use of the XREF profile:

- Settings in a DEFAULT XREF profile apply when no user specific XREF profile exists. Settings in the DEFAULT profile can be changed for the current session.
- If the user that changes an XREF profile is defined in Predict, the changes of the profile are saved and apply to all subsequent LIST XREF sessions of that user.

The LIST XREF profile is displayed with the LIST XREF command `PROFILE`.

Assigning Functions to PF Keys

```
17:03:34            *****  P R E D I C T *****                        2009-09-02
Library: SYSDIC         -   Add  user profile   -       DBnr:   180 Fnr:    64

Profile name : DEF

PF-Key definition

Function                  PF-Nr Text    Function                  PF-Nr Text

Invoked programs           1    Invp    Report programs with XREF  7    Xref
Data areas and variables   2    GDAV    Verify applic. integrity   15   ApCh
Views and fields           8    View    Maintenance profile        16   Prof
Copycodes                  6    Copy    Return to XREF main menu   11   Main
Error messages             23   Erro    Return to last menu        3    Quit
Printers                   13   Prt     Exit                       12   Exit
Workfiles                  14   Work    Show PF-Keys 1-12 / 13-24  10   SPfk
Retained sets              4    Sets    Operate on set             9    OSet
Processing rules           5    Rule    Create new set             18   CSet
External Resources         17   NRes


Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

All choices of the XREF menu and some additional functions can be called with PF keys. The PF key assignment and text that are displayed at the bottom of each menu for each PF key is defined in the first screen of the XREF profile.

In addition to the functions of the XREF menu, the following functions can be assigned to PF keys:

- Display the user's LIST XREF profile for changes (Maintenance profile)
- Return to the last menu that was displayed (the STOP function of Predict)
- Exit: leave the XREF system and return to Natural
- Change the display of PF key settings that appears at the bottom of every Natural LIST XREF menu

The settings for either PF keys 1-12 or 13-24 are displayed. at the bottom of each menu. The function of switching from one display to the other can be assigned to a PF key just like any other function, but this function will then automatically be assigned to the equivalent key in the alternative PF key display (either 12 more or 12 less).

Limiting the Scope of the Report Programs with XRef Data Function

```
17:05:24            *****  P R E D I C T *****                        2009-09-02
Library: SYSDIC         -   Add  user profile   -       DBnr:   180 Fnr:    64

Profile name : DEF

Report definition

  Report                           Nr    Report                           Nr

  Statistical data                  1    Predict description               2
  Natural program list              3    Using/Referenced programs         4
  Views, Da-views and fields        5    Da-areas and variables            6
  Workfiles, printers, errors...    7    format extended description       N

  expand copycodes / rules          N    suppress empty reports            Y

Miscellaneous
  Keep set after cat/stow processing ... N
  Default Library Structure ...........*
  Search usage in Web Services ......... Y
  Verify Structure ..................... Y

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↩
```

The Report programs with xref data function provides up to seven areas of information. The areas of information and the order in which they are displayed can be determined by entering the respective numbers in the column Nr.

A default value (Y or N) can also be specified for the parameters Format extended description, Expand copycodes/rules and Suppress empty reports.

Parameter Keep set after cat/stow processing

If you set this parameter to Y, the set is retained after `CAT` or `STOW`.

Note: If you enter a `CATALOG` or `STOW` command with the **option** `KEEP` **or** `NOKEEP`, this value is set to Y or N respectively for the duration of your current session.

Parameter Default Library Structure

This parameter specifies the default library structure to be used. Enter the ID of a library structure or one of the following values:

Value	Description
blank	No library structure. Only objects in the current library will be evaluated.
*R	The runtime structure will be used.

*	The following selection window appears:
	
	`E D I C T  *****  2009-07-31`
	`  XRef Menu  -  DBnr:  180 Fnr:  54`
	
	`  +-----------------------------------+`
	`  !  *** without structure ***  !`
	`  ------  !  *** run time structure ***  !`
	`  !  LS-NEWDICLX  !`
	`  !  LS-PDLX  !`
	`  !  PD-COB  !`
	`  !  !`
	
	In this window you can specify that
	■ no library structure is to be used for evaluating XRef data the runtime structure is to be used.
	■ the **runtime structure** is to be used.
	■ a library structure defined in Predict is to be used. A list of all library structures in Predict is displayed.
	
	See Library Structure in the Predefined Object Types in Predict documentation

Note: The Default Library Structure will be activated for the current library only if the system which documents the current library is contained in the link list Library Structure > System.

Parameter Search Usage in Web Services

If you set this parameter to Y, it is also considered for Natural members, files and fields whether they are used in Web services.

The following screen shows an example of how this parameter effects the output of the LIST XREF function Program using programs.

```
MORE
 15:54:46            *****  P R E D I C T *****                      2009-07-31
 Library: MUTCST            - Invoked Programs -         DBnr:   180 Fnr:    54
 Command: PROG KKUNDNM4 (*) USING PROG * (*) WITH * VIA *            Page:    1
          SAVE N
      T:Program               using                              via
-----------------------------------------------------------------------------
    1 N:KKUNDNM4            1 P:unknown  <<- nfnd                Call
                              Entry   : C-ENTRY
      Used by Service KundeEclipseUnixClassWS
      registered in
      DEMO
                            2 P:unknown  <<- nfnd                Call
                              Entry   : C-ENTRY2
                            3 P:unknown  <<- nfnd                Call
                              Entry   : C-ENTRY3
                            4 ?:CALC     <<- nfnd                Callnat
                            5 N:KKUNDNM4                         Callnat
                              Used by Service KundeEclipseUnixClassWS
                              registered in
                              DEMO


 ***** END OF LIST *****                                                    ↩
```

Parameter Verify Structure

If you set this parameter to Y, the scope of the function is not only the current library, but also the objects in the current library structure.

Assigning Defaults to the Verify All Function

The functions to be performed if the `Verify all` function is selected, can be specified in the next screen.

```
17:07:07            *****  P R E D I C T *****                      2009-09-02
Library: SYSDIC         -   Add  user profile   -       DBnr:   180 Fnr:    64

Objects and options for function Verify all

   Object consistency        Object not implemented    Object not referenced

 X Source,Module,Xref      X Programs                X Programs
 X Resources used in pgm.  X Externals               X Data areas
                           X Error numbers           X Variables
                           X Sets not built          X Copycodes
                                                     X Error numbers
                                                     X Data areas in Programs


Save set : N (Y,N)





Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
     Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit   ↵
```

Note: The Parameter `Verify Structure` described above is also valid for the `Verify all` function.

65 LIST XREF Commands

- General Information 454
- Using LIST XREF in Batch Mode 455
- Commands for Retrieving Information for Individual Members 457
- Command to Report on Program Active References 462
- Commands for Verifying the Integrity of an Application 462
- Miscellaneous 463

General Information

Functions of the Natural `LIST XREF` command can be called with commands. There are two ways to enter Natural `LIST XREF` commands:

- At the Natural NEXT prompt `LIST XREF` commands must be preceded with `'LIST XREF'`. This can be abbreviated to `L X` (separated by a blank).
- In the command line of any LIST XREF menu `LIST XREF` commands do *not* have to be introduced with `LIST XREF`. Note that any Natural command can be entered after the Command ===> prompt.

In the following sections, the commands are shown as they are to be entered after the Command ===> prompt.

Basic LIST XREF Command

The basic `LIST XREF` command - if entered without any command following - initiates the `LIST XREF` menu dialog.

```
LIST XREF
```

Asking for Online Help

Entering `LIST XREF ?` in response to the NEXT prompt or ? in the Command===> field will cause the XREF command help menu to be displayed:

```
LIST XREF ?
```

Notational Conventions

The following rules apply to the syntax descriptions below and in the online help on `LIST XREF` commands:

- Angle brackets indicate optional parts of the commands; the angle brackets themselves should not be typed. Parentheses may be omitted only when the text they enclose is omitted.
- Keywords shown in upper case can be truncated provided that the input remains unambiguous.
- Text shown in lower case should be replaced by an appropriate value.

In the tables below each syntax description is introduced by the name of the function in the `LIST XREF` menu.

Syntax Conventions

There is a subset of Predict keywords (or abbreviations of keywords) which, when used as names for variables (such as program names, field names, etc.), would be ambiguous. If you use such a keyword as the name of a variable, you may run into syntax errors.

To avoid such syntax errors, you are strongly recommended to enclose variable parts of the command in single quotation marks.

Example

The following command would result in a syntax error, since the variable `REFE` contains characters that are also used in the reserved keyword `REFERENCED`.

```
PROG REFE (*) USING PROG
```

To avoid the syntax error, enter the command as follows:

```
PROG 'REFE' (*) USING PROG
```

Using LIST XREF in Batch Mode

All functions of the `LIST XREF` command can also be executed in batch mode.

The following basic rules apply when entering `LIST XREF` commands in Predict in batch mode:

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.

Continuation Records

If the parameters will not fit in one record,

- the command (`LIST XREF`) and the parameters must be split in separate records
- the parameters must be separated with the input delimiter (the blank is no longer valid as delimiter)
- the last parameter in the record must be followed by a percent character (%). Parameters in the next line are then concatenated.

You can change the continuation character with the Natural parameter CF.

For example:

```
L X
PROG,*,(*),USING,FIELD,PERSONNEL-ID% <-- Continuation record follows
(EMPLOYEES-FILE),USAGE,U
```

LIST XREF-specific Conversations

A `LIST XREF` batch job has to contain commands in the following order:

- Log on to the library where XRef data is to be retrieved (required), e.g. `LOGON MYLIB`
- Change Natural system parameters (optional), e.g. `GLOBALS PS=60`
- Select a profile (optional), e.g. `L X PROFILE JOHN`.

  Profile may contain a Default Library Structure.

  Profile is required to pass a user ID for save sets if Natural Security is not active.
- Select a library structure (optional), e.g. `L X STRUCTURE *R` or via the `PROFILE` command (see above)

  Note: If no profile is selected or the selected profile does not exist, the profile defined as the LIST XREF default profile is used. If the selected profile or the default profile contains a library structure, this library structure is only accepted if it is valid for the current library.

- Enter any number of `LIST XREF` commands, each of them beginning with `LIST XREF` (or its abbreviation `L X`), e.g.

  ```
  L X PROGRAM ACM* USING PROGRAMS VIA CALLNAT

  L X VERIFY ALL
  ```

  Give the `FIN` command to terminate the Natural session.

Identifying Sets in Batch Mode

If sets are to be processed, a user ID must be specified to identify the set. This can be done in one of the following ways:

- If Natural Security is active, the logon user ID is used to identify the set.
- If Natural Security is not active, an `L X PROFILE` command can be issued to pass a user ID

Commands for Retrieving Information for Individual Members

Invoked Programs

Function	Code	Syntax
Count references in program	D	COUNT PROG <program> <(program-type)> USING PROG <used-program> <(used-program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property>
Count references to program	C	COUNT PROG <used-program> <(used-program-type)> REF PROG <program> <(program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property>
Program using programs	U	PROG <program> <(program-type)> USING PROG <used-program> <(used-program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property> IN <set-number> <SAVE Y/N>
Program referenced in programs	R	PROG <used-program> <(used-program-type)> REF PROG <program> <(program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property> IN <set-number> <SAVE Y/N>
Program using programs recursively	E	PROG <program> <(program-type)> USING REC <used-program> <(used-program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property> IN <set-number> <DEPTH depth>
Program referenced in programs recursively	I	PROG <used-program> <(used-program-type)> REF REC <program> <(program-type)> <VIA method> <WITH Subroutine/Entry/Class/Method/Property> IN <set-number> <DEPTH depth>
XREF program using programs	X	XREF PROG <program> <(program-type)> USING PROG <used-program> <(used-program-type)> <WITH Subroutine/Entry/Class/Method/Property> <VIA method> IN <set-number> <DEPTH depth> <SAVE Y/N>
XREF program referenced in programs	Y	XREF PROG <program> <(program-type)> REF PROG <used-program> <WITH Subroutine/Entry/Class/Method/Property> <VIA method> IN <set-number> <DEPTH depth> <SAVE Y/N>

Methods

The method used to invoke programs. For a list of valid values see ***table*** in subsection *Invoked Programs* in the section *Retrieving Information For Individual Members* in this documentation.

Data Areas and Variables

Function	Code	Syntax
Count references to variable	C	COUNT VAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
only for variables defined in global data areas		COUNT GVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
only for variables defined in local data areas		COUNT LVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
only for variables defined in parameter data areas		COUNT PVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage>
Program using variable	U	PROG <program> <(program-type)> USING VAR <variable> <(structure)> <OF data-area> <USAGE variable-usage> IN <set-number>
only for variables defined in global data areas		PROG <program> <(program-type)> USING GVAR <variable> <(structure)> <OF data-area> <USAGE variable-usage> IN <set-number>
only for variables defined in local data areas		PROG <program> <(program-type)> USING LVAR <variable> <(structure)> <OF data-area> <USAGE variable-usage> IN <set-number>
only for variables defined in parameter data areas		PROG <program> <(program-type)> USING PVAR <variable> <(structure)> <OF data-area> <USAGE variable-usage> IN <set-number>
Variable referenced in programs	R	VAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage> IN <set-number> <SAVE Y/N>
only for variables defined in global data areas		GVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage> IN <set-number> <SAVE Y/N>
only for variables defined in local data areas		LVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area> <USAGE variable-usage> IN <set-number> <SAVE Y/N>

Function	Code	Syntax
only for variables defined in parameter data areas		PVAR <variable> <(structure)> REF PROG <program> <(program-type)> <OF data-area><USAGE variable-usage> IN <set-number><SAVE Y/N>
Program using data areas	D	PROG <program> <(program-type)> USING DA-AREA <data-area> <(data-area-type)> <BLOCK data-block> IN <set-number> <SAVE Y/N>
Data area referenced in programs	E	DA-AREA <data-area> <(data-area-type)> REF PROG <program> <(program-type)> <BLOCK variable-usage> IN <set-number> <SAVE Y/N>

Views and their Fields

Function	Code	Syntax
Count references to view	C	COUNT VIEW <view> REF PROG <program> <(program-type)> <USAGE ddm-usage>
Count references to field of view	D	COUNT FIELD <field> <(view)> REF PROG <program> <(program-type)> <USAGE field-usage>
Program using views	U	PROG <program> <(program-type)> USING VIEW <view> <USAGE ddm-usage> IN <set-number>
View referenced in programs	R	VIEW <view> REF PROG <program> <(program-type)> <USAGE ddm-usage> IN <set-number> <SAVE Y/N>
Program using fields of views	F	PROG <program> <(program-type)> USING FIELD <field> <(view)> <USAGE field-usage> IN <set-number>
Field of view referenced in programs	G	FIELD <field> <(view)> REF PROG <program> <(program-type)> <USAGE field-usage> IN <set-number> <SAVE Y/N>
Program using data area views	A	PROG <program> <(program-type)> USING DA-VIEW <data-area-view> <(view)> <USAGE data-area-usage> IN <set-number>
Data area view referenced in programs	B	DA-VIEW <data-area-view> <(view)> REF PROG <program> <(program-type)> <USAGE data-area-usage> IN <set-number> <SAVE Y/N>

Copy Code

Function	Code	Syntax
Count references to copy code	C	COUNT COPYCODE <copycode> REF PROG <program> <(program-type)> <USAGE copycode-usage>
Program using copy code	U	PROG <program> <(program-type)> USING COPYCODE <copycode> <USAGE copycode-usage> IN <set-number> <SAVE Y/N>
Copy code referenced in programs	R	COPYCODE <copycode> REF PROG <program> <(program-type)> <USAGE copycode-usage> IN <set-number> <SAVE Y/N>

Error Numbers

Function	Code	Syntax
Count references to error number	C	COUNT ERROR <from-error-number> <(error-type)> REF PROG <program> <(program-type)> <THRU to-error-number>
Program using error numbers	U	PROG <program> <(program-type)> USING ERROR <from-error-number> <(error-type)> <THRU to-error-number> IN <set-number>
Error number referenced in programs	R	ERROR <from-error-number> <(error-type)> REF PROG <program> <(program-type)> <THRU to-error-number> IN <set-number> <SAVE Y/N>

Printers

Function	Code	Syntax
Program using printers	U	PROG <program> <(program-type)> USING PRINTER <from-printer-number> <THRU to-printer-number> <NAME printer-name> IN <set-number>
Printer referenced in programs	R	PRINTER <from-printer-number> REF PROG <program> <(program-type)> <THRU to-printer-number> <NAME printer-name> IN <set-number> <SAVE Y/N>

Workfiles

Function	Code	Syntax
Program using workfiles	U	PROG <program> <(program-type)> USING WORK <from-work-file-number> <THRU to-work-file-number> <USAGE work-file-usage> IN <set-number>
Workfile referenced in programs	R	WORK <from - work-file-number> REF PROG <program> <program-type)> <THRU to-work-file-number> <USAGE work-file-usage> IN <set-number> <SAVE Y/N>

Retained Sets

Function	Code	Syntax
Program using retained sets	U	PROG <program> <(program-type)> USING SET <set> <USAGE set-usage> IN <set-number>
Retained set referenced in programs	R	SET <set> REF PROG <program> <(program-type)> <USAGE set-usage> IN <set-number> <SAVE Y/N>

Processing Rules

Function	Code	Syntax
Count references to rules	C	COUNT RULE <rule> <(rule-status)> REF PROG <map> <(M)>
Map using rules	U	PROG <map> <(M)> USING RULE <rule> <(rule-status)> <USAGE rule-usage> IN <set-number>
Rule referenced in maps	R	RULE <rule> <(rule-status)> REF PROG <map> <(M)> IN <set-number> <SAVE Y/N>

External Resource

Function	Code	Syntax
Count references to External resources	C	COUNT NRESOURCE <nresource> REF PROG <program> <(program-type)> <USAGE resource-usage>
Program using External resources	U	PROG <program> <(program-type)> USING NRESOURCE <NRESOURCE> <USAGE resource-usage> IN <set-number> <SAVE Y/N>
External resources referenced in programs	R	NRESOURCE <resource> REF PROG <program> <(program-type)> <USAGE resource-usage> IN <set-number> <SAVE Y/N>

Command to Report on Program Active References

The following command syntax sets the Short list parameter to N and produces an extensive report:

```
REPORT<program> <(program-type)> <BY user-id> <ON terminal-id> <FROM
      yy-mm-dd> <THRUyy-mm-dd > <SAVE Y/N>
```

The following command syntax sets the Short list parameter to Y and produces a condensed report:

```
SUMMARIZE<program> <(program-type)> <BY user-id> <ON terminal-id> <FROM
      yy-mm-dd> <THRU yy-mm-dd> <SAVE Y/N>
```

Commands for Verifying the Integrity of an Application

Verify the Consistency of an Application

Function	Code	Syntax
Programs with inconsistent xref data	X	VERIFY XREF-DATA <SAVE Y/N>
Resources used in program	R	VERIFY RESOURCES <SAVE Y/N>

Verify the References of an Application

Function	Code	Syntax
Data areas not referenced	D	VERIFY DATA-AREA REFERENCES <(data-area-type)> <SAVE Y/N>
Variables in data area not referenced	V	VERIFY VARIABLE REFERENCES <data-area> <(data-area-type)>
Copy code not referenced	C	VERIFY COPYCODE REFERENCES
Programs with unused data areas	U	VERIFY PROGRAM DATA-AREA <program> <(program-type)> <(data-area-type)> <SAVE Y/N>
Error numbers not referenced	N	VERIFY ERROR REFERENCES
Programs not impl./ref. starting from one	I	VERIFY PROGRAM STARTING program
Programs not referenced	P	VERIFY PROGRAM REFERENCES <(program-type)> <SAVE Y/N>

Verify the Integrity of an Application

The following command syntax activates all the application verification functions specified in the user's profile:

```
VERIFY ALL <SAVE Y/N>
```

Miscellaneous

Operate on Sets

Function	Code	Syntax
Intersect all objects in the sets.	I	INTERSECT set-number set-number
Merge all objects in the sets.	U	UNION set-number set-number
Subtract set Y from set X.	X,Y	DIFFERENCE set-number set-number
Start a catalog process for all objects in the set.	C	CATALOG set-number [KEEP | NOKEEP]
Start a check process for all objects in the set.	K	CHECK set-number [KEEP | NOKEEP]
Start a stow process for all objects in the set.	W	STOW set-number [KEEP | NOKEEP]
List the source code of all objects in the set.	L	LIST set-number

Option KEEP / NOKEEP

KEEP	The set is retained after processing. The value for parameter Keep set after cat/check/stow processing from the LIST XREF profile is set to Y for the current session.
NOKEEP	The set is not retained after processing. The value for parameter Keep set after cat/check/stow processing from the LIST XREF profile is set to N for the current session.

If neither KEEP nor NOKEEP is specified, the current value for Keep set after cat/check/stow processing is used.

Maintain a User's LIST XREF Profile

`PROFILE`

Specifying the Library Structure to be used

The following commands are used to specify the library structure which is to be used for evaluating XRef data for subsequent LIST XREF commands. The commands below are only available in batch mode in the XRef menu. Online, the library structure is specified using the parameter Structure as described in the section **LIST XREF Main Menu**.

- `STRUCTURE <structure-name> [REVERSE | NOTREVERSE]`

With this command you can specify which library structure is to be used for evaluating XRef data.

- `STRUCTURE *R [REVERSE | NOTREVERSE]`

This command specifies that the runtime structure is to be used for evaluating XRef data.

If `NOTREVERSE` is specified, the library structure as defined in either Natural Security or Predict is used for evaluation. This is the default.

If `REVERSE` is specified, the library structure as defined in either Natural Security or Predict is used for evaluation but in reverse order.

Note: If you do not specify any structure, the LIST XREF functions are evaluated without library structure.

VIII List XRef for Third Generation Languages

66 List XRef for Third Generation Languages

- Creation of XRef Data for 3GL 468
- Contents of XRef Data for 3GL 469
- Types of LIST XREF Functions for 3GL 470
- Calling Functions of LIST XREF for 3GL 471
- Retrieving Information for Individual Members 472
- Verify Application 479
- Operate on Sets 480
- Maintaining a LIST XREF Profile 481

In addition to documentation objects, which establish the metadata model of an application, XRef data is stored in Predict. XRef data document objects of an application that are already implemented. Data of this type mirror the current implementation status of an application.

Note: The concepts of retrieving XRef data and some LIST XREF functions apply both to third generation languages and Natural. The detailed descriptions of the concepts and functions applying both to Natural and third generation languages are contained in the section **LIST XREF for Natural**. This section refers back to the previous section whenever necessary. A general description of how XRef data is created and used is contained in the section *XRef Data* in the *Introduction to Predict documentation*.

Creation of XRef Data for 3GL

XRef data for programs in third generation languages is written by different utilities:

- By Adabas Native SQL (COBOL, FORTRAN, PL/I and ADA).
- During precompilation with the Predict preprocessor (Assembler, COBOL and PL/I). The preprocessor is described in the section *Preprocessor* in the *External Objects in Predict documentation*.
- If Natural DB2 is installed, the function CREATE DBRM of Natural for DB2 creates XRef data for Natural programs that use static SQL.
- If a 3GL program is documented in Predict as a system, XRef data is automatically created when the program is documented in the data dictionary as a program in the library *SYSSYS*. The invocation method is CALL.

Where XRef Data for 3GL Members is Stored

With this version of Predict, you can store XRef data for 3GL members in the default library from the table below, or you can define your own library by documenting this library as a Predict object of type system. The Library parameter is also used by the preprocessor.

Language	Default Library
BAL/Assembler	*SYSBAL*
COBOL	*SYSCOB*
FORTRAN	*SYSFOR*
Language ADA	*SYSADA*
PL/I	*SYSPLI*
Static SQL	*SYSSTA*
Language C	*SYSCCC*

Contents of XRef Data for 3GL

- Which Invocation Methods are Evaluated

The following information is stored for 3GL members:

- the name of the program and the application it belongs to
- the program type (only main program)
- the date and time the program was last cataloged
- the ID of the user who cataloged the program
- the ID of the terminal from which the program was cataloged (in batch mode: the job name)
- the names of 3GL programs invoked and the methods used to invoke them (only CALL, static SQL)
- the names of external programs invoked
- the names of files used in the program and the type of file usage (deleting, reading or updating)
- the names of fields of files used in the program and the type of field usage (for counting, reading, searching or updating) (3GL precompiled by Adabas Native SQL).

Which Invocation Methods are Evaluated

See table below for the invocation methods evaluated when generating XRef data for third generation languages.

Language	Invocation Method
BAL/Assembler	CALL
COBOL	CALL
FORTRAN	CALL
Language ADA	CALL
PL/I	CALL
Static SQL	Static SQL
Language C	CALL

Types of LIST XREF Functions for 3GL

XRef data for third generation languages is retrieved with functions of the Predict XRef menu. There are essentially three groups of functions:

- **Functions that retrieve information on specific types of members in an application**
  All functions from Invoked programs through Report programs with xref data in the XRef Menu are of this type (see screen below). See **Retrieving Information for Individual Members** .
- **Functions that retrieve information on the consistency of an application as a whole**
  The functions are called via a submenu that is called with code A (Verify application). See *Verify Application*.
- **Functions that manage sets**
  Sets allow efficient use of information retrieved from XRef data. Most of these functions can be executed only from a menu. See *Using Sets*.

```
13:00:18              *****  P R E D I C T  *****                2007-05-31
                               -   Xref Menu   -

                           Code Object
                           ---- ------------------------------
                             I  Invoked programs
                             V  Views and fields
                             C  Copycode
                             X  Report programs with xref data
                             A  Verify application
                             O  Operate on sets
                          ---- ------------------------------
   Code....................:   (./?)
   Library ...............*:
   Library structure ID ...:
   System Id ..............:

   Command ===>
  Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
        Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit
```

The following functions are not available for static SQL:

I	Invoked programs
C	Copy Code
A	Verify application integrity

Calling Functions of LIST XREF for 3GL

All LIST XREF Functions for third generation languages can be called from the Predict XRef menu or with a command. The XRef Menu itself can be called with a command ACTIVE XREF or with function code X in a Predict main menu.

A library must be specified, either a default library from the table below or a user-defined library.

Language	Code	Default Library
BAL/Assembler	BA	*SYSBAL*
COBOL	CO	*SYSCOB*
FORTRAN	FO	*SYSFOR*
ADA	LA	*SYSADA*
PL/I	PL	*SYSPLI*
Static SQL	SQ	*SYSSTA*

Example for PL/I:

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

See **LIST XREF Commands** for more information on calling LIST XREF function with commands.

Using Sets

The LIST XREF for third generation languages often produces a list of objects that need further processing. To facilitate further processing Predict will save the output list of LIST XREF for third generation languages in a set, if the Save set switch of a Predict LIST XREF function is set to Y. Sets are saved in the same libraries as the active references for those languages. Up to ninety-nine sets of member types and names can be saved at any time in any library.

See **Using Sets** for more information.

Retrieving Information for Individual Members

All Predict LIST XREF function display information on members written in one specific third generation language. It is not possible to retrieve active reference information for more than one third generation language in one call.

Note: See the respective section under *LIST XREF for Natural* for a description of retrieving information for individual 3GL members.

Invoked Programs

The functions of the Invoked Programs screen display XRef data about programs that are invoked from 3GL programs. The functions do not apply to static SQL. The screen is displayed by selecting code I in the XREF menu.

```
13:23:13             *****  P R E D I C T  *****              2007-05-31
 Library:  LIBRARY          - Invoked Programs -

                Code Function
                ---- -----------------------------------
                  D  Count references in program
                  C  Count references to program
                  U  Program using programs
                  R  Program referenced in programs
                  E  Program using programs recursively
                  I  Program referenced in programs recursively
                  X  Xref program using programs
                  Y  Xref program referenced in programs
                --- -----------------------------------
 Code...........: ? (./?)                          Save set..: N (Y,N)

 Using program..:
 Referenced prg.:
 External.......:                                  Rec. depth: 7 (1-7)

 Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit
```

Function Codes

Code	Function	Answers the Question	Remarks
D	Count references in program	How many programs does the program ABC refer to	
C	Count references to program	How many times is the program ABC referred to by programs	
U	Program using programs	Which programs does the program ABC refer to	
R	Program referenced in programs	Which programs refer to the program ABC	
E	Program using programs recursively	Which programs are referred to by the program ABC, or which programs are referred to by programs which are referred to by the program ABC.	The output is structured to show the direct references to programs on the next level. See ***Truncating the output***, below.
I	Program referenced in programs recursively	Which programs refer to the program ABC, or which programs refer to programs which refer to the program ABC.	
X	XRef program using programs	Which programs are referred to by the program ABC, or which programs are referred to by programs which are referred to by the program ABC.	The total number of programs on each level and the type and ID of each program are output in a report. This function gives a summary of the output of function E but ignores structural information.
Y	XRef program referenced in programs	Which programs refer to the program ABC, or which programs refer to programs which refer to the program ABC.	The total number of programs on each level and the type and ID of each program are output in a report. This function gives a summary of the output of function I but ignores structural information.

Truncating the Output - Functions E and I

In some cases the listing of a LIST XREF function is truncated and marked as follows:

- Recursion is marked with *rec* and the output is terminated.
- The output can be truncated and marked with *trunc* at a level (up to seven) selected by the user.
- The user can enter other parameters that limit the output (for example, to programs of a particular type or with IDs beginning with a particular letter). Output terminated in this way or terminated by a reference to a 3GL program is marked *stop*.

- A program that has already been displayed at a higher level in the retrieval hierarchy is not repeated at a lower level. Instead the program is suppressed and marked *suppr*.

Limiting the Scope of the Functions

The following parameters limit the scope of the functions.

Parameters	Description
Save set	Any user defined in Predict can save the types and names of programs listed by functions U, R, X or Y in a set. See ***Using Sets***. **Note:** Sets only contain the resulting objects of the called function.
Using program	The ID of the 3GL program for which the function is to be executed. Asterisk notation is possible.
Referenced programs	The program or programs that are referred to by the 3GL program selected above.
External	The point of entry into an external program (written in a different programming language) that is called. This parameter applies when a program is invoked without using its ID. Predict uses information from either Adabas Native SQL or the Predict preprocessor to identify the program to which an external entry belongs.
Rec. depth	A number from 1 to 7: the recursion depth for the functions with codes E, I, X and Y.

Views and their Fields

The functions of the Views and Fields screen display XRef data about views and fields. The screen is displayed by selecting code V in the XRef menu for a 3GL.

```
13:24:19            *****  P R E D I C T  *****             2007-05-31
 Library:  LIBRARY        -    Views and Fields   -


                   Code Function
                   ---- ----------------------------------
                     C  Count references to view
                     D  Count references to field of view
                     U  Program using views
                     R  View referenced in programs
                     F  Program using fields of views
                     G  Field of view referenced in programs
                   ---- ----------------------------------
Code...........: ? (./?)
                                                Save set....: N (Y,N)
Program........:
View...........:                                View usage..:   (?)
Field..........:                                Field usage.:   (?)


Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit
```

Function Codes

Code	Function	Answers the Question
C	Count references to view	How many times is the view ABC referred to by programs
D	Count references to field of view	How many times is the field ABC referred to by programs
U	Program using views	Which views does the program ABC refer to
R	View referenced in programs	Which programs refer to the view ABC
F	Program using fields of views	Which views does the program ABC refer to, and which fields of these views does it refer to
G	Field of view referenced in programs	Which programs refer to the field ABC

Parameters Limiting the Scope of the Functions

Code	Answers the Question
Save set	If the user is defined in Predict, the set of program types and names produced by functions R, G and B is saved.
Program	The ID of the 3GL program for which the function is to be executed. Asterisk notation is possible.
View	The Adabas userview that is referred to by the 3GL program selected above. Asterisk notation is possible.
View usage	How the view is used by the 3GL program. One of the following values can be specified:

Code	Answers the Question	
	D	Delete
	M	Modified (either deleted or updated)
	O	Only read (neither deleted nor updated)
	R	Read, deleted or updated
	U	Update
Field	The field or fields that are referred to by the 3GL program. Asterisk notation is possible.	
Field usage	How the field is used by the 3GL program. One of the following values can be specified:	
	C	Used as a counter field
	O	Either used as a counter field or read
	R	Updated, used as a counter field or read
	S	Search
	U	Updated

Copy Code

The functions of the Copy Code screen display XRef data about copy code. The functions do not apply to static SQL. The screen is displayed by selecting C in the XRef menu.

For programs in third generation languages, XRef data is only stored for copy code if is used directly (not via any other copy code).

```
 13:04:13            *****  P R E D I C T  *****              2007-05-31
 Library:  LIBRARY              - Copycode -



                     Code Function
                     ---- -------------------------------
                       C  Count references to copycode
                       U  Program using copycodes
                       R  Copycode referenced in programs
                     ---- -------------------------------

Code...........: ? (./?)
                                                 Save set.... N (Y,N)
Program........:
Copycode.......:




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit
```

Code	Function	Answers the Question
C	Count references to copy code	How many times is the copy code ABC referred to by programs
U	Program using copy codes	Which copy code does the program ABC refer to
R	Copy code referenced in programs	Which programs refer to the copy code ABC

Parameters Limiting the Scope of the Functions	
Save set	If the user is defined in Predict, the set of program types and names produced by functions U or R is saved.
Program	The ID of the 3GL program for which the function is to be executed. Asterisk notation is possible.
Copy code	Copy code to which the programs refer. Asterisk notation is possible.

Report Programs with XRef Data

The function Report Program retrieves and displays a summary of active reference information about programs. The screen is displayed by selecting code X in the XRef menu for a 3GL.

```
  13:07:26            *****  P R E D I C T  *****              2007-05-31
  Library:  LIBRARY           - Report  Program -


          Program: ________  (?)
          User Id: ________  Terminal Id : ________
        from date: __________     to date: 2007-05-31
       short list: N (Y,N) save set: N (Y,N)

       Report                           Nr   Report                           Nr
       ------------------------------- --   ------------------------------- --
       Statistical data                 1   Predict description              2
       Using / Referenced programs      4   Views and fields                5
       Copycodes                        7

       Format extended description      N   Suppress empty reports          Y
       ------------------------------- --   ------------------------------- --

 Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit
```

Up to five types of information can be retrieved and displayed by the function:

Types of Information Reported by the Report Program Function	
Statistical data	The Program ID, the date and time when the program was precompiled, the ID of the user who precompiled it and the ID of the terminal from which it was precompiled.
Predict description	The extended description and abstract that are stored in the Predict object for that program. If Format extended description is set to Y, Con-form, Software AG's text formatting facility, is used whenever a description is output. Any formatting instructions included in the description will be executed by Con-form.
Using / Referenced programs	For each program that either invokes or is invoked by the specified program: the program type, the program ID and the method of invoking.
Views and fields	Each view and each field of a view that is referred to by the specified program.
Copy codes	All copy code referred to by the specified program.

Parameters Limiting the Scope of the Functions		
Program ID		Enter either a unique program ID to limit the function to one program or use asterisk notation to specify a range.
User ID		The scope of a report to programs can be limited by specifying a user ID. Only those programs that have been precompiled from the terminal of the specified user will be processed. Asterisk notation is possible.
From date ... to date		The scope of a report to programs can be limited by specifying a range of dates. Only those programs that have been precompiled in the time specified will be processed.
Short list	Y	Displays a single line of information for each program selected.
	N	A more extensive report is created, containing up to seven categories of information. Enter digits in the Nr column to determine the sequence of the information displayed. Enter zeros to suppress categories that are not to appear.
Save set	Y	Any user defined in Predict can save the types and names of listed programs in a set. This feature is normally used together with the Short list option.See ***Using Sets***.
Format extended description	Y	Any Con-form instructions contained in the extended description of any object will be executed. See *Extended Description Skeleton* in the section *Defaults* in the *Predict administration documentation*.
Suppress empty reports	Y	Suppress reports where no information is found in a particular category.

Verify Application

The following screen is displayed by selecting code A in the XRef menu for a 3GL (not available for static SQL):

```
13:08:58              *****  P R E D I C T  *****              2007-05-31
 Library:  LIBRARY        - Verify Appl. Integrity -




                    Code Function
                    ---- ----------------------------------
                      R  Programs not referenced
                      E  Externals not impl. but referenced
                    ---- ----------------------------------
Code...........: ? (./?)


Save set.......: N (Y,N)




Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Invp  GDAV  Quit  Sets  Rule  Copy  Xref  View  OSet  SPfk  Main  Exit
```

Code	Function	Answers the Question
R	Programs not referenced	Which 3GL programs are not referred to by any other programs written in the same programming language
E	Externals not implemented but referenced	Which external entries of programs written in this 3GL are not referred to by programs written in the same programming language

Operate on Sets

The names and object types of 3GL objects can be saved in sets for further processing. The information contained in the section **Operate on Sets** also applies to 3GL sets.

Maintaining a LIST XREF Profile

Three types of settings can be changed in a LIST XREF profile:

- PF key assignments
- the information to be included in reports created with the Report program function, and the sequence in which the information is to be displayed.
- the functions to be executed by the Verify all function.

See ***Maintaining a LIST XREF Profile***.