software AG

Predict

Introduction

ADABAS & NATURAL

Table of Contents

Preface

This documentation provides an overview of the Predict functions, the objects that can be processed with Predict, and all the handling information you require for using Predict.

Predict Functionality	An overview of the functionality offered by Predict and the objects you can process with these functions. The basic concepts of XRef data are also explained.
Predict User Interface	How to use Predict functions effectively using the tools that are available for routine tasks.

1 About this Documentation

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

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at **https://documenta-tion.softwareag.com**.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at **https://empower.softwareag.com/**.

You can find product information on the Software AG Empower Product Support website at **https://empower.softwareag.com**.

To submit feature/enhancement requests, get information about product availability, and download products, go to **Products**.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the **Knowledge Center**.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at **https://empower.softwareag.com/public_directory.aspx** and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at **http://techcommunity.softwareag.com**. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.

- Access articles, code samples, demos, and tutorials.

- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.

- Link to external websites that discuss open standards and web technology.

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 Functionality

Predict is an active data dictionary system which is used to describe information processing systems. It also provides functions to use this information when designing, implementing, using and maintaining the system.

The following types of information can be stored and maintained:

- What data is stored and where. This data reflects the information structure of an organization.
- Which processing objects work with the data. This data reflects the functional structure of an organization.
- Who creates data and/or objects that process the data. This data reflects the personnel working with the data processing system.

A data dictionary serves as the central repository of information for all people involved in producing, processing and maintaining a software package or data processing application. Information held in the data dictionary can be understood as the submodel of the complete enterprise model dealing with the operational data.

This section contains general information on the object types contained in the Predict metadata structure and the functions provided by Predict to maintain a metadata model and retrieve information from the model. This section covers the following topics:

General Information on Predict Functions

Predict Object Types

Creating Documentation Data

Dummy and Placeholder Objects

Attributes Of Predict Objects

Enforcing Standards

XRef Data

Active Use of Predict Information

Retrieving Information from Predict

Preprocessor

Protecting your Environment with Predict Security

2 General Information on Predict Functions

This diagram provides an overview of the functions available in Predict.

Some functions, for example Retrieval, only process dictionary data. Other functions, for example Administration, process dictionary data and objects in an external environment.

Metastructure of Predict

The following diagrams show the predefined metastructure of the Predict documentation data.

Key

A: is parent of Y

X x ——————→ Y

P: is child of X

A: active association

P: passive association

Y is entered
as Attribute of X

X x – – – – – –→ Y

A:
Decomp
into KY

P:
Composed
by KY

KEYWORD

32

All Object Types

USER

99

OWNER

99

All Object Types

REPORT
LISTING

A: Uses ET

P: Contained in RT

EXTRACT

32
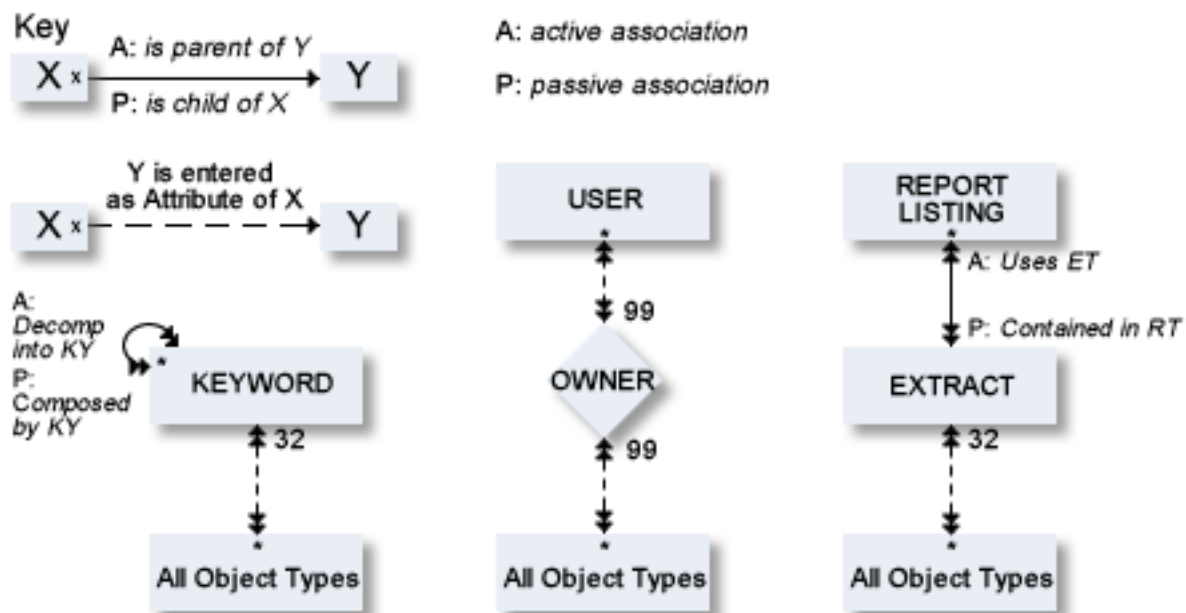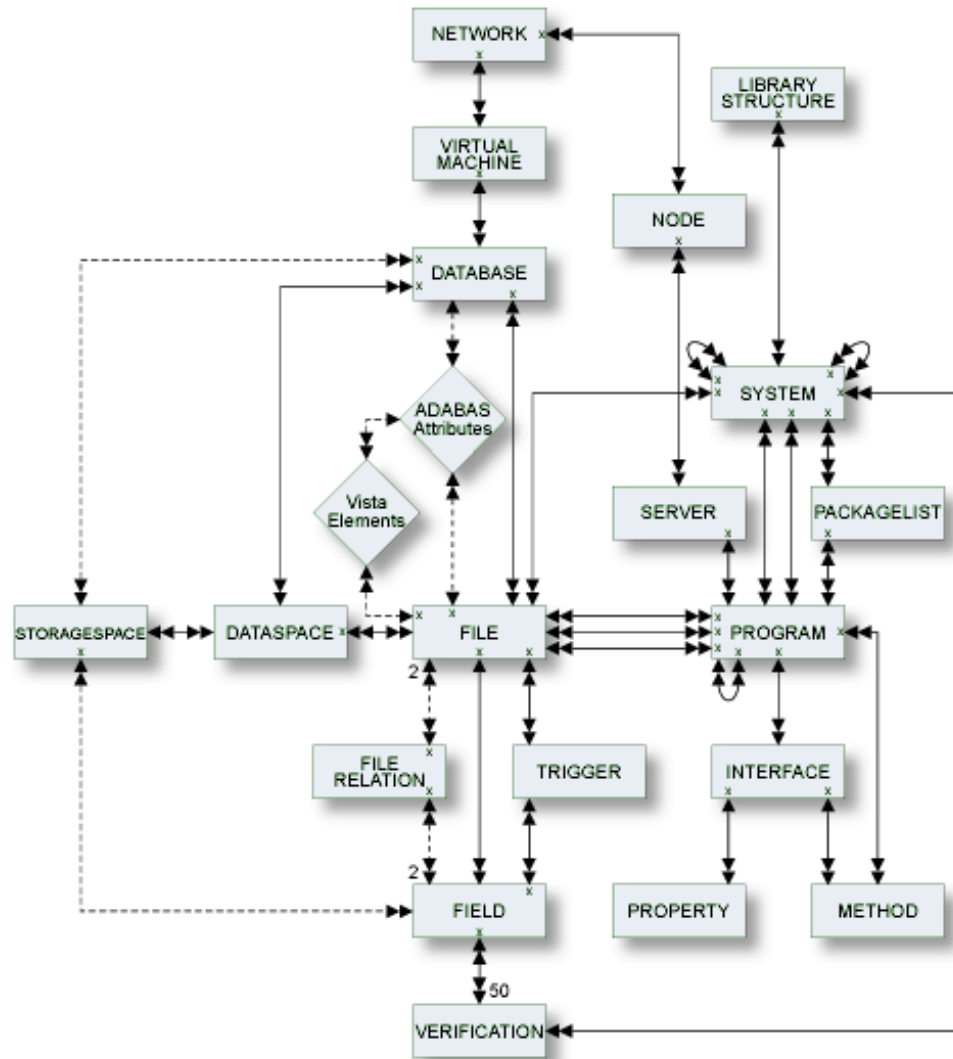
All Object Types

Object Types

Predict object types can be grouped as follows:

- **Objects documenting data structures**
- **Objects documenting data processing objects**
- **Users**
- **Keywords**
- **Extracts**
- **Report Listings**
- **User-defined object types**

Subtypes of Predict Object Types

Predict objects can have subtypes. For example: the Predict object type System has the subtypes Application, Conceptual, 3GL application and DB2 plan. These subtypes of system objects are referred to as system types.

The different types of objects and their subtypes are described in detail in the documentation Predefined Object Types in Predict.

Associations - Documenting the Structure of a System

Objects of an information processing system are organized in parent-child associations.

Associations of objects can be established

- by entering one object in the child list of another object or
- by entering the parent of an object in the parameter in parent of the child object (where in parent is to be substituted by the respective association, for example contained in DA). In this case, an entry in the child list of the parent object is created automatically.

For example:

The file object DEF_1 can be linked as child to Database ABC_1 by adding DEF_1 to the contains FI list of ABC_1.

Alternatively, ABC_1 can be specified as parameter contained in DA when adding or maintaining the file DEF_1.

New types of associations can be defined to link object types.

See *User-Defined Object Types and Association Types*.

Relating Objects Logically

Different objects of an information processing system having the same properties can be related logically in Predict in one of the following ways:

- By assigning the same *owner* to objects, for example by assigning the owner Junior to all the performance-critical programs written by Mr. Junior.

  See *Users/Owners* for more information.
- By assigning the same *keyword* to objects, for example by assigning a keyword performance_critical to programs.

See *Keywords*.

■ By using the same *text string* in the abstracts or descriptions of objects, for example by using performance_critical in the abstract of programs or by using this string in the description of programs.

See *Restrictions*.

■ By placing objects in an *extract*. An extract can contain an unlimited number of other objects. Full retrieval functionality is available for creating extracts, and existing extracts can be processed with set operations Union, Difference, Intersection.

See the section *Extracts*.

Objects Documenting Data Structures

This diagram shows Predict data
and how they are related.
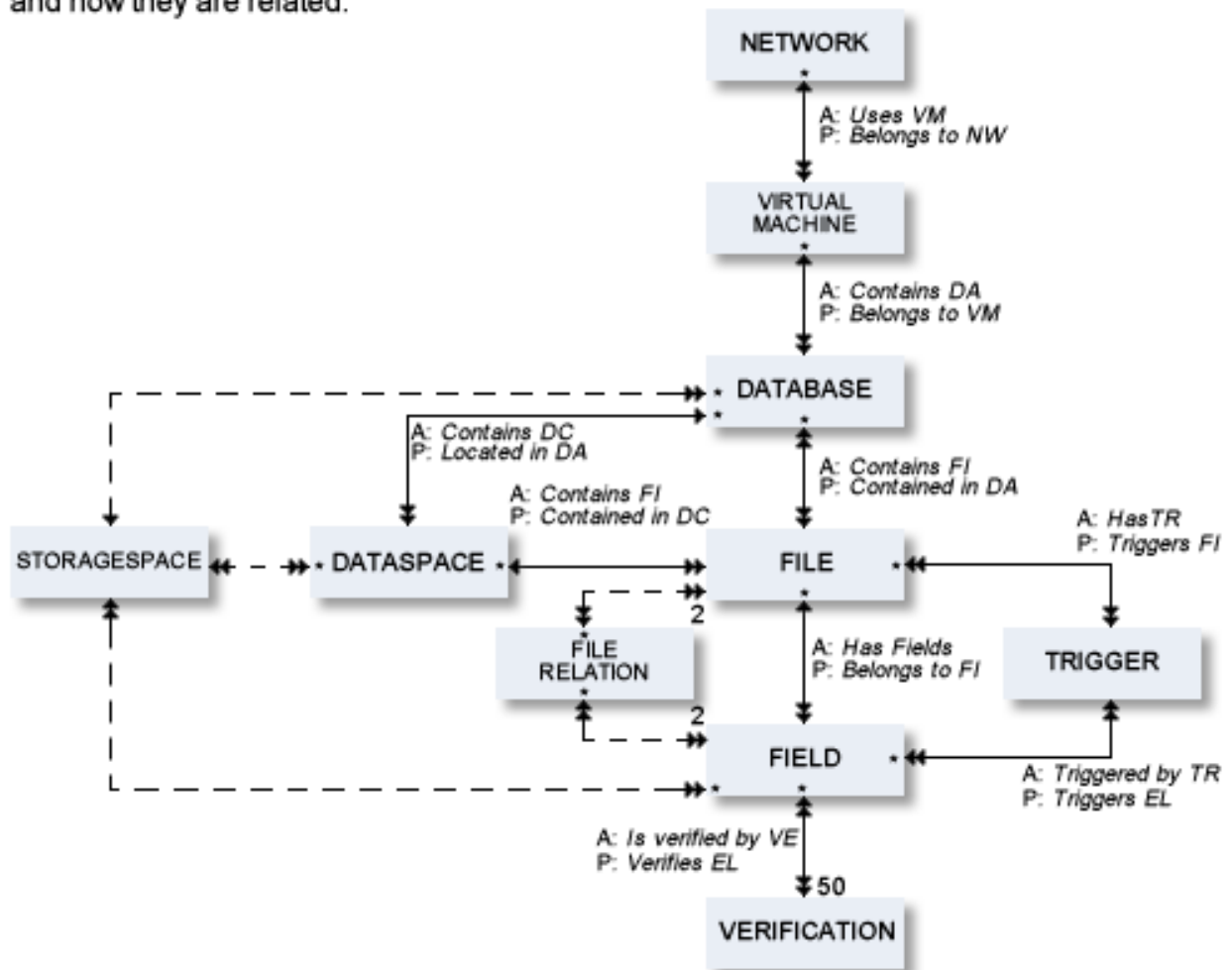
Object Type	Description
Network	All databases belonging to the same network can be accessed. Virtual machines are linked to networks and databases are linked to virtual machines.
Virtual machine	The Predict object virtual machine identifies the hardware and operating system environment of a database.
Database	Various subtypes are provided to document different database systems. The subtype Conceptual can be used to create preliminary Database objects in the design phase.
File	Over two dozen different file types (including userviews) are provided for documenting different data storage systems and application development environments. Different types of files are distinguished:

Object Type	Description
	■ Conceptual files ■ Standard files ■ Physical files ■ Userviews
Field	Field objects can contain all the information on attributes and characteristics of fields. If Verifications of type automatic are linked to a field via *is verified by VE*, these are used automatically in Natural maps which use the field. Fields can be redefined, structured in groups, have synonyms etc.
File relation	Coupling of Adabas files and referential integrity of DB2 tables can be documented in Predict with objects of type File relation.
Dataspace	For the documentation of DB2 tablespaces and SQL dbspaces.
Storagespace	For the documentation of DB2 storagegroups.
Verification	Objects of type verification can contain the Natural code for processing rules. Verifications can be of type documented, conceptual, free, automatic, Natural Construct or SQL. Verifications of type automatic are used automatically by any Natural program using the field (to which the Verification is linked via *verifies EL*) in a map.
Trigger	Triggers of several SQL database systems are documented as trigger objects. They can be linked to tables of type DB2, Ingres, Sybase and Informix and to fields of DB2 tables and views.

Objects Documenting Data Processing Objects

Data dictionary objects of the following types are used to describe objects which process data:
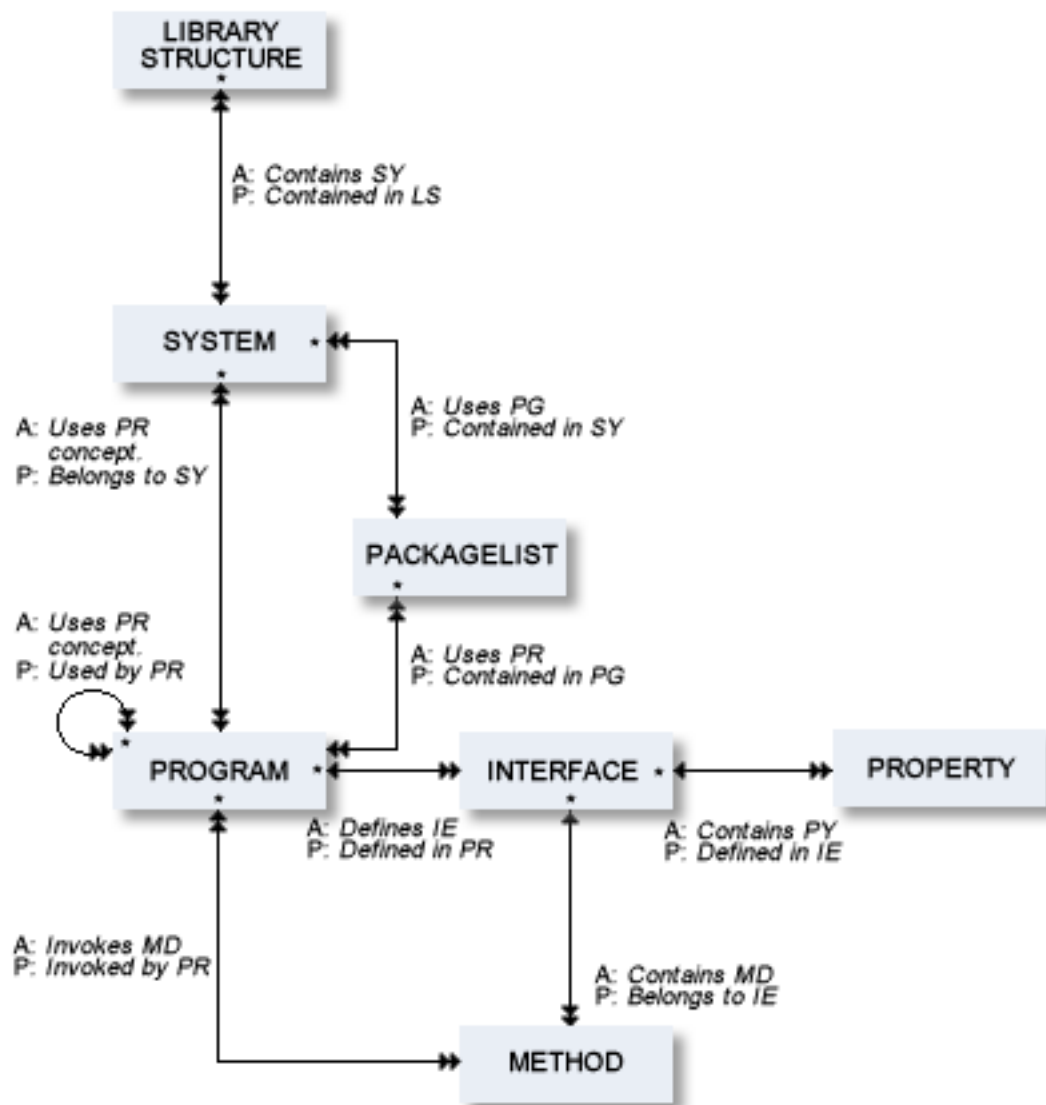
Object Type	Description
System	An application can be documented with a Predict object of type system. The subtype Conceptual can be used to outline a preliminary description of an application in the design phase. Libraries of 3GL programs must be specified as systems.
Packagelist	The Predict object type packagelist is used to document DB2 packages.
Program	Predict differentiates between more than a dozen different types of programs, ranging from Parameter data area to Natural Expert Model. About a dozen different languages are predefined and user-specific languages can be defined.
Interface	Used to document the interfaces of a class.
Method	Used to document the methods of an interface.
Property	Used to document the properties of an interface.

Users/Owners

Information on users and organizational units can be documented using the Predict object type user.

Attributes of the Predict object type user include: ID, name, organizational affiliation, address. One attribute of the Predict object type user is owner. Groups of users that reflect organizational units (for example project teams) can be formed by assigning individual users to an owner. Each user can belong to several owners. Owners can be linked to other types of Predict objects.

Using Owners to Relate Dictionary Objects Logically

It is possible to document who uses an information processing object or is responsible for it by adding an owner to the Owner list of the User and then adding the same owner to the Owner list of the object. This is illustrated in the diagram below. The user Levi-Strauss is related logically to the system Blue, because both the Owner lists of the system object Blue and the User object Levi-Strauss contain the owner Team_2.
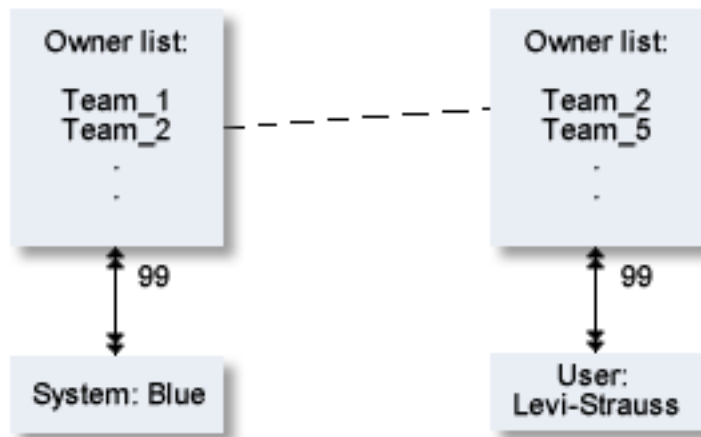
Keywords

Keywords can be used to relate objects logically (for example all objects used in a particular business context or all objects that share certain properties). Objects of different types can be related using keywords.

Using Keywords To Relate Dictionary Objects Logically

The use of keywords to relate objects is illustrated in the following diagram. The program objects SCAN_DESC and REGENER_SYS both have the keyword PERF_CRITICAL in their keyword list.

Predict functions could then be applied to all objects considered performance-critical by specifying the value PERF_CRITICAL with the selection parameter Restrictions / with keyword.

Up to 32 keywords can be assigned to each data dictionary object.
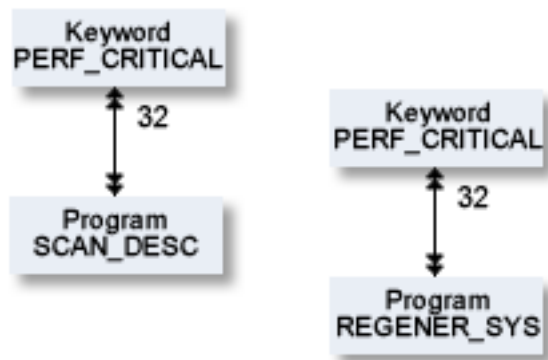
> **Note:** A keyword must have been defined as a Predict object before it can be assigned to other objects.

Extracts

Objects of type extract are used to group objects. This is in particular needed if you want to export objects from Predict using the Predict Coordinator.

Report Listings

Report Listings are log files created automatically during conversion or when a transfer operation of the Predict Coordinator takes place.

User-Defined Object Types and Association Types

In addition to the object and association types delivered with Predict, new object and association types can be defined by data dictionary administrators.
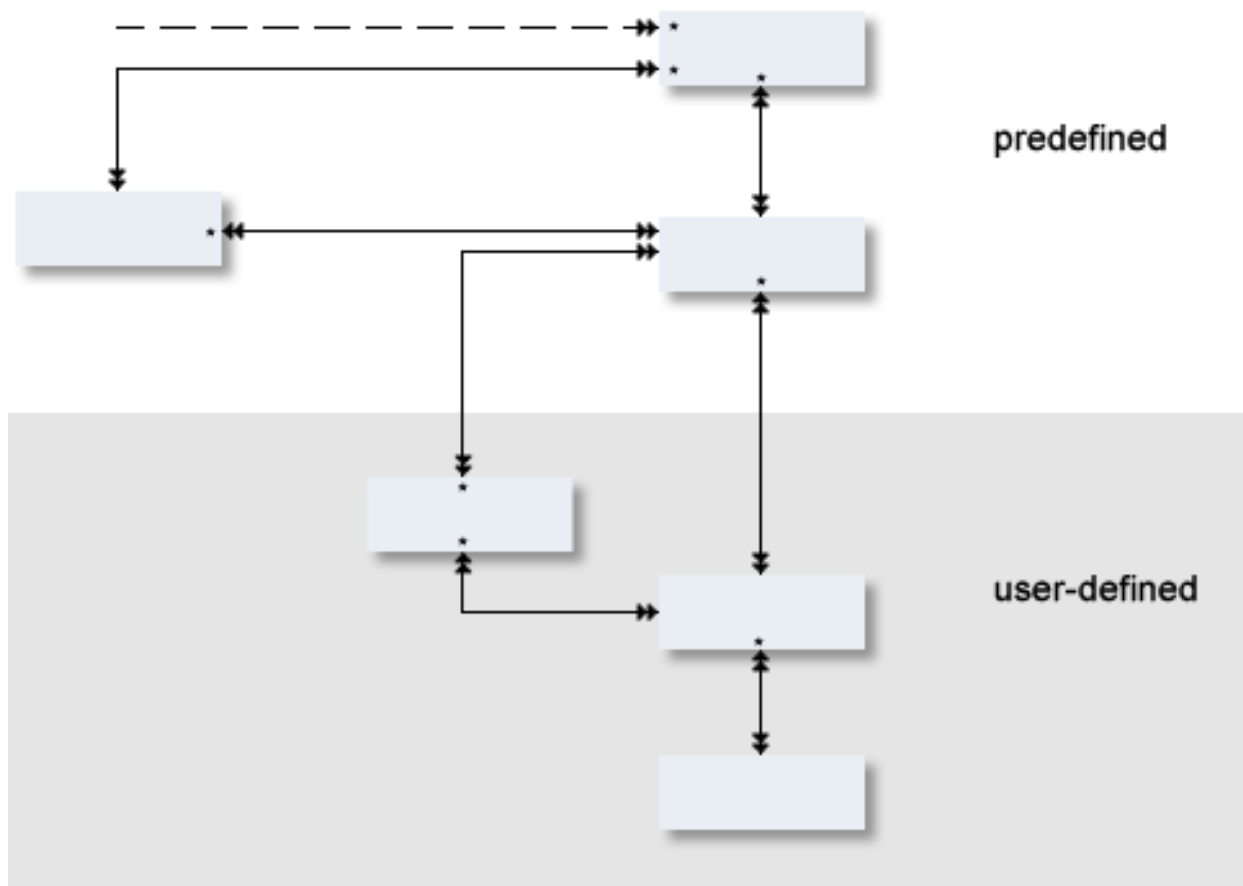
User-defined object types and associations are supported by Predict in the following ways:

- They are administered in a separate library.

- Standard maintenance and retrieval functions can also be applied to objects of user-defined types.

- If the definition of a user-defined object type is changed in the Metadata Administration, all objects are changed accordingly.

See the section *Metadata Administration* in the *Predict Administration documentation* for a description of how user-defined object types and associations are created.

3 Protecting your Environment with Predict Security

The Predict Security System controls access to a Predict environment using security definitions stored in a Natural Security file.

An individual environment can be defined for each user or group of users and protected against unauthorized access.

Predict definitions are not protected in Natural Security as default. This means that when Predict is delivered, each user has access to every object and can execute any Predict function. Predict Security only takes effect when access to objects, object types or functions is explicitly restricted for individual users or groups of users.

Adding Security Definitions

Default Definitions

Standard definitions for Natural Security must be added with the Special Function Maintain NSC Definitions > Add NSC Default Definitions. See *Maintain NSC Definitions* in the section *Special Functions* in the *Predict Administration documentation*.

Additional Security Definitions

If you wish, you can create your own Security definitions for any object in Natural Security - either for an individual user or for a group of users.

Natural Security External Object Types for Predict

The following Natural Security (NSC) external object types and their standard definitions are added in Natural Security with the special function Maintain NSC Definitions > Add NSC Default Definitions.

- Prd-Docu-Objects
- Prd-Ext-Objects
- Prd-3gl-XREF-Library
- Prd-Function

These are described in more detail below.

Prd-Docu-Objects

For this NSC external object type, the default definitions for main object types and object subtypes are automatically added in Natural Security with special function Maintain NSC Definitions > Add NSC Default Definitions. For example: FI and FI-A for files or files of type Adabas. User-defined object types are also added with this function.

All instances must be added manually in Natural Security. With the special function Mass Grant you can create security definitions in Natural Security on the basis of data in Predict. Objects for which you wish to create security definitions must be placed in an extract. See *Mass Grant* in the section *Special Functions* in the *Predict Administration documentation*.

There are three strategies you can follow when protecting objects:

- Protect *Individual objects* Example: The file SALARY can only be read/modified by certain Users.
- Protect a *range of objects* Use naming conventions to group objects and take advantage of asterisk notation in Natural Security.

  Example: User USER1 has been denied READ access to file objects in general, but READ access for files that begin with his User ID. The more specific authorization has priority.
- Protect *all objects* of a particular type.

Other Object Types

With NSC external object type PRD-Docu-Object you can also protect the following:

- Association types
- Object types (for UDEs, only applies to Metadata Administration)
- Retrieval models
- Implementation plans

Prd-Ext-Objects

All external object types that can be processed with Predict functions can be maintained by Predict Security. Instances of this NSC external object type are for example CO (COBOL copy code) or D2 (DB2 database). The instances are automatically added in Natural Security with special function Maintain NSC Definitions > Add NSC Default Definitions.

Security definitions for Prd-Ext-Objects are used to protect functions.

Example: A user without ADD or MODIFY access to object type CO cannot execute the function Generate COBOL Copy Code.

Prd-3gl-XREF-Library

A security check is carried out when you access XRef data in 3GL libraries from Predict (Preprocessor, List XREF for 3GL). This check accesses the security definition for the 8-character library name in Natural Security. If you wish to protect 3GL libraries, you must define security objects of this type manually in Natural Security.

Prd-Function

As a rule, security definitions in Predict are defined at object type or object level. The following areas of Predict do not process any objects in Predict and are therefore protected with objects of NSC external object type Prd-Function in Natural Security.

Resources of the NSC external object type Prd-Function are divided into the following groups:

- Special Functions
- Coordinator
- Defaults, including extended description skeletons
- LIST XREF for 3GL

For this NSC external object type, the instances are automatically added in Natural Security.

Activating Predict Security

The Predict parameter Protect current Predict file in the General Defaults > Protection screen determines whether Predict Security is called. This parameter can be defined for each FDIC file.

Where to find more Information

The Predict Security documentation provides you with all the information you need to set up your Security environment and gives a detailed description of how the individual Predict functions behave when working with objects that are protected with Predict Security.

4 Creating Documentation Data

Data dictionary objects can be created

- manually
- with incorporation functions or the function Redocument program
- with the Load/Import functions of the Predict Coordinator
- using other Software AG systems
- using the Predict Application Programming Interface (API)

The different options are described below.

> **Note:** It is strongly recommended not to write data to the Predict file with any other method. Predict performs comprehensive consistency checks before adding or modifying documentation data. These checks naturally do not apply when writing to the Predict file directly.

Creating Predict Objects Manually

Predict functions to create or maintain object descriptions can be called with commands or using a menu. See *Creating Predict Objects* in the section *Maintenance* in the *Predict Reference documentation*.

Creating Predict Objects from Implemented Objects

- Predict objects documenting data structures and objects of type user can be created from external objects with incorporation functions.
- Predict objects documenting Natural and 3GL programs can be created with the function Redocument program.

The table gives an overview of which objects can be incorporated from implemented code:

System/Language	Type of Object
Adabas	Databases, files
DB2	Databases, Tables/views
Adabas SQL Server	Tables/views
Natural for DL/I	Database descriptions (NDB), User defined fields, (UDF)
Natural	Data definition modules (DDM), Classes, Interfaces, Methods, Programs, Properties, Resources (information is taken from XRef data)
COBOL	Copy Code

System/Language	Type of Object
Super Natural	User information
Natural Security	User information
3GL programs	Programs (information is taken from XRef data)
Following SQL systems Adabas D Informix Ingres Oracle Sybase	Tables/views

Loading Objects with the Predict Coordinator

Data dictionary information which is already available in a suitable machine-readable format can be loaded into a Predict data dictionary with the Predict Coordinator utility. When loading, it is possible to replace matching information which already exists in the data dictionary.

See the Predict Coordinator documentation for more information.

Using Other Software AG Systems

Various other Software AG systems create Predict data, for example Predict Case, or Natural Construct.

See the documentation of the respective products for more information.

5   Dummy and Placeholder Objects

Dummies

Each component of an information processing system is related to other components (for example, a program uses a file, a field is contained in a file). A dummy object is an object which has not yet been created but is referenced by another object.

A dummy 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. A record is also stored physically in the Predict file for this dummy object.

- When an object that is linked to another object is loaded/imported with the Predict Coordinator *without* internal ID, and the referenced object is not loaded/imported 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 below.)

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

See *Support of Dummy and Placeholder Objects* in the section *Maintenance* in the *Predict Reference documentation* for the rules which apply when processing dummies with maintenance functions.

Placeholders

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

This placeholder reserves 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 the "real" object.

6 Attributes Of Predict Objects

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

- identifier

- abstract

- extended description

- Keywords

- owner list

- list of child objects

- type-specific attributes, for example a data dictionary object of type File includes information about the file's organization and attributes; an object of type Program includes information about the programming language and the authors.
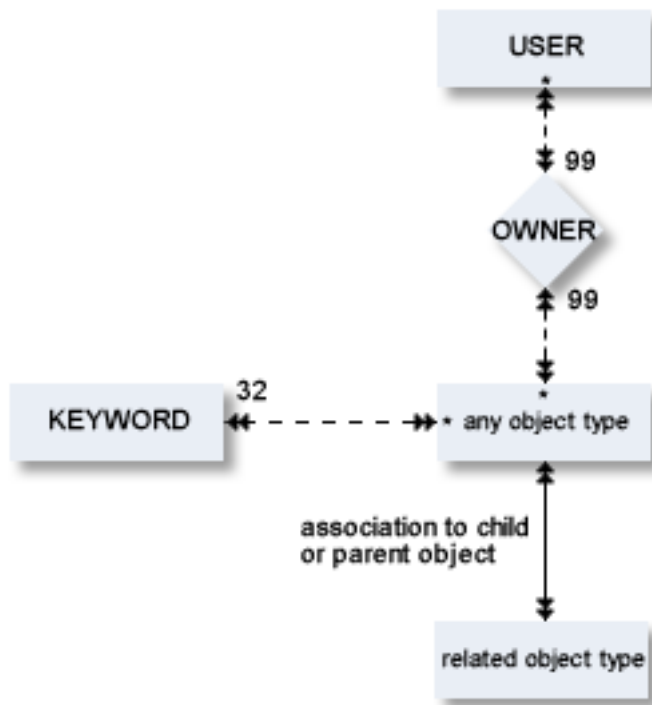
Where to find Detailed Information

See *Creating Predict Objects* and *Creating and Modifying Type-Specific Attributes* in the section *Maintenance* in the *Predict Reference documentation* for a detailed description of the attributes applying to all types of objects. Type-specific attributes are described in the Predefined Object Types in Predict documentation.

7   Enforcing Standards

Predict offers the following options to enforce standards for the documentation of data structures and information processing systems in Predict.

Forcing/Disallowing Extended Descriptions and Owner/Child Lists

```
   09:34:49              *****  P R E D I C T  *****             2007-05-31
                       - Modify object type definition -

   Object type code ... CE                          Modified 2007-05-31 at 13:12
                                                               by ULH


   Object type attributes                    Default related object types
     Internal code ... 1000018                  Default Parent ....*
     Name ........... CHNG-ENHANCEMENT          Default Child .....*
     Title .......... Chng-enhancement
     Object type no .* 1012000
   Edit owner ...............* D   Disallowed
   Edit description .........* F   Force
   Check description ......... Y   (Y,N)
   Object ID length .......... 32  (1-32)
   Disallowed characters .....
   Natural naming convention . N   (Y,N)


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

Creation of owner lists and extended descriptions can be disallowed or enforced with the parameters Edit owner and Edit description. These parameters can be set in the Metadata Administration (see screen above) or in Predict profiles.

If neither force nor disallowed is specified for owner lists, extended descriptions, it is up to the Predict user to create or modify these attributes.

Protecting Parts of Default Extended Descriptions - Skeletons

A default extended description skeleton can be defined for any subtype of each object type with the function Extended description skeleton in the Modify Defaults menu. Parts of a default extended description or complete descriptions can be protected. Protection of descriptions is possible for subtypes of object types.

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

8 XRef Data

XRef data is stored in the Predict system file.

XRef data documents objects of an application that have already been implemented. Data of this type mirrors the "real world" of the implementation, providing a comprehensive summary of information for all implemented members of an application.

Name references will be evaluated during cross reference analysis to reflect the functional structure in an environment you specify.

XRef data can be used to retrieve information on individual members or to retrieve comprehensive information about applications as a whole:

- its invocation structure
- the resources it uses
- its consistency, completeness and correctness.

Contents of XRef Data for Natural

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.

Benefits of XRef Data

The application development process can be advanced significantly at almost all phases by using XRef data:

Phase	Benefits of XRef Data
Design	Identifying programs of other applications that might possibly be used in the planned application.
Implementation	Finding resources that are not (or no longer) used. Finding errors and inconsistencies that cause errors at runtime, for example calls to non-existent subroutines. Supervising the implementation state of an application.
Testing	Detecting incomplete documentation.

Phase	Benefits of XRef Data
	Detecting differences between design and implementation. Detecting programming errors.
Production	Reporting all information about the programs that are in production.
Maintenance	Estimating the impact of changes. Discovering the invocation structure of an application.

Creating XRef Data for Natural Members

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.

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.

See the section *LIST XREF for Natural* in the *Predict Reference documentation* for a complete description of where and when XRef data for Natural members is created.

9 Active Use of Predict Information

Predict offers a variety of functions reducing the effort necessary to implement an application.

External data definition objects generated by Predict and stored in Predict can be maintained with administration functions.

Where to find Detailed Information

Generation functions are described in the section *Generation* in the *External Objects in Predict documentation*.

Section *Administration of External Objects* in the *External Objects in Predict documentation* describes how external objects are administrated.

Generating and Maintaining Natural Processing Rules

Natural processing rules can be generated from verifications objects. If a DDM has been generated from a file containing fields with verifications linked via `is verified by VE`, and processing rules of status automatic have been generated from any of these verifications, Natural will automatically apply the processing rule to a map.

Processing rules are stored in the Predict system file for use by Natural.

Where to find Detailed Information

See the section *Verifications and Processing Rules* in the *Predict and Other Systems documentation* for a detailed description of how processing rules are generated and maintained.

Generating Natural Data Definition Modules

Natural data definition modules (DDMs) can be generated from Predict file objects. Generated DDMs are stored in Predict.

Where to find Detailed Information

See *DDMs* in the section *Generation* in the *External Objects in Predict documentation*.

Generating Other Types of Data Definition Objects

Two types of external objects are distinguished:

- external objects owned by Predict
- external objects not owned by Predict

The main differences between the two types are listed below.

See the section *Handling of External and Documentation Objects* in the *External Objects in Predict documentation* for more information.

Storage/Administrating

- External objects *owned by* Predict are stored in Predict and are administered exclusively with Predict functions.
- External objects *not owned by* Predict are stored in the application environment and could be administered with utilities of the application environment. However, administration of external objects stored in Predict with utilities of the application environment is not recommended.

Purging

- External objects *owned by* Predict are dependent on the Predict documentation objects from which they were generated: if the documentation object is deleted, the generated objects are deleted as well.
- External objects *not owned by* Predict are not deleted automatically if the documentation object is deleted.

When used?

- External objects *owned by* Predict are in card format or Copy Code. External objects of these types are mainly used at compile time.
- Most external objects *not owned by* Predict are implemented in the application environment. External objects of these types are mainly used at run time.

External Objects Owned by Predict

The following types of external objects are owned by Predict:

- 3GL Copy/Include Code (C, COBOL, Assembler, FORTRAN, PL/I)

- Adabas invert, compression and security definitions (ADAINV, ADAWAN/ADACMP/ADAFDU, ADASCR)

- Transparency Table for Adabas Bridge for VSAM.

- SQL CREATE statements

There are different options to output generated external data definition objects of the above types:

- External objects can be generated temporarily (option 1)

- External objects can be stored as members in a Natural library of the Predict system file (option 2). Objects can then later be punched to the application development environment.

- External objects can be stored additionally in a workfile or - with Entire System Server - as an operating system member (option 3)

Where to find Detailed Information

See *Storage of External Objects Owned by Predict* in the section *Generation* in the *External Objects in Predict documentation*.

External Objects Not Owned by Predict

The following types of external data definition objects are not owned by Predict: they belong to an application (development) environment.

- Databases (Adabas, DB2, IMS/DL/I)

- Vista translation tables

- DB2 tablespaces and storagegroups

- Files, tables and views (Adabas, Adabas SQL, DB2)

- Natural for DL/I user defined fields (UDFs)

- Natural DDMs (including Natural security definitions and/or Super Natural files).

Connection of External and Predict Documentation Objects

Predict connects external objects and documentation objects if an external object has been generated from a documentation object or - vice versa - a documentation object has been incorporated from an external object. Connecting external and documentation objects helps ensure the consistency of the documentation and an application: documentation objects and - to a certain extent - external objects that are connected are protected from being deleted or modified.

Where to find Detailed Information

See the section *Generation* in the *External Objects in Predict documentation*.

Impact of the Connection of External and Documentation Objects

The connection of external objects and documentation objects affects the following functions:

- Administration: external objects connected to documentation objects can be protected.
- Maintenance: Predict protects documentation objects that are connected to an external object from being purged accidentally.
- Comparison: to compare an external object and a documentation object, both have to be connected.

Disconnecting Objects

External objects and documentation objects can be disconnected with the administration functions Disconnect implementation and Purge implementation.

See the section *Administration of External Objects* in the *External Objects in Predict documentation*.

External objects owned by Predict need not be connected: because these types of objects can only be administrated with Predict functions, the consistency with documentation objects is not endangered.

Using Documentation Data In Online Help Texts

Predict provides a user-modifiable, generally applicable help facility: routines that make data stored in dictionary objects available as help information both at function level (map, program) and at field level.
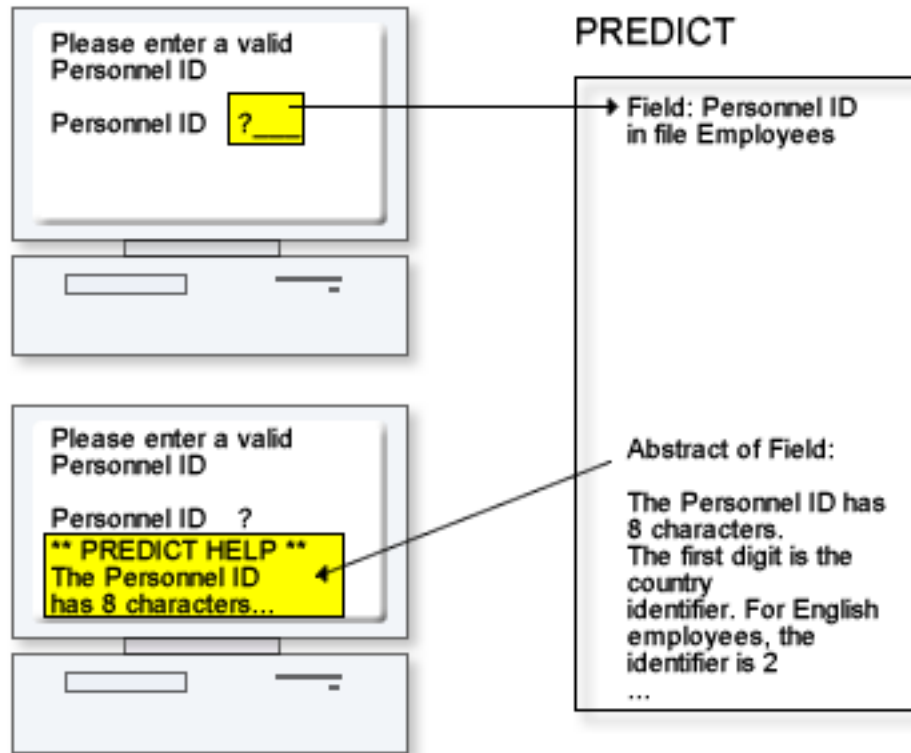
Any combination of the following information stored with Predict objects can be displayed in help screens:

- The abstract and extended description of a map, field, database, file relation and system.
- The abstract, extended descriptions and permitted values of verifications linked to a field via standard link *verifies EL*.

See the section *SYSHELP* in the *Predict Reference documentation*.

10 Retrieving Information from Predict

Retrieving Information from Documentation Data

The Predict data dictionary system helps to manage complex information processing systems by providing functions that retrieve information on its internal structure and properties of its components. Information can be retrieved online or in batch mode. Essentially three types of information can be retrieved from the dictionary.

■ **Information on Attributes of Individual Objects**
Most retrieval types report on the attributes of individual objects. Information on any number of objects can be retrieved in one run.

■ **Information on the Associations of Objects**
Retrieving information on the associations of objects means retrieving information on the structure of the information processing system, for example which programs belong to a system, and which files are used by these programs.

■ **Information on the Implementation of Objects**
For example whether an object that is documented in the dictionary is already implemented. Active retrieval functions as well as features of standard retrieval types provide information on the implementation of objects. The section **Retrieving Information from XRef Data** describes in more detail all options to retrieve information on the implementation of a system and compare this information with documentation data.

Overview of Retrieval Options

To determine precisely which information is to be retrieved from the dictionary the following types of parameters are used:

■ **Selection Criteria**
Retrieval operations can be executed for single objects or for groups of objects. Selection criteria determine which objects are to be included in a report.

■ **Retrieval Type**
The retrieval type determines the type of information to be retrieved from the dictionary.

■ **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.

■ **Output Options**
The selection criteria and the retrieval type determine the information collected by Predict. Output options can then be used to determine which information is actually contained in a report.

- ■ **Retrieval Models**
  The retrieval type Execute retrieval model reports on the structure of an information processing system. It can be specified exactly which part of a metadata model is to be evaluated.

Where to find Detailed Information

See the section *Retrieval* in the *Predict Reference documentation*.

Documentation Data Refers to XRef Data and Vice Versa

Predict retrieval and active retrieval functions evaluate XRef data *and* documentation data, whereby retrieval functions start on the documentation side and active retrieval functions start on the XRef side.

To evaluate (and compare) information on objects stored as documentation data and as XRef data, Predict must know which documentation objects belong to which record in XRef data. This connection between the documentation and the 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 is used in several libraries, multiple documentation of this member can be avoided by omitting the library name. Predict then finds out for itself all the libraries in which this member exists.
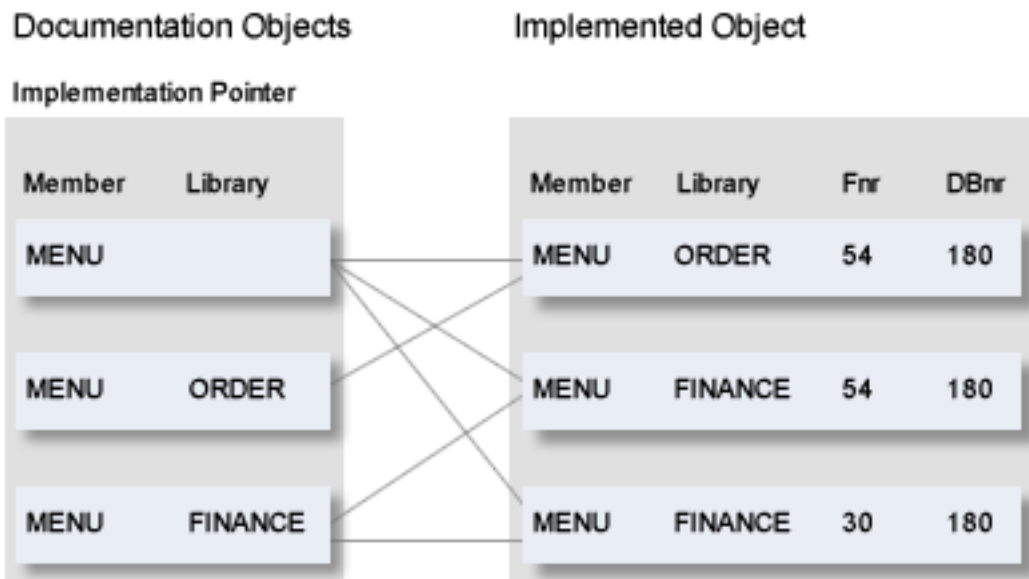
Forcing/Disallowing an Implementation Pointer

The parameters Implementation Library and Implementation DBNR/FNR can be set in the Maintenance Options screen of the General Defaults menu and apply when maintaining program objects.

F

Forced: Library and/or DBnr/Fnr must be specified.

A

Allowed: Library and/or DBnr/Fnr of the implemented program documented with the Predict program object can be specified.

D

Disallowed: Library and/or DBNR/FNR must not be specified. For third generation languages, library *SYSALL* must be used.

Where to find Detailed Information

See *Using Implementation Pointers to Establish a Connection between Documentation and XRef Data* in the section *Active Retrieval* in the *Predict Reference documentation*.

Retrieving Information from XRef Data

Documentation data alone does not necessarily say much about what has been implemented. It can exist without a single line of code having been written or differ significantly from what has been implemented. It is the XRef data that completes the picture by documenting what has actually been implemented.

XRef data is of interest to various groups of users: programmers, system analysts, project leaders, database administrators, data dictionary administrators, computer auditors, etc. However, each of these groups requires the XRef data evaluated in different ways. The following two sections contain some hints on how XRef data might be used.

Evaluating XRef Data for a Specific Application

The `LIST XREF` command provides information on a specific application which can be used to monitor the application. Errors can be detected (unused programs) and opportunities for optimization (reducing the number of variables) are given in early development phases. The impact of program changes can be estimated easily (for example, how many programs must be modified when a field or variable is changed).

The following kinds of evaluation are available:

■ How often an object (program, variable, Copy Code, error number or processing rule) is used;

■ How a program is used, including recursive invocation structures;

- Which objects are used by a program;

- Which objects are available but not used (for example, which programs are not invoked by any other programs);

- Which files are used, which of their fields are used, and how the fields are used (as counter fields or for reading, storing, searching or updating) in data areas;

- All information about a program (use of programs, files, fields, variables, work files, retained sets, Copy Code, printers and error numbers, the source code of the program and its documentation in the Predict data dictionary.

- Which programs were cataloged when, by whom and from which terminal.

The functions of Natural LIST XREF can be called from a menu or with a command.

Monitoring the Implementation Process

Design: PREDICT   **Implementation: NATURAL**

Program:
XYZ

Program:
MAP1

File:
VIEW1

Inconsistencies

FIND VIEW1 WITH

INPUT USING MAP "MAP1"

UPDATE

Field:
FIELD1

Field:
FIELD2

Inconsistencies   Consistent

Xref Data

Compare   Catalog

Program: XYZ

Files	Fields	Programs
VIEW1	FIELD1	MAP1
	FIELD2	
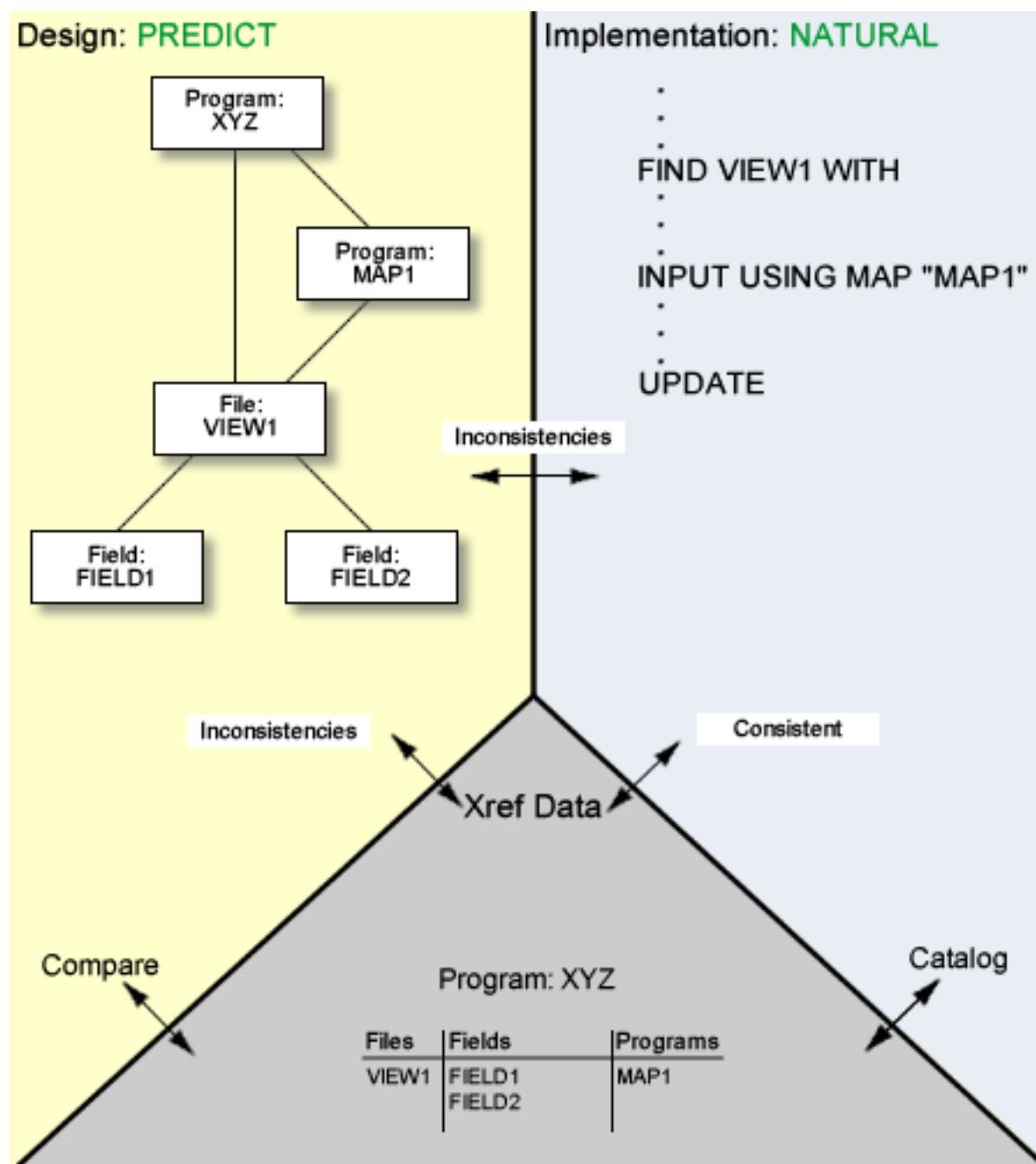
Evaluating XRef Data Across Several Applications

Predict active retrieval functions evaluate XRef data across several applications.

When documentation data is evaluated, implementation data can also be output if required. The date when the documentation was last altered shows if changes to programs have been carried out in the documentation. In this way, the state of both the documentation and the implementation of an application can be evaluated.

Comparing the documentation data and the implementation also allows the detection of possible errors, such as descriptor fields that are not used. Section *Comparison* in the *External Objects in Predict documentation* describes how documentation and implementation are compared.

According to the type of function, the evaluation can be restricted either by specifying implementation data (member name, library name) or documentation data (object name) as selection criteria.

The following types of evaluation are available:

- Which objects are implemented, including an indication of the documented entries;
- How often an object is used (field, file, external program or verification);
- Which objects are implemented but not documented;
- Which objects are documented but not implemented;
- Which members use documented objects (verifications, files, and fields);
- The files defined as being used by a member and the files that are actually used;
- The programs defined as being used in an application and the members that are actually used.

Using these evaluations, differences between the real implementation of a system and its documentation in Predict can be detected and resolved (see also the diagram above).

How XRef Data can be Evaluated and where the Different Methods are Described

XRef data can be evaluated on its own or in conjunction with descriptive information stored in data dictionary objects.

Evaluating XRef Data without Documentation Objects

Two methods are provided for retrieving XRef data independently of information stored in data dictionary objects:

- XRef data written for members in the current Natural library is evaluated with the LIST XREF command. This method is described in the section *LIST XREF for Natural* in the *Predict Reference documentation*.

  See also *Evaluating XRef Data for a Specific Application*.

- XRef data that was generated by Adabas Native SQL, Adabas SQL Server, the Predict preprocessor or the function CREATE DBRM of Natural for DB2 is evaluated with the LIST XREF for 3GL command of Predict. The LIST XREF command for 3GL is described in the section *LIST XREF for Third Generation Languages* in the *Predict Reference documentation*.

Evaluating XRef Data Together with Documentation Data

Again, two methods can be used to evaluate XRef data in connection with information stored in Predict documentation objects:

- Predict active retrieval functions compare XRef data with the corresponding information in Predict documentation objects. This method is described in the section *Active Retrieval* in the *Predict Reference documentation*.

  See also *Evaluating XRef Data Across Several Applications*.

- With Predict retrieval functions, set the parameter Show implementation in the output options to Y to display the implementation of a documentation object, or set the parameter Mark implementation to Y to mark implemented objects with an asterisk. Information on the implementation is taken from XRef data.

When is an Object Regarded as Implemented?

What is meant by implemented depends on the object type.

The following *databases* are regarded as implemented:

- type A and connected to a physical Adabas database,

- type D and connected to a physical DB2 database,

- type P with a DBnr defined in the NTDB macro as an Entire System Server database,

- type I.

dataspace
    A dataspace is regarded as implemented if a DB2 tablespace or SQL/DS DBspace has been generated from the Predict dataspace.

file
    A file is regarded as implemented if

    - any kind of copy code has been generated for a file or

    - the file is connected to an external object.

program
    A program is regarded as implemented if XRef data exists.

> **Note:** This does not apply to programs of type R (SQL procedure) and U (Database function). Both these types are regarded as implemented, if a DB2 procedure or function has been generated from the Predict program object.

storagespace

A storagespace is regarded as implemented if a DB2 storagegroup has been generated from the Predict storagespace.

system

A system is regarded as implemented if XRef data exists for a program in the library documented by the system.

verification

A verification is regarded as implemented if the rule of the verification is used in a map and XRef data exists.

Using Sets

Retrieval functions performed on XRef data often produce a list of objects that needs further processing, such as programs to be maintained or cataloged. Predict can save the output list of XRef data retrieval in a set for further processing. Sets can be used to share work among members of a project team and for communicating exactly what work needs to be done.

The following rules apply when working with sets:

- Sets are created by setting the Save set option in a retrieval menu to Y or by using the function Create new sets of LIST XREF.

- Sets can contain Natural objects of all types. Members stored in sets can be cataloged, checked or stowed, their contents can be edited or listed directly from the set.

- 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 Natural library. Any user defined in Predict can create and use up to ninety-nine sets in each library.

> **Note:** Sets can also be used in the Natural utilities NATUNLD and SYSMAIN and in Natural ISPF.

Example for the Use of Sets

Whenever a Copy Code is changed, all programs that use the Copy Code may need to be changed accordingly. The LIST XREF function Copy code referenced in programs can be used to find out which programs are affected. If the types and names of these programs are saved as a set, a single function can then be used to edit the contents of all the programs, one after another, and change them to match the changed Copy Code.

All members contained in the set can then be recataloged using a single function.

Where to find Detailed Information

Section *LIST XREF for Natural* in the *Predict Reference documentation* contains a complete description of handling sets.

11 Preprocessor

The Predict preprocessor offers two main options:

- XRef data on the use of copy/include code and on the call of external members in 3GL programs is written. For Assembler, preprocessor statements have to be included into source code to perform this operation.

  This option is completely independent from Predict documentation objects.

- Data definitions generated from Predict file objects can be included into COBOL, PL/I and Assembler (BAL, 370-Assembler) programs, and XRef data documenting the use of these definitions is written.

  Data definitions generated from Predict objects can be included in two ways:

  - the preprocessor copies file layouts into a source program that have already been generated in Predict as copy code.

  - the preprocessor generates file layouts from Predict file objects and copies these file layouts into a source program.

  Two types of definitions can be included:

  - record buffer structures
  - format buffer structures

II Predict User Interface

Effective use of a data dictionary system is significantly influenced by the following points:

- The design of the user interface helps the user understand how functions work and how functions are called.

- Information guiding the beginner must be available, but experienced users must be able to access functions directly without having to pass too many menus.

- Functions and utilities are available to help with routine work.

  In accordance with the above objectives, the Predict user interface offers a variety of options to facilitate work in the main aspects of using a data dictionary system:

  - creating and maintaining dictionary data

  - retrieving information from the dictionary

  - using the active functions of Predict that help with the implementation of an information processing system

  - performing data dictionary administration tasks

This section covers the following topics:

Starting and Quitting Predict
Using Menus, Commands or the Workplan
Predict Help System
Menu Dialog
Selecting Objects for Processing
Using the Workplan
Copying Text from Other Sources
Selecting Objects for Link Lists
Customizing Predict with Profiles

12 Starting and Quitting Predict

Predict is called with:

```
LOGON SYSDIC MENU
```

or with:

```
SYSDIC
```

Predict is stopped by entering

```
END
```

or

```
.
```

or by pressing the PF key for quitting in one of the Predict main menus.

13   Using Menus, Commands or the Workplan

Functions can be called from a menu, with a command, from the workplan or in batch jobs.

Using Menus

Executing a function from a menu may be appropriate for users who are less familiar with Predict or who use Predict only infrequently. Valid input options are indicated and users are guided through the sequence in which input is required.

Experienced users may prefer to execute functions from menus when using functions with which they are not familiar.

Any Predict menu can be displayed in two ways:

- With a command. Example: RETRIEVE FILE will display the File Retrieval Menu.
- By entering a function and object code in a main menu Example: function code R and object code FI to display the File Retrieval Menu.

Using Commands

Using commands, functions can be called in a more compact and time-saving form. Complete command sequences, including all parameters, can be entered.

Commands are entered at the Command ===> prompt at the bottom of the screen. Commands are executed by pressing ENTER.

After a function has been executed with a command, Predict displays the menu from which the function is called in menu dialog. For example, if the command DISPLAY FILE is entered, the File Retrieval menu is displayed after successful execution of the function.

A message on the Natural message line indicates successful execution or failure.

Using the Workplan

The workplan is useful whenever a large number of objects is to be processed.

For example: If objects for all dummies (associations to objects that are not yet created in the dictionary) in the dictionary are to be created, the dummies can be listed with a retrieval operation and be put into the workplan together with an Add command. By executing the Add command from the workplan, objects for all dummies are created easily.

See *Using the Workplan*.

Using Menus, Commands and the Workplan Together

There is no obligation to favor menu dialog or entering commands in advance. Use the method which seems more appropriate.

If both a function code of a menu and a command are entered, the command takes precedence.

The menu dialog and the use of the workplan are described in this section. Calling functions with commands is described in the section *Predict Commands* in the *Predict Reference documentation*.

Operating Predict in Batch Mode

Most Predict commands can be called from batch jobs.

See the section *Predict in Batch Mode* in the *Predict Administration documentation*.

Interrupting the Execution of a Function

The execution of a function can be stopped by executing the command STOP. This command must be assigned to a PF key.

See *PF Keys*.

Entering a period (.) in the Code/Retrieval type field of any menu skips back to the next higher menu level.

Handling of Errors

Successful execution of a function (or failure) is indicated with a message on the Natural message line.

By entering ? DIC *nnnn* (*nnnn* = error number) a more detailed description of the error can be displayed. Example: ? DIC1234 gives additional information on Predict error 1234.

? *nnnn* gives additional information on Natural error messages.

14   Predict Help System

Predict offers different forms of passive and active help.

■ **Active Help - Selection Windows**
Active help is given if parameter values needed to execute a function are missing.

■ **Passive Help - Online Help Text**
Passive help provides descriptive information on functions.

Online help information can be displayed in a context-sensitive form

■ by entering a question mark in the first field of any menu. See **Using Online Help Context Specifically**.

■ from the Help Main Menu. See **Using Online Help via the Help Main Menu**.

Using Active Help - Selection Windows

Active help is given if parameter values needed to execute a function are missing. A selection window containing all possible parameter values is displayed.

Active Help is available for all input fields marked with an asterisk. If an asterisk is entered in such a field a selection window is displayed. A selection window will also be displayed if a parameter is mandatory and the field is left blank or an invalid value is entered.

A sample active help window is shown below. An asterisk has been entered in the field File type of the File Maintenance screen. The valid file types are displayed in a selection window.

```
17:06:53                 *****  P R E D I C T  *****                2007-05-31
Plan   0                   - (FI) File Maintenance -             Profile SYSTEM
                                              +Top-------File-types----------+
 Function                            Functio ! _ A     Adabas file          !
                                             ! _ AT    Adabas cluster table  !
A  Add a file                        L  Link ! _ B     Adabas SQL view       !
C  Copy file                         S  Sele ! _ C     Conceptual file       !
M  Modify file                       B  Push ! _ D     DB2 table             !
N  Rename/Renumber file              F  Forc ! _ E     DB2 view              !
P  Purge file                        J  Phys ! _ F     rdb file              !
D  Display file                      K  Modi ! _ I     IMS segment           !
                                             ! _ J     IMS seg. layout       !
Function .........                           ! _ K     IMS userview          !
File ID ..........                           ! _ L     Logical VSAM          !
Copy ID ..........                           ! _ M     ISAM file             !
Contained in DA ..                           ! _ O     Other file            !
External name ....                           ! _ P     Entire Sys. Server    !
                                             ! _ Q     Sys. Server userview  !
Restrictions ....*    Profile Default ,used  ! _ R     Logical VSAM view     !
                                             ! _ S     Sequential file       !
Command ===>                                 ! _ T     RMS file              !
                                             !Command ==> +_____            !
                                             +More-------------------------+
```

The following rules apply when using selection windows.

▪ A list of valid values may exceed the size of the window. All, Top, More or Bottom in the top or bottom line indicate if and how a list of values exceeds a window. The following commands can be entered in the command field of a selection window:

+	scroll down one window
++	scroll to bottom of list
-	scroll up one window
--, top	scroll to top of list
PF3 or .	quit window

> **Note:** The commands ++ and -- are not available if the data displayed in a selection window is read from the dictionary.

▪ Values can be chosen from selection windows in different ways:

  ▪ by placing the cursor on the desired value and pressing ENTER

  ▪ by marking them in the first column and pressing ENTER.

Selection windows are especially useful when selecting objects from the dictionary. Asterisk notation can then be used. All objects that can be specified in the specific context are displayed.

See also the section **Using Selection Windows** for an overview of when selection windows can be used.

■ Selection windows in which more than one code, value or option can be chosen are closed with PF3 or '.'. For example: the Output options window. Selection windows in which one code or value is chosen are closed by confirming the selection with ENTER. For example: the File types selection window shown in the screen above.

> **Note:** It is not possible to select a value by entering the respective code in the command field (==>) in the bottom line of selection windows.

Using Passive Help - Online Help Text

Context-sensitive online help information can be displayed by entering a question mark in the first field of a menu or from the Help Main Menu. Both forms are described in the sections below. See the following list for an overview of topics:

- Online Help and Software AG Editor
- Customizing Online Help
- Using Online Help Context Specifically
- Using Online Help via the Help Main Menu
- Display Help Texts
- Functions in the Help Main Menu

A topic list is displayed at the beginning of most help texts. To get more information on the topic you are interested in, position the cursor on that topic and press ENTER.

```
 13:27:47                   *****  P R E D I C T  *****                    2007-05-31
 Help   1                      - (PR) Program Maintenance -
 -----------------------------------------------------------------------------
    Program Maintenance Menu
    -----------------------
  >> General description <<
  >> Add a program (Code A) <<
  >> Copy program (Code C) <<
  >> Modify program (Code M) <<
  >> Rename program (Code N) <<
  >> Purge program (Code P) <<
  >> Edit description of a program (Code W) <<
  >> Display program (Code D) <<
  >> Link children (Code L) <<
  >> Edit owners of a program (Code O) <<
  >> Edit entry-points (Code R) <<
  >> Select program from a list (Code S) <<
  >> Redocument program (Code X) <<
  >> Edit procedure code of a program (Code Y) <<


  Command ===>                                                 Scroll ==> CSR
  Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                 Quit          RFind Flip  -     +          Left  Right
                                                   line 1 of ... (line 4 on screen)
```

Online Help and Software AG Editor

The Predict online help system uses the enhanced scrolling and scan functionality of the Software AG Editor. For more information see *Software AG Editor* in the section *Editors in Predict* in the *Predict Reference documentation*.

Customizing Online Help

Online help texts can be changed by users. A special editor is provided for this purpose (see *Maintain Predict Help Texts* in the section *Special Functions* in the *Predict Administration documentation*).

Using Online Help Context Specifically

■ Each menu has an associated help text that describes the function or functions invoked from that screen. The help text can be displayed by entering a question mark in the first field of the menu.

■ A question mark entered in the command line displays a series of help screens which explain the syntax and functions of the commands.

Using Online Help via the Help Main Menu

A more general access to the Predict help system is provided via the Help Main Menu which is displayed by entering a question mark in the Function field of the Predict main menu.

```
  13:35:34              *****  P R E D I C T  *****                    2007-05-31
  Plan  10                      - Help main menu -

   Help information                        Help information

   G    PREDICT - general overview         C    PREDICT direct commands
   W    What is new ?
   ?    Help with Help                     P    PREDICT error messages
                                           N    NATURAL error messages
   D    Documentation object types
   X    External object types             M    PREDICT coordinator
                                           O    PREDICT conversion utility
   F    Function help
   E    Editor help                        A    PREDICT metadata administration

  Help information ..
  Function .........*
  Object type ......* FI
  Message number ....

  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
```

Display Help Texts

With the Software AG Editor

Enter a period (.) or PF3 at the *end* of a help text to skip to the *next* text.

Enter a period (.) or PF3 *within* the help text to skip all remaining help texts and return to the Help Main Menu.

Without the Software AG Editor

Enter PF3 to skip remaining texts.

Functions in the Help Main Menu

Predict - General Overview - Code G

Displays a short description of the overall structure of Predict and its functions. Parameters entered in the fields Function, Object type and Message number are ignored.

What is new - Code W

Displays a description of new functions. Values entered in the fields Function, Object type and Message number are ignored.

Help with Help - Code ?

Displays help information on the Predict help system. Other parameters are ignored.

Documentation object types - Code D

Displays help texts of Predict functions that apply to specified Predict object types. The parameters Function and Object type determine which help texts are displayed. Enter an asterisk to display possible values.

If fields are left blank, help for all functions/object types is displayed.

The example below displays the retrieval help texts for all documentation object types:

Code: D

Function: R

Object type: blank

External object types - Code X

Displays help texts of Predict functions that apply to specified external object types. The parameters Function and Object type determine which help texts are displayed. Enter an asterisk to display possible values. Object type must be specified. If Function is left blank, help for all functions for the specified external object type is displayed.

The example below displays the help text of the function Generate PL/I Include Code.

Code: X

Function: G

Object type: PL

Function help - Code F

Displays help texts on different functional groups: Defaults, File Implementation and special functions. Leave Function field blank to display all help texts for these functional groups. Enter an asterisk to display valid values. Object type and Message number are ignored.

Editor Help - Code E

Displays help text on Predict editors. Leave Object type field blank to display help texts on all Predict editors. Function and Message number are ignored.

Predict direct commands - Code C

Displays information on how to enter Predict commands and on the use and syntax of individual commands. To find information on a specific command use the SCAN function or the topic lists in the help text provided. Other parameters are ignored.

Predict error messages - Code P

Enter four-digit Predict error number to display the full Predict error description. Fields Function and Object type are ignored.

Predict error descriptions can also be called with the following command from any Command ===> line: ? DICnnnn

Natural error messages - Code N

Enter four-digit Natural error number to display the Natural error description. Fields Function and Object type are ignored.

Natural error descriptions can also be called with the following command from any Command ===> line: ? nnnn

Predict Coordinator - Code M

Displays help on the Predict Coordinator. Other parameters are ignored.

Predict conversion utility - Code O

Displays help on the Predict Conversion Utility. Other parameters are ignored.

Predict metadata administration - Code A

Displays help on Predict Metadata Administration. Other parameters are ignored.

15　Menu Dialog

The usual way of using Predict is to call functions from a menu. This section covers the following aspects of working with menus:

Main Menus

Predict offers three types of main menu. The default main menu is specified in the Modify User Defaults > Handling screen, but either of the alternate menu types can be called by entering the respective code in any main menu.

In all three menus, functions are executed by specifying a function code and - in most cases - an object type. Predict then skips to the respective function menu. If a required value is not specified, a window appears from which a value can be selected.

Function Main Menu

Contains a choice of function groups. This menu is displayed by entering code F in the field Menu type which appears in all main menus.

```
   13:40:16              *****  P R E D I C T  *****              2007-05-31
   Plan  10                   - Function Main Menu -              Profile HNO

        Documentation                    External objects

        A  Active retrieval              F  File implementation
        M  Maintenance                   G  Generation
        R  Retrieval                     I  Incorporation
        X  List xref for 3GL             C  Comparison
                                         L  Administration


        DDA services                     Miscellaneous

        D  Defaults                      W  What is new ?
        S  Special functions             ?  Help system
                                         .  Leave PREDICT


   Function ........
   Object type ....*                    Menu type ....* F

   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
```

Documentation Menu

Lists the object types defined in the Predict metastructure. Function menus can then be called for individual object types. The Documentation Menu is displayed by entering code D in the field Menu type of any main menu.

External Object Menu

lists the types of external objects that can be created and maintained. Function menus can then be called to maintain individual types of external objects. The External Object Menu is displayed by entering code E in the field Menu type of any main menu.

Layout of Predict Menus and Screens

General conventions apply to the layout of Predict input and output screens.
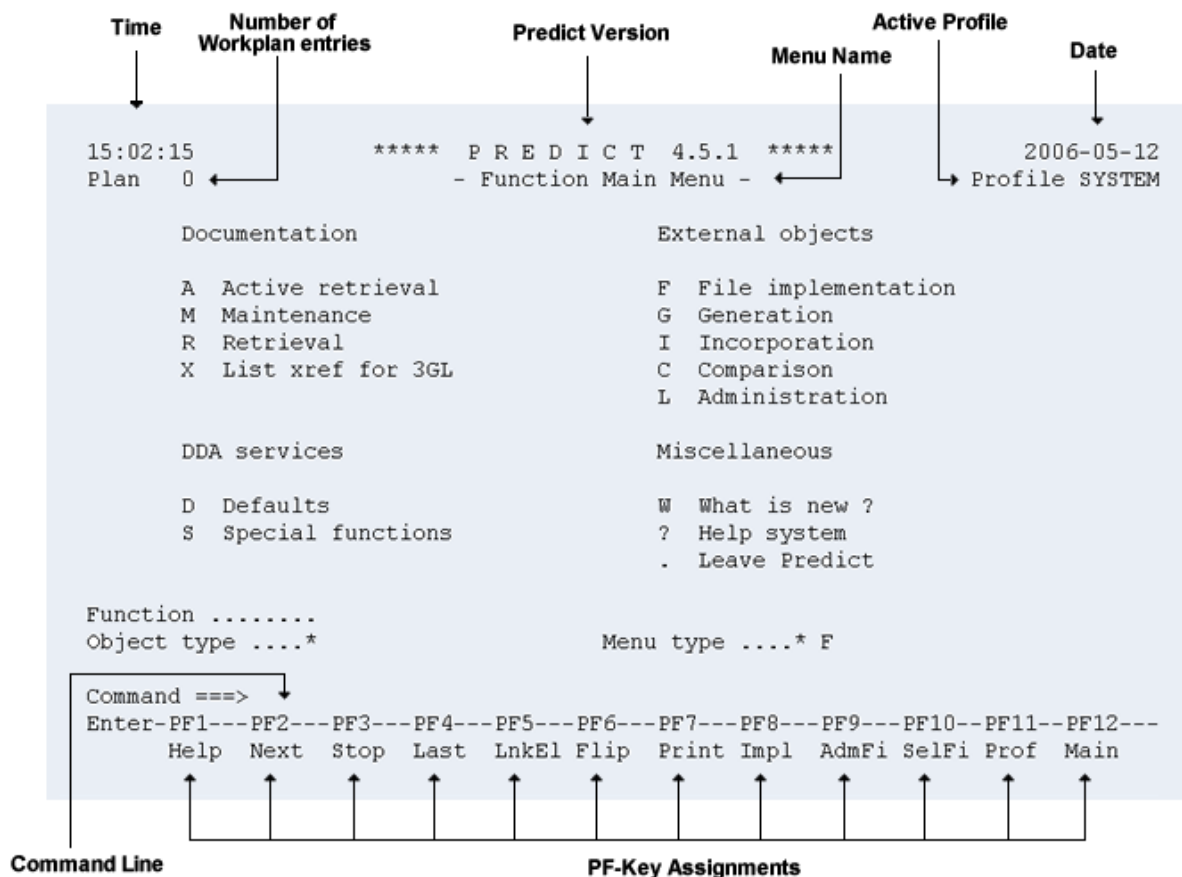
- The first line contains the current time, the Predict version and the current date (from left to right).

- The information in the second line is different for input menus and output screens:

    - In input menus the second line indicates how many commands are stored in the workplan (in the field Plan), the name of the current menu and which profile is active.

    - In output screens this line contains the name of the function and the page number of the current page in the report.

- The three last lines contain the input fields for commands (Command ===>) and the PF key line showing the current PF key assignment. This assignment can be changed in the profile.

Specifying Parameter Values

With many Predict functions, parameter values must be specified. Predict stores parameter values temporarily in session variables.

Session variables is a term which covers the following:

- Hot Objects
- Resetting Session Variables
- Taking the Object Type from the Menu with Commands
- Parameter Zoom

Whenever possible, parameter values that have been omitted are taken from these session variables. The parameter values that can be chosen from depend on both the function and the parameter type.

Using menus, parameter values that are taken from session variables are displayed in the respective input fields.

Values stored in session variables remain when terminating a Predict session.

Hot Objects

If applicable, the following information about objects is stored in the session variables for the 10 objects of different types (including user-defined types) that were processed last:

- object ID

- subtype (in the case of fields, a file type specified with the parameter in files of type)

- database number

- file number

- entry points

- language

- verification status

- association type

- a retrieval model to be applied to the object type

Whenever any of this information is needed to carry out a function and has not been entered explicitly, it is taken from the session variables.

For example: the command DISPLAY ELEMENT A* B* sets the session variable field ID to "A*" and the session variable file name to "B*". If the command DISPLAY FILE is then entered, the file ID is taken from the session variable file name, and the command is interpreted as DISPLAY FILE B*. To override values in session variables, simply enter a parameter value explicitly.

The values listed above are stored in the user's profile and are available in the next session.

To reset hot objects

Enter command RESETHOT to delete all hot objects currently defined.

Resetting Session Variables

Predict resets all parameter variables after each command when running in batch mode. When running Predict online, session variables are reset after the MENU command. Session variables are reset to the values valid when the Predict session was started.

Taking the Object Type from the Menu with Commands

If a command requires an object type and no object type is specified, Predict takes the object type from the menu where the command has been entered (if this is possible).

For example: if the command MAINTAIN is entered in the File Retrieval menu, Predict directly displays the File Maintenance menu.

If the object type of the current menu is not applicable for the command, the message OBJECT TYPE MUST BE SPECIFIED is displayed and Predict returns to the Main Menu.

Parameter Zoom

If all possible parameter values do not fit into a screen, a Zoom parameter is provided to display a window for entering values (see example below).

```
17:00:50              *****  P R E D I C T  *****                2007-05-31
                          - Modify Dataspace -
Dataspace ID .... HNO-TEST                +------------ Abstract ------------+
Type ............ DB2 mainframe           !  THIS DATASPACE WAS INCORPORATE  !
Located in DA ... HNO-DB2                  !  D FROM DB2-CATALOG ON 2004-07-  !
Keys ..                                    !  31.                            !
                                           !  IT INCLUDES TEST DATA FOR      !
Dataspace attributes                       !  EVALUATION PURPOSES.           !
  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                       +---------------------------------+
  THIS DATASPACE WAS INCORPORATE
  D FROM DB2-CATALOG ON 2004-07-
  Additional attributes ..* N        Associations ..* N
```

16   Selecting Objects for Processing

Most Predict functions process either documentation objects or external objects. Individual objects or several objects can be processed in one run. Predict provides a variety of options to select objects for processing:

Selecting Individual Objects

Individual objects are selected by entering an ID in the respective input field (for example File ID, User ID, ...). For fields, the ID of the file to which the field belongs also has to be specified (because fields in different files can have the same ID).

If a function can only be applied to individual objects, entering a name in asterisk notation (for example ABRA*) will cause Predict to display an active help window containing the names matching that asterisk notation (for example ABRAHAM or ABRACADABRA). See also **Using Active Help - Selection Windows**. This option only applies to fields marked with an asterisk (*).

Another option is to select objects from selection lists. Enter / (slash), S or X in the column Cmd of the selection list.

> **Note:** Whenever an individual object has been selected by specifying a unique object ID, additional selection criteria are ignored.

Selecting Several Objects

With functions that can be applied to several objects, asterisk notation can be used to select several objects for processing. For example ABRA* will cause Predict to apply a function to all objects starting with ABRA (for example ABRAHAM or ABRACADABRA). If just an asterisk or blank is entered, all objects for which the function is applicable are processed.

A selection of several objects can be restricted by specifying additional selection criteria. The following types of additional selection criteria are described below:

■ type-dependent additional selection criteria

■ type-independent additional selection criteria (keywords, owners, extracts, date and strings).

Type-dependent Selection Criteria

The following parameters can be used in different type-specific functions. Many of the parameters refer to the hierarchical structure of a metadata model.

Parameter	Applicable to
object of type	Object types where subtype is possible. Only objects with the given subtype are included in the selection. For example: a file of type A (Adabas file).
default passive association	Object types for which a default passive association is defined. Only objects associated as child objects to the given parent object (via the default passive association) are included in the report. For example: only files linked to a database via the default passive association `Contained in DA` are selected.
Database number	Database
External name	File
File number	File
Synonyms	Field
Language	Program
Implementation pointer - Library - Member - Fnr - DBnr	Program
Implementation pointer - Library - Fnr - DBnr	System
Using file	File relation
User name	User
Verif. of status	Verification
Format	Verification

The selection criteria above are described in detail in the retrieval sections of the *Predefined Object Types in Predict documentation*.

Restrictions

Restrictions can be specified for all functions where objects can be selected for processing. Only objects which meet the criteria specified are evaluated.

Restrictions can either be specified temporarily by entering code T in the field Restrictions or they can be taken from a profile. See *Specifying Restrictions*. Temporary restrictions are lost when a session is terminated.

The following topics are covered:

- Overview of Restrictions
- Specifying Restrictions
- Specifying Restriction Values Temporarily
- Restriction Parameters
- Saving Restrictions for Later Use
- Using the Restriction Values of another User
- Personal Default Restriction Values
- How the Use of Restrictions is Indicated
- Deactivating Restrictions

Overview of Restrictions

- **Keywords**
  Up to five Keywords can be specified, either uniquely qualified or with asterisk notation. See **Relating Objects Logically** in the section "Predict Functionality" in the "Introduction to Predict documentation" and section *Keyword* in the *Predefined Object Types in Predict documentation*.

- **Owner**
  You can restrict the selection to objects that are assigned to a particular owner. See **Relating Objects Logically** in the section "Predict Functionality" 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.

Specifying Restrictions

Enter an asterisk in the field Restrictions to display the screen below, or enter one of the codes U, A, D, T or E to specify restrictions directly.

```
   13:09:47              *****  P R E D I C T  *****                  2007-05-31
   Plan   7                   - (FI) File Maintenance -             Profile HNO
                                              +All-----Restrictions---------+
    Function                        Functio ! _ A   Active profile       !
                                            ! _ D   Profile Default      !
    A  Add a file                   L  Link ! _ T   Temporary profile    !
    C  Copy file                    S  Sele ! _ E   Empty restrictions   !
    M  Modify file                  B  Push ! _ ' ' Confirm current      !
    N  Rename/Renumber file         F  Forc !       profile and execute  !
    P  Purge file                   J  Phys !       function             !
    D  Display file                 K  Modi !                            !
                                            !                            !
                                            !                            !
    Function .......                        !                            !
    File ID ........                        !                            !
    Copy ID ........                        !                            !
    Contained in DA.                        !                            !
    External name ..                        !                            !
    Restrictions ..* * Profile HNO ,used    !                            !
                                            !                            !
    Command ===>                            !                            !
                                            !Command ==> _____        !
                                            +----------------------------+
```

Restriction values can be stored in profiles. To use restrictions of a profile, the profile does not necessarily have to be active as the current session profile. The following options are possible:

Code	Profile	Result
U	Your profile	Use restrictions in the profile of the current user
A	Active profile	Use the restrictions of the profile that is currently active. The active profile is indicated in the upper right corner the screen (in the field Profile).
D	Profile default	Use the restrictions of the Predict default profile.
T	Temporary profile	Use temporary restrictions. The Restrictions window is then displayed containing the values of the currently active profile. The values can be overwritten for use in the current session.
E	Empty restrictions	Deactivate restrictions.

Specifying Restriction Values Temporarily

Enter T in the field Restrictions to display the following screens:

```
 13:00:16              ***** P R E D I C T *****              2007-05-31
 Plan   7                 - (FI) File +Top-----------Restrictions-------------
                                     ! With keyword(s)                       !
  Function                          !                                       !
                                     !                                       !
  A  Add a file                      !                                       !
  C  Copy file                       !                                       !
  M  Modify file                     !                                       !
  N  Rename/Renumber file            ! combined by (AND/OR) OR              !
  P  Purge file                      ! AND with owner                       !
  D  Display file                    !                                       !
                                     ! BUT NOT with keyword                 !
                                     !                                       !
  Function .......                   ! AND included in extract              !
  File ID ........                   !                                       !
  Copy ID ........                   ! AND containing the string            !
  External name ..                   !                                       !
  Contained in DA.                   ! Scan options:                        !
  Restrictions ..* T Profile HNO ,used !   Abstract Y (Y,N) Rules ..... N (Y,N) !
                                     !   Descr.   N (Y,N) Object ID . Y (Y,N) !
  Command ===>                       !   Absolute Y (Y,N) Ignore case Y (Y,N) !
                                     !Command ==> +___                      !
                                     +More-------------------------------
```

```
 13:00:16              ***** P R E D I C T *****              2007-05-31
 Plan   7                 - (FI) File +More-----------Restrictions------------+
                                     ! AND from date (YYYY-MM-DD HH:II)       !
  Function                          ! 2002-01-01 00:00                      !
                                     !                                       !
  A  Add a file                      !                                       !
  C  Copy file                       !                                       !
  D  Display file                    !                                       !
  M  Modify file                     !                                       !
  N  Rename/Renumber file            !                                       !
```

The individual parameters are described below.

Restriction Parameters

For a list of valid parameters refer to *Restrictions* in the section *Retrieval* in the *Predict Reference documentation*.

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 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 that is not defined in Predict with an object of type User is rejected.

Using the Restriction Values of another User

To use the restriction profile of another User, activate the profile of that User with the command PROFILE user-ID.

Personal Default Restriction Values

Personal default values for restrictions are defined with the function Modify user defaults > Restrictions. See the section **Customizing Predict with Profiles** for more information.

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.

> **Note:** If the result of a selection operation is smaller than expected, it is possible that restrictions that were set previously are still active.

Using Selection Windows

In many situations when one of several possible objects in the dictionary is to be selected, a selection mechanism can be used. The list gives an overview:

- Objects of specific types can be selected from the dictionary by executing a retrieval operation with the output mode Select.

- Objects of a specific type can be selected for maintenance by executing the function Select object from a list in any Maintenance menu.

- Objects of a certain type can be selected to assign them as child objects by executing the SEL command in the list editor.

- Fields used in the definition of a derived field (sub/super/field/descriptor) can be selected.

  See *Selection Mechanism* in the section *Field* in the *Predefined Object Types in Predict documentation*.

- A physical file definition of a logical file can be selected when using the Modify Adabas attributes function (code J)

- A Vista element can be selected when using the Modify Vista elements function (code K).

- Sections of text can be copied from different sources into all attributes of Predict objects that have the form of text. See **Copying Text from Other Sources**.

See the section *Using Active Help* for a description of how to use selection windows.

17   Using the Workplan

Calling Predict functions using the workplan is a two-step process:

1. Select the objects to be processed by specifying the type of function to be applied to them in the Cmd column of a selection list. Enter an asterisk in this column to display the valid functions for the selected object. Confirm the combinations of function and object with ENTER.

2. Execute the commands in the workplan, one after another or in a specified order. The two methods of calling workplan commands are described in the section **Step 2: Executing Commands from a Workplan**.

> **Note:** When you create a dummy (by creating a link to an object that does not yet exist), an entry in the workplan is created automatically and a message is given if the parameter Workplan commands/Add task for dummy in the maintenance options of the active profile is set to Y. Only one command is put into the workplan for each dummy. If the command already exists in the workplan, a corresponding message is given.

Step 1: Adding Entries to the Workplan

Entries are added to the workplan by entering one or several function codes in the column Cmd in an object selection list. Enter an asterisk in Cmd column for a list of codes valid for the object selected. The example below shows the result of entering an asterisk in the Cmd column alongside an Adabas file.

```
 13:00:53                ***** P R E D I C T *****                 2007-05-31
 Plan  10                     - Select File -
                                               +All--Possible functions------+
 Cmd  File ID                        Type    ! _ /   Select                 !
                                             ! _ C   Copy                   !
 ->   * ARHTESTCHEN                   A       ! _ D   Display                !
      ARTCHD-ZICKELID                 I       ! _ E   Edit Elements          !
      ARTICLE-A                       I       ! _ B   Push Backward          !
      ARTICLE-B                       I       ! _ J   Modify Adabas Attr.    !
      ARTICLE-C                       I       ! _ K   Modify Vista element   !
      ARTICLE-CONTAIND                I       ! _ L   Link Children          !
      ARTICLE-CONTAINS                I       ! _ M   Modify                 !
      ARTICLE-PART                    I       ! _ N   Rename                 !
      ARTICLE-POSORDER                I       ! _ O   Edit Owner             !
      ARTICLE-SUPPLIER                I       ! _ P   Purge                  !
      AUTOMOBILES-VS                  V       ! _ S   Select                 !
      AUTOMOBILES-VS-VIEW             W       ! _ W   Edit Description       !
      AZ-AVB-L-ART-1                  L       ! _ X   Select                 !
      AZ-AVB-L-ART-111                L       ! _ ' ' No command             !
      AZ-AVB-L-ART-2                  L       !                              !
      AZ-AVB-U-ART                    U       !                              !
      AZ-AVB-V-ART                    V       !                              !
      * AZ-LOW                        A       !                              !
                                              !Command ==> _____           !
   __ ('.' = EXIT, 'T' = TOP, 'C' = Specify asso  +---------------------------+
```

> **Note:** To select an object for immediate processing, enter / (slash), S or X in the column Cmd.

Rules

The following rules apply when adding commands to the workplan:

- Up to 999 commands can be added to a workplan.

- When marking objects in a list produced by the Select function, commands are written to the workplan as soon as ENTER is pressed.

- The position of new entries in the workplan can be determined with the profile parameter *Position of storage* (Maintenance Options > Workplan Commands > *Position of storage*). New entries can be positioned at the beginning and at the end of the workplan.

- If you leave a list with PF3 or call another function with a different PF key, selections that have not been confirmed with ENTER are lost.

- If you leave a list with '.' or scroll to the top of the list with T in the bottom line of the selection list, selections in the current screen are added to the workplan.

- For the Copy function: Enter the ID of the new object to be created in the window provided.

- For the function Push backward: Enter name of standard file in the window provided.

- For the Link children function: Enter C in the bottom line of the screen to display a window of valid associations for selection. Association must be specified.

- To select a single object for immediate processing, enter S in the Cmd column or position the cursor to the line containing the object and press ENTER. This will not result in a command being added to the workplan.

Step 2: Executing Commands from a Workplan

An example of a workplan is shown below.

```
 13:09:31                  *****  P R E D I C T  *****                 2007-05-31
 Plan   3                        - Plan menu -

 Cmd   Plan-commands

   __    DISPL FI      HNO-FI-S-1
   __    LINK  FI EL   HNO-FI-C
   __    MODIF FI      HNO-E

   __

   __


 _____  ('.' = EXIT, 'T' = TOP, 'CLEAR' = DELETE all Plan-commands)
```

Commands contained in a workplan can be executed in two ways:

- By entering the command NEXT in the command line of any Predict menu. The first command in the workplan will then be executed. The NEXT command can be executed most easily by assigning it to a PF key. If the NEXT command is issued with a hyphen following ("NEXT -") the last command in a workplan is executed.

- By opening the workplan with the command PLAN and then by entering X in the Cmd column of the workplan. Commands can then be executed in any order.

Administrating a Workplan

The following rules apply for the administration of a workplan:

- To use a workplan, a users ID has to be documented in Predict with an object of type user.

- The number of commands in a workplan is displayed in the upper left corner of every Predict menu.

- If a user is deleted, the workplan of this user is deleted as well.

- If a user is renamed, the link to the workplan remains.

A workplan can be opened with the command PLAN. The following actions can then be executed:

- Individual commands can be purged from the workplan by entering a hyphen ("-") in the Cmd column of the workplan.

- The contents of a workplan can be cleared with the command `CLEAR`.

- Commands can be executed from within the workplan by marking them with X and pressing ENTER. One command can be executed at a time.

18 Copying Text from Other Sources

Text can be copied from the following types of sources into text attributes of Predict objects (extended descriptions, processing rules, field expressions and subqueries).

With Command SEL

- extended description
- default extended description
- abstract
- processing rule
- subquery
- procedure code
- derived field expression

With Command IMPORT

- Natural source code
- Con-nect document
- User defined source
- PC ASCII file

Text can be copied in the following Predict editors:

- the description editor
- the rule editor
- the expression editor
- the subquery editor
- the trigger code editor

If the position where the text is to be copied to is not specified, the copied text is added at the end of text.

See also *Transferring Text* in the section *Editors in Predict* in the *Predict Reference documentation*.

19   Selecting Objects for Link Lists

If you enter the command SEL in a link editor, Predict retrieval functionality is available for adding objects to the link list. Perform the following steps.

1. Enter SEL in the command field of the link editor to call a selection menu.

```
   >
SEL

           > + PR: HNO-PR1                              L: 1    S: 2
   ALL  Program                          Type  Member    Library
        ------------------------------   ----  --------  --------
        HNO-PR1                           P
        ARH-PR-TEST-TYPE-A                P     ARH-PARA  ARH2
        ARH-PR-TEST-TYPE-C                C     ARH-COPY
        ARH-PR-TEST-1                     P     ARH-PR    ARH1
```

2. Enter a retrieval type in the selection menu. You can also enter additional search criteria and restrictions to limit the scope of the selection. See example below.

3. If only one object meets the parameters specified, this object is automatically added to the link list and marked with < inserted. If several objects meet the parameters specified, these are displayed for selection. Mark the objects you wish to include in the link list with /, S or X.

> Note: This function also enables you to add tasks to the workplan. Enter an asterisk in the Cmd column to display a list of commands valid for the current object for selection. See also **Using the Workplan**.

4. Enter a period (.) in the field at the bottom of the Cmd column to return to the link list. Newly added objects are marked < inserted.

```
     >
CAT

             > + PR: HNO-PR1                               L: 1    S: 6
    ALL   Program                          Type  Member   Library
          ------------------------------- ---- -------- --------
          HNO-PR1                           P
          ARH-PR-TEST-TYPE-A                P   ARH-PARA ARH2
          ARH-PR-TEST-TYPE-C                C   ARH-COPY
          ARH-PR-TEST-1                     P   ARH-PR   ARH1
          ARH-PR-TEST-2                     J   ARH-PR2  ARH2
< inserted


          ARH-PR-TEST-3                     P   ARH-PR3
< inserted
```

5. Enter CAT in the command field of the link editor to store the new link list.

Example

The following screen appears if you enter the SEL command in the Program list editor for association *Uses PR concept.*:

```
   10:04:23                  *****  P R E D I C T  *****                    2007-05-31
   Plan   0                      - Program Selection Menu -              Profile HNO


   Program ID ......... HNO-PR1
                                                          Added 2007-05-31 at 10:03
                                                               by HNO
   Select object type .............. PR  ( Program )

   Retrieval type .....* D
   Output mode ........* S Select

   Search criteria
     Program ID ........                            Program of type *
     Belongs to SY .....                            Language .......*
     Member ............                            User system Fnr .
     Library ...........                            User system DBnr.




   Restrictions .......*   Profile HNO ,empty                 Association ....*
```

Parameters	
Program ID	ID of the Program processed with the function Link children. This field cannot be modified.
Select object type	Child object type. This value is specified with the function Link children and cannot be modified here.
Retrieval type	The valid retrieval types depend on the object type. For object type Program, for example, the following types are possible.

D	Programs (Default)
C	Dummy/Placeholder programs
O	Programs with no parent
U	Programs with no child

Note: If Dummy/Placeholder programs is selected, an additional window is displayed in which you must select whether dummies or placeholder objects are to be listed for selection.
These retrieval types are described in the section *Standard Retrieval Types* in the *Predict Reference documentation*.

Parameters	
Output mode	Output mode is always Select. This field cannot be modified.
Search criteria	Depending on the object type, the selection can be limited by additional parameters. For example: the following additional search criteria are offered for object type Program: Program ID, Belongs to SY, Member, Library, Program of type, Language, User system Fnr/DBnr. See *Selection criteria* in the section *Retrieval* in the *Predict Reference documentation*.

Parameters	
Restrictions	See *Restrictions*.
Association	Only valid for retrieval types Objects with no parent, Objects with no child. These retrieval types are described in the section *Standard Retrieval Types* in the *Predict Reference documentation*.

20 Customizing Predict with Profiles

The way Predict works can be changed to a certain extent with session parameters.

Session parameters and parameters that determine how functions work are usually set differently, depending on who uses data dictionary functions and for which purpose. Sets of parameter settings (User Defaults) can be stored in user-specific profiles. Different types of profiles are distinguished.

Predict Profile

The Predict Profile has different sections containing a wide variety of parameters ranging from Maintenance Options to Colour Definition for colour terminals. The currently active profile is indicated in the upper right corner of Predict input screens.

LIST XREF Profile

The LIST XREF profile allows the user to define PF-keys and the scope of the functions Report Programs and Verify All. See the section *LIST XREF for Natural* in the *Predict Reference* documentation for more information.

Editor Profiles

Predict editors can be customized with profiles. The use of editors is described in the section *Editors in Predict* in the *Predict Reference* documentation.

Maintaining User-specific Profiles

Users documented with an object of type user in Predict can define their own profile with the command `PROFILE`, which displays the Add User Defaults menu. A user-specific profile is created as soon as one of the functions in the Add user profile menu has been executed successfully. A user-specific profile is used automatically after logging on to Predict.

Using the Default SYSTEM Profile

Users who have not yet defined their own profile use the DEFAULT system profile. Values in the DEFAULT system profile can be changed for the current session. Changes to the DEFAULT system profile do not remain after a Predict session has been quit.

To activate the DEFAULT system profile explicitly, enter:

`PROFILE SYSTEM.`

Using the Profile of Another User

To activate the profile of another user, enter command PROFILE <user-id>.

Modify User Defaults Menu

Profiles can be modified via the Modify User Defaults menu that is displayed by entering the PROFILE command without <user-id>.

The following rules apply when changing a profile:

- It is not allowed to modify the profile of another user.

- Changes to a user-specific profile cannot deactivate restrictions specified in the default profile (SYSTEM) or with functions of the metadata administration. Example: if the metadata administration parameter EDIT description is set to F (force), this setting cannot be deactivated in a user profile.

- Parameter settings contained in the profile sections Restrictions and Output options can be changed temporarily or permanently when using retrieval functions. See the section *Retrieval* in the *Predict Reference* documentation.

The Modify User Defaults menu offers a selection of different sections of the profile. Each section contains a different type of default. After choosing the type of default by entering one of the codes, one or several input screens are then displayed to specify the default values. All options are described in the following sections.

```
   13:55:33                ***** P R E D I C T *****               2009-05-31
   Plan  3                    - Modify User Defaults -             Profile HNO


                    Function

                    M  Maintenance options
                    R  Restrictions
                    O  Output options
                    H  Handling
                    P  PF-Keys
                    C  Colour definition


   Function ..........




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

Maintenance Options

The Maintenance Options screen is called by entering code M in the Modify User Defaults screen.

```
17:20:57                  *****  P R E D I C T  *****              2017-08-15
                             - Maintenance Options -
                                                 Modified 2017-05-15 at 11:13
Profile .........  SYSTEM                              by SYSTEM
Edit owner                               Edit description
  Add/Modify ............* O  Optional     Add/Modify .........* O  Optional
  Copy .................* O  Optional      Copy ..............* O  Optional


Edit default associations                Edit additional attributes
  Add/Modify ............* O  Optional     Add/Modify .........* O  Optional
  Copy .................* O  Optional      Copy ..............* O  Optional


Workplan commands                        Protect extended description
  Position of storage ...* E  End           Start character .....
  Add task for dummy ..... Y  (Y,N)         End character .......


List action .............. Y  (Y,N)
Purge mode ............... Y  (Y,N)
Full maintenance check ... N  (Y,N)
Current virtual machine .* HOME
SQL Server ..............*
Oracle Handler ..........* XYZ-ORA1
For HELP enter '?' in the first field.    MORE Type dependent options N (Y,N) ↵
```

General Maintenance Options	
Edit owner, description, default child association	The values specified in these fields are used as default values for the fields in the Additional attributes / Associations line. The Additional attributes / Associations line is displayed at the bottom of Add/Copy/Modify functions. Settings in this line determine whether the owner list, extended description or the child list of an object is to be created or changed.
	Example: the Additional attributes / Associations line of Add/Copy/Modify database screens looks as follows:
	`Additional attributes ..* N Associations ..* N`
	Note: The parameters Edit description and Edit owner are also set in the metadata administration and override any settings in a user profile.
	■ If disallow has been set in the metadata administration for either Edit description or Edit owner of an object type, this setting will determine that N (no) is set in the respective field of the Additional attributes / Associations line. This setting cannot be changed by values specified in the profile.
	■ If force has been set in the metadata administration for either Edit description or Edit owner of an object type, Y is set in the respective field of the Additional attributes / Associations line if an extended description or owner list does not yet exist. If an extended description or owner list exists, Y is set if the respective parameter in the profile is set to Y or to O. If the respective parameter in the profile is set to A or to N, the input field is set to N however.

General Maintenance Options	

	See **Forcing/Disallowing Extended Descriptions and Owner/Child Lists** for a more detailed description of the concepts behind these parameters.
	
	The following values can be specified for these parameters:
	
	Y
	
	    Yes. The respective fields in the Additional attributes / Associations line are preset to Y (yes).
	
	N
	
	    No. The respective fields in the Additional attributes / Associations line are preset to N (no).
	
	O
	
	    Optional. If the object already has owners, an extended description or sub objects, the respective fields in the Additional attribute / Associationss line are preset to Y (yes); otherwise, they are preset to N (no).
	
	A
	
	    Add. Only applicable to the Add/Modify function. If the object being maintained already has owners, an extended description or sub objects, the respective field in the Additional attributes / Associations line is preset to N (no), otherwise, it is preset to Y (yes).
	
	Note: If verifications are created with the Natural map editor, the Edit owner parameter has no effect. To ensure that owners are added to a verification that has been created with the Natural map editor, the user exit U-ACMR must be used.

Workplan commands	Determines the position of new workplan commands. Valid values:
	
	<table><tr><td>**Position of storage**</td><td>**B**
	
	  commands are added to the beginning of the workplan
	
	E
	
	  commands are added to the end of the workplan.</td></tr><tr><td>**Add task for dummy**</td><td>**Y**
	
	  An entry is added to the workplan for each dummy created when a link is established to an object that has not yet been created.</td></tr></table>

List action	**Y** Messages are displayed on the terminal whenever data dictionary objects are changed.
Purge mode	**Y** An additional check is performed before objects are deleted. The logic of the check is as follows: if a database is to be scratched, its dependent files (linked via `Contains FI`) are deleted only if they are not linked to another database via `Contained in DA`. Similar logic applies to the scratching of systems and programs that are linked via `Uses PR concept`.

General Maintenance Options	
Full maintenance check	**Y**
	Predict will check the validity of all input values on a screen whenever any of them is changed.
	N
	Predict will only check the validity of input values that have changed.
Protect extended description, Start character, End character	Parts of the default extended description (extended description skeletons) can be protected by enclosing them in special characters. These characters are defined with these parameters.
	See *Extended Description Skeleton* in the section *Defaults* in the *Predict Administration* documentation.
Current virtual machine	Default for the parameter Current VM. See the section Defaults in the *Predict Administration* documentation. This parameter is used to determine the target environment for Generate/ Incorporate/ Compare/ Administration Implementation functions.
	The virtual machine entered here must be linked to the current network. Enter an asterisk to display all virtual machines linked to the current network. A Default current virtual machine and the Current network can be specified in the second Miscellaneous screen of the General Defaults function.
	Note: After changing the Current virtual machine, implementation plans created with a different virtual machine can only be purged or displayed. See the section *File Implementation Plans* in the *External Objects in Predict* documentation for details.
	The current virtual machine can also be set with the command: SET VM<virtual machine ID>.
SQL Server	When adding or modifying the documentation of a DB2 object, the physical attributes of the object in the denoted server are offered for editing.
Oracle Handler	When adding or modifying the documentation of an Oracle object, the physical attributes of the object in the denoted server are offered for editing.

Additional Maintenance Options for Files		
Modify Adabas attributes	Y	
	With the functions Add/Copy/Modify file for files of type A or U, the parameter Additional attributes is set to Y.	
Additional Maintenance Options for Fields		
Automatic Adabas names	Y	
	Predict automatically generates Adabas names for fields that are added to the data dictionary.	
Automatic null value	Y	
	Predict automatically sets suppression/null value options for fields that are added to the dictionary. The value depends on the type of file:	

Additional Maintenance Options for Files				
	Parameter	**File Types**		
		All SQL Types* except X X	**X**	**Other File Types**
	Unique option = Unique or Desc. type = Primary or Field format = serial	R	R	N
	Others	U	blank	N
	Note: SQL file types include files of type A with parameter Adabas SQL usage set to Y.			
Copy description	**Y** The extended description is also copied when a field is copied from the field list of one file into the field list of another file (using either the SEL command or the .H command of the field list editor). **N** The extended description is not copied.			
Ripple abstract	This option defines whether the abstract of a field object is rippled, i.e., whether changes to the abstract of a field are pulled forward from a physical file to a userview. Valid values: **N** Abstract is not rippled. **T** Abstract is rippled even if the abstract in the real file and the corresponding userview are different. **L** Abstract is rippled only if the entire block of abstract in the real file and the corresponding userview is identical. See *Rippling* in the section *File* of the *Predefined Object Types in Predict* documentation.			
List rippling	Y Messages are displayed at the terminal whenever rippling is performed.			
File for select	Fields to be linked to a file can be selected from an existing field list by using the SEL command in Link children. The File for select parameter is then used as default setting for the *Belongs to FI* parameter.			

Restrictions

The restrictions screen is called by entering R in the *Modify User Defaults Menu*. The valid parameters are described in the section *Restrictions*.

Output Options

- General output options applying to all object types are specified in the first screen.

  For a detailed description see *Output Options Valid For All Objects* in the section *Retrieval* in the *Predict Reference* documentation.

- Output options for specific object types can be defined in subsequent screens be entering Y in the field MORE Type dependent options. For a detailed description see *Type-specific Output Options* in the section *Retrieval* in the *Predict Reference* documentation.

See also *Using Restrictions and Output Options from Profiles* in the section *Retrieval* in the *Predict Reference* documentation for the different ways of using output options.

```
15:59:56            *****  P R E D I C T  *****                    2009-07-31
                         - Output options -
                                                      Added 2008-09-25 at 11:13
Profile ......... SYSTEM                            Modified 2009-07-21 at 11:56


                     Current    related                        Current and
                     object     object                         related object
Type independent                              Type independent
  Description .......   Y  (Y,N)   Y             Association attr. .*   Y
  No. abstract lines  16 (0-16)  16             Mark implementation    N  (Y,N)
  Keywords ..........  24 (0-32)  24             Show implementation    Y  (Y,N)
  Extracts ..........  32 (0-32)  32             Display modifier ...   N  (Y,N)
  Owners ............  99 (0-99)  99             Connecting character   :
    with users .....*   Y          Y             Cover page .........   N  (Y,N)
  Attributes ........   Y  (Y,N)   Y             Page size ..........   60
  Dummy/Placeholder *   D          Y


Miscellaneous                                 Conform options
  Web Service usage .   Y  (Y,N)                 Use Conform ........   N  (Y,N)
                                                   Predict header ...   N  (Y,N)
                                                   User FPROFILE ....   N  (Y,N)
                                                   List error message   N  (Y,N)
For HELP enter '?' in the first field.    MORE Type dependent options: N      ↵
```

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.
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.
Synonyms	Display language-specific synonyms of field IDs.
	#
	Display synonyms for all languages.
	*
	A selection window for language code appears.
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 Program	
Entry points	**Y**
	Entry points are displayed.

Introduction

Additional Output Options for Field	
Procedure call	**Y** Display procedure calls.

Additional Output Option for Verification	
Rules	**Y** Display rules of a verification.

Handling

```
  13:09:01                   *****  P R E D I C T  *****                        2007-05-31
                                  - Handling -
                                                             Modified 2007-05-31 at 13:06
  Profile ......... HNO                                             by HNO


  Display first screen (logo screen) ....... N  (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) .. C:\PREDICT\                              -

  Stay after modify
     Documentation objects .................. Y  (Y,N)
     External objects ....................... N  (Y,N)
     Special function ....................... N  (Y,N)
     Menu ................................... N  (Y,N)

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

Handling Defaults	

Display first screen (logo screen)	**Y**
	
	    The Predict banner is displayed when Predict is invoked.
	
	N
	
	    The Predict banner is skipped and the Predict Main Menu is displayed immediately.

Menu type	Predict offers three different main menus. The default menu can be chosen by entering one of the following codes:

Handling Defaults	
	F
	Function Main Menu
	D
	Documentation Menu
	E
	External Object Menu
Use as editor (first, second, third choice)	Definition of the editor to be called. Up to three preferences are possible. At least one editor must be specified. The editor actually used depends on the environment from which the editor is called.
	W
	Word for Windows is called.
	S
	The Software AG Editor is called.
	C
	A Natural-based editor is called.
	Note: We recommend you set your editor preferences in the order given above.
	See *Defining Editor Defaults* in the section *Editors in Predict* in the *Predict Reference* documentation for more details.
Additional parameters for Word for Windows	
Download PC part	**Y**
	If all prerequisites are met, the Predict-Winword interface is downloaded to your PC.
	See *Maintaining Extended Descriptions with Word for Windows* in the section *Editors in Predict* in the *Predict Reference* documentation for more information.
Entire Connection task name	The name of a task defined in Entire Connection. See the Entire Connection documentation for detailed information on tasks.
Predict Word for Windows directory	Enter here the drive and directory to which the template file PRD.DOT of the Predict-Winword interface is to be loaded.
	See *Maintaining Extended Descriptions with Word for Windows* in the section *Editors in Predict* in the *Predict Reference* documentation for more information.
Stay after modify	**Y**
	ENTER must be pressed twice to confirm input in any Predict menu or function screen.
	N
	ENTER must be pressed once to confirm input in any Predict menu or function screen.
	This option can be set differently for documentation objects, external objects or special function menus. If Y is specified in the field Menu, the setting of Stay after modify applies to all kinds of object types and functions.

PF Keys

PF keys can be assigned in two screens. If applicable, the assignments are also valid in other Predict applications (SYSDICBE, SYSDICCO, SYSDICMA). Change definitions by simply overwriting them.

```
   13:35:33              *****  P R E D I C T  *****                    2007-05-31
                              - PF-Keys -
                                                        Modified 2007-05-31 at 13:20
   Profile ......... HNO                                         by HNO

   Text     PF-nr   Function
   Help     1       CMD ??
   Menu     2       MENU
   Canc     3       STOP
   S-fi     4       CMD SELECT FILE
   E-el     5       CMD EDIT ELEMENT
   M-pr     6       CMD MAINTAIN PROGRAM
   Print    7       +H
   Impl     8       CMD IMPLEMENT FILE
   Last     9       CMD LAST
   FLIP     10      +YL
   PROF     11      CMD PROFILE
   Next     12      CMD NEXT
   Use '+' as control character for assigning Natural terminal commands to
     PF-KEYS (e.g. +H for hard copy)


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

Parameters	
Text	Up to 5 characters can be specified as text that is displayed in the PF key line.
Pf-nr	PF key number.
Function	**Command to be triggered by the PF key. The following commands can be assigned to PF keys.**
STOP	Backs out the current transaction, cancels successive actions and returns (normally) to the last menu that was active. Under some circumstances, it may return to the Predict Main Menu.
CMD MAIN	Returns control to the main menu without resetting the session variables.
CMD MENU	Resets all session variables (file name, owner ID, etc.). It then returns control to the main menu.

Parameters	
CMD xyz	CMD calls the Predict command processor. This program is executed with the parameter xyz, which is any valid Predict command. Predict commands can be abbreviated.
+x (Terminal Command)	The plus sign (+) is substituted for the leading percent sign (%) in Natural terminal commands such as "%H".
B-CMD xyz	The command processor of the Predict Coordinator. This program is executed with the parameter xyz, which is any valid Predict command. Abbreviations may be used as required.

Colour Definitions

The colours used for the different types of output on colour terminals can be changed with this screen.

```
   13:42:23              *****  P R E D I C T  *****                    2007-05-31
                            - Colour Definition -
                                                      Modified 2007-05-31 at 13:20
   Profile ......... HNO                                       by HNO

   Colour Definition                   - Valid Colours -
     Normal output ....... TU            BL   Blue
     Intense output ...... YE            GR   Green
     Normal input ........ RE            NE   Neutral
     Intense input ....... NE            PI   Pink
                                         RE   Red
                                         TU   Turquoise
                                         YE   Yellow




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