

# **Natural**

## **External Objects in Predict**

Version 9.1.3 for Windows

October 2020

This document applies to Natural Version 9.1.3 for Windows and all subsequent releases.

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

Copyright © 2003-2020 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

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

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

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

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

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

**Document ID: SGEN-OPRDEXTOBJECTS-913-20201002**

## Table of Contents

Preface .....	vii
1 About this Documentation .....	1
Document Conventions .....	2
Online Information and Support .....	2
Data Protection .....	3
2 Handling of External and Documentation Objects .....	5
What is an External Object .....	6
Connection of External and Documentation Objects .....	7
I Generation of External Objects .....	9
3 Generation of External Objects .....	11
II About Generation .....	13
4 About Generation .....	15
Overview of Generation Functions .....	16
Generation Defaults .....	17
Common Parameters .....	18
III Generate for Adabas - Adabas File .....	19
5 Generate for Adabas - Adabas File .....	21
Parameters .....	23
Keywords .....	25
Sample Output .....	26
IV Generate for Natural .....	29
6 Generate for Natural .....	31
Replace Verification Rule .....	32
Data Definition Module .....	35
V Generate for DB2 .....	47
7 Generate for DB2 .....	49
Common Parameters for All DB2 Types .....	50
DB2 Database .....	50
DB2 Procedure/Function .....	53
DB2 Table/View .....	56
DB2 Storagegroup .....	62
DB2 Tablespace .....	64
VI Incorporation .....	69
8 Incorporation .....	71
VII Concepts of Incorporation .....	73
9 Concepts of Incorporation .....	75
Overview of Incorporation Options .....	76
Connecting External and Predict Objects .....	76
Calling Incorporation Functions .....	78
VIII Incorporating Adabas Databases .....	81
10 Incorporating Adabas Databases .....	83
Prerequisites .....	84
Restrictions .....	84

Selecting Adabas Databases .....	85
Incorporation Functions for Adabas Databases .....	86
Keywords .....	87
IX Incorporating Adabas Files .....	89
11 Incorporating Adabas Files .....	91
Prerequisites and Restrictions .....	92
Selecting Adabas Files .....	93
Incorporation Functions for Adabas Files .....	95
Keywords .....	96
X Common Parameters for All DB2 Types .....	97
12 Common Parameters for All DB2 Types .....	99
XI Incorporating DB2 Storagegroups .....	101
13 Incorporating DB2 Storagegroups .....	103
Prerequisites .....	104
Selecting DB2 Storagegroups .....	105
Incorporation Functions for DB2 Storagegroups .....	106
Keywords .....	107
XII Incorporating DB2 Databases .....	109
14 Incorporating DB2 Databases .....	111
Prerequisites .....	112
Selecting DB2 Databases .....	113
Incorporation Functions for DB2 Databases .....	114
Keywords .....	115
XIII Incorporating DB2 Tablespaces and SQL/DS DBspaces .....	117
15 Incorporating DB2 Tablespaces and SQL/DS DBspaces .....	119
Prerequisites .....	120
Selecting Tablespaces .....	121
Incorporation Functions for Tablespaces .....	122
Keywords .....	123
XIV Incorporating DB2 and SQL/DS Tables/Views .....	125
16 Incorporating DB2 and SQL/DS Tables/Views .....	127
Prerequisites .....	128
Selecting DB2 and SQL/DS Tables/Views .....	129
Incorporation Functions for Tables/Views .....	130
Keywords .....	133
XV Comparison .....	135
17 Comparison .....	137
XVI Concepts of Comparison .....	139
18 Concepts of Comparison .....	141
Functional Scope .....	142
Resetting the Modified after generation Flag .....	143
Format of Reports Created by Comparison Functions .....	143
Calling Comparison Functions .....	143
XVII Comparing Adabas Databases .....	145
19 Comparing Adabas Databases .....	147

Prerequisites .....	148
Parameters .....	149
XVIII Comparing Adabas Files .....	151
20 Comparing Adabas Files .....	153
Prerequisites and Restrictions .....	154
Parameters .....	155
XIX Comparing DDMs .....	159
21 Comparing DDMs .....	161
Prerequisites .....	162
Parameters .....	163
XX Comparing DB2 Databases .....	165
22 Comparing DB2 Databases .....	167
Prerequisites .....	168
Parameters .....	169
XXI Comparing DB2 Storagegroups .....	171
23 Comparing DB2 Storagegroups .....	173
Prerequisites .....	174
Parameters .....	175
XXII Comparing DB2 Tablespaces .....	177
24 Comparing DB2 Tablespaces .....	179
Prerequisites .....	180
Parameters .....	181
XXIII Comparing DB2 Tables/Views .....	183
25 Comparing DB2 Tables/Views .....	185
Prerequisites and Restrictions .....	186
Parameters .....	187
XXIV Administration of External Objects .....	189
26 Administration of External Objects .....	191
Overview of Options .....	192
Administrating External Databases .....	193
Administrating External Dataspaces .....	194
Administrating External Files .....	195
Administrating External Programs .....	196
Administrating External Storagespaces .....	197
XXV Using the Preprocessor .....	199
27 Using the Preprocessor .....	201
Calling the Preprocessor .....	202
Prerequisites - Work Files .....	203
Reports - Print Files .....	203
XXVI Preprocessor Statements .....	205
28 Preprocessor Statements .....	207
Overview of Preprocessor Statements .....	208
Format of Preprocessor Statements .....	209
CALL .....	209
COPY .....	210

ENTRY .....	211
FORMAT-BUFFER .....	212
GENERATE .....	213
PROGRAM .....	224
XXVII Creation of XRef Data .....	227
29 Creation of XRef Data .....	229
What Types of Code Use are Documented in XRef Data .....	230

---

## Preface

---

Many Predict functions process Predict documentation data as well as implemented code. This simplifies creation and maintenance of documentation and external objects and guarantees consistency between documentation and implementation. This documentation describes in detail the functions which process both documentation and external objects.

The functions described in this documentation are available from two different plug-ins: Schema Generation and Object Description. See the table below to find out which plug-in to use for which purpose.

This documentation covers the following topics:

<b>Handling of External and Documentation Objects</b>	An overview of the functions which process documentation and external objects. The information provided in this section is essential for understanding many functions described in later sections of this documentation.
<b>Generation of External Objects</b>	A wide variety of external objects can be generated from documentation objects stored in Predict.  Only available with Schema Generation. When Schema Generation has been installed and activated, the generation function can also be invoked directly from Object Description.
<b>Incorporation</b>	Predict objects can be created by incorporating information from implemented objects which are stored in an external environment.  Only available with Schema Generation.
<b>Comparison</b>	Predict objects and external objects can be compared, and if any inconsistencies are detected, either the documentation or external object can be updated.  Only available with Object Description.
<b>Administration of External Objects</b>	Implemented objects that are connected to Predict objects can be administered using Predict functions.  Only available with Object Description.

---



# 1 About this Documentation

---

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

## Document Conventions

---

Convention	Description
<b>Bold</b>	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	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 <http://documentation.softwareag.com>. The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

### Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to [empower@softwareag.com](mailto: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](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.

---

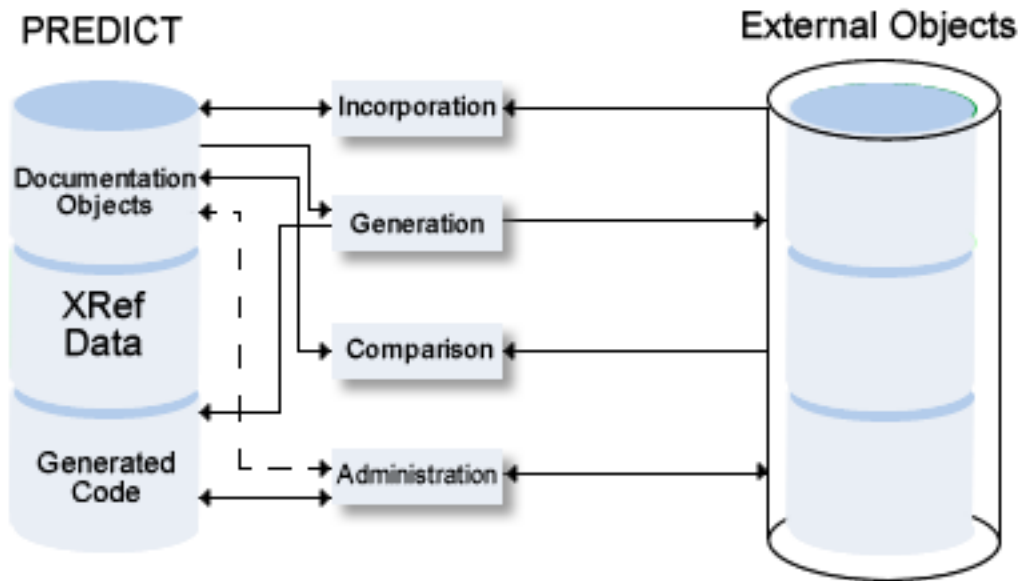
# 2 Handling of External and Documentation Objects

---

- What is an External Object ..... 6
- Connection of External and Documentation Objects ..... 7

Information stored in Predict objects can be used to generate external objects, and documentation objects can be incorporated from external objects. The concepts of handling external and documentation objects in Predict are described in this chapter.

Information provided in this chapter is needed to understand many options of generation, incorporation and comparison functions.



This chapter covers the following topics:

## What is an External Object

---

In Predict, data definition objects for use in applications that can be connected to Predict documentation objects are called “external objects”.

The following rules apply.

### Administration

- External objects can be generated from Predict documentation or be processed with incorporation functions.
- Each external object can be connected to a documentation object with a generation or incorporation function. For the impact of connecting external and documentation objects, see [Connection of External and Documentation Objects](#).

- For external objects implemented with SQL (DB2, Oracle, SQL/DS storagegroups, databases, tablespaces, dbspaces, tables and views) Predict stores the SQL statements that have been generated in a generation protocol. If several generation runs are executed, the protocol is extended for each generation.
- The objects can be administered with utilities of the application environment (for example SYSDDM, SYSAOS, SYSDB2, Natural map editor).

### Use

- Most of the object types are used at run time. Some of these objects are stored directly in the application environment, others are stored in the Predict system file. For objects that are stored in the Predict system file, Predict data must therefore be accessible at run time of the application which uses the external objects.

## Connection of External and 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.

### Impact of Connecting External and Documentation Objects

The connection of external objects and documentation objects affects the following activities.

- **Administration of External Objects**

The administration of external objects connected to documentation objects can be restricted with the Predict parameters of the SYSDDM utility, AOS, SYSDB2 utility and Rule in map editor. See *Protection* in the section *General Defaults* in the *Predict Administration* documentation. If these parameters are set to C (connected) the respective utilities cannot be used to administrate external objects that are connected to a Predict object.

- **Modification of Predict Objects**

Modification of Predict objects is affected as follows:

- Predict file objects of type Adabas that are connected to an implemented file cannot be unlinked from the Predict database object they belong to.

■ **Deleting Predict Objects**

Deleting Predict objects is affected as follows:

- Predict File and Verification objects connected to an external object cannot be deleted. To delete a Predict object connected to an external object, the two objects must be disconnected.
- A generation protocol created by a Generate DB2 ... function can only be purged by purging or disconnecting the implemented DB2 object.

■ **Incorporation of External Objects**

As a general rule, only external objects not yet connected to a documentation object can be processed with incorporation functions.

■ **Comparison of External and Documentation Objects**

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 Object Description commands **Disconnect** and **Delete**. See [Administration of External Objects](#).



# I Generation of External Objects

---



# 3

## Generation of External Objects

---

Only available with Schema Generation. When Schema Generation has been installed and activated, the generation function can also be invoked directly from Object Description.

This section covers the following topics:

[About Generation](#)

[Generate for Adabas](#)

[Generate for Natural](#)

[Generate for DB2](#)



# II

## About Generation

---

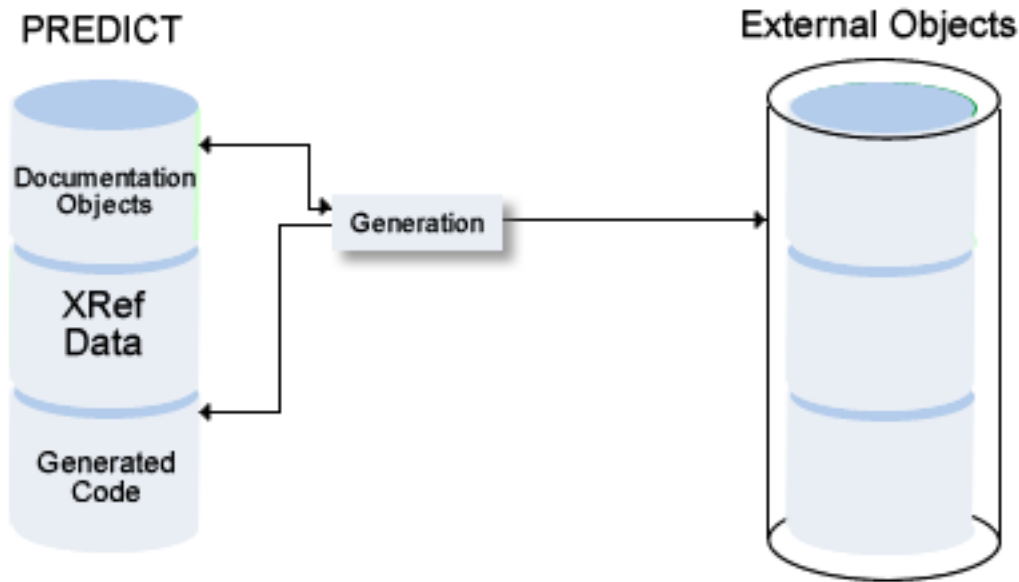


# 4 About Generation

---

- Overview of Generation Functions ..... 16
- Generation Defaults ..... 17
- Common Parameters ..... 18

External objects can be generated from Predict objects.



## Overview of Generation Functions

Object	Description of External Object
Adabas File	Adabas files are loaded directly into an Adabas database. If the file is already loaded, the differences of the implemented file and the documented file are determined and update commands are generated in order to transform the implemented Adabas file according to the documented file.
Data Definition Module	A collection of field definitions used by Natural for accessing a database.
DB2 database	A DB2 database is implemented directly as a physical DB2 database. Not applicable to SQL/DS.
DB2 procedure/function	This function requires an object of type Program as input, from which then either a procedure or a function is generated.
Verification rule	A rule must already have been generated using the Generate DDM function. Only the code of the rule is changed.  The new rule will automatically be used by Natural maps that are cataloged.



Object	Description of External Object
DB2-Storagegroup	Storagegroups are implemented directly from Predict storagespaces. If a storagegroup has already been implemented from the storagespace, the differences of the implemented DB2 storagegroup and the documented storagespace are determined and update commands are generated to adapt the implementation to the documentation.
DB2 table/view	DB tables/views are implemented directly from Predict file objects of type D or E.  If a table/view has already been implemented from the Predict table/view definition, the differences of the implemented DB2 table/view and the Predict table/view definition are determined and update commands are generated in order to change the implementation according to the documentation.
DB2 tablespace SQL/DS DBspace	Tablespaces/dbspaces are implemented directly from Predict dataspace. If a tablespace/dbspace has already been implemented from the dataspace, the differences of the implemented DB2 tablespace/ SQL/DS DBspace and the documented dataspace are determined and update commands are generated to transform the implementation according to the documentation.
Oracle table/view	Oracle tables/views are implemented directly from Predict file objects of type OT or OV.  If a table/view has already been implemented from the Predict table/view definition, the differences of the implemented Oracle table/view and the Predict table/view definition are determined and update commands are generated in order to change the implementation according to the documentation.

## Generation Defaults

Default generation values are set at installation. Most default values are displayed in the input screen of the respective generation function and can then be overwritten for temporary use.

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

- Default values of generation parameters can be changed in Predict: with the function Generation Defaults in the Modify Defaults menu or the command `DEFAULT object-code`.
- Generation defaults can be protected in Predict: by blanking out the "X" preceding the parameter in Modify ... Defaults screens. Protected default values cannot be changed when executing a generation function. The fields of protected parameters are locked in the **Generation Parameters** page of the schema generation wizard. These fields are skipped when positioning the cursor with the TAB key.
- Some default values are not displayed in the **Generation Parameters** page of the schema generation wizard and can therefore only be changed using the Modify Generation Defaults function in Predict. These parameters are described under *Presetting* in the descriptions of individual generation functions later in this section.

## Common Parameters

### Adabas Version



**Note:** If a new Adabas version is released that does not have any effect on Schema Generation functions, this new version will not appear in the selection menu. Use the code for the old version.

Code	Version	Remarks
I1	V 5.1 for IBM/Siemens	Applicable to all external object types for which this parameter can be specified. When generating copy/include code, sub/superdescriptors are not included in the record buffer layout.
I3	V 5.3 for IBM/Siemens	As above.
I6	V 6.1 for IBM/Siemens	As above. This Adabas version supports larger database and file numbers.
I7	V 7.1 for IBM/Siemens	As above.
I8	V 8.1 for IBM/Siemens	
I9	V 8.2 for IBM/Siemens	
O4	V 4.1 for IBM/Siemens	
U1	V 1.1 for UNIX	
U4	V 2.2 for UNIX	
U5	V 3.1 for UNIX/Windows	
U6	V 3.2 for UNIX/Windows	
U7	V 5.1 for UNIX/Windows	
U8	V 6.1 for UNIX/Windows	
U9	V 6.2 for UNIX/Windows	
V2	V 2.1 for VMS	
V3	V 3.1 for VMS	
V4	V 3.2 for VMS	
V5	V 4.1 for VMS	

# III

## Generate for Adabas - Adabas File

---



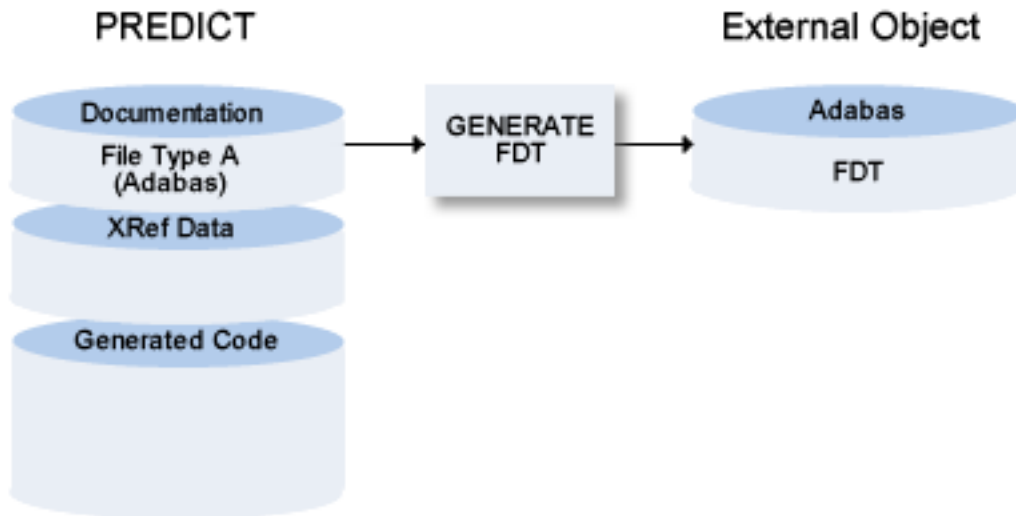
# 5

## Generate for Adabas - Adabas File

---

■ Parameters .....	23
■ Keywords .....	25
■ Sample Output .....	26

The function generates an Adabas FDT from a Predict file object of type A (Adabas).



#### Notes:

1. If Natural Security and Adabas Online Services are installed and function protection for Predict is activated, permission is required to generate an Adabas file.
2. With this function you can generate input for the ADAFDU utility to create a new file in your Adabas database. For existing files the FDT and the documentation of the file are compared and the input, necessary to adapt the existing FDT for the utilities ADAINV and ADADBM, is generated. The utilities are started immediately. This method automatically creates a connection between external and documentation object. Before you can delete the documentation object, you must either delete the external object (with the administration function Purge) or disconnect the external and documentation object (with function Disconnect). See also section [Handling of External and Documentation Objects](#) in this documentation.
3. When using Adabas Version 6.2 or above, it is possible and recommended to generate Adabas files using the Adabas call interface. This simplifies the generation of Adabas files if the Adabas versions of the source and destination differ from each other. In addition, using the Adabas call interface also offers the generation of Adabas files on remote Adabas databases.
4. An associated LOB file can be specified for a base file containing fields with format LO. A dedicated file object named SAG-ADA-LOB is created during installation. Whenever the number of an associated LOB file is entered in the Adabas attributes of a base file, an own set of Adabas attributes is stored for SAG-ADA-LOB having the specified file number. These settings can be customized. The function Generate Adabas file will automatically create the LOB file in Adabas, when the base file is generated.

5. Before an existing Adabas file will be updated, the file will be locked for general use with the ADAOPR LOCK utility. After generation (or in case of an abnormal end of a action) the file is unlocked.
6. When generating Adabas files, the value created for the MUPEX parameter is taken from the default settings.

A file generated with this method is handled like a file created with Adabas Online Services.

## Parameters

Before the following page is shown, another page may appear in which you have to select a physical file from a drop-down list box.

Adabas File for GER-FI-A of DB188 - Page 3 of 5 [IBM2 (7319)]

**ADABAS & NATURAL**

**Schema Generation**

software AG

Generation Parameters

Load file into database:       Replace if only FDT exists:

Update/add fields (FDT):

Update file parameters:       Replace if no records loaded:

Change file number:

Format DS/Index:

Adabas password:

List actions:

Stop users using file:

Cancel    <Back    Next >    Finish

Values for fields which have been locked by your data dictionary administrator cannot be overwritten. These fields are skipped when positioning the cursor with the TAB key. See [Generation Defaults](#).

The following parameters can only be set in Predict with the function Defaults > Generation Defaults for object type AF.

Default Parameters					
Delay after stop users	<p>Enter a value of up to 99 seconds. This is the time the system waits before actually performing the generation if parameter Stop users using file is set to Y. See below.</p> <p>Due to Adabas load or too much network traffic, it is possible that not all users have been stopped before the function Generate Adabas file is executed. This delay makes certain that all users really have been stopped before an Adabas file is generated.</p>				
Execute online invert	<table border="1"> <tr> <td>Y</td> <td>If set to N, no online invert function is executed.</td> </tr> </table> <p><b>Note:</b> This parameter applies only when generating for Adabas databases of version 7 or above.</p>	Y	If set to N, no online invert function is executed.		
Y	If set to N, no online invert function is executed.				
Only if TOPISN less than	<p>This function is only available if Execute online invert is set to Y. You can specify a value for TOPISN. The default value is 9999999999.</p> <p><b>Note:</b> This parameter applies only when generating for Adabas databases of version 7 or above.</p>				
Delay after online invert	<p>The online invert function of Adabas version 7 is executed as an asynchronous task. Only one online invert task may be active at a given time. Enter a value of up to 99 seconds. This is the time the system waits after one online invert before performing the next.</p> <p><b>Note:</b> This parameter applies only when generating for Adabas databases of version 7 or above.</p>				
Stop users using file	<table border="1"> <tr> <td>Y</td> <td>All current use of the Adabas file is stopped before a generation run is executed. This parameter must be set to Y on a separate terminal if a file currently used by other users is to be regenerated.</td> </tr> <tr> <td>N</td> <td>The function is terminated if other users are accessing the file.</td> </tr> </table> <p><b>Note:</b> If Natural Security and Adabas Online Services are installed and function protection for Predict is activated, permission is required to stop users using an Adabas file.</p>	Y	All current use of the Adabas file is stopped before a generation run is executed. This parameter must be set to Y on a separate terminal if a file currently used by other users is to be regenerated.	N	The function is terminated if other users are accessing the file.
Y	All current use of the Adabas file is stopped before a generation run is executed. This parameter must be set to Y on a separate terminal if a file currently used by other users is to be regenerated.				
N	The function is terminated if other users are accessing the file.				
Current VM	<p>Adabas files can only be generated from Predict file objects contained in databases accessible from the current virtual machine.</p>				
Parameters					
Load file into database	<table border="1"> <tr> <td>Y</td> <td>If no file already exists with given file number, a new file is created in the Adabas database.</td> </tr> </table>	Y	If no file already exists with given file number, a new file is created in the Adabas database.		
Y	If no file already exists with given file number, a new file is created in the Adabas database.				
Update/add fields (FDT)	<table border="1"> <tr> <td>Y</td> <td>Fields documented in Predict and missing in the FDT are added to the Adabas file. The field length in the FDT is adapted to the value in the Predict object, and fields defined as descriptors in the FDT but not in the Predict documentation are released from descriptor status.</td> </tr> </table>	Y	Fields documented in Predict and missing in the FDT are added to the Adabas file. The field length in the FDT is adapted to the value in the Predict object, and fields defined as descriptors in the FDT but not in the Predict documentation are released from descriptor status.		
Y	Fields documented in Predict and missing in the FDT are added to the Adabas file. The field length in the FDT is adapted to the value in the Predict object, and fields defined as descriptors in the FDT but not in the Predict documentation are released from descriptor status.				



Update file parameters	Y	File parameters, including the file space allocation, are adapted to the documentation. If the Adabas file is physically coupled to other files and these relationships are not documented, the files are uncoupled.
Change file number	Y	If the Predict file object was renumbered, the Adabas file is renumbered.
Format DS/Index		DATA storage is to be physically formatted.
Adabas password		Must be provided if the Adabas file (or only an FDT) already exists in the Adabas database and is protected by an Adabas security definition.
List actions	Y	Actions which are to be performed to adapt the Adabas file to the Predict specification are listed. Some of the actions may not yet be performed by this generation function: they must be run as batch utilities. The list shows all parameters which are used for generation. Parameters which are invalid or currently not able to be implemented are marked accordingly.
Replace if only FDT exists	Y	<p>If a file with this file number once existed in the database and was deleted with the KEEPFDI option, the remaining FDT is overwritten.</p> <p><b>Note:</b> The FDT of the file will not be deleted if the generation terminates abnormally, for example if the field list in Predict contains invalid field definitions.</p>
Replace if no records loaded	Y	The Adabas file parameters are replaced, because it is not possible to modify all parameters of an existing file. Replacement is only possible if the Adabas file is empty and is not physically coupled to other Adabas files. The current FDT is kept.
<b>Additional Parameters for UNIX</b>		
Adabas version		The version of Adabas for which the external object is to be generated (see <a href="#">Adabas Version</a> for more information).
Contiguous AC, DS, NI, UI		Setting of space allocations in Adabas utility ADAFDU for Address Converter, Data Storage, Normal Index and Upper Index. See your <i>Adabas Utilities</i> documentation for more information.

## Keywords

The schema generation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
File ID	FILE-ID
Phys. file number	FNR
Database ID	DB
Phys. database number	DBNR
Load file into database	LOAD
Replace if only FDT exists	REPLACE-FDT
Update/add fields (FDT)	UPD-FDT
Update file parameters	UPD-PARM
Replace if no records loaded	REPLACE-EMPTY
Change file number	RENUMBER
Format DATA storage	DATAFRM
Adabas password	PSW
List actions	LIST
Stop users using file	STOPF

If a parameter is not specified, the default value is taken.

## Sample Output

```
File ID ..... GENERATION-EXAMPLE          PFnr ...   171
Database ID .. DEMO-DB                      PDBnr ..   180
```

Action	Parameter	Remark
--------	-----------	--------

```
DEFFDT  FILE=171
        FNDEF='01,AA'
        Field: GROUP-1
        FNDEF='02,AB,14,U,NU'
        Field: ELE-N-9V5
        FNDEF='02,AC'
        Field: GR-IN-GROUP
        FNDEF='03,AD,4,B,DE,FI'
        Field: ELE-B-4
        FNDEF='03,AE,4,P,NU'
        Field: ELE-PS-5V2
        FNDEF='03,AF,4,B,NU,MU'
        Field: MU-B-4
        FNDEF='02,AG,42,A,NU'
        Field: ELE-A-42
        FNDEF='02,AH,8,G,FI'
        Field: ELE-F-8
```

```

FNDEF='02,AI,3,B,NU'
Field: ELE-B-3
FNDEF='01,AL,PE'
Field: PC-OCC-7
FNDEF='02,AM,10,U,DE,NU'
Field: PC-ELE-DE-NS-7V3
FNDEF='02,AN,4,P,DE,NU,MU'
Field: PC-MC-PS-6V1
FNDEF='02,AO'
Field: PC-GR
FNDEF='03,AP,2,B,FI'
Field: PC-ELE-I-2
FNDEF='03,AQ,14,P,NU'
Field: PC-PS-20V7
FNDEF='02,AR,4,G,FI'
Field: PC-ELE-F-4
FNDEF='01,AS,4,P,NU'
Field: ELE-D
FNDEF='01,AT,7,P,NU'
Field: ELE-T
FNDEF='01,AU,1,B,NU'
Field: ELE-L
SUBFN='AJ=AG(10,14)'
Field: SB-ELE-A-5
SUPDE='S1=AI(1,3),AG(20,36),-'
      'AE(1,4)'
Field: SP-DE-A-24
HYPDE='04,S2,4,F,MU,PE=AP,AF'
Field: HQ-DE-I-4
LOADFILE NOACEXTENSION=NO
MIXDSDEV=NO
DATAFRM=YES
MAXISN=3391
NAME=GENERATION-EXAMPLE
MINISN=1
ASSOPFAC=10
DATAPFAC=10
PGMREFRESH=NO
ISNREUSE=NO
LOADFILE DSREUSE=YES
MAXDS=0
MAXNI=0
MAXUI=0
CIPHER=NO
MAXRECL=5060
UISIZE=41B
NISIZE=80B
DSSIZE=233B
DSDEV=3390

```



# IV

## Generate for Natural

---



# 6 Generate for Natural

---

- Replace Verification Rule ..... 32
- Data Definition Module ..... 35

## Replace Verification Rule

---

The Replace Verification Rule function can be applied only to rules of status automatic. A rule is assigned status automatic by applying the function Generate DDM to the file containing the field that the corresponding Verification object is linked to via "Is verified by VE".

The function Replace Verification Rule replaces the code of the rule; all links to fields remain unchanged.

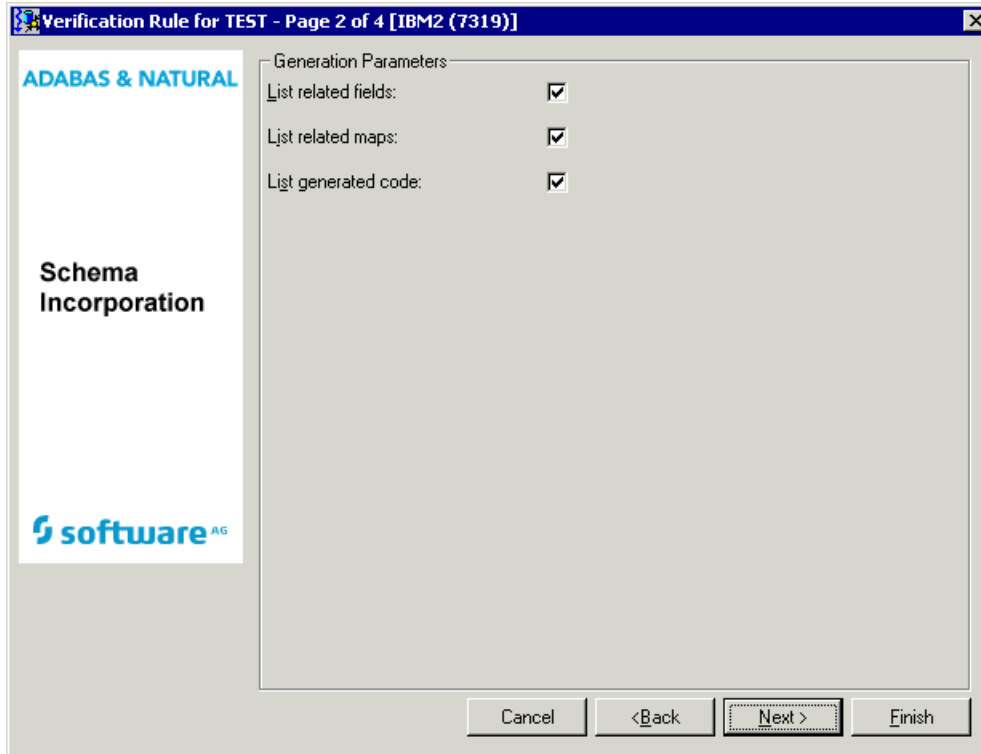
The following topics are covered below:

- [Parameters](#)
- [Keywords](#)



- Sample Output

## Parameters



Values for fields which have been locked by the data dictionary administrator cannot be overwritten. These fields are skipped when positioning the cursor with the TAB key. See [Generation Defaults](#).

Parameters	
List related fields	Y   The IDs of all fields connected to this verification rule via "Is verified by VE" are to be listed.
List related maps	Natural maps using replaced verifications are to be recataloged so that they use the new verification rules. List related maps helps to find these maps.
	Y   The IDs of all maps connected to this verification rule are to be listed.
List generated code	Y   The generated code is to be listed.

## Keywords

The schema generation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
Rule ID	RULE-ID
List related fields	RELATE
List generated code	LIST
List related maps	RELATED-MAPS

If a parameter is not specified, the default value is taken.

## Sample Output

```

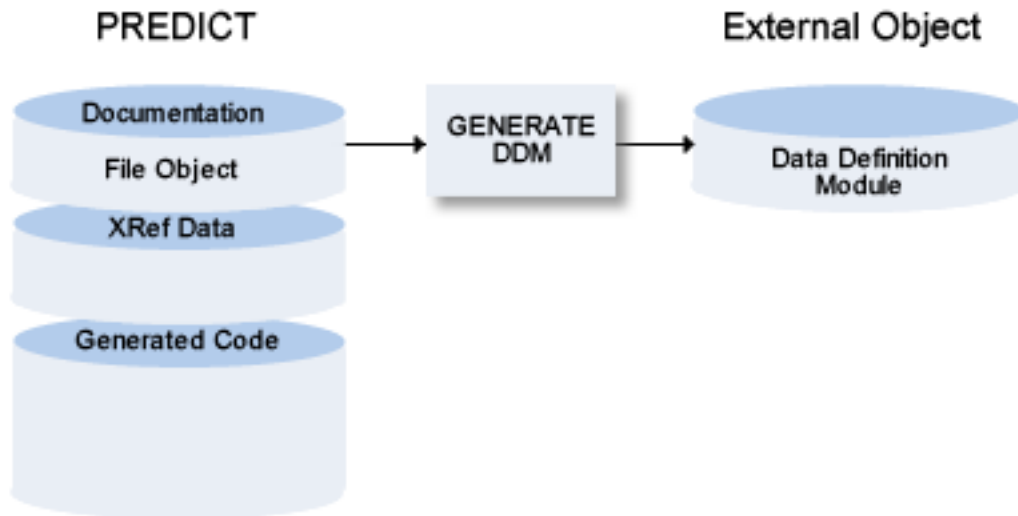
Link of Verification GEN-CHECK-ZERO to Elementary Fields
File Name                               Field Name                               Remark
-----
GENERATION-EXAMPLE                       ELE-N-9V5                                activ
                                           ELE-PS-5V2                                activ
                                           PC-MC-PS-6V1                              activ
Gen-examples                             Ele-N-9V5                                 not activ
                                           Ele-PS-5V2                                 not activ
                                           PC-MU-PS-6V1                              not activ

Rule: GEN-CHECK-ZERO

0010 * *****
0020 * Verification: GEN-CHECK-ZERO generated by Predict          *
0030 * with format: Numeric; Type: Not Equal;                    *
0040 * on: 2007-05-31; at: 13:03:46; from user: SMR;            *
0050 * *****
0060 IF & = 0
0070 REINPUT 'Invalid value(s): ''0'' .'
0080 MARK *&

```

## Data Definition Module



The following topics are covered below:

- [Rules Applying to the Generation of DDMs](#)
- [Rules Applying to the Generation of DDMs When Using Unicode](#)
- [Parameters](#)
- [Generating DDMs for Use with IMS](#)
- [Generating a DDM for Use with Adabas Vista](#)
- [Recommendations when Using Adabas Vista](#)
- [Keywords](#)
- [Sample Output](#)

### Rules Applying to the Generation of DDMs

- If Natural Security is installed and the specified file is defined to Natural Security, the same checks are performed as are performed before a DDM can be regenerated or a file can be added. See description of the parameter DDM Modifier in the section *File Maintenance* of the Natural Security documentation.
- A DDM generated for a physically coupled file contains the ID of the file to which it is coupled and the short names of the fields via which they are coupled as a comment. For example: "Coupled via AA to GA of FINANCE".

- The names of the Predict file and its userviews begin with SYSDIC. When a DDM is generated for a file whose name begins with SYSDIC, the logical database number (255) and logical file number (253) that point to the FDIC are taken.
- If the file (which has a DDM) was renamed, the old DDM is purged in the Generate DDM function and the new DDM is added. If Natural Security objects or Super Natural objects exist for the old DDM, these objects are renamed too.



**Note:** Rules applying to the generation of DDMs for use with IMS or Adabas Vista are described in the sections *Generating DDMs for Use with IMS* and *Generating a DDM for Use with Adabas Vista*.

When generating DDMs in Windows or UNIX environments and the FDIC file is located on the mainframe, the DDM will not be available on the mainframe. It will be stored only in the file system where FDDM is located. When generating DDMs on the mainframe and the FDIC file is also located on mainframe, the DDM will not be available in the Windows or UNIX environment. The function must be executed twice.

### Rules Applying to the Generation of DDMs When Using Unicode

Database fields containing Unicode data are represented with the format U in the DDM, when running with Natural Version 4.2 for Mainframes or above, Natural Version 6.2 for UNIX or above and Natural Version 6.2 for Windows or above.

Whether a field contains Unicode data, is determined in the following way:

- For DB2 files (file types D, E, MT), the CCSID definition of the table (or of the underlying tables in case of views and materialized query tables) in the SQL Server specified in the DDM Generation defaults is evaluated.

If it is set to U, the fields with format A, AV, AL or LO, and character set option set to S or to M and fields with format G, GV, or GL are represented with format U in the DDM. If no CCSID value is given, the CCSID definition of the related dataspace in the given SQL Server is checked.

- For Adabas files (file types A, U, AT) each field with format A, AV, AL or LO and character set option M is represented with format U in the DDM.

### Parameters

Before the following page is shown, another page may appear in which you have to select a database from a drop-down list box. Its appearance depends on specific defaults defined by the administrator and/or the object that has been selected on the first page of the wizard.

Example for Adabas:

Values for fields which have been locked by your data dictionary administrator cannot be overwritten. These fields are skipped when positioning the cursor with the TAB key. See [Generation Defaults](#).

Presettings	
The parameters below can be changed in the Modify DDM Defaults screen of Predict. See <a href="#">Generation Defaults</a> .	
Specification DB ID	Only applicable to Adabas files and userviews.
	Y Database ID <i>must</i> be specified.
	N Database ID <i>may</i> be specified.
	D Database ID <i>may not</i> be specified.
This parameter is set to N at installation.	
Money format (for SQL)	N Unpacked numeric
	P Packed numeric
Date/Time representation	A Alphanumeric / Char
	D Date/Time
<b>Note:</b> A representation as format D or T is only taken into account for DB2 tables and views and when executing the generation with Natural Version 4.1 for mainframes or above.	

SQL Server	The server that should be used for data access from Natural.  <b>Note:</b> It is possible to define an index as clustered in one server and as not clustered in a second server. This influences whether the component fields of the index may be updated or not. Whether a field may be updated or not is indicated by special short names in the DDM for Natural.
Super Natural file DBNR	The database number of the Super Natural system file must be supplied by the DDA if the Super Natural file is not the current FUSER file at generation time. This parameter is set to 0 at installation.
VSAM trailing BLANK char	If a VSAM prefix name contains trailing blanks, these blanks must be represented in Predict with a special character (because trailing blanks cannot be entered in Predict). When generating a DDM for Natural VSAM the special character specified is removed and the correct representation of a blank inserted. A VSAM trailing BLANK char. is not defined at installation. We recommend setting this parameter to \$.
Super Natural file FNR	The file number of the Super Natural system file must be supplied by the DDA if the Super Natural file is not the current FUSER file at generation time. This parameter is set to 0 at installation.
<b>Parameters</b>	
Replace	Y   An existing DDM is overwritten.
List generated code	Y   Generated code is listed.
Adabas version	The version of Adabas for which the DDM is to be generated (see <a href="#">Adabas Version</a> for more information). The Adabas version you specify for DDM generation is independent of your installation platform.
Field name prefix	Prefix to be used for each generated field name.  <b>Note:</b> This parameter does not apply if you are generating a DDM from an SQL file type.
Line comments	Y   The first abstract line from each Predict field object is truncated to 20 characters and appended as a line comment to the generated description.
	N   No line comments are generated from Predict object.
	O   Occurrences defined in Predict.
Abstract	The number of Predict abstract lines to be included in the generated code as lines beginning with an asterisk.
General comments	Y   Comment lines in the Predict file object (marked with ** or */ in field Type) are included. If the field synonym is used, the original field name is displayed. The date and time of the generation and the ID of the user who initiated it are inserted. Source fields of sub/super/hyperdescriptors are also listed.
	N   Comment lines are not included. Source fields of sub/super/hyperdescriptors are not listed.

	D	Derivation only. Similar to Y, but comment lines are only included, where a derivation occurs. Source fields of sub/super/hyperdescriptors are also listed.
Generate verification rules	Y	The status of verification objects connected to fields of this file via "Is verified by VE" is changed to automatic.
Replace verification rules	Y	Verification objects of status automatic connected to fields of this file via "Is verified by VE" are regenerated if they have been modified since generation.
	S	Verification objects of status automatic connected to fields of this file via "Is verified by VE" are regenerated only if they are not connected to any other file.
List verification rules	Y	List the IDs of all verification objects connected to each field.
<b>For Adabas</b>		
Use Vista access-nr	See table <a href="#">Meaning of Parameter Use Vista access-nr</a> .  <b>Note:</b> This parameter is only applicable to Adabas files and userviews. See also section <i>Adabas Vista</i> in the <i>Predict and Other Systems</i> documentation.	
Long fields as DYNAMIC	Controls whether fields with format LO or AV in files of type A/U are generated with dynamic or fixed length in the DDM.	
	Y	Yes.
	N	No.
	S	As defined in Predict. This is the default value.
<b>For SQL</b>		
Truncate creator	Y	The name of the generated DDM is the original SQL table/view name <i>without</i> the part that identifies the creator.
	<b>Note:</b> Only applicable to SQL file types.	
NAT-I for long fields	Y	Fields longer than 253 and with Natural field length specified are represented as multiple fields (M) in the generated DDM.
	N	The specified field length is taken.
<b>For IMS</b>		
Generate UDFs	Y	User-defined fields for Natural for DL/I are generated or regenerated.
	Only applicable to files of type I, J, and K.	
Replace UDFs	Y	Any user-defined fields are replaced.
List UDFs	Y	User-defined fields are to be listed.
IMS field suffix	Y	The DDM field name is created using a compressed field name (or Natural synonym), hyphen and IMS segment name.
<b>For Natural subsystems</b>		

Define in security	Only applicable if Natural Security is installed and the user is authorized to create profiles. An authorized user is an administrator who either has no owners or is an owner of himself without countersignatures.	
	Y	An authorized user can create a Natural Security profile for the file whose DDM has just been generated.

### Generating DDMs for Use with IMS

When generating a DDM for one of the IMS file types, the following fields are included:

For all file types (I, J, K)

- KEY fields from the higher level IMS segments
- KEY fields from the current IMS segment
- KEY fields from the lower level IMS segment if data is accessed via an AIX index

Additionally for IMS Segments (file type I)

- user-defined fields of all IMS segment layouts (file type J)
- user-defined fields of the related logical child or logical parent segment

IMS Segment (file type J)

- user-defined fields of the named IMS segment layout
- user-defined fields of the related logical child or logical parent segment

IMS userview (file type K)

- user-defined fields of the named IMS User View file
- user-defined fields of the related logical child or logical parent segment

### Generating a DDM for Use with Adabas Vista

#### Prerequisites

If an Vista access number is used for generating a DDM, the virtual machine of the Vista element used for generation must be linked to the current network.

Changing the Current VM and Current Network is described under *Miscellaneous* in the *Predict Administration* documentation.



## Recommendations when Using Adabas Vista

If using Schema Generation with Adabas Vista distributed data processing, we recommend the following:

- Always enter Vista access numbers for Adabas files and userviews (Add, Modify file in Predict).
- Set the parameter Specification DB ID in the Modify DDM defaults screen of Predict to D (database ID may not be specified when generating a DDM from an Adabas file/userview) or to N (database ID may be specified but is not mandatory). If this parameter is set to Y, the user is forced to enter a database ID when generating a DDM from an Adabas file/userview.
- Set the parameter Use Vista access-nr in the Modify DDM defaults screen of Predict to T, but do not protect the field.
- Do not enter a database ID when generating a DDM. This ensures the Vista access number is used for generation.

If you do not want to use Adabas Vista:

- Set the parameter Use Vista access-nr in the Modify DDM defaults screen of Predict to N and deactivate the option by blanking out the preceding X.

The table below illustrates the use of the parameter Use Vista access-nr.

### Meaning of Parameter Use Vista access-nr

Specification DB ID	Use Vista Access-Nr	Database ID specified	Result
D,N	N	no	DDM is generated with DB ID=0
D,N	Y	no	If the file does not contain Vista access numbers, an error message is given.  If the file contains Vista access numbers, a DDM is generated with the access numbers taken from the file.
D,N	T	no	If the file does not contain Vista access numbers, an error message is given.  If the file contains Vista access numbers, the system tests whether a Vista element with the same numbers exists for this file. If so, a DDM is generated with this access numbers.
N,Y	N	yes	[1] If the file is not defined as partitioned in this database and the Adabas attribute Vista access only is not set to Y: the DDM is generated with the physical database number and physical file number (defined with file maintenance function Modify Adabas attributes).
N,Y	N	yes	[2] If the file is defined as partitioned in this database or the Adabas attribute Vista access only=Y: the system checks whether a Vista element for this file and database exists.

Specification DB ID	Use Vista Access-Nr	Database ID specified	Result
			<p>If exactly one element exists, the DDM is generated with the Vista numbers defined in this Vista element.</p> <p>If more than one Vista element is found, no DDM is generated and an error message is given.</p>
N,Y	T,Y	yes	<p>Same as above, except that if several Vista elements exist for the file and database, the system checks whether the Vista access numbers of the file are identical to the Vista numbers of one of these Vista elements.</p> <p>If identical, the DDM is generated.If not, no DDM is generated and an error message is given.</p>

### Keywords

The schema generation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
File ID	FILE-ID
Database ID	DATABASE-ID
Replace	REPLACE
Field name prefix	PREFIX
Line comments	LINE-COM
General comments	GENERAL-COM
Abstract	SHORT-COM
List generated code	LIST
Generate verification rules	GENERATE-RULE
Replace verification rules	REPLACE-RULE
List verification rules	LIST-RULE
Generate UDFs	GENERATE-UDF
Replace UDFs	REPLACE-UDF
List UDFs	LIST-UDF
Adabas version	ADA-VER
Truncate creator	TRUNC-CREATOR
IMS field suffix	IMS-SUFFIX
Use Vista access nr.	USE-ACCESS-NR
Define in security	ADD-SECURITY
Status <sup>1)</sup>	STATUS
DDM Modifiers <sup>1)</sup>	MODIFIER1

Field	Keyword
DDM Modifiers <sup>1)</sup>	MODIFIER1-COUNTER-SIGN
DDM Modifiers <sup>1)</sup>	MODIFIER2
DDM Modifiers <sup>1)</sup>	MODIFIER2-COUNTER-SIGN
DDM Modifiers <sup>1)</sup>	MODIFIER3
DDM Modifiers <sup>1)</sup>	MODIFIER3-COUNTER-SIGN
DDM Modifiers <sup>1)</sup>	MODIFIER4
DDM Modifiers <sup>1)</sup>	MODIFIER4-COUNTER-SIGN
DDM Modifiers <sup>1)</sup>	MODIFIER5
DDM Modifiers <sup>1)</sup>	MODIFIER5-COUNTER-SIGN
DDM Modifiers <sup>1)</sup>	MODIFIER6
DDM Modifiers <sup>1)</sup>	MODIFIER6-COUNTER-SIGN
DDM Modifiers <sup>1)</sup>	MODIFIER7
DDM Modifiers <sup>1)</sup>	MODIFIER7-COUNTER-SIGN
DDM Modifiers <sup>1)</sup>	MODIFIER8
DDM Modifiers <sup>1)</sup>	MODIFIER8-COUNTER-SIGN
Owner <sup>1)</sup>	OWNER1
Owner <sup>1)</sup>	OWNER1-COUNTER-SIGN
Owner <sup>1)</sup>	OWNER2
Owner <sup>1)</sup>	OWNER2-COUNTER-SIGN
Owner <sup>1)</sup>	OWNER3
Owner <sup>1)</sup>	OWNER3-COUNTER-SIGN
Owner <sup>1)</sup>	OWNER4
Owner <sup>1)</sup>	OWNER4-COUNTER-SIGN
Owner <sup>1)</sup>	OWNER5
Owner <sup>1)</sup>	OWNER5-COUNTER-SIGN
Owner <sup>1)</sup>	OWNER6
Owner <sup>1)</sup>	OWNER6-COUNTER-SIGN
Owner <sup>1)</sup>	OWNER7
Owner <sup>1)</sup>	OWNER7-COUNTER-SIGN
Owner <sup>1)</sup>	OWNER8
Owner <sup>1)</sup>	OWNER8-COUNTER-SIGN
Security Notes <sup>1)</sup>	NOTE1
Security Notes <sup>1)</sup>	NOTE2
Security Notes <sup>1)</sup>	NOTE3
Security Notes <sup>1)</sup>	NOTE4
Security Notes <sup>1)</sup>	NOTE5

Field	Keyword
Security Notes <sup>1)</sup>	NOTE6
Security Notes <sup>1)</sup>	NOTE7
Security Notes <sup>1)</sup>	NOTE8
Security Notes <sup>1)</sup>	NOTE9
Security Notes <sup>1)</sup>	NOTE10
NAT Length	NAT-LENGTH
Long fields as DYNAMIC	LONG-DYNAMIC

If a parameter is not specified, the default value is taken.

<sup>1)</sup> Only available in Natural Security. For further details and possible values see the Natural Security documentation.

### Sample Output

```

          DBID: 180  FNR: 171  DDM: GENERATION-EXAMPLE          DEF.SEQ:
T L  DB  NAME                                F LENG  S D  REMARKS
-----
*      Generation started
*      at 2007-05-31 13:19:05
*      by user MSZ
*
*      EXAMPLE FILE FOR THE
*      GENERATION SUBSYSTEM
*
G 1  AA  GROUP-1
    2  AB  ELE-N-9V5                N 9.5   N
        HD=ELE/HEADER

    VER:  GEN-CHECK-ZERO

G 2  AC  GR-IN-GROUP
    3  AD  ELE-B-4                  B 4.0   F D

    VER:  GEN-VER-EXAM

    3  AE  ELE-PS-5V2                P 5.2   N

    VER:  GEN-CHECK-ZERO

*      MU field in group redefined two
*      times
M 3  AF  MU-B-4                    B 4.0   N
    2  AG  ELE-A-42                 A 42.0  N

```

```

VER:  GEN-CHECK-BLANK
VER:  GEN-VER-EXAM

2  AH  ELE-F-8                F 8.0  F
2  AI  ELE-B-3                B 3.0  N
*   Superfield and superdescriptor
1  AJ  SB-ELE-A-5            A 5.0  N
*   ----- SOURCE FIELD(S) -----
*   ELE-A-42(10-14)

1  S1  SP-DE-A-24            A 24.0  N S
*   ----- SOURCE FIELD(S) -----
*   ELE-B-3(1-3)
*   ELE-A-42(20-36)
*   ELE-PS-5V2(1-4)
*
*   PE-group with automatic counter
*
P 1  AL  PC-OCC-7
2  AM  PC-ELE-DE-NS-7V3      N 7.3  N D
*   counter for MU-field in PE-group
M 2  AN  PC-MC-PS-6V1        P 6.1  N D

VER:  GEN-CHECK-ZERO

*
G 2  A0  PC-GR
3  AP  PC-ELE-I-2            B 2.0  F
Format 'INTEGER' changed to 'BINARY'.
3  AQ  PC-PS-20V7            P 20.7  N
2  AR  PC-ELE-F-4            F 4.0  F
*   Formats: date,time,logical
1  AS  ELE-D                  D 6.0  N
1  AT  ELE-T                  T 12.0  N
1  AU  ELE-L                  L 1.0  N
*   Hyperdescriptor with MU field
*   and source field of a PE-group
M 1  S2  HQ-DE-I-4            I 4.0  H
*   ----- SOURCE FIELD(S) -----
*   PC-ELE-I-2
*   MU-B-4
***** DDM replaced *****

```



**V**

**Generate for DB2**

---





# 7 Generate for DB2

---

▪ Common Parameters for All DB2 Types .....	50
▪ DB2 Database .....	50
▪ DB2 Procedure/Function .....	53
▪ DB2 Table/View .....	56
▪ DB2 Storagegroup .....	62
▪ DB2 Tablespace .....	64

## Prerequisites

Generation of DB2 objects is subject to DB2 security mechanisms. See the section *DB2 and SQL/DS* in the *Predict and Other Systems* documentation for more information.

## Common Parameters for All DB2 Types

---

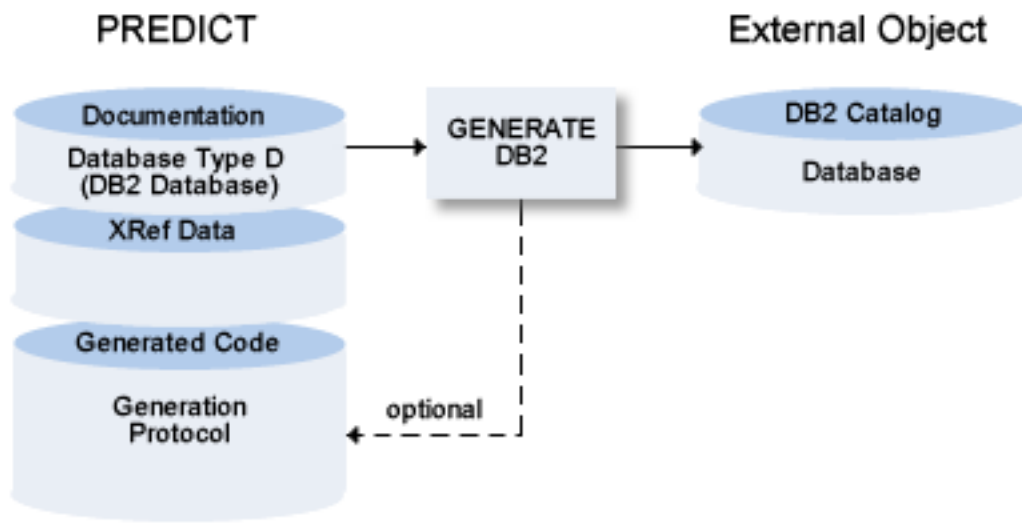
The following parameters can be set in all DB2 object types.


Parameters	
SQL Server	This parameter is defined on page 2 of the schema generation wizard. This page does not appear when only the default server can be used. The SQL server, the object will be implemented in. The value is preset with the value specified in the user or system profile.
Use SQLID	To switch from the logon user ID to a group ID in DB2, you can enter a different SQL ID in this field. The statement <code>SET CURRENT SQLID</code> will then be submitted.
List DB2 statement	<input type="checkbox"/> List the generated DB2 SQL statements.
Protocol saved in member	SQL statements generated for the implementation of the database can be saved in a Natural source member (protocol) on the FDIC file. A member name for the protocol is only required when the DB2 object is generated for the first time. SQL statements generated in subsequent generation runs are concatenated to the protocol. The protocol can be displayed using the Display function of the Administration Implemented <i>object-type</i> menu of Predict.
Protocol saved in library	Library where the protocol is to be saved.

## DB2 Database

---

A DB2 database can be implemented with Schema Generation by generating the necessary SQL statements. The generated SQL statements can be saved in a generation protocol.



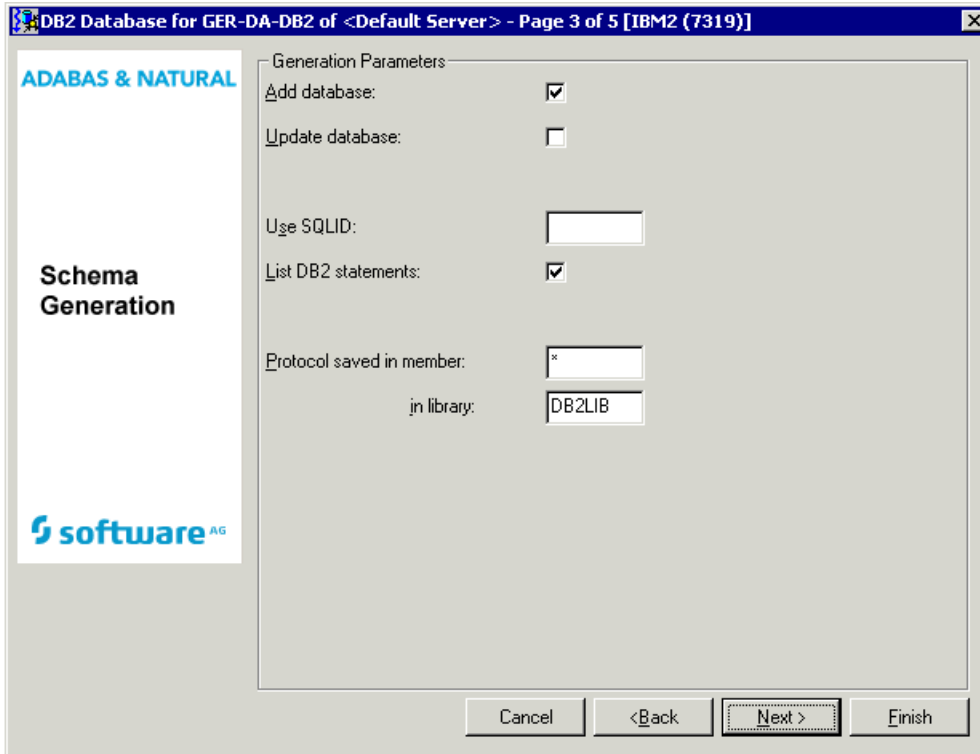
 **Note:** This function is only available if DB2 is installed in your server environment. The function is not applicable to SQL/DS.

The following topics are covered below:

- [Parameters](#)
- [Keywords](#)

- [Sample Output](#)

## Parameters



Values for fields which have been locked by your data dictionary administrator cannot be overwritten. These fields are skipped when positioning the cursor with the TAB key. See [Generation Defaults](#).

Common parameters which are not included in the table below are described under [Common Parameters for All DB2 Types](#).

Presetting	
The parameter below can be changed in the Modify DB2 Database Defaults screen of Predict. See <a href="#">Generation Defaults</a> .	
Count of saved actions	Number (0-99) of generation actions which are stored in the protocol specified under Protocol saved in member/library. If 99 is entered, all generation actions are stored. This is the default setting at installation.
Parameters	
Add database	Y   A new DB2 database is created. If the DB2 database already exists, this parameter must be set to N.
Update database	Y   If the database is already implemented, the differences of the implemented database and the Database object in Predict are

Presetting	
	determined and update commands are generated to modify the implemented database according to the Predict object.

## Keywords

The schema generation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
Database ID	DB
List DB2 statement	LIST
Protocol saved in member	MEM
Protocol saved in library	LIB
Add database	ADD-DB
Update database	UPD-DB
Use SQLID	SQLID
SQL Server	SQL-SERVER

If a parameter is not specified, the default value is taken.

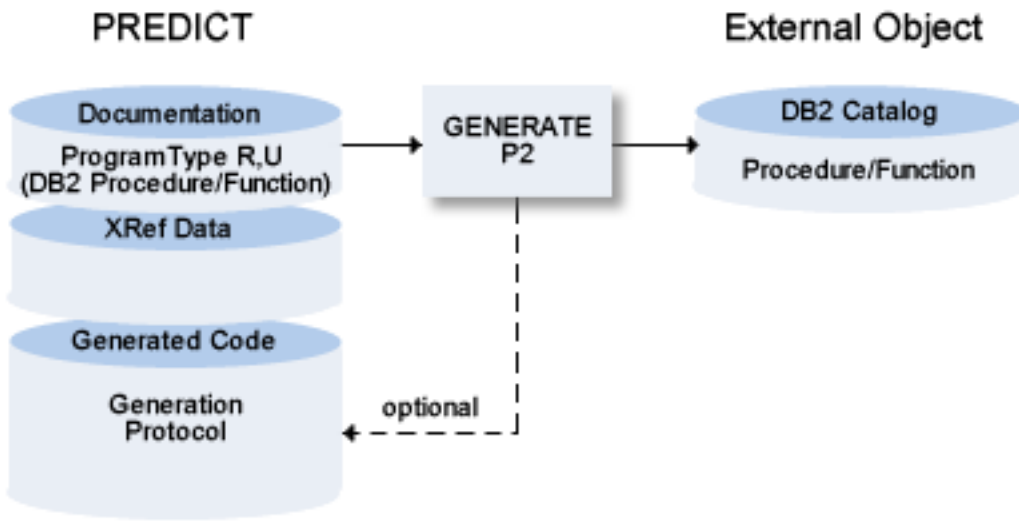
## Sample Output

```
Database-ID: EXAM-DATABASE
```

```
CREATE DATABASE EXAMDB
  STOGROUP EXAM001
  BUFFERPOOL BP1;
```

## DB2 Procedure/Function

A DB2 procedure or function requires an object of type Program as input, from which then either a procedure or a function is generated.



 **Note:** This function is only available if DB2 is installed in your server environment. The function is not applicable to SQL/DS.

The following topics are covered below:

- [Parameters](#)

- Keywords

## Parameters

Values for fields which have been locked by your data dictionary administrator cannot be overwritten. These fields are skipped when positioning the cursor with the TAB key. See [Generation Defaults](#).

Common parameters which are not included in the table below are described under [Common Parameters for All DB2 Types](#).

Presetting	
The parameter below can be changed in the Modify DB2 Proc/Function Defaults screen of Predict. See <a href="#">Generation Defaults</a> .	
Count of saved actions	Number (0-99) of generation actions which are stored in the protocol specified under Protocol saved in member/library. If 99 is entered, all generation actions are stored. This is the default setting at installation.
Parameters	
Add procedure/function	Y   A new DB2 procedure/function is created. If the DB2 procedure/function already exists, this parameter must be set to N.
Replace	Y   An existing procedure/function in DB2 is dropped and created newly.

Update procedure/function	Y	If the procedure/function is already implemented, the differences of the implemented procedure/function and the program object in Predict are determined and update commands are generated to modify the implemented procedure/function according to the Predict object.
Comment on	Y	The first eight abstract lines of Predict field and file objects are used as DB2 comments for the columns and the table (SQL statement COMMENT ON).



**Note:** Links concerning the associations IN (Input FI) and RE (Returns FI) define the input and output parameters of the procedure or function.

## Keywords

The schema generation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
Program ID	PROGRAM-ID
Add procedure/function	ADD-PROCEDURE
Update procedure/function	UPD-PROCEDURE
Replace	REPLACE
Comment on	COMMENT
List DB2 statement	LIST
Protocol saved in member	MEM
Protocol saved in library	LIB
Use SQLID	SQLID
SQL Server	SQL-SERVER

If a parameter is not specified, the default value is taken.

## DB2 Table/View

The Generate DB2 Table / View function:

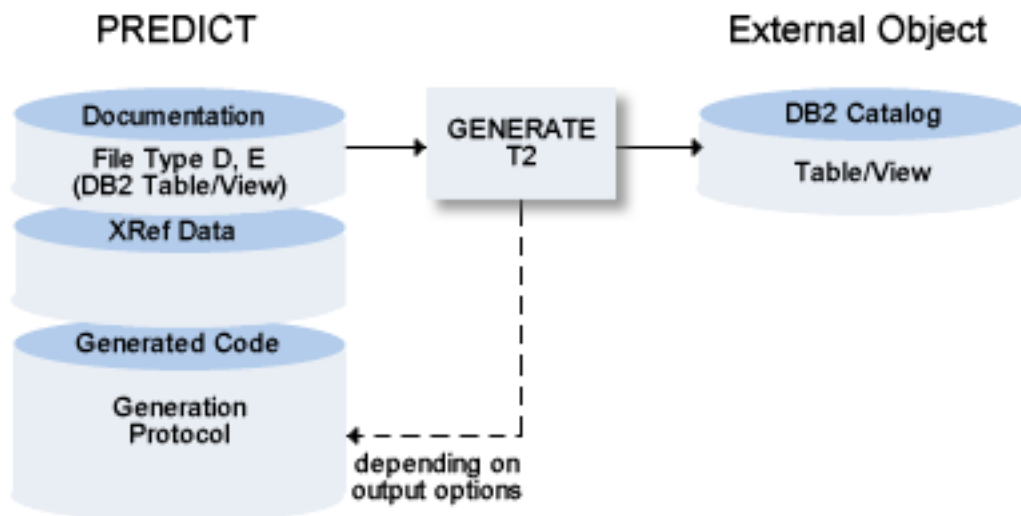
- Generates DB2 and SQL/DS Table/View/Indexes. If a table contains a LOB column, the function generates a SET CURRENT RULE='STD' statement which allows DB2 to create the necessary auxiliary tablespaces, tables and indexes for these objects.
- Generates distinct types.
- Generates check expressions. A check expression that differs from the documentation in Predict is dropped and recreated.



- Generates triggers. If a trigger is linked to a DB2 view, an `INSTEAD OF` trigger is generated. A trigger that differs from the documentation in Predict is dropped and recreated.
- Modifies existing DB2 and SQL/DS Table/Indexes, replaces DB2 and SQL/DS Views/Indexes or
- Purges indexes from DB2 or SQL/DS tables if no longer documented with a corresponding Predict file object of file type D (DB2 Table).
- Purges masks or permissions from DB2 tables if no longer documented with a corresponding Predict file object of file type D (DB2 Table).



**Note:** If an XML column (Field format LO and Character Set X) is added for the first time to a DB2 table, you are recommended to run the function Compare DB2 table after the generation of the DB2 table. The function Compare DB2 table must be executed on the server with the update options Add fields and Field attributes set to Y.

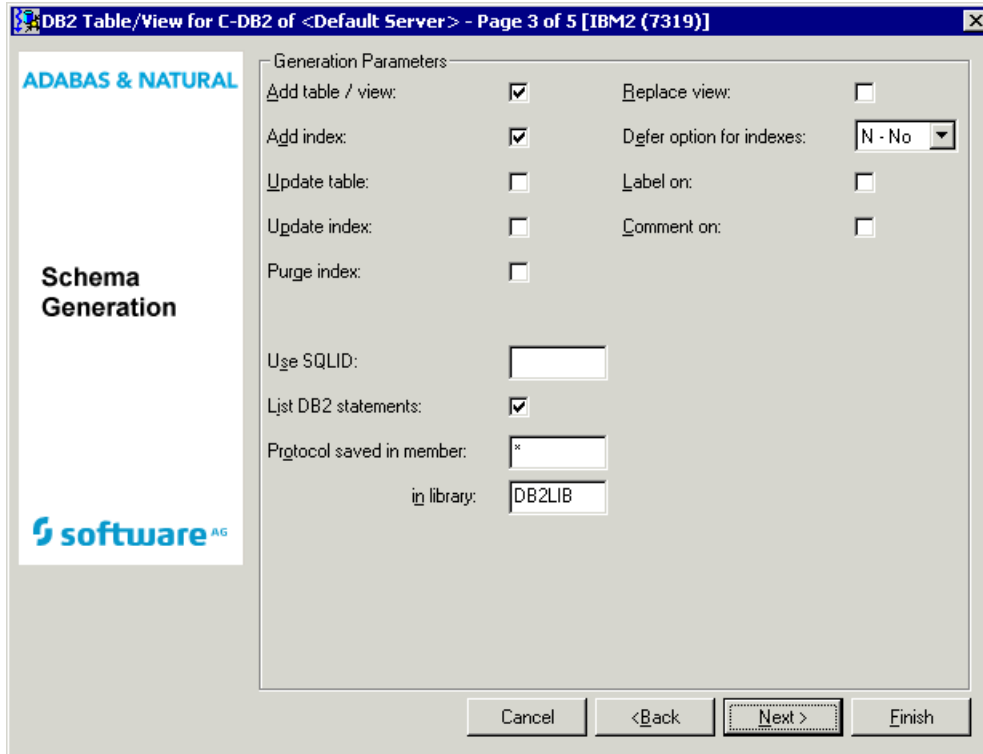


The following topics are covered below:

- [Parameters](#)
- [Keywords](#)

- [Sample Output](#)

## Parameters



Values for fields which have been locked by your data dictionary administrator cannot be overwritten. These fields are skipped when positioning the cursor with the TAB key. See [Generation Defaults](#).

Common parameters which are not included in the table below are described under [Common Parameters for All DB2 Types](#).

Presetting	
The parameter below can be changed in the Modify DB2 Table/View Defaults screen of Predict. See <a href="#">Generation Defaults</a> .	
Count of saved actions	Number (0-99) of generation actions which are stored in the protocol specified under Protocol saved in member/library. If 99 is entered, all generation actions are stored. This is the default setting at installation.
Parameters	
Add table/view	Y   A new DB2 table/view is created. If the DB2 table already exists, this parameter must be N.
Replace view	Y   If the DB2 view already exists, the DB2 statements DROP VIEW and COMMIT are generated first, followed by the CREATE VIEW statement.

Add index	If DB2 indexes are defined for the DB2 table in Predict, set this option to Y to generate the CREATE INDEX statements.																	
Update table	Y	If the DB2 table/view is already implemented, commands are generated to adapt the DB2 table/view to the documentation (if differences were found). This includes the generation of necessary ALTER TABLE DROP COLUMN statements for fields that are no longer documented.																
Update index	Y	<p>If the DB2 index is already implemented, update commands are generated to adapt the DB2 index to the documentation (if differences were found).</p> <p>Updating DB2 indexes is only possible for differences of the following attributes: buffer pool, close, freepage, pctfree, VSAM catalog, storage group, primary allocation, secondary allocation, erase, index type, GBPCACHE, piece size.</p> <p>If any other attributes differ, Update index and Purge index must be set to Y. Predict will then update the index by issuing the statements DROP INDEX, COMMIT and CREATE INDEX.</p>																
Purge index	Y	<p>A DROP INDEX and a COMMIT statement are generated if a DB2 index exists and no corresponding index is defined in Predict.</p> <p><b>Note:</b> Partitioned indexes cannot be purged.</p>																
Column masks	<p>Options for column masks. Possible values:</p> <table border="1"> <tr> <td>E</td> <td>Enable</td> </tr> <tr> <td>D</td> <td>Disable</td> </tr> <tr> <td>A</td> <td>Activate</td> </tr> <tr> <td>X</td> <td>Deactivate</td> </tr> <tr> <td>EA</td> <td>Enable &amp; Activate</td> </tr> <tr> <td>EX</td> <td>Enable &amp; Deactivate</td> </tr> <tr> <td>DA</td> <td>Disable &amp; Activate</td> </tr> <tr> <td>DX</td> <td>Disable &amp; Deactivate</td> </tr> </table>		E	Enable	D	Disable	A	Activate	X	Deactivate	EA	Enable & Activate	EX	Enable & Deactivate	DA	Disable & Activate	DX	Disable & Deactivate
E	Enable																	
D	Disable																	
A	Activate																	
X	Deactivate																	
EA	Enable & Activate																	
EX	Enable & Deactivate																	
DA	Disable & Activate																	
DX	Disable & Deactivate																	
Row permissions	Options for row permissions. Possible values: See Column masks option above.																	
Defer option for indexes	Y	The SQL statement CREATE INDEX is generated with the option DEFER YES.																
	N	The SQL statement CREATE INDEX is generated without the option DEFER YES.																
Label on	Y	<p>The string specified in the Predict field attribute Header1 is used as DB2 label for the columns (SQL statement LABEL ON).</p> <p><b>Note:</b> If Label on is set to Y, existing labels are generally overwritten even if Header1 is not specified in Predict for each column in the table.</p>																

Comment on	Y The first eight abstract lines of Predict field and file objects are used as DB2 comments for the columns and the table (SQL statement COMMENT ON).  <b>Note:</b> Comment on is set to Y, existing comments are generally overwritten even if no abstract is specified in Predict.
------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Special Considerations for Temporal Tables and Archive-Enabled Tables

When generating temporal tables, Predict only requires the name of the history or archive table to be used. This history or archive table is then automatically generated and connected to the base table. The history or archive table name can be provided when documenting a temporal table in Predict.

If you want to use a history or archive table with an index of its own then this history or archive table must be documented separately in Predict. Indexes can then be defined manually where required.

Such a separate history or archive table must then also be generated separately. The connection between the base table and the history or archive table, that is normally maintained automatically in Predict, will then be dropped. You will have to manually and individually apply changes made to the base table also in the history or archive table.

After the changes have been applied, you will have to generate both, the base table and the history or archive table, individually in order for the changes to take effect.

A third generation process is then required to reestablish the connection between the base table and the history or archive table.

### Keywords

The schema generation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
File ID	FILE-ID
Add table/view	ADD-TABLE
Add index	ADD-INDEX
Update table	UPD-TABLE
Update index	UPD-INDEX
Purge index	PURGE-INDEX
Replace view	REPLACE
List DB2 statements	LIST
Protocol saved in member	MEM
Protocol saved in library	LIB

Field	Keyword
Label on	LABEL
Comment on	COMMENT
Defer option for indexes	DEFER
Use SQLID	SQLID
SQL Server	SQL-SERVER
Column masks	MASK
Row permissions	PERMISSION

If a parameter is not specified, the default value is taken.

### Sample Output

```

File-ID: EXAM-GEN_TABLE
CREATE TABLE EXAM.GEN_TABLE
  (EMPNO              CHAR(6)          NOT NULL,
   FIRSTNME          VARCHAR(12)       NOT NULL,
   MIDINIT           CHAR(1)          NOT NULL,
   LASTNAME           VARCHAR(15)      NOT NULL,
   WORKDEPT          CHAR(3)          NOT NULL WITH DEFAULT,
   PHONENO           CHAR(4)          NOT NULL WITH DEFAULT,
   HIREDATE           DATE             NOT NULL WITH DEFAULT,
   JOB               CHAR(8)          NOT NULL WITH DEFAULT,
   EDLEVEL           SMALLINT         NOT NULL WITH DEFAULT,
   SEX               CHAR(1)          NOT NULL WITH DEFAULT,
   BIRTHDATE         DATE             NOT NULL WITH DEFAULT,
   SALARY            DECIMAL(9, 2)    NOT NULL WITH DEFAULT,
   BONUS             DECIMAL(9, 2)    NOT NULL WITH DEFAULT,
   COMM             DECIMAL(9, 2)    NOT NULL WITH DEFAULT,
   PRIMARY KEY (EMPNO))
IN DATABASE EXAMDB
EDITPROC PROGDB
AUDIT NONE;
CREATE UNIQUE INDEX PRDDBA.EMPNO
ON EXAM.GEN_TABLE
  (EMPNO              ASC)
SUBPAGES 4
BUFFERPOOL BPO
CLOSE YES;
CREATE INDEX DSN8210.XEMP2
ON EXAM.GEN_TABLE
  (WORKDEPT          ASC)
SUBPAGES 4
BUFFERPOOL BPO
CLOSE YES;

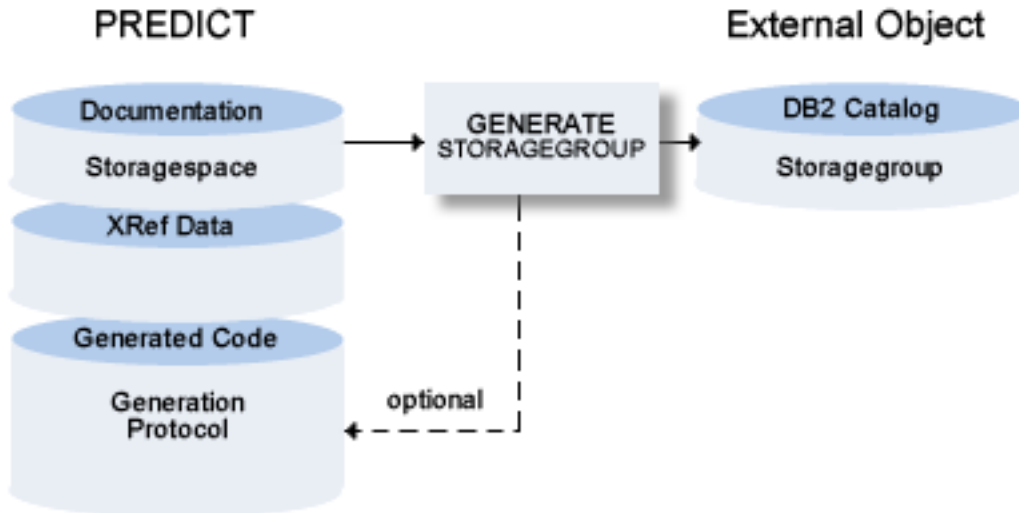
DIC1880 SUMMARY:      14 FIELD(S) PROCESSED

```

## DB2 Storagegroup

---

A DB2 storagegroup can be implemented with Schema Generation by generating the necessary SQL statements. The generated SQL statements can be saved in a generation protocol.



The following topics are covered below:

- [Parameters](#)
- [Keywords](#)

- Sample Output

## Parameters

Values for fields which have been locked by your data dictionary administrator cannot be overwritten. These fields are skipped when positioning the cursor with the TAB key. See [Generation Defaults](#).

Common parameters which are not included in the table below are described under [Common Parameters for All DB2 Types](#).

Presetting	
The parameter below can be changed in the Modify DB2 Storagegroup Defaults screen of Predict. See <a href="#">Generation Defaults</a> .	
Count of saved actions	Number (0-99) of generation actions which are stored in the protocol specified under Protocol saved in member/library. If 99 is entered, all generation actions are stored. This is the default setting at installation.
Parameters	
Add storagegroup	Y   A new DB2 storagegroup is created. If the DB2 storagegroup already exists, this parameter must be set to N.

Presetting	
Update storagegroup	Y   If the DB2 storagegroup is already implemented, commands are generated to adapt the DB2 storagegroup to the documentation (if differences were found).

## Keywords

The schema generation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
Storagespace ID	STORAGESPACE-ID
Add Storagegroup	ADD-STORAGEGROUP
Update Storagegroup	UPD-STORAGEGROUP
List DB2 statements	LIST
Protocol saved in member	MEM
Protocol saved in library	LIB
Use SQLID	SQLID
SQL Server	SQL-SERVER

If a parameter is not specified, the default value is taken.

## Sample Output

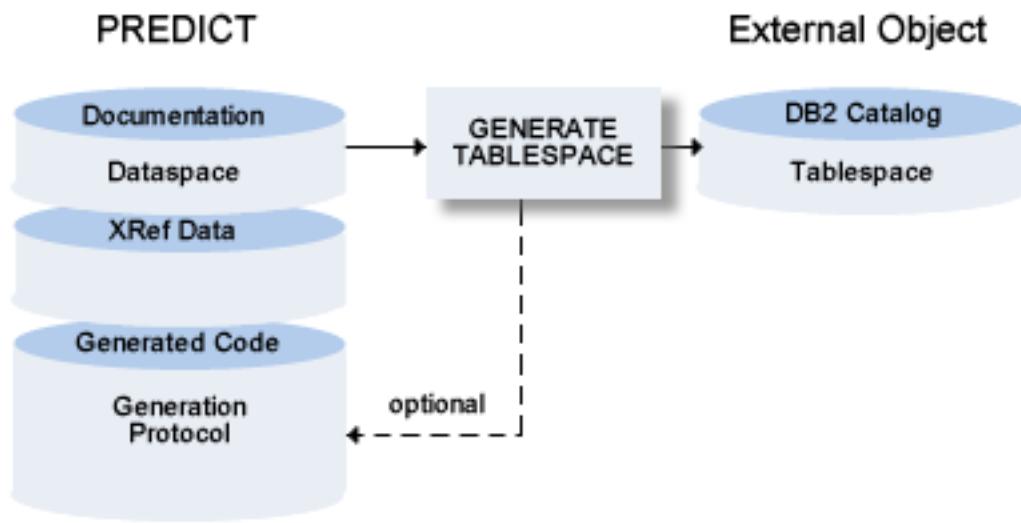
```
Storagespace-ID: EXAM-STORAGEGROUP

CREATE STOGROUP EXAM001
  VOLUMES (VOL05 )
  VCAT DB2;
```

## DB2 Tablespace

A DB2 tablespace can be implemented with Schema Generation by generating the necessary SQL statements. The generated SQL statements can be saved in a generation protocol.



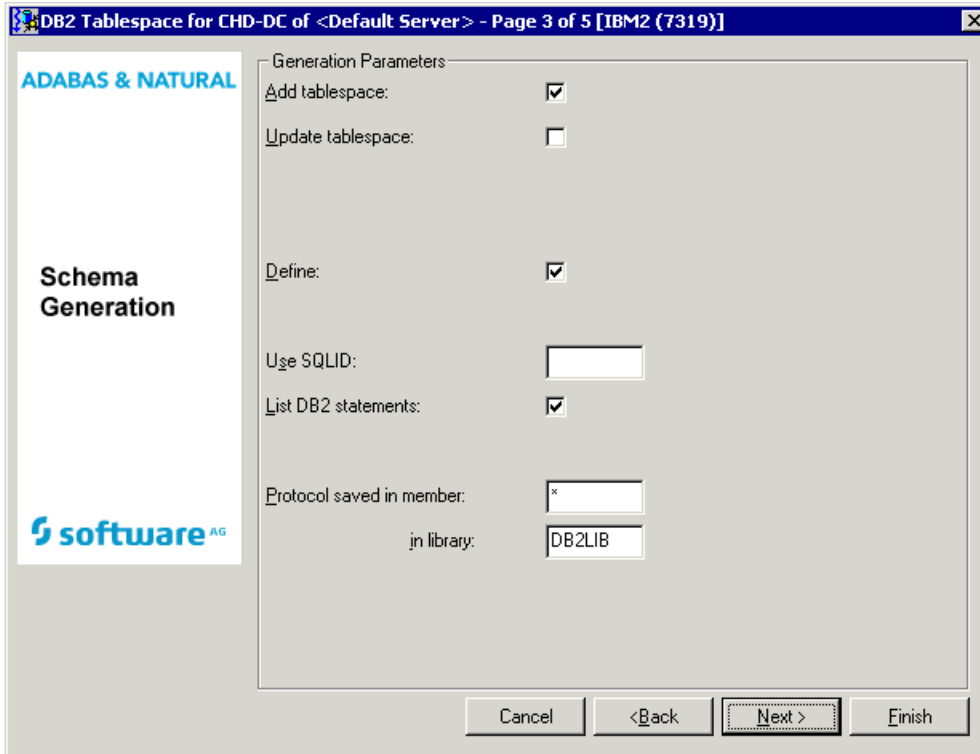


The following topics are covered below:

- [Parameters](#)
- [Keywords](#)

- Sample Output

## Parameters



Values for fields which have been locked by your data dictionary administrator cannot be overwritten. These fields are skipped when positioning the cursor with the TAB key. See [Generation Defaults](#).

Common parameters which are not included in the table below are described under [Common Parameters for All DB2 Types](#).

Presetting	
The parameter below can be changed in the Modify DB2 Tablespace Defaults screen of Predict. See <a href="#">Generation Defaults</a> .	
Count of saved actions	Number (0-99) of generation actions which are stored in the protocol specified under Protocol saved in member/library. If 99 is entered, all generation actions are stored. This is the default setting at installation.
Parameters	
Add tablespace	Y   A new DB2 tablespace is created. If the DB2 tablespace already exists, this parameter must be set to N.
Update tablespace	Y   If the tablespace is already implemented, the differences of the implemented tablespace and

Presetting	
	<p>the documented dataspace are determined and update commands are generated to modify the implemented tablespace according to the documented dataspace.</p> <p><b>Note:</b> Certain parameters may be modified with the ALTER TABLESPACE statement only when the database is active, other parameters only when the database is stopped. For this reason it may be necessary to submit the generation job twice.</p>
Define	N   A DEFINE NO clause is generated
Partitions initialized	Only for tablespaces with type G (Partition by growth). Specifies the number of partitions to be initialized (corresponds to the NUMPARTS option in DB2) and must not exceed the total number of partitions defined for that tablespace.

## Keywords

The schema generation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
Dataspace ID	DATASPACE-ID
Add tablespace	ADD-TABLESPACE
Update tablespace	UPD-TABLESPACE
List DB2 statements	LIST
Protocol saved in member	MEM
Protocol saved in library	LIB
Define	DEFINE
Use SQLID	SQLID
SQL Server	SQL-SERVER
Partitions initialized	NUMPARTS

If a parameter is not specified, the default value is taken.

## Sample Output

```
Dataspace-ID: EXAM-TABLESPACE
```

```
CREATE TABLESPACE EXAMTS IN EXAMDB  
  NUMPARTS 4  
  BUFFERPOOL BPO  
  LOCKSIZE ANY  
  CLOSE NO;
```

# VI

## Incorporation

---

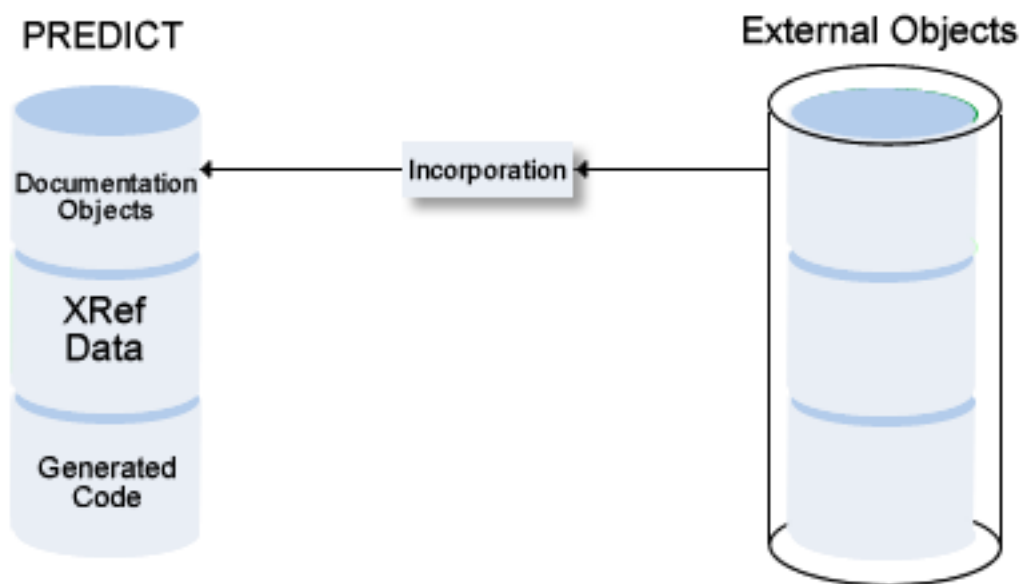


# 8 Incorporation

---

Only available with Schema Generation.

Predict objects can be created or updated by incorporating information from external objects.



This section covers the following topics:

**Concepts of Incorporation**

**Incorporating Adabas Databases**

**Incorporating Adabas Files**

**Common Parameters for All DB2 Types**

**Incorporating DB2 Storagegroups**

**Incorporating DB2 Databases**

**Incorporating DB2 Tablespaces and SQL/DS DBspaces**

**Incorporating DB2 and SQL/DS Tables/Views**



# VII

## Concepts of Incorporation

---



# 9 Concepts of Incorporation

---

- Overview of Incorporation Options ..... 76
- Connecting External and Predict Objects ..... 76
- Calling Incorporation Functions ..... 78

Incorporation functions are used

- to find external objects that are not yet documented/connected to a documentation object,
- to document applications that are not yet documented in Predict,
- to redocument applications that are not completely documented in Predict.

## Overview of Incorporation Options

---

Incorporation functions perform the following tasks:

### ■ Test

Compare external and documentation objects. The function finds implemented objects already documented with Predict objects, where the implemented object and the Predict object are not connected.



**Note:** In Predict, this incorporation function is called Compare.

### ■ Connect

Connect external and documentation objects. The function connects documentation and implementation objects and - if appropriate - the documentation object is partially updated. For example: physical file attributes are updated, but file lists are not.

### ■ Display Masters

Display IDs of objects that must be documented in Predict before objects can be incorporated.

- Display DB2 databases, tablespaces and storagegroups that must be documented in Predict before tables/views or tablespaces can be incorporated.

### ■ Incorporate

Incorporate external objects. A Predict object is created, its attributes are taken from the implemented object. The implemented object and the new documentation object are connected.

## Connecting External and Predict Objects

---

The incorporation functions Test, Connect and Incorporate can be used to connect Predict objects and external objects.

See the section [Handling of External and Documentation Objects](#) in this documentation for a description of the basic principles of connecting external and Predict objects.

## Identifying Corresponding External and Documentation Objects

Incorporation functions read information from Predict and from external sources to try and find objects that correspond. Corresponding objects are identified as follows:

Predict Object	Corresponding Object Identified with
Adabas Database	Physical database number (P-DBnr)
Adabas File	Physical database and file number (P-DBnr/P-Fnr); see also note below.
DB2 Database	DB2 database name
DB2,SQL/DS Tables/Views	DB2,SQL/DS table/view name - Predict file ID
DB2 Storagegroup	DB2 storagegroup name
DB2 Tablespace	DB2 tablespace name



**Note:** With partitioned files, a Predict file object can be implemented with several physical files: In this case the database/file number of implemented files and corresponding Predict file objects need not be identical. Predict evaluates the Vista translation table to determine implemented files documented with Predict file object when the database/file numbers are not identical.

## Sources of Information on External Objects

Predict reads the following external sources of information:

Documentation Object	External Source of Information
Adabas Database	Adabas GCB
Adabas file	Adabas FCB and FDT
DB2 Database	DB2 catalog
DB2,SQL/DS Tables/Views	DB2 catalog
DB2 Storagegroup	DB2 catalog
DB2 Tablespace	DB2 catalog

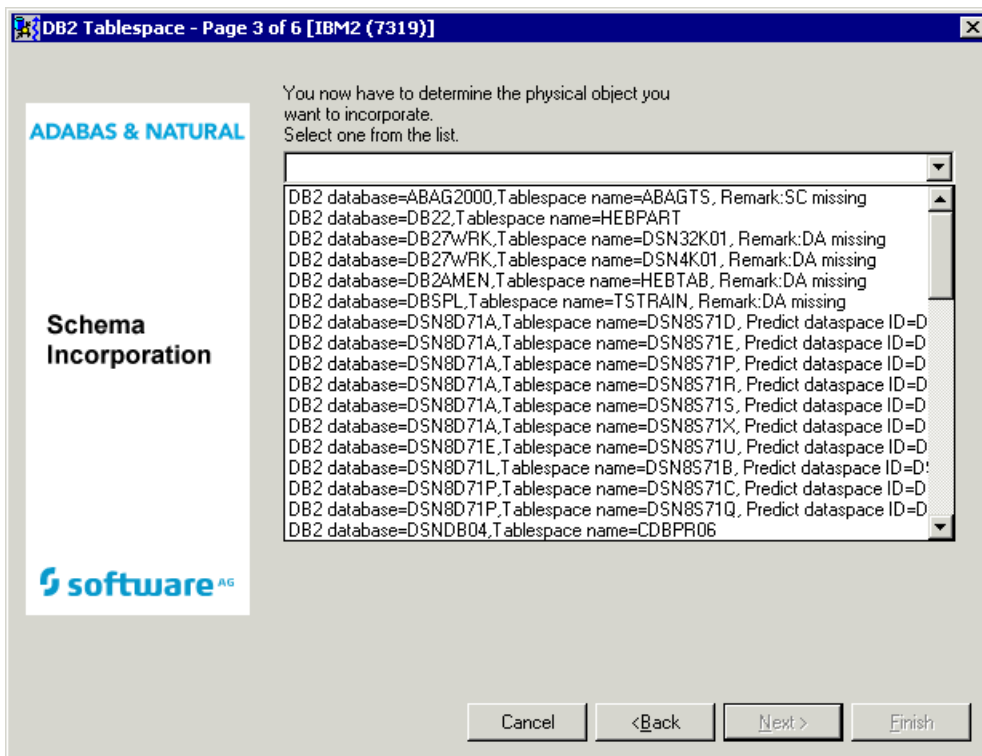
## Calling Incorporation Functions

### Incorporation Functions are Executed in Two Steps


- Create a list of external objects not yet connected to any Predict documentation object by specifying selection criteria.
- Process the selected objects with incorporation functions.

#### Step 1: Selecting the Objects to be Processed

A drop-down list box containing external objects is provided. Selection criteria determine which objects are contained in the list. The content of the list depends on the external object type.



**Remarks which apply to all Object Types**

 **Note:** Type-specific remarks are described with the respective external object types.

Remark	Function	Description
Equal	Test	No significant differences between external and documentation object. Connection between the two can be established.
Different		Significant differences between external and documentation object. No connection between the two possible.
Connected	After execution of incorporation function Connect	Connection between external and documentation object was established.
Not connected		No connection established due to significant differences between external and documentation object.
Incorporated	After execution of function Incorporate	New Predict object created with information from external object.
Not incorporated		No new Predict object created.

**Step 2: Processing Objects with Incorporation Functions**

**Test**

Compares the external objects with Predict objects and reports differences between attributes and lists of children (if applicable). Requires that both an implemented object and a corresponding Predict object exist.

**Connect**

Connects the external object to an existing Predict object and

- copies current information from external object to the Predict object;
- writes to the generation log of the object that a corresponding implemented database existed when the connection was established;
- connects child objects to the corresponding external objects.

**Display Masters**

When incorporating tables/views/tablespaces, higher-level objects (databases, storagegroups, tablespaces) must be documented in Predict. This function finds higher-level objects that are not documented.

### **Incorporate**

Creates Predict objects from information of external objects and connects the Predict objects to the external objects. Applicable to all external object types.

- When incorporating files/tables/views, Predict objects for the files in the database and links from files to fields are also created.

Type-specific parameters determine how the different incorporation functions work.



# VIII

## Incorporating Adabas Databases

---



# 10 Incorporating Adabas Databases

---

■ Prerequisites .....	84
■ Restrictions .....	84
■ Selecting Adabas Databases .....	85
■ Incorporation Functions for Adabas Databases .....	86
■ Keywords .....	87

## Prerequisites

---

- Adabas Online Services must be active.
- Databases must be Adabas Version 6 databases or above.
- If Natural Security is installed and function protection for Predict is activated, access rights must be given.

## Restrictions

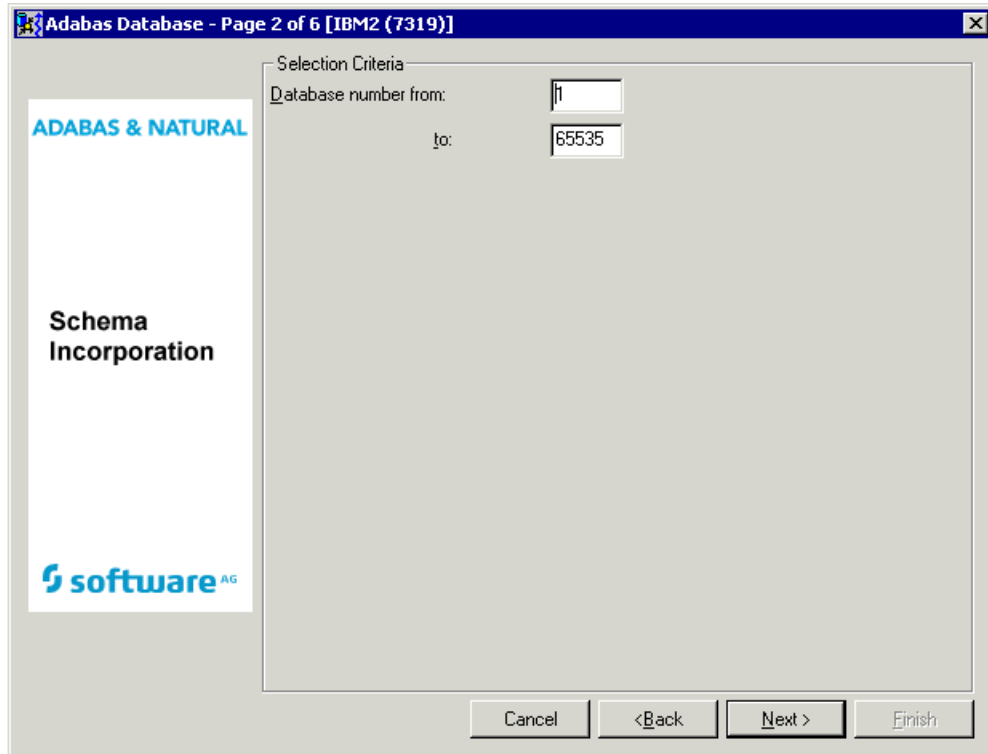
---

- Only databases accessible from the current virtual machine can be processed



**Note:** See [Concepts of Incorporation](#) for basic information on how to use incorporation functions.

## Selecting Adabas Databases



Parameters	
Database number from/to	The physical numbers of the Adabas databases to be incorporated. A range of up to 23 databases can be specified.
Update DB attributes (connect option)	Provided on a later page of the schema incorporation wizard (after the physical object has been selected from the drop-down list box). Y - The attributes of the Predict database object are to be updated.

Databases which meet the given selection criteria and are not already connected to a Predict database are then provided in the drop-down list box on the next page of the schema incorporation wizard.

Information in the drop-down list box	
DBnr	Number of the database.
Adabas name	The Adabas name of the database (the name specified with the ADADEF or the ADAREF utility).
Predict database ID	The IDs of Predict database object.

Information in the drop-down list box	
Remark	Any of the following type-specific remarks or blank. For type-independent remarks see <a href="#">Remarks which apply to all Object Types</a> .  <b>Note:</b> If any of the following remarks is given, no incorporation function is possible.
DB error: nnnn	Where nnnn is an Adabas response code between 3001 and 3255. The database is not defined in NTDB as type Adabas.
Vista element exists	Incorporating the database would destroy the consistency of the Predict documentation.
Database with different type exists	A Predict database object with a different database type exists. Incorporating the database would destroy the consistency of the Predict documentation.
Exists in other Network	Unique database number is required and the database already is exists in another network. Incorporating the database would destroy the consistency of the Predict documentation.
Access via Vista	The database to be incorporated is only accessible using Vista, but Vista support is not activated in Predict.

## Incorporation Functions for Adabas Databases

Valid functions:

- Test
- Connect
- Incorporate

### Test

Compares (tests) the database and the Predict database object and reports differences between attributes and file lists.

### Connect

Connects the database to the existing Predict database object, and

- copies current information from Adabas to Predict object
- writes to the generation log of the object that a corresponding implemented database existed when the connection was established
- connects the checkpoint and security file (if any) of the Adabas database to Predict file objects in the file list of the database with the same file number.

Specify the following parameters in an additional page of the schema incorporation wizard.

Connect Options	
Update DB attributes	See the parameter table in <i>Selecting Adabas Databases</i> for a description of this parameter.

## Incorporate

### Scope of the Incorporation

- A Predict database object of type A is created and connected to the external object.
- Vista elements for the database are created
  - if Vista is installed and
  - the Vista translation table contains translation elements with this database specified as target.
- The new database object is linked to the current virtual machine.
- If this database has a checkpoint or security file, the following files are added to the file list of the database: *SAG-ADA-CHECKPOINT* and *SAG-ADA-SECURITY*.

Specify the following parameters in an additional page of the schema incorporation wizard.

Incorporation Options	
Database ID	The ID of the new database object in Predict (mandatory). The Adabas name of the database is displayed and can be accepted or overwritten. The Adabas name is specified in the ADADEF or the ADAREF utility when creating the database.
List incorporated code	Y   The values of all attributes incorporated for the database are listed (optional).

## Keywords

The schema incorporation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
Database number from	DBNR-FROM
Database number to	DBNR-TO
Function	SUB-FUNCTION
Database ID	DATABASE-ID
List incorporated code	LIST
Update DB attributes	UPD-DB





# IX

## Incorporating Adabas Files

---



# 11 Incorporating Adabas Files

---

■ Prerequisites and Restrictions .....	92
■ Selecting Adabas Files .....	93
■ Incorporation Functions for Adabas Files .....	95
■ Keywords .....	96

## Prerequisites and Restrictions

---

- A Predict database object for the Adabas database of the file must exist.
- If Natural Security is installed and function protection for Predict is activated, access rights must be given.
- Certain features of the Connect and Incorporate functions require Adabas Online Services to be active and Adabas Version 6 or above (see descriptions below).



**Note:** See *Concepts of Incorporation* for basic information on how to use incorporation functions.

With partitioned or replicated files, a Predict file object can be implemented as several physical files. In this case, the database/file number of implemented files and corresponding Predict file objects need not be identical. Predict evaluates the Vista translation table to determine if an implemented file is documented with a Predict file object, even if the database/file numbers are not identical. Therefore, when incorporating partitioned files, only one implemented Adabas file can be incorporated using the Incorporate function. Other components of that file can then be connected with the incorporation function Connect. Both the functions Incorporate and Connect update the Vista translation table of the file.

## Selecting Adabas Files

Adabas File - Page 2 of 6 [IBM2 (7319)]

**ADABAS & NATURAL**

**Schema Incorporation**

**software AG**

Selection Criteria

Database number:

Physical file number from:

to:

Even if only FDT exists:

Adabas password:

Cancel <Back Next > Finish

Parameters	
Database number	The physical number of the Adabas database in which the Adabas files to be incorporated reside.
Physical file number from/to	Range of physical file numbers of the Adabas files to be incorporated.
Even if only FDT exists	<p>Y A file is to be incorporated even if it has been deleted with the Keep FDT option.</p> <p>N Only valid if the following conditions are met:</p> <ul style="list-style-type: none"> <li>■ Adabas Online Services must be available;</li> <li>■ The file to be incorporated must reside in a database of Adabas Version 6 or above;</li> <li>■ The file to be incorporated must not have been deleted with the Keep FDT option, since it must have implementation data.</li> </ul>
Update file attributes (connect option) / File attributes (incorporation option)	Provided on a later page of the schema incorporation wizard (after the physical object has been selected from the drop-down list box). Determines which attributes of the data dictionary file object are to be updated. Provided that

	Adabas Online Services are available and the file to be connected or incorporated resides in a database of Adabas Version 6 or above, the following values can be specified:	
	R	File attributes required when loading a file and any other file attributes specified in the data dictionary object.
	O	All file attributes including optional attributes not specified in the data dictionary object.
	N	No file attributes are to be updated.
Adabas password	The password required to access the file if the file is protected by Adabas security. The password is not converted to upper case.	

Files which meet the given selection criteria and are not already connected to a Predict file object are then provided in the drop-down list box on the next page of the schema incorporation wizard.

Information in the drop-down list box		
PFnr	Physical number of the file in the database.	
Adabas name	The name that was specified, either to the ADALOD utility or to Adabas Online Services, when the file was loaded into the database.	
Predict File ID	The ID of Predict file object of type A that may correspond to the Adabas files.	
Remark	Any of the following type-specific remarks or blank. In any of the following messages is displayed, the connect function cannot be executed. For type-independent remarks see <a href="#">Remarks which apply to all Object Types</a> .	
	Partitioned master missing	Documentation of Vista elements and entries in Vista translation table are different.
	ISN range overlapping	Predict file object with physical distribution type expanded is different.
	Different components	Predict file object with physical distribution type expanded is different. Vista elements and entries in Vista translation table are different.
	Different Vista type	Documentation of Vista elements and entries in Vista translation are different.
	Different local definition	Documentation of Vista elements and entries in Vista translation are different.

## Incorporation Functions for Adabas Files

Valid functions:

- Test
- Connect
- Incorporate

### Test

Compares Adabas files and corresponding Predict file objects.

### Connect

Connects Adabas file to the corresponding Predict file object and

- copies information from the Adabas FCB to the Predict file object,
- writes to the generation log of the object that a corresponding Adabas file existed when the connection was established.
- adds or updates Vista elements according to Vista translation table (if applicable).

Specify the following parameters in an additional page of the schema incorporation wizard.

Connect Options	
Update file attributes	See the parameter table in <a href="#">Selecting Adabas Files</a> for a description of this parameter.

### Incorporate

#### Scope of the Incorporation

- One Predict file object is created for each Adabas file. The Predict file objects are connected to the Adabas files.
- The new file object is linked to the database object via association "Contains FI".
- One field object is created for each field in the file. The ID assigned to each field object is created from the name of the field followed by either -GROUP for a group or -FIELD for a single field. A prefix and/or suffix specified with the parameters Field name prefix/suffix is/are added.
- If the file is physically coupled, a relationship object is created for each coupling definition.
- One Vista element (if applicable). If the file is accessible via several entries in the Vista translation table, the Vista element is created for the lowest logical file number.

- If Adabas Online Services are active and the file is implemented in a database of Adabas Version 6 or above, information on the implementation of the file (space allocation, for example) is incorporated.

Specify the following parameters in an additional page of the schema incorporation wizard. Parameter File ID is mandatory, all other parameters are optional.

Incorporation Options	
File ID	ID of the new file object (mandatory). The Adabas name of the file is displayed and can be accepted or overwritten.
File attributes	See the parameter table in <a href="#">Selecting Adabas Files</a> for a description of this parameter.
Field name prefix	Prefix of each field ID.
Field name suffix	Suffix of each field ID.
List incorporated code	Y   The values of all attributes incorporated for the file are to be listed.

## Keywords

---

The schema incorporation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
Database number	DBNR
Physical file number from	FROM-FNR
Physical file number to	TO-FNR
Even if only FDT exists	EVEN-ONLY-FDT
Update file attributes (connect option) / File attributes (incorporation option)	FILE-ATTR
Adabas password	PSW
Function	SUB-FUNCTION
File ID	FILE-ID
Field name prefix	PREFIX
Field name suffix	SUFFIX
List incorporated code	LIST



# **X** Common Parameters for All DB2 Types

---



# 12

## Common Parameters for All DB2 Types

---

The following parameters can be set in all DB2 object types.

Parameters	
SQL Server	The SQL server from which the object will be incorporated from.

---

# XI Incorporating DB2 Storagegroups

---



# 13

## Incorporating DB2 Storagegroups

---

▪ Prerequisites .....	104
▪ Selecting DB2 Storagegroups .....	105
▪ Incorporation Functions for DB2 Storagegroups .....	106
▪ Keywords .....	107

## Prerequisites

---

Incorporation of DB2 objects is subject to DB2 security mechanisms. See the section *DB2 and SQL/DS* in the *Predict and Other Systems* documentation.



**Note:** See [Concepts of Incorporation](#) for basic information on how to use incorporation functions.



## Selecting DB2 Storagegroups

Common parameters which are not included in the table below are described under *Common Parameters for All DB2 Types*.

Parameters	
Storagegroup name	DB2 storagegroup name. Asterisk notation is allowed.
Creator name	DB2 creator name. Asterisk notation is allowed.
Update storagespace attributes (connect option)	Provided on a later page of the schema incorporation wizard (after the physical object has been selected from the drop-down list box). Y - Attributes of the Predict storagespace that differ from the DB2 storagegroup are updated.

Information in the drop-down list box	
Storagegroup name	Name of the storagegroup.
Predict storagespace ID	ID of a Predict storagespace documenting the storagegroup.
Remark	A type-independent remark (see <i>Remarks which apply to all Object Types</i> or blank).

## Incorporation Functions for DB2 Storagegroups

---

Valid functions:

- [Test](#)
- [Connect](#)
- [Incorporate](#)



**Note:** Corresponding DB2 storagegroups and Predict objects of type storagespace are identified by the DB2 storagegroup name.

### Test

Compares the storagegroup and the Predict storagespace and reports differences between attributes.

### Connect

Connects the storagegroup to the existing Predict storagespace, and

- updates the Predict storagespace;
- writes to the generation log of the Predict object that a corresponding implemented storagegroup existed when the connection was established.

Specify the following parameters in an additional page of the schema incorporation wizard.

Connect Options	
Update storagespace attributes	See the parameter table in <a href="#">Selecting DB2 Storagegroups</a> for a description of this parameter.

### Incorporate

#### Scope of the Incorporation

- Creates a Predict storagespace with attributes documenting the storagegroup.
- Connects the storagegroup and the Predict storagespace.

Specify the following parameters in an additional page of the schema incorporation wizard. Storagespace ID is mandatory, List incorporated code is optional

Incorporation Options	
Storagespace ID	ID to be assigned to the new Predict storagespace.
List incorporated code	Y   The values of all attributes incorporated for the table or view are to be listed.

## Keywords

The schema incorporation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
Storagegroup name	STOGROUP
Creator name	CREATOR
Function	SUB-FUNCTION
Storagespace ID	STOSPACE-ID
List incorporated code	LIST
Update storagespace attributes	UPD-ST
SQL Server	SQL-SERVER

---

# XII

## Incorporating DB2 Databases

---



# 14

## Incorporating DB2 Databases

---

- Prerequisites ..... 112
- Selecting DB2 Databases ..... 113
- Incorporation Functions for DB2 Databases ..... 114
- Keywords ..... 115

## Prerequisites

---

Incorporation of DB2 objects is subject to DB2 security mechanisms. See the section *DB2 and SQL/DS* in the *Predict and Other Systems* documentation.



**Note:** See [Concepts of Incorporation](#) for basic information on how to use incorporation functions.



## Selecting DB2 Databases

Common parameters which are not included in the table below are described under *Common Parameters for All DB2 Types*.

Parameters	
DB2 database name	DB2 database name. Asterisk notation is allowed.
Creator name	Creator name. Asterisk notation is allowed.
Update DB attributes (connect option)	Provided on a later page of the schema incorporation wizard (after the physical object has been selected from the drop-down list box). Y - Attributes of the Predict database object are to be updated.

Databases which meet the selection criteria and are not already connected to a Predict database object are then provided in the drop-down list box on the next page of the schema incorporation wizard.

Information in the drop-down list box	
Creator	Creator of the database.
Database name	Name of the database
Predict database ID	ID of Predict database object for a DB2 database with this DB2 name.
Remark	Either the following type-specific remark or blank. For type-independent remarks see <a href="#">Remarks which apply to all Object Types</a> .
	SC missing   Default storagegroup is not documented. Incorporation is not possible.

## Incorporation Functions for DB2 Databases

Valid functions:

- [Test](#)
- [Connect](#)
- [Incorporate](#)

### Test

Compares the database and the Predict database object and reports differences between attributes and file lists.

### Connect

Connects the DB2 database to the corresponding Predict database object and

- copies current information from the DB2 catalog into the dictionary object;
- writes to the generation log of the Predict database object that a corresponding implemented database existed when the connection was established.

Specify the following parameters in an additional page of the schema incorporation wizard.

Connect Options	
Update DB attributes	See the parameter table in <a href="#">Selecting DB2 Databases</a> for a description of this parameter.

## Incorporate

### Scope of the Incorporation

- Creates a database object of type D with attributes documenting the database.
- The new database object is linked to the Predict object documenting the current virtual machine via association "Contains DA".
- Connects the database and the Predict database object.

Specify the following parameters in an additional page of the schema incorporation wizard. The parameter Database ID is mandatory, the other parameter is optional

Incorporation Options	
Database ID	ID of the new Predict database object. The creator and DB2 name of the database joined by a hyphen is displayed and can be accepted or overwritten.
List incorporated code	Y   The values of all attributes incorporated for the database are to be listed.

## Keywords

The schema incorporation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
DB2 database name	DB2NAME
Creator name	CREATOR
Function	SUB-FUNCTION
Database ID	DATABASE-ID
List incorporated code	LIST
Update DB attributes	UPD-DB
SQL Server	SQL-SERVER



# XIII

## Incorporating DB2 Tablespaces and SQL/DS

### DBspaces

---



# 15 Incorporating DB2 Tablespaces and SQL/DS DBspaces

---

- Prerequisites ..... 120
- Selecting Tablespaces ..... 121
- Incorporation Functions for Tablespaces ..... 122
- Keywords ..... 123

## Prerequisites

---

Incorporation of DB2 objects is subject to DB2 security mechanisms. See the section *DB2 and SQL/DS* in the *Predict and Other Systems* documentation.



**Note:** See [Concepts of Incorporation](#) for basic information on how to use incorporation functions.



## Selecting Tablespaces

Common parameters which are not included in the table below are described under *Common Parameters for All DB2 Types*.

Parameters	
Tablespace name	Name of DB2 tablespace. Asterisk notation is allowed.
Creator name	Name of tablespace creator. Asterisk notation is allowed.
DB2 database name	Name of DB2 database. Protected when processing SQL/DS DBspaces. Asterisk notation is allowed.
Update dataspace attributes (connect option)	Provided on a later page of the schema incorporation wizard (after the physical object has been selected from the drop-down list box). Y - Attributes of the Predict dataspace that differ from the DB2 tablespace are changed.

Information in the drop-down list box	
DB2 database	Name of the database containing the tablespace.
Tablespace name	Name of the tablespace.
Predict dataspace ID	ID of the Predict dataspace object documenting the DB2 tablespace.
Remark	Any of the following type-specific remarks or blank. For type-independent remarks see <i>Remarks which apply to all Object Types</i> .
	SC missing   Default storagegroup is not documented. Incorporation is not possible.
	DA missing   The DB2 database or storagegroup is not documented in Predict.
	ST missing   Incorporation is not possible.

## Incorporation Functions for Tablespaces

Valid functions:

- [Test](#)
- [Connect](#)
- [Display Masters](#)
- [Incorporate](#)

### Test

Compares DB2 tablespaces and corresponding Predict dataspace objects. Differences between file lists and tablespace/dataspace attributes are reported.

### Connect

Connects DB2 tablespaces to Predict dataspace object and

- copies current information from the DB2/SQL/DS catalog to the dictionary object
- writes to the generation log of the Predict dataspace object that a corresponding implemented database existed when the connection was established.

Specify the following parameters in an additional page of the schema incorporation wizard.

Connect Options	
Update dataspace attributes	See the parameter table in <i>Selecting Tablespaces</i> for a description of this parameter.

## Display Masters

The DB2 databases and storagegroups a tablespace is related to must be documented in Predict before the tablespace can be incorporated. The incorporation function M can be used to identify all objects that are missing. These can then be incorporated before the tablespace is incorporated.

## Incorporate

### Scope of the Incorporation

- A Predict dataspace object is created for each tablespace and connected to the tablespace.
- The dataspace object is linked to the corresponding database in Predict via association "Contains DC".

Specify the following parameters in an additional page of the schema incorporation wizard. Dataspace ID is mandatory, List incorporated code is optional.

Incorporation Options	
Dataspace ID	ID of the Predict dataspace object. The DB2 tablespace name is displayed and can be accepted or overwritten.
List incorporated code	Y   Display what has been incorporated as attributes of the dataspace object.

## Keywords

The schema incorporation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
Tablespace name	TBSPACE
Creator name	CREATOR
DB2 database name	DB2NAME
Function	SUB-FUNCTION
Dataspace ID	DSSPACE-ID
List incorporated code	LIST
Update dataspace attributes	UPD-DS
SQL Server	SQL-SERVER



# XIV

## Incorporating DB2 and SQL/DS Tables/Views

---



# 16

## Incorporating DB2 and SQL/DS Tables/Views

---

■ Prerequisites .....	128
■ Selecting DB2 and SQL/DS Tables/Views .....	129
■ Incorporation Functions for Tables/Views .....	130
■ Keywords .....	133

## Prerequisites

---

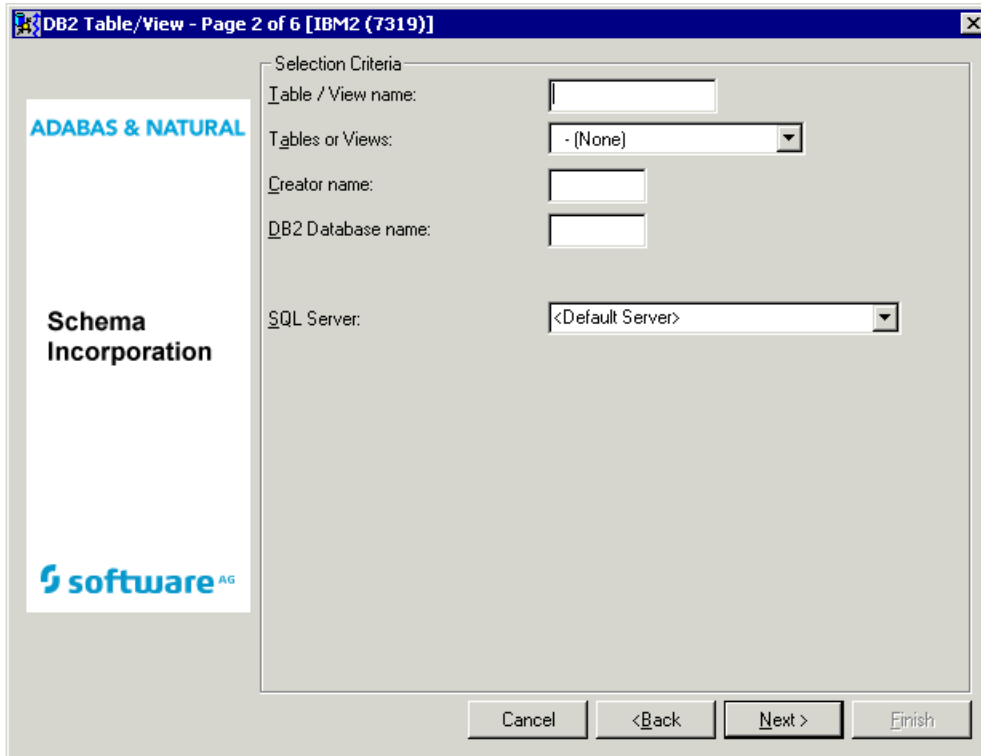
- For DB2 tables and views, a Predict object documenting the DB2 database containing the tables/views must exist.
- For SQL/DS tables and views, a corresponding database object in Predict is not required. See the section *DB2 and SQL/DS* in the *Predict and Other Systems* documentation.



**Note:** See [Concepts of Incorporation](#) for basic information on how to use incorporation functions.



## Selecting DB2 and SQL/DS Tables/Views



Common parameters which are not included in the table below are described under *Common Parameters for All DB2 Types*.

Parameters							
Table / view name	The name in DB2 of the tables and views from which the data dictionary objects are to be created. Asterisk notation is allowed.						
Tables or views	Limit the function to files of one type. Valid values: <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;">T</td> <td>DB2 tables</td> </tr> <tr> <td>V</td> <td>DB2 views</td> </tr> <tr> <td>blank</td> <td>All types</td> </tr> </table>	T	DB2 tables	V	DB2 views	blank	All types
T	DB2 tables						
V	DB2 views						
blank	All types						
Creator name	The DB2 "creator" of the tables and views from which the data dictionary objects are to be created. Asterisk notation is allowed.						
DB2 database name	The name in DB2 of the database or databases in which the specified tables and views reside. Asterisk notation is allowed. When incorporating Tables/Views from SQL/DS, this parameter is protected.						
Update file attributes (connect option)	Provided on a later page of the schema incorporation wizard (after the physical object has been selected from the drop-down list box). Y - Attributes of the Predict file object and, for files of type E, the subselect specification of the SQL						

Parameters	
	create statement are to be updated. The triggers linked to the file via association "Has TR" are also updated.

Files which meet the given selection criteria and are not already connected to a Predict file object are then provided in the drop-down list box on the next page of the schema incorporation wizard.

Information in the drop-down list box									
Creator	Creator in DB2.								
Table/view name	Name of the table/view.								
Database	Name of DB2 database(s) where the table/view is implemented.								
T	Type of the DB2 file: <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px;">T</td> <td>DB2 table</td> </tr> <tr> <td>V</td> <td>DB2 view</td> </tr> <tr> <td>M</td> <td>DB2 query table</td> </tr> </table>	T	DB2 table	V	DB2 view	M	DB2 query table		
T	DB2 table								
V	DB2 view								
M	DB2 query table								
Predict file ID	The ID of Predict file object for a DB2 table or view with this DB2 name.								
Remark	Any of the following type-specific remarks or blank. For type-independent remarks see <a href="#">Remarks which apply to all Object Types</a> . <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 200px;">DA missing</td> <td rowspan="3">The DB2 database, storagegroup or tablespace the table/view belongs to are not documented in Predict. Incorporation is not possible.</td> </tr> <tr> <td>ST missing</td> </tr> <tr> <td>DS missing</td> </tr> <tr> <td>Master missing</td> <td>The specified DB2 view cannot be incorporated as a data dictionary object until a dictionary object for a related DB2 table or view has been created.</td> </tr> <tr> <td>Invalid file-type</td> <td>The type of the specified DB2 table or view does not match the type of the corresponding Predict object: the file type of the data dictionary object is either not D (DB2 table) or not E (DB2 view).</td> </tr> </table>	DA missing	The DB2 database, storagegroup or tablespace the table/view belongs to are not documented in Predict. Incorporation is not possible.	ST missing	DS missing	Master missing	The specified DB2 view cannot be incorporated as a data dictionary object until a dictionary object for a related DB2 table or view has been created.	Invalid file-type	The type of the specified DB2 table or view does not match the type of the corresponding Predict object: the file type of the data dictionary object is either not D (DB2 table) or not E (DB2 view).
DA missing	The DB2 database, storagegroup or tablespace the table/view belongs to are not documented in Predict. Incorporation is not possible.								
ST missing									
DS missing									
Master missing	The specified DB2 view cannot be incorporated as a data dictionary object until a dictionary object for a related DB2 table or view has been created.								
Invalid file-type	The type of the specified DB2 table or view does not match the type of the corresponding Predict object: the file type of the data dictionary object is either not D (DB2 table) or not E (DB2 view).								

## Incorporation Functions for Tables/Views

Valid functions:

- [Test](#)
- [Connect](#)
- [Display Masters](#)
- [Incorporate](#)



**Note:** Corresponding DB2 tables/views and Predict file objects of type D and E are identified by DB2 table/view name.

## Test

Compares DB2 table/views and corresponding Predict file object. Differences between field lists and attributes of the tables/views are reported.

## Connect

Connects DB2 tables/views to Predict file object and

- copies current information from the DB2 catalog to the dictionary object;
- writes to the generation log of the Predict file object that a corresponding implemented table/view existed when the connection was established.

Specify the following parameters in an additional page of the schema incorporation wizard.

Connect Options	
Update file attributes	See the parameter table in <a href="#">Selecting DB2 and SQL/DS Tables/Views</a> for a description of this parameter.

## Display Masters

DB2 databases, storagegroups and tablespace and tables/views a table/view is related to must be documented in Predict before the table/view can be incorporated. This function finds all objects that are not documented. These can then be incorporated before the table/view is incorporated.

## Incorporate

### Scope of the Incorporation

- One Predict file object is created for each table or view. The ID of the new file object consists of the creator name and DB2 name of the table or view joined by a hyphen.
- One field object is created for each field in the table or view. The ID assigned to each new field object will be the name of the field in DB2. For each view that is incorporated, the appropriate parts of the subselect specification in the SQL statement CREATE will be copied to the file and field objects created.
- One field object is added to file *SAG-DISTINCT-TYPE* for each distinct type that is not already documented. The ID assigned to each new field object consists of the schema name and the type name concatenated by a hyphen. The connection between the standard fields of *SAG-DISTINCT-TYPE* and the table field is established if the field in DB2 is distinct type.
- One Predict file object with file type IV (Intermediate view) is created for each subselect clause in the from clause of a view definition. For each column in the select list of subselect a Predict field object is created. For each intermediate view that is created, the appropriate parts of the

subselect specification in the SQL statement CREATE will be copied to the file and field objects created.

- One Predict field object (type SP) for each composite index is created.
- One Predict field object (type SP) for each composite foreign key is created if there is not already a Predict field object for a composite index built from the same columns.
- One Predict relationship object is created for each foreign key.
- Labels defined for columns are documented as field headers.
- A trigger object is added for each trigger defined for a table or view. A link is established either to the file object via association "Has TR" or to the corresponding field objects in case of an update trigger.
- A verification is added and linked to the corresponding field for each check constraint that is defined for one column.
- Attributes of the corresponding file object are updated for each table check constraint (check expression for more than one column).

Specify the following parameters in an additional page of the schema incorporation wizard.

Incorporation Options		
Comments as abstract	Y	String constants assigned with COMMENT ON are to be incorporated as the abstract of the corresponding Predict object: <ul style="list-style-type: none"> <li>■ Comments assigned to the table or view are to be copied to the file object.</li> <li>■ Comments assigned to a field of the table or view are to be copied to the field object.</li> </ul>
List incorporated code	Y	The values of all attributes incorporated for the table or view are to be listed.
Document as table	Y	For views and materialized-query tables only.  This incorporation option can be used if you are not interested in the derivation of a view from the underlying base table or if the used query involves the usage of compiled SQL scalar or table functions.

## Keywords

The schema incorporation wizard uses the following keywords when showing the command that is to be processed:

Field	Keyword
Table/View name	TABLE
Tables or Views	TYPE
Creator name	CREATOR
DB2 Database name	DB2NAME
Function	SUB-FUNCTION
Incorporate comments	COMMENT
List incorporated code	LIST
Incorporate comments	ABSTRACT
Update file attributes	UPD-FILE
SQL Server	SQL-SERVER
Document as table	AS-TABLE



**XV**

**Comparison**

---



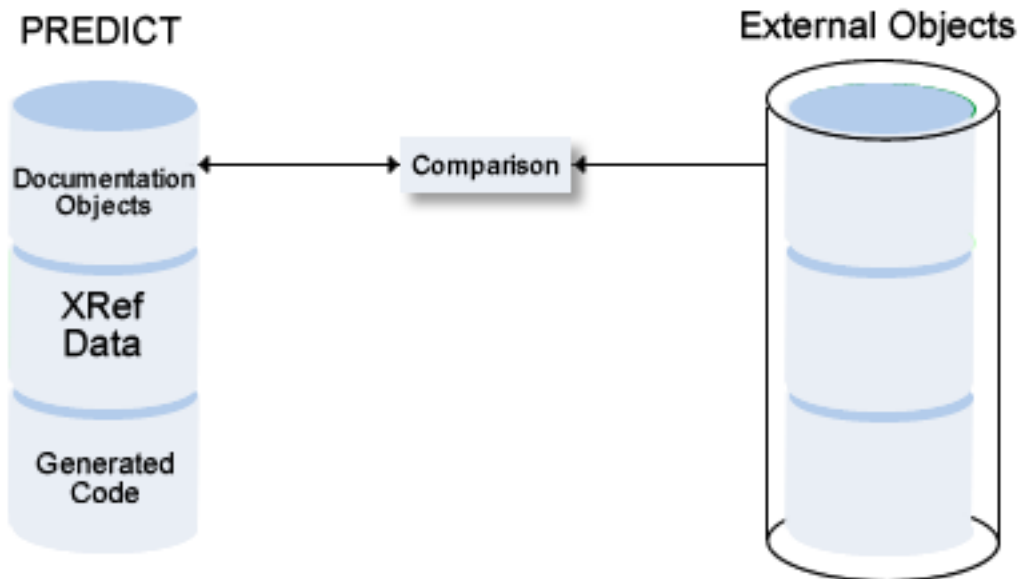


# 17 Comparison

---

Only available with Object Description.

Comparison functions help to keep Predict documentation and the actual implementation of an information processing system consistent. Comparison functions are needed because the external objects can be changed with utilities of the programming environment independent of Predict, even if protection options for external objects are activated. See function *Protection* in the section *General Defaults* in the *Predict Administration* documentation.



This section covers the following topics:

**Concepts of Comparison**

**Comparing Adabas Databases**

**Comparing Adabas Files**

**Comparing DDMs**

**Comparing DB2 Databases**

**Comparing DB2 Storagegroups**

**Comparing DB2 Tablespaces**

**Comparing DB2 Tables/Views**

# XVI

## Concepts of Comparison

---



# 18 Concepts of Comparison

---

- Functional Scope ..... 142
- Resetting the Modified after generation Flag ..... 143
- Format of Reports Created by Comparison Functions ..... 143
- Calling Comparison Functions ..... 143

## Functional Scope

---

The following Predict objects and external objects can be compared:

Predict Object	External Object
File	DDM Adabas File DB2 Tables and Views Oracle Tables and Views
Database	Adabas Database DB2 Database
Storagespace	DB2 Storagegroup
Dataspace	DB2 Tablespace SQL/DS Dataspace

If comparison functions find differences between an external object and its corresponding documentation object, either the implementation or the documentation can be changed to restore consistency.

Actions to restore consistency are either performed by the Predict comparison function or have to be executed manually. The following actions can be executed by Predict comparison functions:

- **Update**  
Update documentation objects according to external objects.
- **Reset**  
Reset the Modified after generation flag for documentation objects that do not differ from corresponding external objects. See [Resetting the Modified after generation Flag](#) below.

The following actions must be executed manually:

- **Generate**  
Generate a new external object from the documentation object.
- **Disconnect**  
Disconnect the external object and the documentation object.
- **Purge**  
Purge the external object or the documentation object.

## Resetting the Modified after generation Flag

---

The Modified after generation flag of Predict objects indicates if a Predict object has been changed after an external object was generated from it.

With Predict file objects of type A, separate Modified after generation flags are set for changes to file attributes and changes to the field list of the file.

If compare functions find that a Modified after generation flag is set for a Predict object that does not differ from the external object, the flag is reset.

Modified after generation flags are reset if the respective check options of the compare function are activated:

- When comparing Adabas files, the Modified after generation flags for file attributes and for field lists can be reset separately. Predict evaluates the check options File attributes and Fields.
- For all other object types, only one Modified after generation flag is set. All check options must be set to Y if the flag is to be reset.

## Format of Reports Created by Comparison Functions

---

When attributes of documentation and external objects differ, the attribute values of documentation objects are reported first and the information on external objects is indented, together with a brief description of the difference.

## Calling Comparison Functions

---

The following topics are covered below:

- [General Prerequisites for Comparison Functions](#)

- [Check, List, and Update Options](#)

## General Prerequisites for Comparison Functions

Predict objects and external objects to be compared must be connected. See the section [Handling of External and Documentation Objects](#) in this documentation.



**Note:** There is one exception: If Adabas Online Services are not installed, implemented Adabas files and Predict file objects of type A can be compared even if they are not connected.

A variety of Adabas attributes can only be compared if Adabas version 6 or above is used and Adabas Online Services are installed.

## Check, List, and Update Options

When comparing external objects and Predict objects, the following options determine the scope of the operation:

### ■ Check Options

Determine how attributes are to be compared. If check options are not set, compare functions will not reset the Modified after generation flag. See [Resetting the Modified after generation Flag](#).

### ■ List Options

Determine the scope of results of the comparison to be displayed.

### ■ Update Options

Determine which attributes of the Predict object are to be updated if differences were found.

The options available depend on the object type. All options are described in detail with the respective object type later in this documentation.



# XVII

## Comparing Adabas Databases

---



# 19 Comparing Adabas Databases

---

- Prerequisites ..... 148
- Parameters ..... 149

## Prerequisites

---

Adabas Online Services must be available if a database of Adabas Version 6 or above is to be compared.

If Natural Security function protection for Predict is activated, permission is required to compare Adabas databases.

See also *General Prerequisites for Comparison Functions*.

## Parameters

**Compare Adabas database, DBnr:188 belongs to DB188**

List Options

Database attributes: D - Differences only

File list: D - Differences only

Update Options

Database attributes:

File list:

OK Cancel Help

List Options	
Database attributes	How attributes of the Predict database object are to be displayed.
	Y All attributes
	N No attributes
	D Different: only database attributes whose values differ.
File list	How files contained in the file list of the Predict database object are to be displayed:
	Y All files
	N No files
	D Different: only files that are either not implemented in the Adabas database or are not linked to a Predict documentation object.

Update options	
Database attributes	Y The attributes of the Predict database object are to be updated.
File list	Y Files that are not implemented in the Adabas database are to be removed from the file list of the Predict database object.

---

# XVIII

## Comparing Adabas Files

---





# 20

## Comparing Adabas Files

---

- Prerequisites and Restrictions ..... 154
- Parameters ..... 155

## Prerequisites and Restrictions

---

- Adabas files and Predict files to be compared must be connected. There is one exception to this rule: if AOS is not installed, implemented Adabas files and Predict file objects can be compared even if they are not connected.
- If Natural Security function protection for Predict is activated, permission is required to compare Adabas files.

See also *General Prerequisites for Comparison Functions*.

## Parameters

### Check options

If the check options File attributes or Fields are not set, compare functions will not reset the Modified after generation flag. See [Resetting the Modified after generation Flag](#) for more information.

File attributes	Y	Attributes of the file are to be compared with the implementation of the file in the database (space allocation, for example).  Attributes not required when loading a file are compared only if they are specified in the Predict object.
-----------------	---	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	This setting is valid only when Adabas Online Services is available and the file that is to be compared resides in a database of Adabas Version 6 or above.	
Fields	Y	The field list of the file object is to be compared with the Adabas file definition table (FDT).
Even if only FDT exists	Y	A file is to be compared even if it has been deleted with the Keep FDT option.
	N	File is not compared. For this setting, three conditions must be met: <ul style="list-style-type: none"> <li>■ the Adabas Online Services selectable unit must be installed</li> <li>■ the file to be compared must reside in a database of Adabas Version 6 or above</li> <li>■ the file to be compared must not have been deleted with the Keep FDT option.</li> </ul>
Adabas password	Password required to access the file if the file is protected by Adabas security. Passwords are not converted to upper case.	
<b>List options</b>		
File attributes	How attributes of the Predict file object are to be displayed.	
	Y	All file attributes.
	N	No file attributes.
	D	Different: only file attributes whose Predict values differ from the implementation.
Fields	How fields linked to the Predict file object are to be displayed.	
	Y	All fields.
	N	No fields.
	D	Different: only fields whose Predict definitions differ from the FDT.
Members using differences	Determines the amount of information to be displayed on members using external objects that differ from the corresponding Predict objects. The following differences are possible: <ul style="list-style-type: none"> <li>■ file numbers of Predict file objects differ from the FDT or a userview of such a file.</li> <li>■ Predict field objects differ from the FDT.</li> </ul> Valid values:	
	Y	Display types and names of all members using files that differ in any of the above ways.
	N	Do not display members.
	C	Count the number of members per library.
	<b>Update options</b>	
Update options are effective only if parameter Members using differences is set to N.		

File attributes	How attributes of the Predict file object are to be updated. Provided that the Adabas Online Services selectable unit is available and the file that was compared resides in a database of Adabas Version 6 or above, one of the following can be specified:	
	R	File attributes required when loading a file and any other file attributes specified in the Predict object.
	N	No file attributes.
	O	File attributes not required when loading a file are updated only if they are specified in the Predict object.
Field sequences	Y	The field list is to be rearranged to match the sequence of the fields in the FDT.
Add fields	Y	Fields in the FDT that are missing in the field list are to be inserted in the field list.
Purge fields	Y	Fields in the field list that are missing from the FDT are to be removed from the field list.
Field attributes	How attributes of the fields linked to the Predict file object are to be changed to match the FDT.	
	A	All field attributes.
	N	No field attributes.
	I	Field attributes that can only be changed in the FDT by deleting and re-implementing the file. After fields have been updated, the function Edit elements of a file is executed automatically and a catalog is performed.
<p><b>Note:</b> We recommend that you set all or none of the field update parameters to Y. Otherwise confusing results may occur, especially if the Field sequence parameter is not set to Y. For example, a periodic group may be inserted out of sequence. A cursory inspection might then give the false impression that the group's member fields are on the wrong level.</p>		



# XIX

## Comparing DDMs

---





# 21

## Comparing DDMs

---

- Prerequisites ..... 162
- Parameters ..... 163

## Prerequisites

---

See also *General Prerequisites for Comparison Functions*.

If Natural Security is installed and the file (DDM) has been defined to the Security system, only administrators may compare the file. Also, if the file object in Natural Security has an Owner list, the DDM may be compared only by one of its owners. No countersignatures are necessary.

## Parameters

Check options	
If check options are not set, compare functions will not reset the Modified after generation flag. See <a href="#">Resetting the Modified after generation Flag</a> for more information.	
File attributes	Y   Attributes of the file are to be compared with the implementation of the file in the DDM and any differences in file number, database ID and file type-specific attributes are to be displayed.
Fields	Y   The field list of the file object is to be compared with the DDM.
Verifications	Y   The verifications of the fields are to be compared with the DDM to find any that <ul style="list-style-type: none"> <li>■ have incompatible formats,</li> <li>■ are not activated, or</li> <li>■ have been modified, disconnected or resequenced since the DDM was generated.</li> </ul>
Adabas version	The version of Adabas for which the DDM was generated. This determines the representation of integers and floating point numbers, hyperdescriptors and the size of the DBnr and Fnr in the DDM.

List options	
Fields	How fields linked to the Predict file object are to be displayed.
	Y All fields.
	N No fields.
	D Different: only fields whose Predict definitions differ from the DDM.
Verifications	How verifications of the fields are to be displayed.
	Y All verifications.
	N No verifications.
	D Different: only verifications that are not activated or have been modified, disconnected or resequenced since the DDM was generated.
Members using differences	Determines the amount of information to be displayed on members using external objects that differ from the corresponding Predict objects. The following differences are possible: <ul style="list-style-type: none"> <li>■ A verification has been modified, disconnected or resequenced since the DDM was generated.</li> <li>■ A Predict field definition differs from the DDM.</li> <li>■ The file number of a Predict file object differs from the DDM or a userview of such a file.</li> </ul> Valid values:
	Y Display types and names of all members using objects that differ in any of the above ways.
	N Do not display members.
	C Count the number of members per library.

**XX**

## Comparing DB2 Databases

---



# 22 Comparing DB2 Databases

---

- Prerequisites ..... 168
- Parameters ..... 169

## Prerequisites

---

DB2 must be installed. See the section *DB2 and SQL/DS* in the *Predict and Other Systems* documentation.

See also *General Prerequisites for Comparison Functions*.



## Parameters

List options	
Database attributes	How attributes of the Predict database object are to be displayed.
	Y All database attributes
	N No database attributes
	D Different: only database attributes whose Predict values differ from the implementation.
Dataspace list	How Predict dataspace objects linked to the Predict database object via association "Contains DC" are to be displayed.
	Y All dataspace.
	N No dataspace.
	D Different: only dataspace that are not implemented as DB2 tablespaces and DB2 tablespaces that are not documented with Predict dataspace are contained in the list.
File list	How files linked to the Predict database object are to be displayed.
	Y All files.
	N No files.

	D	Different: only files that are either not implemented in the DB2 database or not defined in Predict.
<b>Update options</b>		
Database attributes	Y	The attributes of the Predict database object are to be updated.
Unlink dataspace	Y	Dataspace not implemented as DB2 tablespaces are to be removed from the dataspace list of the Predict database object.
Unlink files	Y	Files that are not implemented in the DB2 database are to be removed from the file list of the Predict database object.  <b>Note:</b> Unlink files can only be set to Y if Unlink dataspace is also set to Y.

# XXI

## Comparing DB2 Storagegroups

---



# 23 Comparing DB2 Storagegroups

---

- Prerequisites ..... 174
- Parameters ..... 175

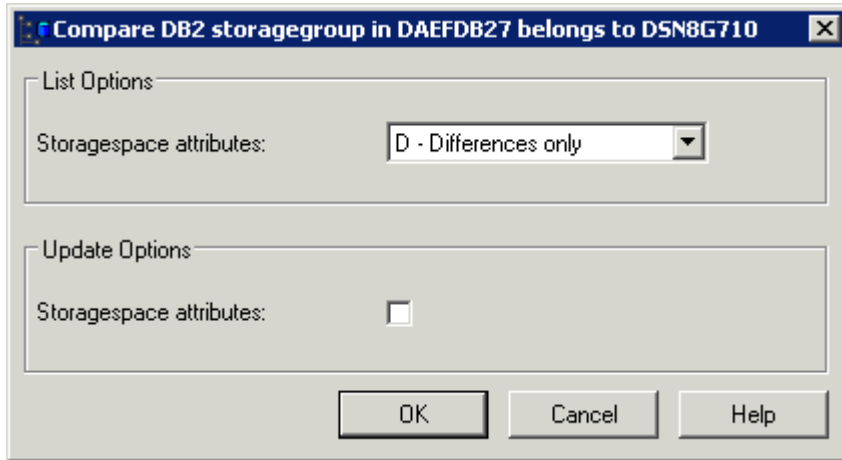
## Prerequisites

---

DB2 must be installed. See the section *DB2 and SQL/DS* in the *Predict and Other Systems* documentation.

See also [General Prerequisites for Comparison Functions](#).

## Parameters



List option		
Storage space attributes	How attributes of the Predict storage space object are to be displayed.	
	Y	All attributes are listed.
	N	No attributes are listed.
	D	Different: only attributes of storage space that are different from the storage group are listed.

Update option		
Storage space attributes	Y	Attributes of the Predict storage space that differ from the DB2 storage group are updated.





# XXII

## Comparing DB2 Tablespaces

---

---

# 24 Comparing DB2 Tablespaces

---

- Prerequisites ..... 180
- Parameters ..... 181

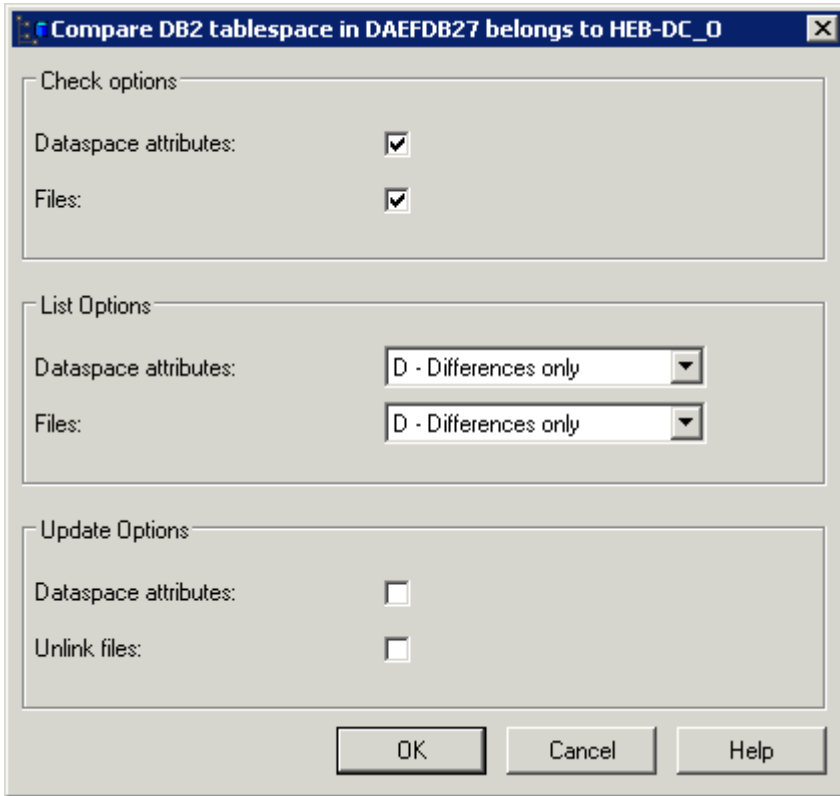
## Prerequisites

---

DB2 must be installed. See the section *DB2 and SQL/DS* in the *Predict and Other Systems* documentation.

See also [General Prerequisites for Comparison Functions](#).

## Parameters



Check options	
If check options are not set, compare functions will not reset the Modified after generation flag. See <a href="#">Resetting the Modified after generation Flag</a> for more information.	
Dataspace attributes	Y   Attributes of the Predict dataspace are compared with the DB2 tablespace.
Files	Y   The file list of the dataspace is to be compared with the DB2 tablespace.
List options	
Dataspace attributes	How dataspace attributes are reported.
	Y   All attributes of the Predict dataspace are reported.
	N   No attributes are reported.
	D   Different: only attributes of dataspace that are different from the tablespace are reported.
Files	How files linked to the dataspace via association "Contains FI" are to be reported:
	Y   All files

	N	No files
	D	Different: only files that are either not implemented in the DB2 tablespace or not defined in Predict.
<b>Update options</b>		
Dataspace attributes	Y	Attributes of the Predict dataspace that differ from the DB2 tablespace are changed.
Unlink files	Y	Files that are not implemented in the DB2 tablespace are removed from the file list of the Predict dataspace object and from the file list of the database object to which the dataspace belongs.

# XXIII

## Comparing DB2 Tables/Views

---





# 25 Comparing DB2 Tables/Views

---

- Prerequisites and Restrictions ..... 186
- Parameters ..... 187

## Prerequisites and Restrictions

---

DB2 must be installed. See the section *DB2 and SQL/DS* in the *Predict and Other Systems* documentation.

See also [General Prerequisites for Comparison Functions](#).

## Parameters

**Compare DB2 table/view in SAMPLE8 belongs to SMR-DB2\_SQL**

**Check options**

File attributes:

Fields:

**List Options**

Files: D - Differences only

Fields: D - Differences only

Members using differences: N - None

**Update Options**

File attributes:

Field sequences:

Add fields:

Purge fields:

Field attributes:

OK Cancel Help

Check options	
If check options are not set, compare functions will not reset the Modified after generation flag. See <a href="#">Resetting the Modified after generation Flag</a> for more information.	
File attributes	Y   Attributes of the file are to be compared with the implementation of the file in DB2.
Fields	Y   The field list of the file object is to be compared with the DB2 table or view.
List options	
Files	How attributes of the Predict file object are to be displayed.
	Y   All file attributes.
	N   No file attributes.

	D	Different: only file attributes whose Predict values differ from the implementation.
Fields	How fields linked to the Predict file object are to be displayed.	
	Y	All fields.
	N	No fields.
	D	Different: only fields whose Predict definitions differ from the DB2 table or view.
Members using differences	Determines the amount of information to be displayed on members using fields that have been compared and whose Predict definition differs from the DB2 table or view.	
	Y	Display the names and types of all members.
	N	Do not display members.
	C	Count the number of members per library.
<b>Update options</b>		
File attributes	Y	Attributes of the Predict file object and, for files of type E, the subselect specification of the SQL create statement are to be updated. The triggers linked to the file via association "Has TR" are also updated.
Field sequence	Y	The field list is to be rearranged to match the sequence of the fields in the DB2 table or view.
Add fields	Y	Fields in the DB2 table or view missing in the field list are to be inserted in the field list. Distinct types not already documented in standard file SAG-DISTINCT-TYPE will be added as fields. The connection between the standard field and the table field will be established.
Purge fields	Y	Fields that are missing in the DB2 table or view are to be removed from the field list.
Field attributes	Y	Attributes of the fields linked to the Predict file object are to be changed to match the DB2 table or view.  If a field has a connection to a standard field of SAG-DISTINCT-TYPE and the type definition differs from the DB2 catalog, the field attributes will be updated and the field will be marked non-standard.  For files of type E, the attributes that are to be updated include the subselect specification of the SQL create statement for each field. Comments of fields are not updated.

# XXIV

## Administration of External Objects

---

---

# 26 Administration of External Objects

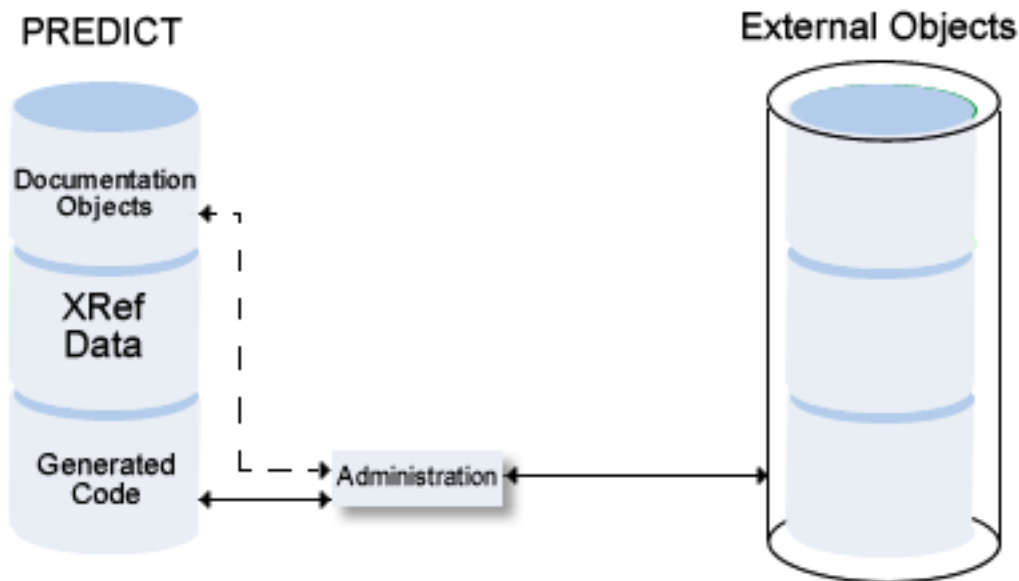
---

- Overview of Options ..... 192
- Administrating External Databases ..... 193
- Administrating External Dataspaces ..... 194
- Administrating External Files ..... 195
- Administrating External Programs ..... 196
- Administrating External Storagespaces ..... 197

Only available with Object Description.

Information stored in Predict objects can be used to generate external objects, and documentation objects can be incorporated from external objects. External objects generated from documentation objects and documentation objects incorporated from external objects are connected. Predict provides functions to administrate external objects connected to documentation objects. These functions are described in this section.

See the section [Handling of External and Documentation Objects](#) in this documentation for a more detailed description of how external and Predict objects are handled.



## Overview of Options

---

The following topics are covered below:

- [Overview of Administration Functions](#)



- [General Rules](#)

## Overview of Administration Functions

Administration functions are used to perform the following tasks.

- **Disconnect implementation - Disconnect command**

Disconnects external objects from their corresponding Predict objects. The implementation pointer from the documentation object to the external object is deleted, but the objects themselves remain intact.

- **Display implementation - Report command**

Displays external objects.

The following information is displayed:

- Generated code.

- **Purge implementation - Delete command**

Purges external objects. Any dependent objects are physically deleted. It is sometimes necessary to delete an external object before the connected Predict object can be deleted.

- **Refresh file - Refresh External command**

Deletes all records stored in Adabas files, DB2 tables/views or Oracle tables/views.

## General Rules

- External objects to be processed are identified by the Predict documentation object to which they are connected.
- If the Predict object was incorporated or connected with an Incorporation function, no generation protocol is stored in Predict.

## Administrating External Databases

---

## Functions

### Disconnect implementation - Disconnect command

Disconnects implemented databases connected to the Predict database object by deleting the implementation pointer of the documentation object.

### Display implementation - Report command

For DB2 databases: displays the generation log. For Adabas and IMS databases: only the implementation pointer is displayed.

### Purge implementation - Delete command

For DB2 databases: Deletes the database and all dependent tablespaces, tables and views. The database ID must be entered again to confirm deletion. If a table holds the last reference to a distinct type, the distinct type is also deleted.

The function is not available for Adabas and IMS databases.



**Caution:** Database and all dependent tablespaces, tables and views are *physically deleted* in DB2. Recovery from this action is not possible.

## Administrating External Dataspaces

---

## Functions

### Disconnect implementation - Disconnect command

Disconnects tablespaces and Predict dataspaces by deleting the implementation pointer of the Predict dataspace object.

### Display implementation - Report command

Displays the generation log.

**Purge implementation - Delete command**

Deletes the tablespace and all dependent tables and views. The dataspace ID must be entered again to confirm deletion. If a table holds the last reference to a distinct type, the distinct type is also deleted.

If dependent tables and views exist, they are listed. The deletion of these dependent objects has to be confirmed.



**Caution:** Tablespace and dependent tables and views are *physically deleted* in DB2. Recovery from this action is not possible.

## Administrating External Files

---

**Functions****Disconnect implementation - Disconnect command**

Disconnects implemented external object(s) from the Predict file object by deleting the implementation pointer of the file object.

**Display implementation - Report command**

Displays generated code.

**Purge implementation - Delete command**

Deletes generated code. If a DB2 table holds the last reference to a distinct type, the distinct type is also deleted.



**Caution:** File is physically deleted in Adabas, DB2, Oracle or Adabas SQL Server directory. All dependent objects are deleted, too.

Special rules apply when deleting Adabas files or DDMs:

**Adabas files**

The following rules apply:

- If the file to be deleted is in use, the Adabas option Stop user using file is used.
- If Predict/AOS Security is active, access rights are required to delete an Adabas file or to stop users using the file. See *Protecting Adabas Databases and Files* in the section *Protecting External Objects in Predict with Natural Security* in the *Predict Security* documentation.

## DDMs

- For DDMs defined in Natural Security, access rights to maintain the security definition of the file in Natural Security are required. See *Protecting DDMs* in the section *Protecting External Objects in Predict with Natural Security* in the *Predict Security* documentation.

### Refresh file - Refresh External command

Deletes all records stored in Adabas files, DB2 tables or Oracle tables. Data structure remains intact. Refresh operation must be confirmed.

Special rules apply when refreshing Adabas files:

- If the file to be refreshed is in use, the Adabas option Stop user using file can be used.



**Note:** If Predict/AOS Security is active, access rights are required to delete an Adabas file or to stop users using the file. See *Protecting Adabas Databases and Files* in the section *Protecting External Objects in Predict with Natural Security* in the *Predict Security* documentation.



**Caution:** This function deletes the entire contents of an Adabas file, DB2 table/view or Oracle table/view. Recovery from this action is not possible.

## Administrating External Programs

---

### Functions

#### Disconnect implementation - Disconnect command

Disconnects implemented external object(s) from the Predict program. The implementation pointer from the documentation object to the external object(s) is deleted, but the objects themselves remain intact.

#### Display implementation - Report command

Displays generated code.

**Purge implementation - Delete command**

Deletes the procedure/function in DB2.

## Administrating External Storagespaces

---

**Functions****Disconnect implementation - Disconnect command**

Disconnects implemented external object(s) from the Predict storagespace by deleting the implementation pointer of the documentation object.

**Display implementation - Report command**

Displays generated code.

**Purge implementation - Delete command**

Deletes generated code.

A storagespace which is used by any tablespace or index space cannot be deleted. In this case, a list of all the tablespaces, tables and views used by the storagespace is displayed.



**Caution:** Storagegroup is physically deleted in DB2.

---

# XXV

## Using the Preprocessor

---





# 27

## Using the Preprocessor

---

- Calling the Preprocessor ..... 202
- Prerequisites - Work Files ..... 203
- Reports - Print Files ..... 203

Using the preprocessor is a two-step process:

1. Write preprocessor statements to the source code of programs to be processed. See also [Preprocessor Statements](#).
2. Call the preprocessor for the programs to be processed. See [Calling the Preprocessor](#) below.

The preprocessor

- scans the source program for preprocessor statements and for certain other statements. See the section [Creation of XRef Data](#).
- generates copy code from Predict file objects and inserts this code into the source program (or inserts already generated code into the source program)
- writes corresponding XRef data.

To use the preprocessor in combination with Adabas Native SQL, it must be called before Adabas Native SQL. An asterisk must then be specified as Member name.

## Calling the Preprocessor

The preprocessor is called with the following command:

```
PREPROCESS <language> [<member name>][<library>] [<user-ID>]
```

If the generation default parameter Preprocessor force is set to Y, the member must be documented in Predict as a 3GL program with fully qualified implementation pointer (member name and library name).

Parameters	
language	Language in which the program to be processed is written: <ul style="list-style-type: none"> <li>■ COBOL</li> <li>■ PL/I</li> <li>■ BAL - Identifies 370-Assembler.</li> </ul>
member name	The member name must not be longer than eight characters. The member name can be supplied in three ways. If the member name is specified in more than one way, it is evaluated in the order below: <ul style="list-style-type: none"> <li>■ When calling the Preprocessor;</li> <li>■ Using the preprocessor statement PROGRAM;</li> <li>■ Using a normal source program statement <ul style="list-style-type: none"> <li>■ in COBOL, the PROGRAM-ID in the Identification division;</li> <li>■ in PL/I, the label preceding the first PROC or PROCEDURE statement</li> </ul> </li> </ul>

Parameters	
	<ul style="list-style-type: none"> <li>■ in BAL, the symbol in the name field of the START statement or the first CSECT statement.</li> </ul> <p>If the preprocessor is used in combination with Adabas Native SQL, an asterisk must be specified.</p>
library	If the parameter library is specified, a system of type G (3GL application) containing this library name in its implementation pointer must already have been defined. If no library is specified, the default libraries *SYSCOB*, *SYSBAL* or *SYSPLI* are used.
user-ID	The parameter user ID is only evaluated in batch mode. Online, the session user ID is evaluated.

## Prerequisites - Work Files

The PREPROCESS command requires the first three Natural workfiles (DD-names CMWKF01, CMWKF02 and CMWKF03; BS2000-link-names W01, W02 and W03).

- The first workfile is the input data set for the preprocessor.
- The second workfile is the output data set; it should be declared with a record length of 80.
- The third workfile is a temporary workfile; it should have record format F (fixed) or FB (fixed blocked) and the record size must be 91.



**Note:** For an explanation of the Preprocessor Condition Codes see the section *Predict in Batch Mode* in the *Predict Administration* documentation.

## Reports - Print Files

The preprocessor produces three types of reports:

- a report that the run was successful or unsuccessful
- a list of preprocessor commands found
- a list of errors.

In batch, the reports are written to the first three Natural reports (DD-names CMPRT01, CMPRT02 and CMPRT03; BS2000-link-names P01, P02 and P03).



# XXVI

## Preprocessor Statements

---



# 28 Preprocessor Statements

---

▪ Overview of Preprocessor Statements .....	208
▪ Format of Preprocessor Statements .....	209
▪ CALL .....	209
▪ COPY .....	210
▪ ENTRY .....	211
▪ FORMAT-BUFFER .....	212
▪ GENERATE .....	213
▪ PROGRAM .....	224

## Overview of Preprocessor Statements

---

### Statements to Include Copy Code Generated from Predict File Objects

The preprocessor statements `COPY`, `FORMAT-BUFFER` and `GENERATE` in COBOL, PL/I and Assembler programs instruct the preprocessor to include data definitions or a format buffer.

#### **COPY**

Instructs the preprocessor to insert copy code which has previously been generated by Predict. XRef data is written for the file and each field in the file.

#### **FORMAT-BUFFER**

Instructs the preprocessor to generate an Adabas format buffer and insert it at the position of the statement. XRef data is written for the file and each field in the file.

#### **GENERATE**

Instructs the preprocessor to generate a record buffer and optionally a format buffer and insert it at the position of the statement. XRef data is written for the file and each field in the file.

### Statements to Write XRef Data for 3GL Copy/Include Code or Function Calls

The preprocessor statements `ENTRY` and `CALL` - if included in Assembler programs - instruct the preprocessor to write XRef data for entry points or the call of external programs. Information to be written to XRef data is specified in parameters of the statement.

#### **CALL**

Specifies the name of a called external program or function that is to be stored in the active reference records. This command can only be used for Assembler programs.

#### **ENTRY**

Specifies the name of a program entry that is to be stored in the active reference records. This command can only be used for Assembler programs.

#### **PROGRAM**

Member ID used in XRef data. This statement is interpreted only if no member name is specified when the Preprocessor is called.



---

## Format of Preprocessor Statements

---

The following rules apply to preprocessor statements:

- Preprocessor statements start with `EXEC ADABAS` (there may be any number of blanks between the two words, but both must be coded on the same line).
- Preprocessor statements can be terminated in any of the following ways:
  - with the statement `END-EXEC`
  - in COBOL by a period (`.`)
  - in PL/I by a semicolon (`;`)
  - In BAL, the preprocessor terminates processing at the end of the current card unless a continuation character is punched in column 72.
- The preprocessor assumes standard statement layout in BAL, for example the Assembler statement `ICTL` is not valid.
- Preprocessor control statements are left in the source program as comments.

### Using Keyword and/or Positional Parameters

- Parameters of preprocessor statements can be specified in positional or keyword form. Both forms are described below.
- Keyword and positional parameters can be mixed. This allows the use of a keyword parameter as a starting point for subsequent positional parameters, as shown in the following example: For Assembler, the `INIT` parameter is the eighth parameter of the `EXEC ADABAS GENERATE` statement. `VALIDATION` is the next parameter in the list and can therefore be specified as a positional parameter directly behind the `INIT` parameter.

```
EXEC ADABAS GENERATE <file-name>,INIT=<init>,<validation>
```

---

## CALL

---

The name of a called external program or function can be specified for Assembler programs. The name is stored in the active reference records.

### Syntax with Positional Parameters

```
EXEC ADABAS CALL <function-name>  
                END-EXEC
```

### Syntax with Keyword Parameters

```
EXEC ADABAS CALL FUNCTION=<function-name>  
                END-EXEC
```

Parameters	
function-name	The name of a called external program or function that should be stored in the active reference records. Maximum length: 8 characters.

## COPY

---

Instructs the preprocessor to include copy code previously generated by Predict. If more than one copy code member has been generated for the appropriate file and language, the copy code name must be specified. XRef data is written for the file and each field in the file.

### Copying a Member from a File that has been Modified after Generation

This statement can also be used if the file has been modified after the corresponding member was generated. The system behavior depends on the Predict version with which the member was generated and on parameter Ignore changes. See table below.

### Syntax with Positional Parameters

```
EXEC ADABAS COPY <file-name><copycode-name> <ignore-changes>  
                END-EXEC
```

### Syntax with Keyword Parameters

```
EXEC ADABAS COPY FILE=<file-name>  
                MEM=<copycode-name>  
                IGNORE-CHANGES=<ignore-changes>  
                END-EXEC
```



**Note:** All parameters except <file-name> are optional.

Parameters		
file name	The ID of the Predict file object from which the copy code was generated. See <a href="#">GENERATE</a> for a list of file types that can be used for generating copy code.	
member name	The member name which was supplied during copy code generation. Maximum length: 8 characters.	
ignore changes	This parameter determines whether a member connected to a file that has been modified after generation is copied.	
	N	Default. A member connected to a file that has been modified after generation is not copied. The Preprocessor issues an error message and terminates with condition code 107.
	Y	A member connected to a modified file is copied - irrespective of whether the member contains usage information. With this option, the preprocessor writes the XRef data on the basis of the current documentation of the file in Predict. A warning is given if the file object has been modified after generation, since using this option may result in inconsistent XRef data.
	X	A member of a modified file is only copied if it was generated with Predict version 3.3 or above. With this version, additional information on the usage of fields in the file is available. This information is used by the Preprocessor when writing XRef data. If the file has been modified since generation and the usage information is not available because the member was generated with an earlier version of Predict, the Preprocessor issues an error message and terminates with condition code 107.

## ENTRY

Declares the name of an entry point in Assembler programs. The name is stored in the active reference records.

### Syntax with Positional Parameters

```
EXEC ADABAS ENTRY <entry-name>
      END-EXEC
```

## Syntax with Keyword Parameters

```
EXEC ADABAS ENTRY FUNCTION=<entry-name>
      END-EXEC
```

Parameters	
entry-name	The name of an entry point to be stored in the XRef data. Maximum length: 8 characters.

## FORMAT-BUFFER

The `FORMAT-BUFFER` statement instructs the preprocessor to generate an Adabas format buffer and insert it into the 3GL member at the position of the statement. XRef data is written for the file and each field in the file.

### Syntax with Positional Parameters

```
EXEC ADABAS FORMAT-BUFFER
      <file-name><format-buffer-name>
      <sync><offset><dcl>
      <adabas-version><buffer-format>
      <literal-delimiter>
      END-EXEC
```

### Syntax with Keyword Parameters

```
EXEC ADABAS FORMAT-BUFFER
      FILE=<file-name>
      FORMAT-BUFFER-NAME=<format-buffer-name>
      SYNC=<sync>
      OFFSET=<offset>
      DCL=<dcl>
      ADA-VER=<adabas-version>
      FORMAT=<buffer-format>
      DELIMITER=<literal-delimiter>
      END-EXEC
```



**Note:** All parameters except `<file-name>` are optional.

Parameters	
file-name	Specifies the ID of the file object in Predict. This parameter is mandatory. The file can be one of the following types: <ul style="list-style-type: none"> <li>■ Adabas file (type A), where parameter Adabas SQL usage may not be set to Y.</li> <li>■ Adabas userview (type U)</li> </ul>
format-buffer-name	The name to be given to the format buffer in the target program. In COBOL, this may be up to 30 characters long; in PL/I, up to 31; and in BAL, up to 8 characters. The default value is the same as in the respective generation function.
sync	Y All appropriate fields will be aligned.
	N No fields will be aligned.
	S Fields will be aligned only if the corresponding Predict object has the 3GL specification Synchronized=S.
offset	L,Y,P The total length of the code will be included in the format buffer.
	V A constant will be generated for the format buffer length.
dcl	Only applies to PL/I include code.
	Y The generated code will be preceded by a declare statement DCL and will end with a semicolon instead of a comma.
adabas-version	The version of Adabas for which the copy code of the Adabas files and userviews is to be generated. See list of possible values in the section <a href="#">Adabas Version</a> .
buffer-format	Y Normal format for format buffer. Adabas groups, standard formats and lengths are used whenever possible. The resulting format buffers are then as short as possible.
	F Full format for format buffer. The format buffer will include field length and format.
literal-delimiter	Only valid for COBOL copy code.
	S single quotes
	D double quotes

## GENERATE

The statement `GENERATE` instructs the preprocessor to generate copy code from Predict file objects and insert it into the 3GL member at the position of the statement. XRef data is written for the file and each field in the file.

Generation can be performed for Assembler, COBOL and PL/I. See table of valid file types and languages below.

The generated copy code will be written to the source area and then copied into the source program.

Code	File Type	Assembler	COBOL	PL/I
A	Adabas file	Y	Y	Y
B	Adabas SQL view	Y	Y	Y
BT, BV	Adabas D table/view		Y	
D, E	DB2 table/view	Y	Y	Y
F	rdb file	Y	Y	Y
J	IMS segment layout	Y	Y	Y
JT, JV	Ingres table/view		Y	Y
L	logical VSAM file	Y	Y	Y
M	ISAM file	Y	Y	Y
O	other file	Y	Y	Y
OT, OV	Oracle table/view		Y	Y
S	sequential file	Y	Y	Y
T	rms file	Y	Y	Y
U	Adabas user view	Y	Y	Y
V	VSAM file	Y	Y	Y
X	General SQL file	Y	Y	Y
XT, XV	Informix table/view		Y	
YT, YV	Sybase table/view		Y	
1	LEASY	Y	Y	Y
2	ISAM BS2000	Y	Y	Y

## Generating Assembler Copy Code

### Syntax with Positional Parameters

```
EXEC ADABAS GENERATE <filename><prefix><suffix><dsect>
                    <dc-ds><nr-comments><offset><init>
                    <validation><truncation><dsect-name><align>
                    <equ><adabas-version><generate-format-buffer>
                    <format-buffer-name><counter-length>
END-EXEC
```

## Syntax with Keyword Parameters

```
EXEC ADABAS GENERATE FILE=<file-name>
    PREFIX=<prefix>
    SUFFIX=<suffix>
    DSECT=<dsect>
    DC-DS=<dc-ds>
    NR-COMMENTS=<nr-comments>
    OFFSET=<offset>
    INIT=<init>
    VALIDATION=<validation>
    TRUNCATION=<truncation>
    RECORD-BUFFER-NAME=<dsect-name>
    SYNC=<align>
    EQU=<equ>
    ADA-VER=<adabas-version>
    FORMAT-BUFFER=<generate-format-buffer>
    FORMAT-BUFFER-NAME=<format-buffer-name>
    COUNTER-LENGTH=<counter-length>
END-EXEC
```



**Note:** All parameters except <file-name> are optional.

Details of the parameters are given in the following table and also in the section *Generation of External Objects* in this documentation. If a parameter is omitted, the default value defined in Predict by the DDA will be used. See the section *Defaults* in the *Predict Administration* documentation.

If the statement is entered using positional parameters and no prefix or no suffix is required, an asterisk (\*) should be substituted for the parameter <prefix> or <suffix>.

Parameters	
file-name	ID of the Predict file object from which the Assembler copy code is to be generated. This parameter is mandatory. See table of valid file types in the description of parameter <b>GENERATE</b> .
prefix	A prefix for the field names. Maximum length: 8 characters.
suffix	A suffix for the field names. Maximum length: 8 characters.
dsect	Y   The copy code will be generated as an ASSEMBLER DSECT (dummy section). The DSECT will have the name specified by the <dsect-name> parameter, or the file ID if no <dsect-name> parameter is supplied.
dc-ds	DC   Assembler DC (define constant) instructions will be generated.
	DS   Assembler DS (define storage) instructions will be generated.
nr-comments	Specifies the number of abstract lines per field (0-16) which will be included in the generated code.
offset	Y   Include the offset of each item in the record buffer structure (relative to the beginning of the structure) in decimal and hexadecimal formats as a comment. The total length of each buffer is also included.

Parameters		
	P	Include the absolute position (offset+1) as a comment.
	L	Include the total lengths of the record buffer and the format buffer as a comment.
	V	Only allowed if parameters As DSECT=N and With DC or DS=DC. The file number and the calculated lengths of the record buffer and the format buffer are to be generated as constants in the copy code. The name of the file number constant is the record buffer name with N as prefix. The name of each length constant is the appropriate buffer name with L as prefix. Each name is prefixed, suffixed, validated and truncated in the same way as any other field name.
	N	No offset.
init	This option takes effect only when With DC or DS=DC and As DSECT is set to N.	
	N	No initialization.
	Y	Statements are generated to initialize the structure with the value specified for Init value in the corresponding field object in Predict. Fields with no value for Init value are initialized with zeros or blanks.  Aligned 8-digit fields with format B or I are not initialized.  In a PE group with the 3GL specification Gr.structur set to blank, only the first occurrence of each field is initialized.
validation	Determines how invalid characters in a field name are handled.	
	blank	Invalid characters will result in an error message but will not be deleted.
	rep.char	Invalid characters will be replaced by this character. Valid values: letters A-Z, digits 0-9, \$, § or #.
	*	Invalid characters will be deleted.
truncation	Specifies which characters are deleted if a generated field name is longer than 8 characters:	
	L	truncate from the left
	R	truncate from the right
	M	truncate from the middle
dsect-name	Specifies the name of the record buffer in the generated structure. The effect of this parameter depends on parameter dsect.	
align	Y	All appropriate fields will be aligned.
	N	No fields will be aligned.
	S	Fields will be aligned only if the corresponding Predict object has the 3GL specification Synchronized= S.
equ	Y	EQU statements are to be generated for fields of length 1 whose format is not P and comment lines are to be generated for other fields, using any condition



Parameters							
	<p>names defined as attributes of the Predict field objects. These names are prefixed, suffixed, validated and truncated in the same way as field names.</p> <p>For fields of format L where no condition name was specified, a condition name is generated by concatenating the field name "Example" to the prefix N. In this case the following statement is generated: NEXAMPLE EQU X'00'.</p>						
adabas-version	The version of Adabas for which the copy code of the Adabas files and userviews is to be generated. See list of possible values in the section <i>Adabas Version</i> .						
generate-format-buffer	<p>Format buffer generation for Assembler copy code is only allowed if parameters As DSECT=N and With DC or DS=DC. The contents of the format buffer will correspond exactly to the contents of the record buffer. Only valid for files of type A (with parameter Adabas SQL usage set to N) or for files of type U.</p> <p>Valid values:</p> <table border="1"> <tbody> <tr> <td>Y</td> <td>Adabas format buffer is to be generated. Adabas groups, standard formats and lengths are used whenever possible. The resulting format buffers are then as short as possible.</td> </tr> <tr> <td>F</td> <td>Full format buffer is to be generated. Length and format of Adabas fields are included.</td> </tr> <tr> <td>N</td> <td>No format buffer is to be generated.</td> </tr> </tbody> </table> <p><b>Note:</b> If you are generating for a WANG environment, set this parameter to F or N.</p>	Y	Adabas format buffer is to be generated. Adabas groups, standard formats and lengths are used whenever possible. The resulting format buffers are then as short as possible.	F	Full format buffer is to be generated. Length and format of Adabas fields are included.	N	No format buffer is to be generated.
Y	Adabas format buffer is to be generated. Adabas groups, standard formats and lengths are used whenever possible. The resulting format buffers are then as short as possible.						
F	Full format buffer is to be generated. Length and format of Adabas fields are included.						
N	No format buffer is to be generated.						
format-buffer-name	Specifies the label (name) of the format buffer in the generated structure. If omitted, the file ID prefixed by F is used.						
counter-length	Length of additional counter fields: Valid values: 1, 2.						

## Generating COBOL Copy Code

### Syntax with Positional Parameters

```
EXEC ADABAS GENERATE <file-name><prefix><suffix> <start-level>
                    <level-increment><shift-number><nr-comments>
                    <offset><init><validation><truncation>
                    <record-buffer-name><cond-name><sync>
                    <indexed><depending>
                    <adabas-version><generate-format-buffer>
                    <format-buffer-name>
                    <check-name>
                    <literal-delimiter>
                    <decimal-char>
                    <redefine-name>
END-EXEC
```

## Syntax with Keyword Parameters

```
EXEC ADABAS GENERATE FILE=<file-name>
                        PREFIX=<prefix>
                        SUFFIX=<suffix>
                        START-LEVEL=<start-level>
                        LEVEL-INCREMENT=<level-increment>
                        SHIFT-NUMBER=<shift-number>
                        NR-COMMENTS=<nr-comments>
                        OFFSET=<offset>
                        INIT=<init>
                        VALIDATION=<validation>
                        TRUNCATION=<truncation>
                        RECORD-BUFFER-NAME=<record-buffer-name>
                        COND-NAME=<cond-name>
                        SYNC=<sync>
                        INDEXED=<indexed>
                        DEPENDING=<depending>
                        ADA-VER=<adabas-version>
                        FORMAT-BUFFER=<generate-format-buffer>
                        FORMAT-BUFFER-NAME=<format-buffer-name>
                        CHECK-NAME=<check-name>
                        DELIMITER=<literal-delimiter>
                        DEC-CHAR=<decimal-char>
                        REDEFINE-NAME=<redefine-name>
                        END-EXEC
```



**Note:** All parameters except <file-name> are optional.

Details of the parameters are given in the following table and also in the section [Generation of External Objects](#) in this documentation. If a parameter is omitted, the default value defined in Predict by the DDA will be used. See the section [Defaults](#) in the *Predict Administration* documentation.

If the statement is entered using positional parameters and no prefix or no suffix is required, an asterisk (\*) should be substituted for the parameter <prefix> or <suffix>.

Parameters	
file-name	ID of the Predict file object from which the COBOL copy code is to be generated. This parameter is mandatory. See table of valid file types in the description of parameter <a href="#">GENERATE</a> .
prefix	A prefix for the field names. Maximum length: 16 characters.
suffix	A suffix for the field names. Maximum length: 16 characters.
start-level	Specifies the starting level of the generated record buffer. Valid values are in the range 1 - 40.
level-increment	Specifies the level-increment. Valid values are in the range 1 - 40.
shift-number	The number of positions to be shifted right when a level number which is higher than the current level number is encountered. Valid values are in the range 0 - 9.

Parameters		
nr-comments	Specifies the number of abstract lines per field that will be included in the generated code. Valid values are in the range 0 - 16.	
offset	Y The offset of each item in the record buffer structure (relative to the beginning of the structure) in decimal and hexadecimal formats is to be included as a comment. The total length of each buffer is also included.	
	P As above, but the absolute position (offset+1) is included as a comment.	
	L The total lengths of the record buffer and the format buffer are to be included as a comment.	
	V The file number and the calculated lengths of the record buffer and the format buffer are to be generated as constants in the copy code. The name of the file number constant is the record buffer name prefixed by N-. The name of each length constant is the appropriate buffer name prefixed by L-. Each name is prefixed, suffixed, validated and truncated in the same way as any other field name.	
	N No offset.	
init	Y The fields will be initialized wherever possible using a COBOL VALUE clause. Any fields with INIT VALUES in their Predict objects will be initialized with those values; other fields will be initialized with low values (zeros or spaces).	
	S Only fields with INIT VALUES in the corresponding Predict object will be initialized.	
	N No initialization.	
validation	Determines how invalid characters in a field name are handled.	
	blank	Invalid characters will result in an error message but will not be deleted.
	rep.char.	Invalid characters will be replaced by this character. Valid values: letters A-Z, digits 0-9 or hyphen.
	*	Invalid characters will be deleted.
truncation	Specifies which characters are deleted if a generated field name is longer than 30 characters:	
	L	Truncate from the left.
	R	Truncate from the right.
	M	Truncate from the middle.
record-buffer-name	Specifies the name of the record buffer in the generated structure. If omitted, the file ID is used.	
cond-name	Y	Any condition names defined in the Predict field objects are to be generated on level 88, provided that the respective field objects have one of the following formats: <ul style="list-style-type: none"> <li>■ A - All lengths</li> <li>■ N or P - Less than 19 digits</li> </ul>

Parameters	
	<ul style="list-style-type: none"> <li>■ I or B - 2, 4 or 8 digits</li> <li>■ L - A FALSE-condition will always be generated. The Condition name is then generated by concatenating the field name to the prefix N- (if not specified explicitly).</li> </ul> <p>These names are prefixed, suffixed, validated and truncated in the same way as field names.</p>
sync	Y   All appropriate fields will be aligned.
	N   No fields will be aligned.
	S   Fields will be aligned only if the corresponding Predict object has the 3GL specification Synchronized = S.
indexed	Y   The COBOL clause INDEXED BY will be generated for all repetitive fields (MU/MC and PE/PC).
	S   This clause will be generated only for repetitive fields which have INDEXED BY NAMEs in their Predict objects.
	N   This clause will not be generated for any field.
depending	Y   COBOL attribute OCCURS DEPENDING ON is generated for a field or field group in a file if it has type PE or MU and DEPENDING ON NAME is specified for this field. These names are prefixed, suffixed, validated and truncated in the same way as field names.
	<p><b>Note:</b></p> <ol style="list-style-type: none"> <li>1. This option is not allowed for files of type A or U.</li> <li>2. This option is ignored when using a WANG COBOL compiler.</li> </ol>
adabas-version	The version of Adabas for which the copy code of the Adabas files and userviews is to be generated. See table of valid values in the section <a href="#">Adabas Version</a> .
generate-format-buffer	The contents of the format buffer will correspond exactly to the contents of the record buffer. Only valid for files of type A (with parameter Adabas SQL usage set to N) or for files of type U.
	Valid values:
	Y   Adabas format buffer is to be generated. Adabas groups, standard formats and lengths are used whenever possible. The resulting format buffers are then as short as possible.
	F   Full format buffer is to be generated. Length and format of Adabas fields are included.
	N   No format buffer is to be generated.
<b>Note:</b> If you are generating for a WANG environment, set this parameter to F or N.	

Parameters		
format-buffer-name	Specifies the name of the format buffer in the generated structure. If omitted, the file ID prefixed by FORMAT-BUFFER- is used.	
check-name	A COBOL field names are checked for uniqueness throughout the whole structure.	
	Y Structure levels are included in the validation check of the field names: if two fields have the same name, they must be separated by at least one field with a different name and a lower-level number.	
	N No check for duplicate field names is performed.	
literal-delimiter	S Single quotes.	
	D Double quotes.	
decimal-character	P Point.	
	C Comma.	
redefine-name	Determines how COBOL field names for Predict fields of type RE are generated:	
	F	The string FILLER is used as redefinition name.
	S	The suffix REGR is added to the Predict field ID. If a field is redefined more than once, the suffix will have the form REGRn, where n is an integer incremented by 1 for each field of type RE.

## Generating PL/I Include Code

### Syntax with Positional Parameters

```
EXEC ADABAS GENERATE <file-name><prefix><suffix>
                    <start-level><level-increment><shift-number>
                    <nr-comments><offset><init><struct-as-char>
                    <static><validation><truncation>
                    <record-buffer-name><align><dc1>
                    <adabas-version><generate-format-buffer>
                    <format-buffer-name><check-name>
                    <numeric sign><position of sign>
END-EXEC
```

### Syntax with Keyword Parameters

```
EXEC ADABAS GENERATE FILE=<file-name>
                    PREFIX=<prefix>
                    SUFFIX=<suffix>
                    START-LEVEL=<start-level>
                    LEVEL-INCREMENT=<level-increment>
                    SHIFT-NUMBER=<shift-number>
                    NR-COMMENTS=<nr-comments>
                    OFFSET=<offset>
                    INIT=<init>
```

```

STRUCTURE=<struct-as-char>
STATIC=<static>
VALIDATION=<validation>
TRUNCATION=<truncation>
RECORD-BUFFER-NAME=<record-buffer-name>
SYNC=<align>
DCL=<dcl>
ADA-VER=<adabas-version>
FORMAT-BUFFER=<generate-format-buffer>
FORMAT-BUFFER-NAME=<format-buffer-name>
CHECK-NAME=<check-name>
NUM-SIGN=<numeric sign>
POS-SIGN=<position of sign>
END-EXEC
    
```

 **Note:** All parameters except <file-name> are optional.

Details of the parameters are given in the following table and also in the section *Generation of External Objects* in this documentation. If a parameter is omitted, the default value defined in Predict by the DDA will be used. See the section *Defaults* in the *Predict Administration* documentation.

If the statement is entered using positional parameters and no prefix or no suffix is required, an asterisk (\*) should be substituted for the parameter <prefix> or <suffix>.

Parameters	
file-name	ID of the Predict file object from which the PL/I include code is to be generated. This parameter is mandatory. See table of valid file types in the description of parameter <b>GENERATE</b> .
prefix	Specifies a prefix for the field names. Maximum length: 16 characters.
suffix	Specifies a suffix for the field names. Maximum length: 16 characters.
start-level	Specifies the starting level of the generated record buffer. Valid values are in the range 1 - 40.
level-increment	Specifies the level-increment. Valid values are in the range 1 - 40.
shift-number	The number of positions to be shifted right when a level number which is higher than the current level number is encountered. Valid values are in the range 0 - 9.
nr-comments	Specifies the number of abstract lines per field that will be included in the generated code. Valid values are in the range 0 - 16.
offset	Y The offset of each item in the record buffer structure (relative to the beginning of the structure) in decimal and hexadecimal formats is to be included as a comment. The total length of each buffer is also included.
	P As above, but the absolute position (offset+1) is included as a comment.
	L The total lengths of the record buffer and the format buffer are to be included as a comment.
	V The file number and the calculated lengths of the record buffer and the format buffer are to be generated as constants in the include code. The name of the file

Parameters		
	number constant will be the record buffer name prefixed by N_. The name of each length constant will be the appropriate buffer name prefixed by L_. Each name is prefixed, suffixed, validated and truncated in the same way as any other field name.	
	N No offset.	
init	Y The fields will be initialized wherever possible. Any fields with INIT VALUEs in their Predict objects will be initialized with those values; other fields will be initialized with low values (zeros or spaces).	
	S Only fields with INIT VALUEs in the corresponding Predict object will be initialized.	
	N No initialization.	
struct-as-char	Y The entire generated structure will be declared at the end of the record buffer as a single character-string.	
static	Y The structure will be declared with the attribute STATIC.	
validation	Determines how invalid characters in a field name are handled.	
	blank	Invalid characters will result in an error message but will not be deleted.
	rep.char	Invalid characters will be replaced by this character. Valid values: letters A-Z, digits 0-9, \$, # or _ (underscore).
	*	Invalid characters will be deleted.
truncation	Specifies which characters are deleted if a generated field name is longer than 31 characters:	
	L	Truncate from the left.
	R	Truncate from the right.
	M	Truncate from the middle.
record-buffer-name	Specifies the name of the record buffer in the generated structure. If omitted, the file ID is used.	
align	Y	All appropriate fields will be aligned.
	N	No fields will be aligned.
	S	Fields will be aligned only if the corresponding Predict object has the 3GL specification Synchronized= S.
	<b>Note:</b> This parameter only takes effect with fields that have the PL/I attribute FIXED BIN or FLOAT DEC.	
dcl	Y	The generated code will be preceded by a declare statement DCL and will end with a semicolon instead of a comma.
adabas-version	The version of Adabas for which the include code of the Adabas files and userviews is to be generated. See table of valid values in the section <a href="#">Adabas Version</a> .	

Parameters		
generate-format-buffer	The contents of the format buffer will correspond exactly to the contents of the record buffer. Only valid for files of type A (with parameter Adabas SQL usage set to N) or for files of type U.	
	Valid values:	
	Y	Adabas format buffer is to be generated. Adabas groups, standard formats and lengths are used whenever possible. The resulting format buffers are then as short as possible.
	F	Full format buffer is to be generated. Length and format of Adabas fields are included.
	N	No format buffer is to be generated.
<b>Note:</b> If you are generating for a WANG environment, set this parameter to F or N.		
format-buffer-name	Specifies the name of the format buffer in the generated structure. If omitted, the file ID prefixed by FORBUF_ is used.	
check-name	A	Field names are checked for uniqueness throughout the whole structure.
	Y	Structure levels are included in the validation check of the field names: if two fields have the same name, they must be separated by at least one field with a different name and a lower level number.
	N	No check for duplicate field names is performed.
numeric sign	Specifies which of the PL/I picture characters T, I or R is to be used for the representation of numeric values of format packed with sign or unpacked with sign.	
position of sign	Defines the position of the sign in a numeric field:	
	L	Left.
	R	Right.

## PROGRAM

The member name can be passed to the preprocessor with the PROGRAM statement:



## Syntax with Positional Parameters

```
EXEC ADABAS PROGRAM <member-name><library>
END-EXEC
```

## Syntax with Keyword Parameters

```
EXEC ADABAS PROGRAM
PROGRAM-ID=<member-name>
LIBRARY-ID=<library>
END-EXEC
```



**Note:** If member and library are specified when the Preprocessor is called, these values are taken. The statement EXEC ADABAS PROGRAM is then not necessary.

Parameters	
member-name	The name used to identify the XRef data. Maximum length: 8 characters.
library	If the parameter library is specified, a system of type G (3GL application) that contains this library name in its implementation pointer must have been defined before. If no library is specified, the *SYSCOB*, *SYSBAL* or *SYSPLI* libraries are used.



# XXVII

## Creation of XRef Data

---



# 29

## Creation of XRef Data

---

- What Types of Code Use are Documented in XRef Data ..... 230

XRef data is written for Assembler, COBOL, or PL/I programs. The following general rules apply when writing XRef data for these types of programs:

- The programs are declared as programs of type P.
- One XRef record is written for each program with the member name as the entry into the program.
- XRef data is written for the file and each field in the file that are accessed in the program using an EXEC ADABAS GENERATE or a COPY statement.

## What Types of Code Use are Documented in XRef Data

---

XRef data is written for the following types of code use:

- CALL and ENTRY statements. For COBOL and PL/I, the Preprocessor scans the source code for these statements. For Assembler (BAL, 370-Assembler), preprocessor statements have to be included in the source code to identify the call of external programs and functions.
- Use of copy code and include code.
- EXEC CICS statements: All Assembler, COBOL and PL/I programs are scanned for several types of EXEC CICS statements.

The different types are described in the sections below.

### Documenting CALL Statements and Entry Points

COBOL or PL/I programs are scanned for CALL statements and entry points.

The following rules apply:

- The name of the entry or external program can be up to 8 characters long.
- In COBOL a maximum amount of 50 entries and 300 called modules are allowed.
- A COBOL entry point is defined by an ENTRY statement.
- A PL/I entry point is defined as the label before a PROC, PROCEDURE or ENTRY statement.
- Equalization of module/entry names in PL/I will be performed. If an external program and an entry are found with both the same name, the existence of an internal procedure will be presumed.
- Assembler programs will be scanned for the macros CALL and CM\$CALL. The names following these statements will be written as the external member names. The entry points and called external programs may also be specified in separate preprocessor statements.

## Documenting the Use of Copy Code

All Assembler, COBOL and PL/I programs are scanned for copy code.

- Assembler and COBOL programs are scanned for the COPY statement; the first eight characters of the word following the COPY statement (in the same line) are stored as copy code name.
- PL/I programs are scanned for the INCLUDE statement. All copy/include code names within an INCLUDE statement are stored as XRef data.

## Documenting EXEC CICS Statements

All Assembler, COBOL and PL/I programs are scanned for a variety of EXEC CICS statements. The following types of EXEC CICS statements are found and documented:

```
EXEC CICS LINK PROGRAM ('literal')
EXEC CICS XCTL PROGRAM ('literal')
EXEC CICS LOAD PROGRAM ('literal')
EXEC CICS SEND MAP ('literal')
EXEC CICS RECEIVE MAP ('literal')
EXEC CICS HANDLE ABEND PROGRAM ('literal')
```

The literal will be written into XRef data as the external member name. If the literal is longer than eight characters, or the brackets contain a variable without apostrophes or quotes, \*DYNAMIC will be written.

