NaturalONE

Predefined Object Types in Predict

Version 9.3.4

October 2026

ADABAS & NATURAL

This document applies to NaturalONE Version 9.3.4 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 © 2009-2026 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software GmbH product names are either trademarks or registered trademarks of Software GmbH 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 GmbH and/or its subsidiaries is located at https://softwareag.com/licenses.

Use of this software is subject to adherence to Software GmbH's licensing conditions and terms. These terms are part of the product documentation, located at https://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 GmbH Products / Copyright and Trademark Notices of Software GmbH Products". These documents are part of the product documentation, located at https://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 GmbH.

Document ID: ONE-OPRDPREDEF-934-20261004

Table of Contents

Preface ix
1 About this Documentation 1
Document Conventions 2
Online Information and Support 2
Data Protection 3
2 General Information 5
Metastructure of the Predict Data Dictionary 6
Global Attributes 7
I Access Definition 11
3 Access Definition 13
Add an Access Definition 15
II Database 17
4 Maintaining Objects of Type Database 19
Database Types 20
Defining Basic Attributes of Databases 20
5 Documenting Databases of Different Types 23
Database Type A - Adabas 24
Specifying the Size of an Adabas Database 26
Database Types C, E, P - Conceptual, General SQL Handler, Entire System Server Nodes 28
Database Type D - DB2 29
Database Type O - Oracle Handler 30
Database Types Q, M, R, H - Adabas SQL Handler, RMS Handler, rdb Handler, Other Handler 30
Database Type I - IMS 31
Database Type S - SQL Server 32
Database Type T - Target Node 33
Database Type V - VSAM Handler 34
Other SQL Database Types 35
6 Database-Specific Maintenance 37
Purge Database 38
Changing Database Attributes 38
III Dataspace 41
7 Maintaining Objects of Type Dataspace 43
Adding a Dataspace 44
Defining Basic Attributes of Dataspace - DB2 Mainframe 44
Defining Basic Attributes of Dataspace - SQL/DS 50
Defining Basic Attributes of Dataspace - DB2 Open Systems 51
8 Dataspace-Specific Maintenance 55
Purge Dataspace 56
IV Extract 57
9 Maintaining Objects of Type Extract 59
Defining Basic Attributes of Extract 60

V Field 61
10 Defining Basic Attributes of Fields 63
Field List Tab 64
Field Type 65
Level Number 66
Field Format 67
Character Set 68
Character Set - Adabas 69
Field Length 69
Descriptor Type 78
Descriptor Type - continued 79
Maximum Number of Values / Occurrences 80
Unique Option 81
Field Short Name 81
Suppression / Null Value Option 82
Variable Length Option - IMS 83
Null Default Option 83
Natural Field Length 83
Do Not Convert Option 84
Related Standard File 84
Check against standard 84
Natural Attributes 84
Hidden 86
11 Defining Derived Fields 89
General Rules for Defining Derived Fields 90
Defining Derived Fields of Special Types 93
Rules Applying to Format Changes 96
Rules Applying to Suppression/Length Changes 98
Validation of Derived Field Definitions 98
12 Defining Additional Attributes of Fields 109
3GL Specification 111
Conditions 113
Field Name Synonyms 114
Adabas Security and Edit mask 115
DBMS Extensions 117
Field Procedure 121
Derived Field Expression 122
Index Definition - DB2 123
Index Definition - Oracle 128
Default value 134
Constraint name 135
Identity definition / Change log 136
Platform Compatibilities 138
Base Extensions 139
VI File 145

13 Maintaining Objects of Type File 149
Common File Attributes 150
Defining Basic File Attributes 153
14 Adabas Files, File Type A 155
Add/Modify a File 156
Modifying Adabas Attributes 157
Phys. distribution attr. 159
Adabas Security Definition 159
Extent Allocation - Size Specifications For More Than One Extent 161
Modifying ADAM Descriptor Definition 162
Encodings 163
15 File Types Conceptual, Standard and Other 165
16 SQL File Types 167
Naming Conventions for SQL Objects 168
Common Parameters for SQL File Types 169
Field Lists of SQL Views 175
Editing the Subquery of an SQL View 176
17 Adabas SQL Server 177
Adabas SQL Server - Overview 178
Naming Conventions for Adabas SQL Server 178
Adabas Cluster Table 180
Adabas SQL View 181
18 Adabas D 183
Naming Conventions for Adabas D 184
Adabas D Table, File Type BT 185
Adabas D View, File Type BV 186
19 DB2 187
Naming Conventions for DB2 188
DB2 Table, File Type D 189
DB2 View, File Type E 192
Intermediate View, File Type IV 192
Intermediate Table, File Type IT 193
DB2 Query Table, File Type MT 194
20 Informix 199
Naming Conventions for Informix 200
Informix Table, File Type XT 201
Informix View, File Type XV 203
21 Ingres 205
Naming Conventions for Ingres 206
Ingres Table, File Type JT 207
Ingres View, File Type JV 208
22 Oracle 209
Naming Conventions for Oracle 210
Oracle Table, File Type OT 211
Oracle View, File Type OV 215

23 Sybase 217
Naming Conventions for Sybase 218
Sybase Table, File Type YT 219
Sybase View, File Type YV 220
24 General SQL File, File Type X 221
25 RDB 223
26 IMS 225
IMS Segment Layouts and Userviews - File Types J and K 226
Editing Field Lists of IMS Files 227
27 VSAM 229
Physical VSAM File - File Type V 230
VSAM Logical Files, VSAM Userviews - File Types L, W and R 232
28 ISAM 233
29 Entire System Server 237
30 File-Specific Maintenance 239
Purge File 240
31 Rippling - Ensuring Consistent Data Definitions 241
Rippling - Overview 242
Rippling from Standard Files 243
Rippling from Master Files to Views/Userviews 247
VII 253
32 File Relation 255
File Relation Maintenance 256
33 Interface 263
Defining Basic Attributes of an Interface 265
VIII Keyword 267
34 Maintaining Objects of Type Keyword 269
Defining Basic Attributes of Keyword 270
Keyword Maintenance Functions 270
IX 271
35 Library Structure 273
Add/Modify Library Structure 275
36 Method 277
Add/Modify a Method 279
37 Network 281
Add a Network 283
Network-Specific Maintenance 283
38 Node 285
Add a Node 287
39 Packagelist 289
Packagelist Types 290
Add a Packagelist 291
Packagelist-Specific Maintenance 292
X Program 293
40 Maintaining Objects of Type Program 295

Program Types 296
Languages 297
Program-Specific Libraries 297
Add a Program 298
41 Defining Additional Attributes of Programs 301
Programs of Type Class 303
Programs of Type Resource 304
Programs of Type SQL Procedure 305
Programs of Type Database function 313
System Programs 316
Programs of Type dynamic 316
42 Program-Specific Maintenance 317
Editing Entry Points 318
Overview of Language-Specific Program Types 318
SQL Procedure Code Tab 320
XI 321
43 Property 323
Add a Property 325
44 Report Listing 327
Report Listing ID 328
Modify Report Listing 330
45 Server 331
Add a Server 333
XII Storagespace 335
46 Maintaining Objects of Type Storagespace 337
Add a Storagespace 338
Storagespace-Specific Maintenance 339
XIII 341
47 System 343
System Types 344
Add/Modify System 345
System-Specific Maintenance 347
48 Trigger 349
Add a Trigger 351
Editing the Trigger Code of a Trigger 352
XIV User 353
49 Maintaining Objects of Type User 355
Add/Modify a User 356
User Maintenance 357
XV Verification 359
50 Maintaining Objects of Type Verification 361
Verification Status 362
Verification Formats 362
Add a Verification 363
51 Verification-Specific Maintenance 367

Purge Verification 368
XVI Virtual Machine 369
52 Virtual Machine 371
Add a Virtual Machine 373

Preface

This documentation describes all the predefined object types, provided by Predict. Type-specific attributes of the respective object type and the type-specific maintenance and retrieval functions are explained. Each object type is described in a separate section. The object types are arranged in alphabetical order.

This documentation covers the following topics:

General Information	Provides general information on the predefined object types in Predict. It describes global attributes such as object ID and keywords. This general information is not repeated in the descriptions of the individual objects.
Access Definition	Objects of type Access Definition document masks and permissions of DB2 columns and tables.
Database	Objects of type Database document a collection of physical and/or logical files.
Dataspace	Objects of type Dataspace document DB2 tablespaces and SQL/DS DBspaces.
Extract	With this object type you can create sets of objects. An extract is used primarily for transferring data with the Predict Coordinator.
Field	With the object type Field you can document field definitions for a wide range of database management systems.
File	With the object type File you can document file structures for a wide range of database management systems. This section also describes the process of rippling.
File Relation	With file relations you can document the relationships between fields in a file.
Interface	Together with objects of type Method, Property and Program, interfaces document the Natural program object class.
Keyword	You can assign objects of type Keyword to other objects in order to link objects logically.
Library Structure	This object type supports the Steplib concept in Natural.
Method	This object type documents the methods of an interface.
Network	Together with objects of type Virtual Machine, networks document the hardware and operating system environment of a data processing system.
Node	This object type together with object type Server documents Remote Procedure Calls.
Packagelist	This object type documents DB2 packages.
Program	With objects of type Program you can document nearly 20 types of programs. Many different programming languages are supported.
Property	This object type documents the properties of an interface.
Report Listing	With this object type Predict Coordinator transfer operations and conversion functions are logged.

Server	This object type is used together with object type Node to document Remote Procedure Calls.
Storagespace	This object type documents DB2 storagegroups.
System	With this object type you can document complex applications.
Trigger	This object type documents SQL triggers.
User	An object of type User documents an individual user.
Verification	Objects of type Verification document the processing rules for validating field values.
Virtual Machine	Together with objects of type Network, objects of type Virtual Machine document the hardware and operating system environment of a data processing system.

1 About this Documentation

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

Document Conventions

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

Online Information and Support

Product Documentation

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

Product Training

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

Tech Community

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

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

Product Support

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

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

Data Protection

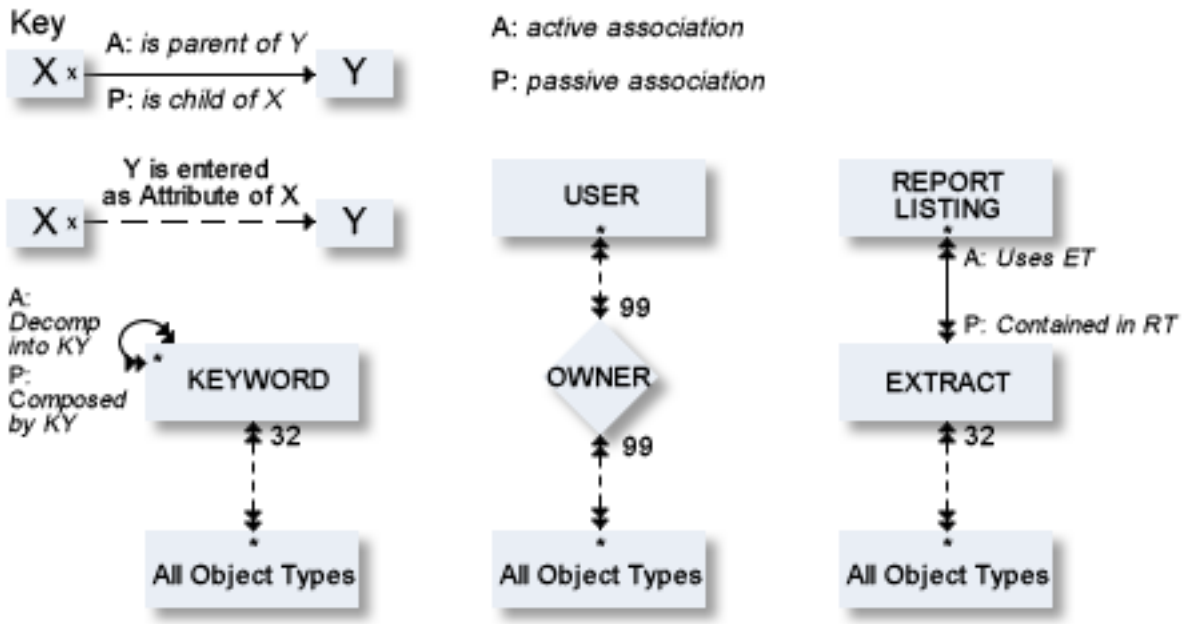
Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

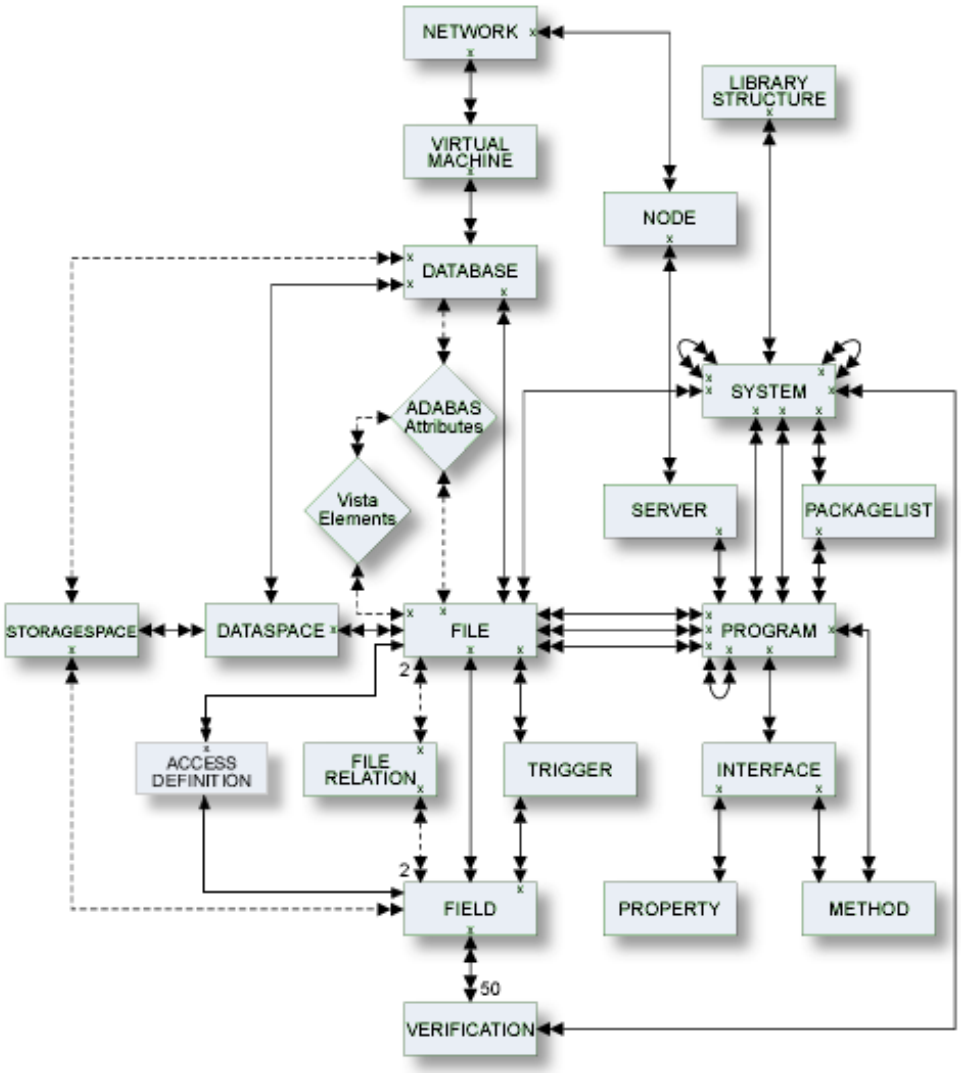
2 General Information

- Metastructure of the Predict Data Dictionary 6
- Global Attributes 7

Metastructure of the Predict Data Dictionary

The metastructure of the Predict data dictionary is illustrated below. Additional object types and association types can be defined with Metadata Administration functions. These objects are referred to as User Defined Object/Association Types or User Defined Entities (UDEs). See the section *Metadata Administration* in the *Predict Administration* documentation.





Global Attributes

The following sections in the object type-specific windows are available for all predefined and user-defined object types.

- General
- Keywords
- Abstract
- Owner

- Extended Description

General

When an object has been added and named, its object ID (name) and object type are shown in this section. Date and time when this object was added and last modified are also shown.

Each object in Predict is identified by its ID. This ID must be unique for objects of the same type.

Note: Field objects can have the same ID if they belong to different files.

To change the ID of an object, use the **Rename** command.

Naming Conventions

IDs of all objects apart from verification are checked against the following naming conventions. (IDs of verifications are checked against Natural naming conventions.)

- The ID of all object types except user can be up to 32 characters long.
- Objects of type user can have IDs of up to 8 characters.
- The maximum length of object IDs (both for predefined and user-defined object types) can be specified with the metadata administration function Modify object type. See the section *Metadata Administration* in the *Predict Administration* documentation for more information.
- There is no minimum length for object IDs: one and two character IDs are also possible.
- An object ID must start with a letter (A - Z or a - z).
- The subsequent characters must be alphanumeric, i.e.
  - letters A - Z or a - z
  - digits 0 - 9
  - any special character except blank, asterisk, comma, question mark.
  - Up to 20 additional disallowed characters can be specified with the Metadata Administration function Modify object type. See the section *Metadata Administration* in the *Predict Administration* documentation for more information.
  - The Predict administrator can specify with the parameter General Defaults>Miscellaneous>Upper/lower case whether alpha characters in object IDs are converted to upper case. Use of lower-case letters for object IDs is not recommended.

Naming Conventions for Natural

If the object ID is also to be referenced by a Natural subsystem, the Natural naming conventions should also be observed.

SQL Naming Conventions

Naming conventions for SQL objects are given in the part dealing with object type **File** in this documentation.

Naming Conventions for Extracts

The following extracts are added automatically with the Coordinator:

- #SAG-TRANSFER
- #SAG-ERROR

See the Predict Coordinator documentation for more information. These IDs are reserved.

Keywords

Up to 32 keywords can be assigned to any Predict object, including keywords.

This section provides a table. You can add new keywords to the table and/or delete existing keywords.

- A keyword must exist as a Predict object before it can be assigned to another object.

See the part on object type **Keyword** in this documentation for more information.

Abstract

Each object in Predict can have an abstract providing short comments on the object.

This section provides a simple text editor. Edit commands such as **Copy** and **Paste** are available in a context menu.

- An abstract can have up to 16 lines of up to 30 characters.
- Abstracts can contain upper and lower-case letters. If the general default parameter Miscellaneous > Upper/Lower case > Abstract is set to U, all alphabetic characters are converted to upper-case.

Owner

Each object in Predict can have an owner list.

This section provides a table. You can add new owners to the table and/or delete existing owners.

Extended Description

Each object in Predict can have a description.

This section provides a text editor. See *Text Editor* in *Managing Documentation Objects,* which is part of the *Predict Description and Generation* documentation.

I Access Definition

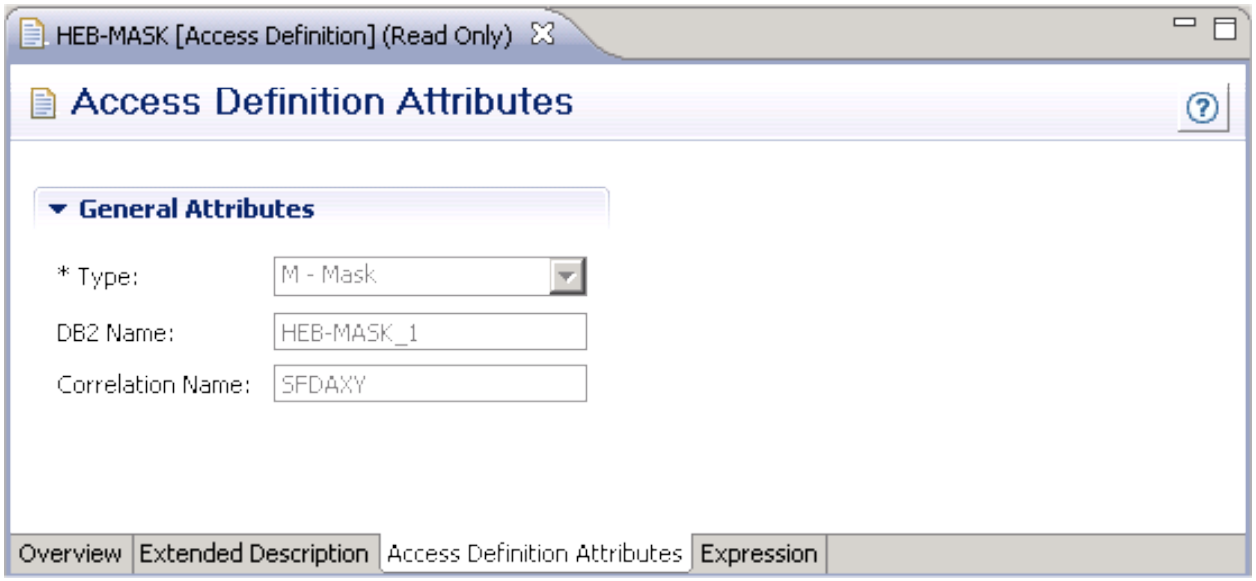
3 Access Definition

- Add an Access Definition 15

Masks and permissions for DB2 columns and tables are documented with objects of type Access Definition.

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Add an Access Definition

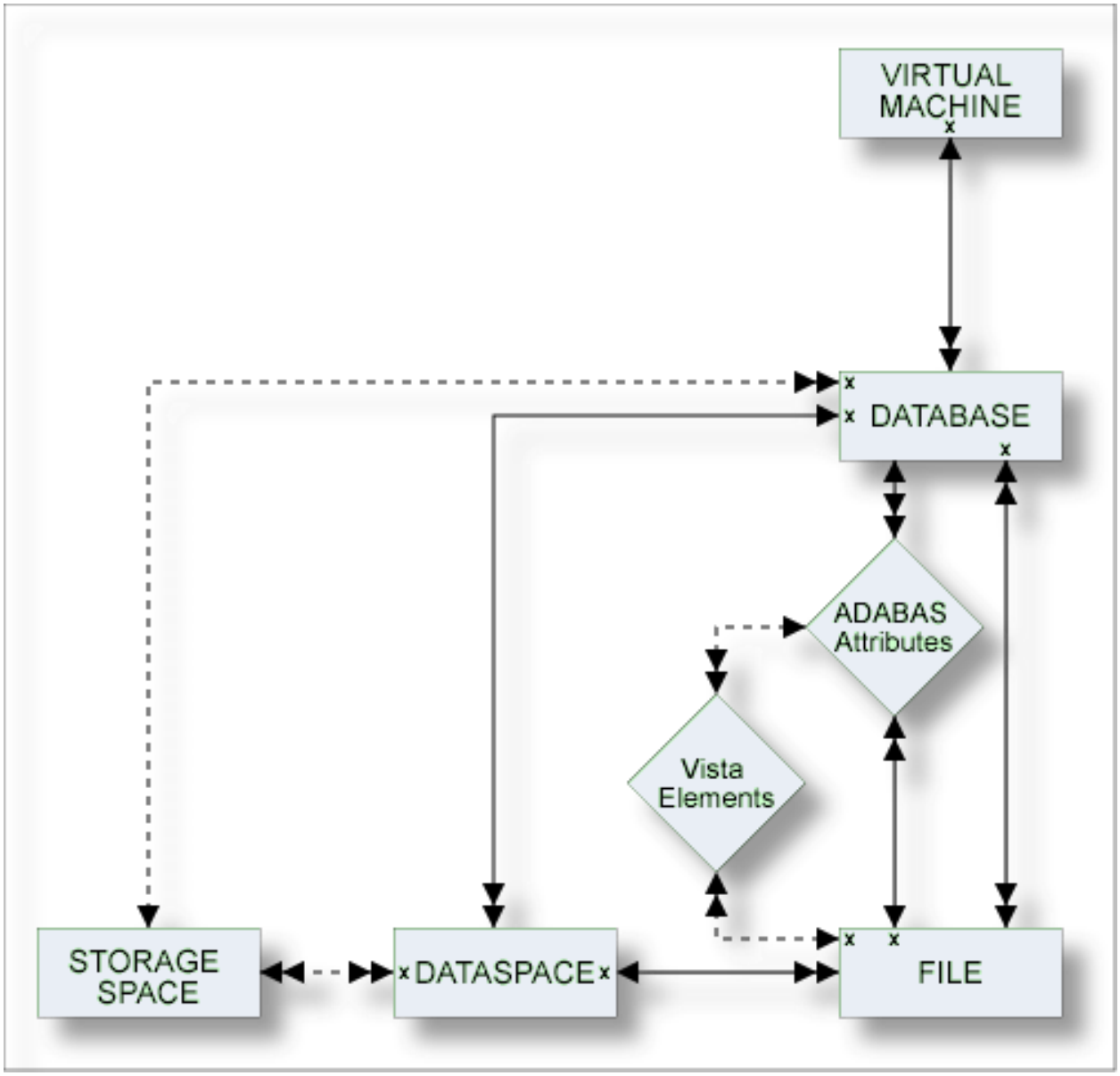


Note: Parameters not listed here are described under *Global Attributes*.

Parameters		
Type	The type of the access definition. Valid values:	
	M	Mask. A column mask in DB2 is used for column access control and specifies the value that should be returned for a specified column. Exactly one mask per column is allowed. In DB2 an enabled column mask does not take effect until the `ALTER TABLE` statement with the `ACTIVATE COLUMN ACCESS CONTROL` clause is used to activate column access control for the table.
	R	Permission. Multiple row permissions can be created for a table. In DB2 an enabled row permission does not take effect until the `ALTER TABLE` statement with the `ACTIVATE ROW ACCESS CONTROL` clause is used to activate row access control for the table.
DB2 Name	The name of the DB2 table or column.	
Correlation Name	The correlation name defined in the subselect clause of a view.	

II Database

Databases and data storage systems of different types are documented with objects of type Database.



The description of object type Database is organized under the following headings:

Maintaining Objects of Type Database

Documenting Databases of Different Types

Database Specific Maintenance

4 Maintaining Objects of Type Database

- Database Types 20
- Defining Basic Attributes of Databases 20

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

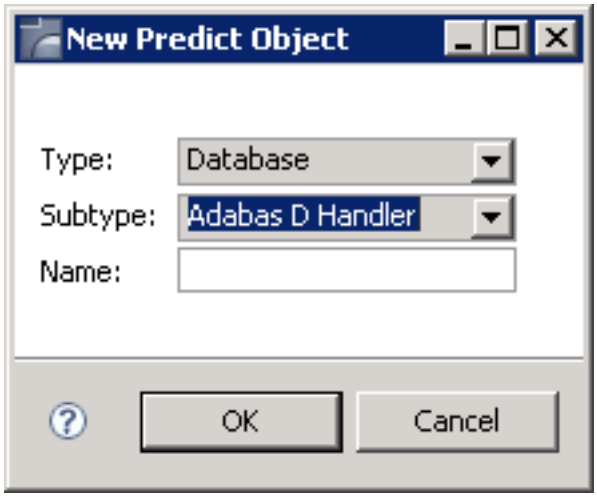
Database Types

The table below contains a list of all valid database types.

Code	Database Type
A	Adabas
B	Adabas D handler
C	Conceptual
D	DB2
E	Gen. SQL handler
H	Other handler
I	IMS
J	Ingres handler
M	RMS handler
O	Oracle handler
P	Entire System Server
Q	Adabas SQL handler
R	rdb handler
S	SQL Server
T	Target node
V	VSAM handler
X	Informix handler
Y	Sybase handler

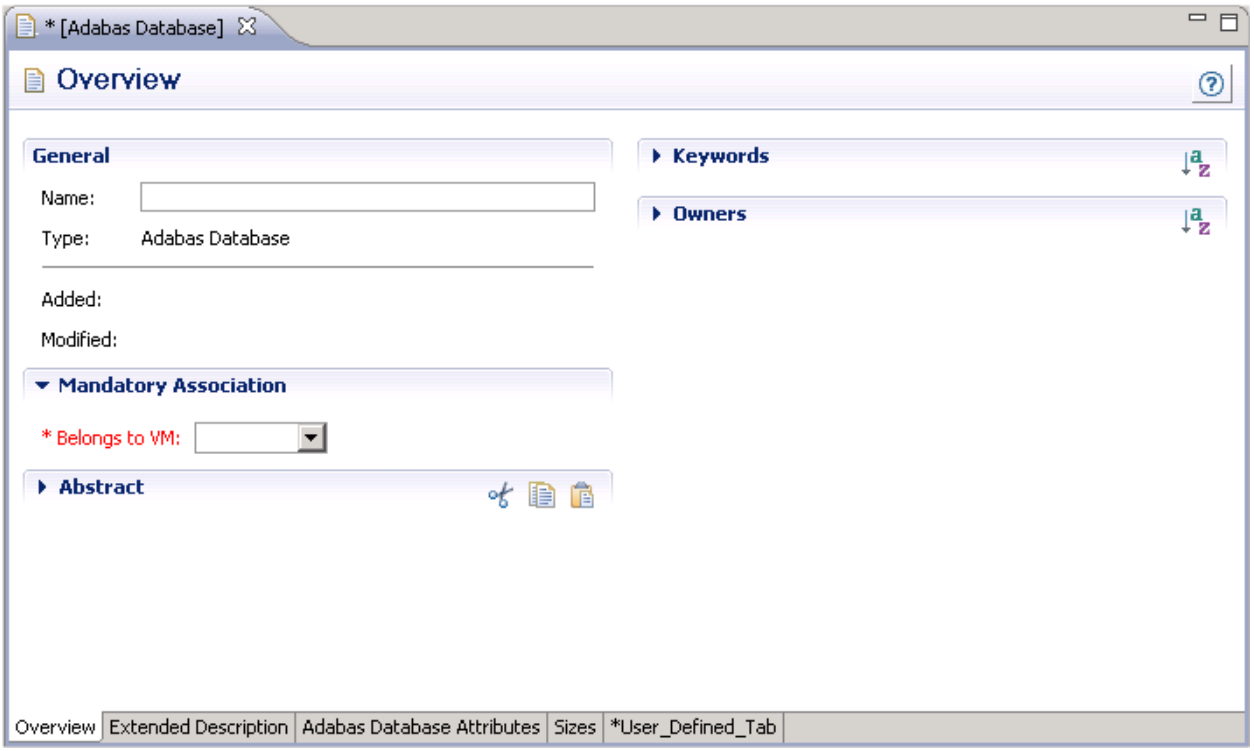
Defining Basic Attributes of Databases

When you add a database, you first have to specify the database type in the **New Predict Object** dialog box.



When you choose the **OK** button, a database type-specific window appears. The database type is indicated in the title bar.

The following sample window shows the parameters which apply to most types of databases.



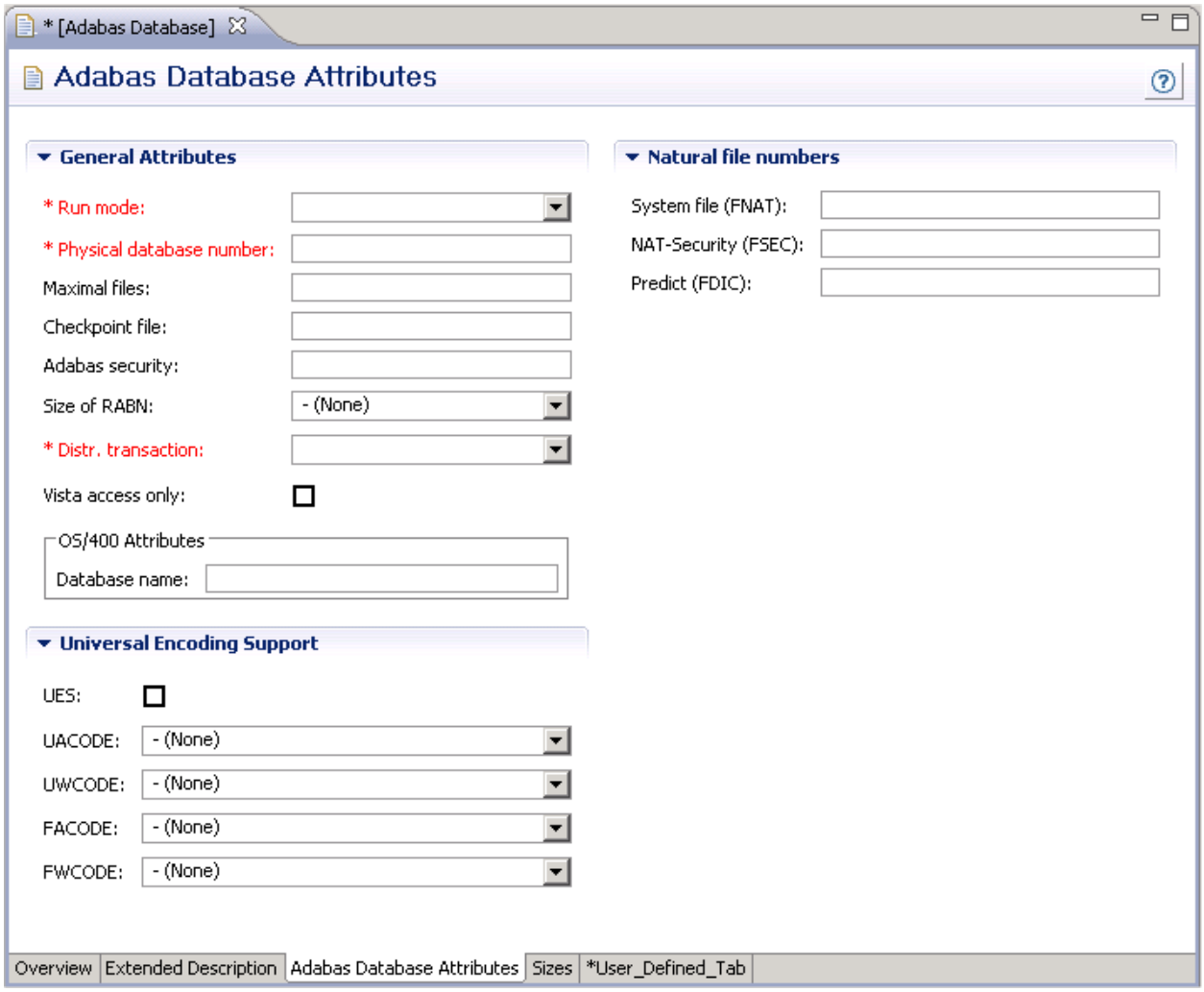
Attributes		
Belongs to VM	This attribute can be found on the **Mandatory Association** tab. Predict virtual machine object documenting the hardware and operating system environment of the database. See also *Defining the Distribution of Data in Predict* in the section *Vista* in the *Predict and Other Systems* documentation.	
Run mode (available from the Adabas Database Attributes tab)	Use of the database with respect to the distribution of data with Adabas Vista.	
	I	Isolated. Adabas Vista is not used. The database is isolated.
	L	Local. The database cannot be accessed using Entire Net-Work.
	V	Vista. Adabas Vista is used. Only valid for Adabas databases.
	See *Defining the Distribution of Data in Predict* in the section *Vista* in the *Predict and Other Systems* documentation for a detailed description of the meaning of the Vista parameter.	
Physical database number (available from the Adabas Database Attributes tab)	Valid values depend on **database type**:	
	Database Type	Range of Database Numbers
	B, E, J, O, Q, R, X, Y	1 - 255
	A, H, M, P, T, V	1 - 65535
	Others	not applicable

5 Documenting Databases of Different Types

- Database Type A - Adabas 24
- Specifying the Size of an Adabas Database 26
- Database Types C, E, P - Conceptual, General SQL Handler, Entire System Server Nodes 28
- Database Type D - DB2 29
- Database Type O - Oracle Handler 30
- Database Types Q, M, R, H - Adabas SQL Handler, RMS Handler, rdb Handler, Other Handler 30
- Database Type I - IMS 31
- Database Type S - SQL Server 32
- Database Type T - Target Node 33
- Database Type V - VSAM Handler 34
- Other SQL Database Types 35

Note: The window for the object type Database contains common sections that are always the same for all object types. For detailed information on these sections, see *Global Attributes*.

Database Type A - Adabas

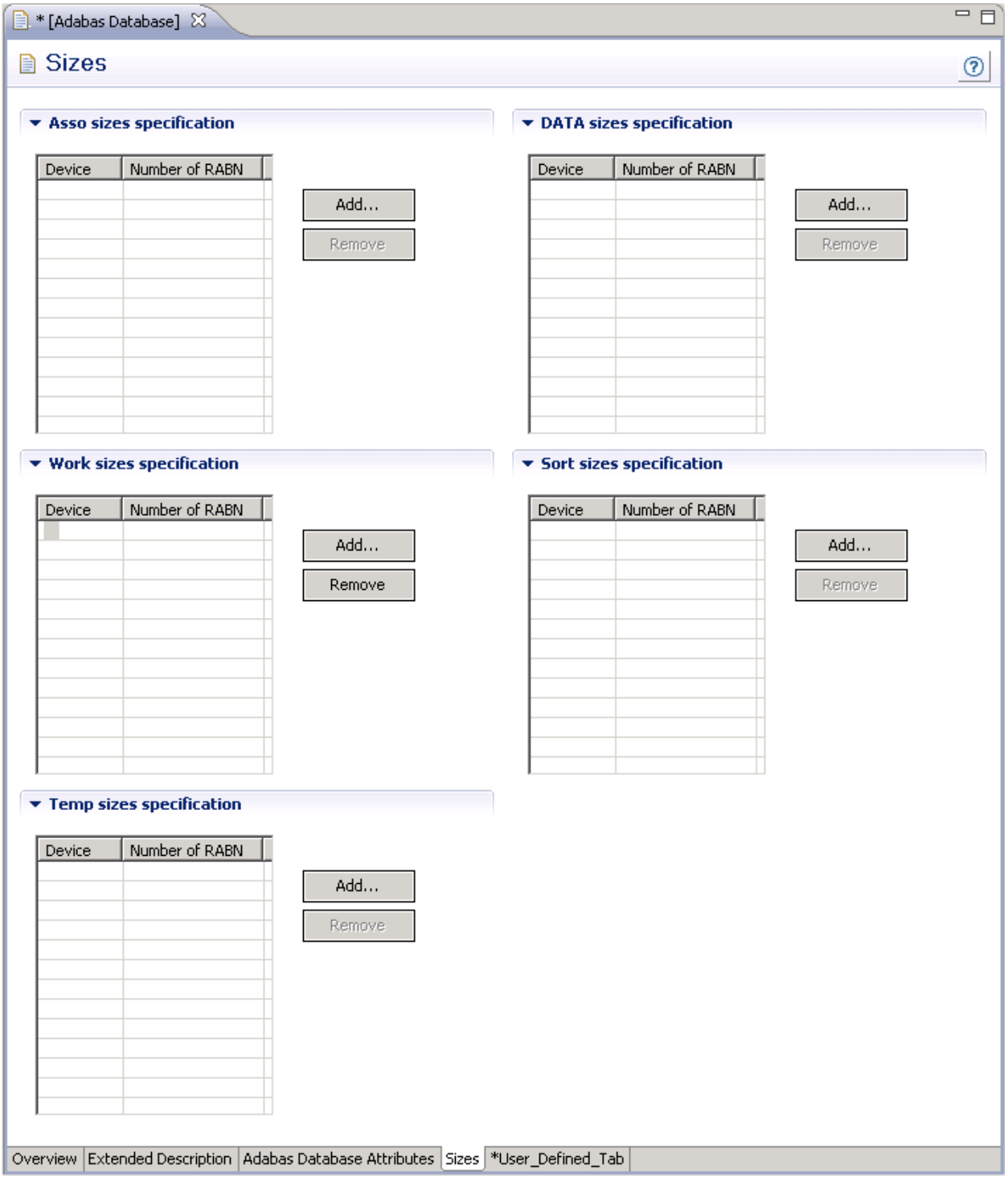


Note: Attributes that are not in the table below are described in the section *Defining Basic Attributes of Databases*.

Attributes	
Adabas attributes	
Maximal files	Number of files permitted in the database (ADADEF parameter MAXFILES). This number must either be 0 or at least 5 but not more than 32767.
Checkpoint file	The number of the Adabas file which contains checkpoint information for the database. Predict automatically creates a data dictionary object with the file ID SAG-ADA-CHECKPOINT for this file.
Adabas security	The number of the Adabas file which contains Adabas security information for the database. Predict automatically creates a data dictionary object with the file ID SAG-ADA-SECURITY for this file.
Size of RABN	Specifies the length of RABNs in the database. 0 — not specified 3 — 3 Byte for 24-bit RABNs 4 — 4 Byte for 31-bit RABNs
Distr. transaction	N — No (Default) RM — Resource Manager TM — Transaction Manager Field must be filled if the database is part of the distributed transaction processing (DTP) of the Adabas Transaction Manager
Vista access only	Y — If the attributes of the database are such that files in the database can only be accessed using Adabas Vista. Vista access only is set by Predict.
OS/400 Attributes	Specify an OS/400 database in the field **Database name**.
Natural file numbers	
System file (FNAT)	The number of the Natural system file.
NAT Security (FSEC)	The number of the Adabas file which contains Natural Security information.
Predict (FDIC)	The number of the Adabas file which contains the dictionary data.
Universal Encoding Support	
Universal encoding support of an Adabas Database can be defined in this section. See the Adabas documentation for detailed information on this topic.	

Specifying the Size of an Adabas Database

Physical properties of a database (device types and sizes of the data sets containing the Adabas ASSO, DATA, WORK, SORT and TEMP) can be defined on the **Sizes** tab.



To specify the size, choose the **Add** button. A new row appears in the table and you can now select a device type and enter the size in RABNs. You can also select an existing entry and change its values.

Parameters	
Device	Devices are identified with a four-letter code that must have been defined in Predict on the server side with the function Adabas device types in the Special functions menu. If a device type is changed, the change should also be made in each file object that is linked to the database.
Number of RABN	The number of RABNs (relative Adabas block numbers) of the specified device that are occupied by the specified extent of the specified database.

Specifying the Encodings of an Adabas Database

Universal encoding support of an Adabas Database can be defined

Note: See the Adabas documentation for detailed information on this topic.

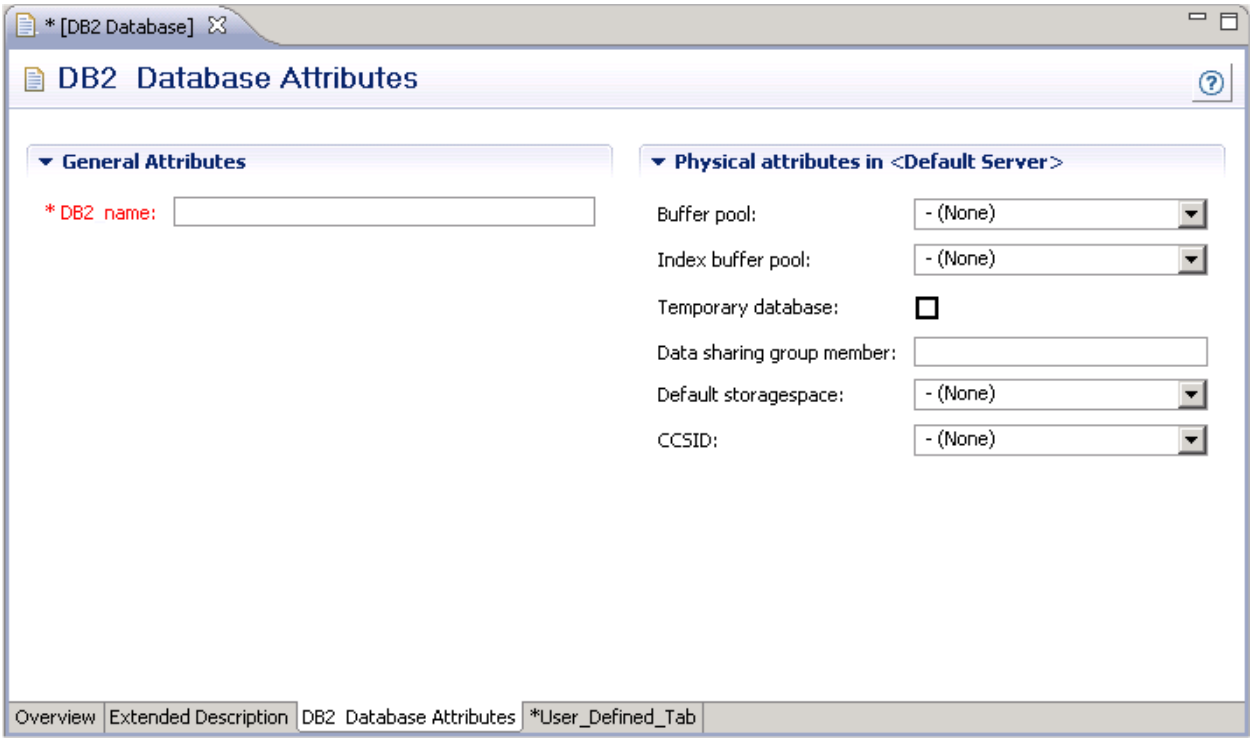
Database Types C, E, P - Conceptual, General SQL Handler, Entire System Server Nodes

A window similar to the following is displayed when adding or modifying databases of the types C, E and P.

The database-specific parameters are described in the section *Defining Basic Attributes of Databases*.

Database Type D - DB2

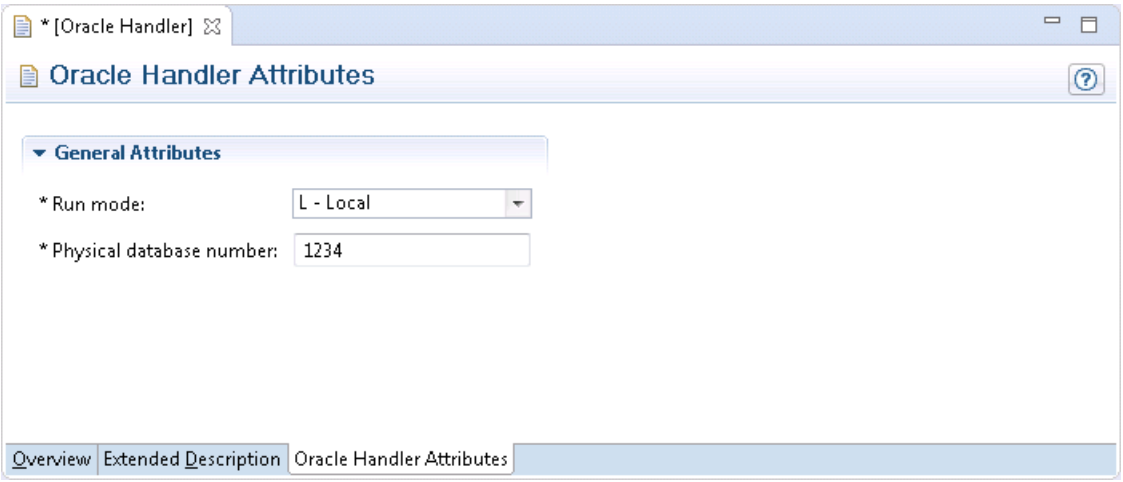
The following attributes apply to databases of type D.



Attributes	
DB2 name	The name of the database in DB2.
Buffer pool	The buffer pool of the database.
Index buffer pool	Buffer pool which is used for the indexes created within the database.
Temporary database	N: No (Default). Y: Database is used for declared temporary tables.
Data sharing group member	Name of the member of the data sharing group. Leave blank or specify name with up to eight characters (letters A-Z, digits 0-9 and special characters $, # and @).
Default storagespace	DB2 tables of the database will be implemented in this storage group if no other storage group is explicitly specified.
CCSID	Defines the encoding scheme of the database. blank: not specified A: ASCII E: EBCDIC

Database Type O - Oracle Handler

A window similar to the following is displayed when adding or modifying databases of the type O.



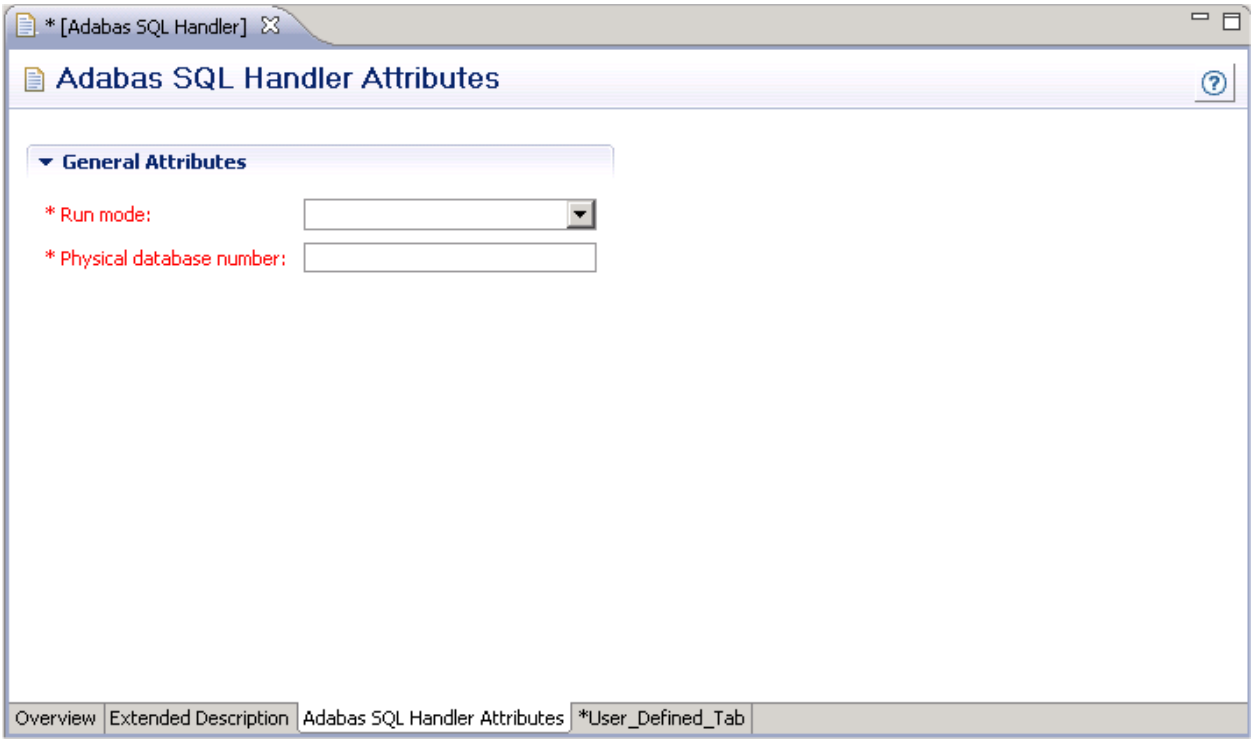
The database-specific parameters are described in the section *Defining Basic Attributes of Databases*.

Database Types Q, M, R, H - Adabas SQL Handler, RMS Handler, rdb Handler, Other Handler

Database type Q is used to document databases of type Adabas SQL handler. See the section *Adabas SQL Server* in the *Predict and Other Systems* documentation for more information.

Database type M is used to document RMS databases; database type R is used to document rdb databases.

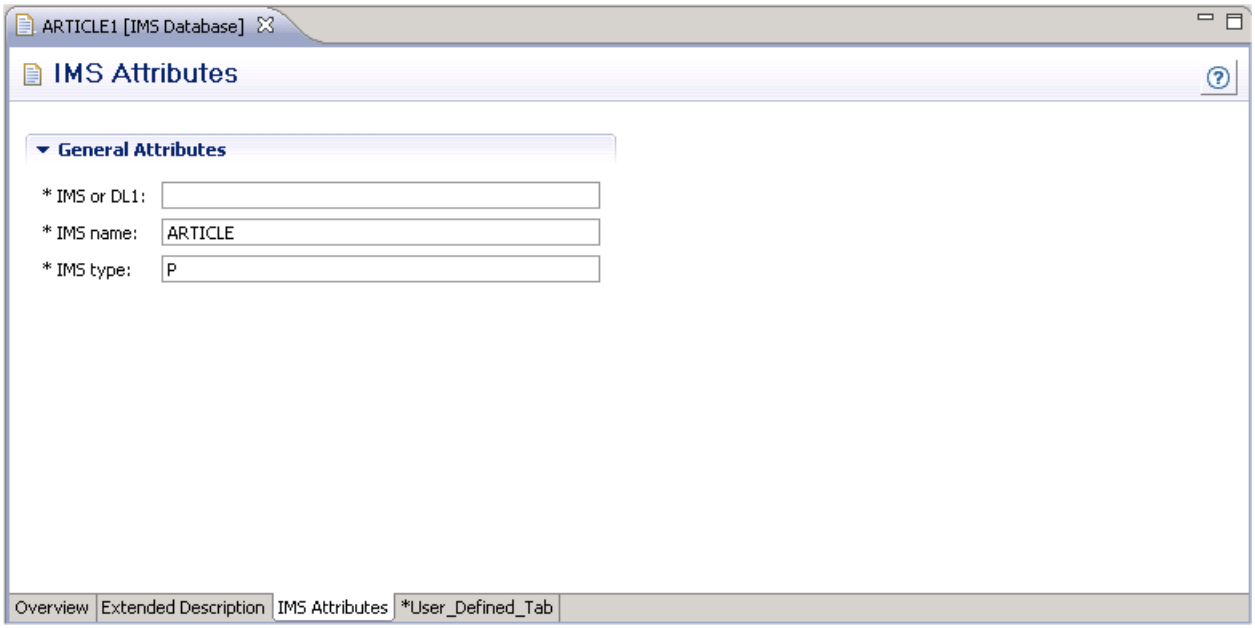
Database type H is used to represent database handlers, such as USER-DB, SESAM, DL/I, WIZZARD, TRS etc. Database type other handler can be used to reserve a database number (prevent it from being used by Adabas).



Parameters	
Physical database number	For database type RMS Handler: the database number must be declared in the Natural NATPARM parameter file as an RMS database number if DDMs for RMS files contained in the database are to be generated. See table in the section *Defining Basic Attributes of Databases* for range of permitted values.

Database Type I - IMS

IMS databases cannot be added using Predict Description and Generation. For further information on how to create an IMS Database object in Predict, see *External Objects in Predict*.



The following attributes apply to databases of type I.

Attributes	
IMS or DL1	The kind of database. Valid values: IMS DL1
IMS name	The name of the database in IMS.
IMS type	The type of the database in IMS. Valid values: LOGICAL PHYSICAL

Database Type S - SQL Server

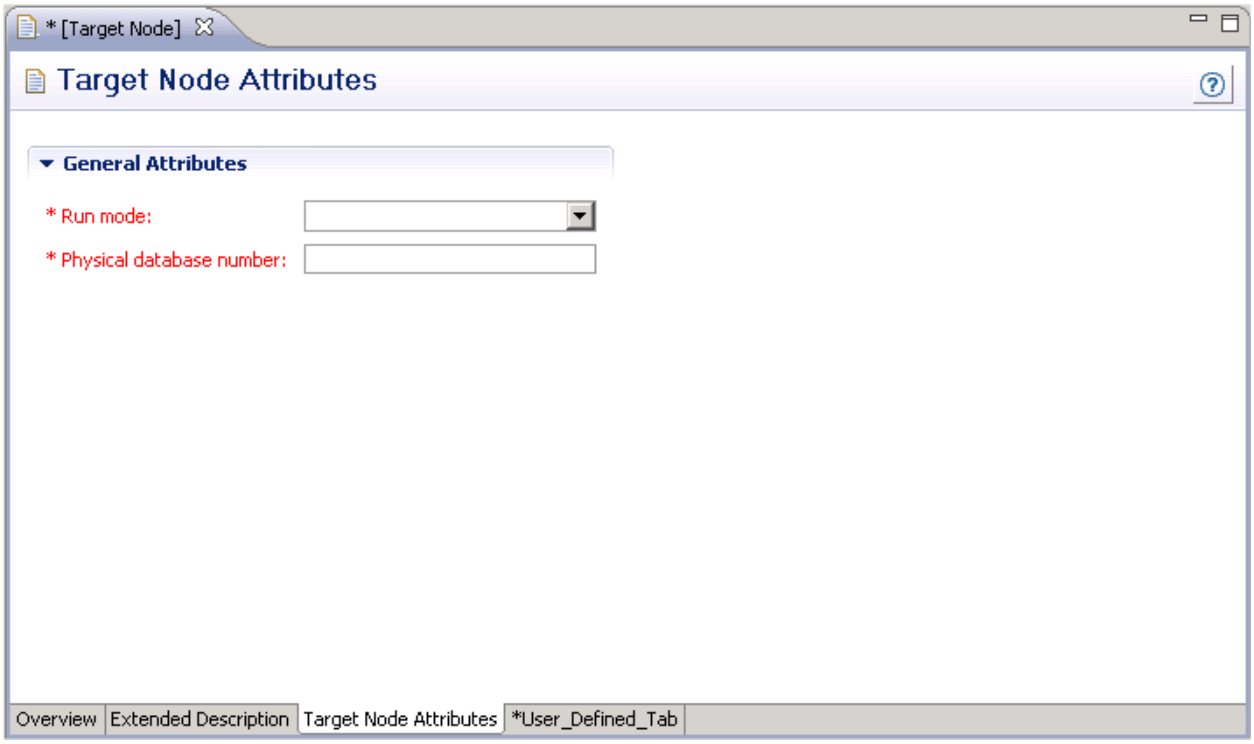
Database type S is used to represent objects of type SQL server.

Attributes	
Connection type	The type of connection: N - Natural for DB2 O - Entire Access C - Natural SQL Gateway
Server name / catalog name	The name that was specified when the data source was created.
Port number	Identifies the server daemon.
Host name / Gateway	Identifies the host machine or gateway on which the server runs.
Driver / CDD data source	Specifies the database driver or data source to be used.

Database Type T - Target Node

Database type T is used to represent database nodes entered in the ID table of an SVC which cannot be documented with a corresponding database type: BROKER, NATURAL GLOBAL BUFFER POOL etc.

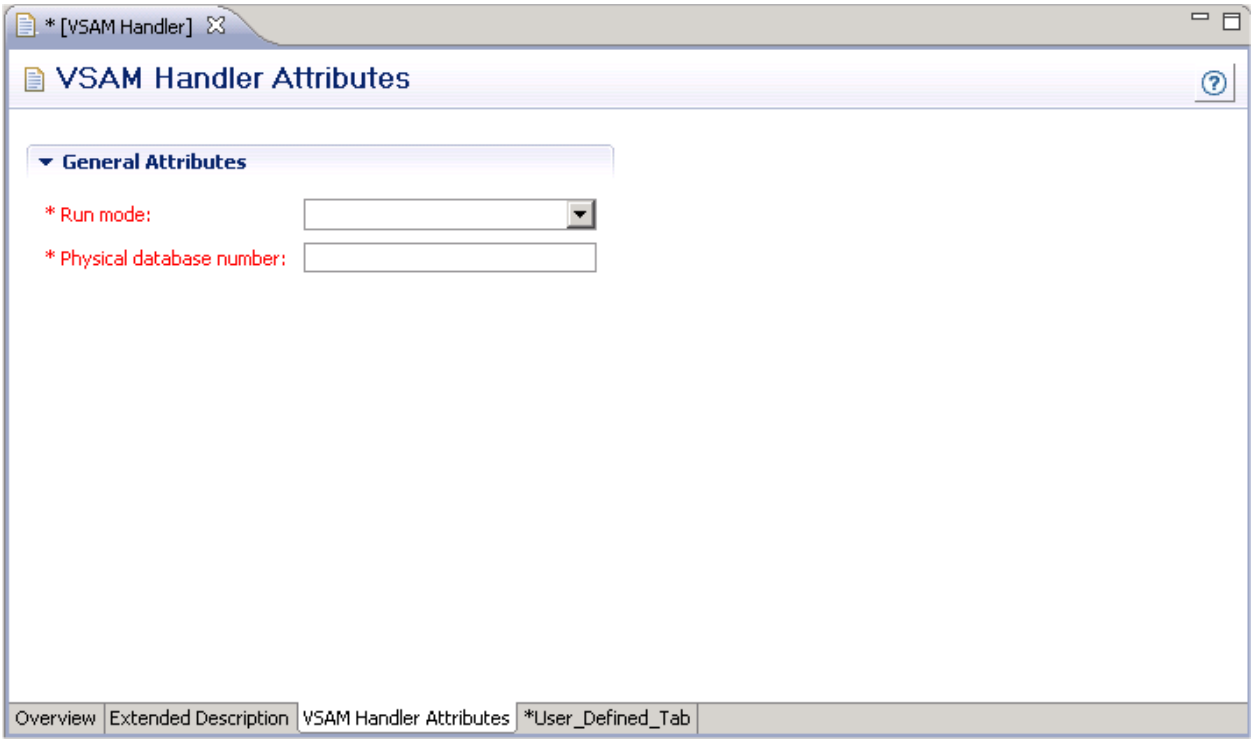
This type of database is used to reserve the corresponding database number and thus prevent this number being used for an Adabas database.



Attributes		
Run mode	Must be specified for databases of this type. Valid values:	
	I	Isolated
	L	Local
Physical database number	The physical database number must be in range 1-65535.	

Database Type V - VSAM Handler

Database objects of type V are used to collect all definitions of VSAM clusters which are accessed by the same Natural VSAM handler. The database number defined in a database object of type V is used by the `GENERATE DDM` function.



Attributes	
Run mode	Must be local for databases of this type.
Physical database number	The physical database number must be in the range from 1 - 65535.

Other SQL Database Types

The windows used to maintain database objects of the following types are the same as for VSAM databases above. The physical database number must be less than or equal to 254.

Type	Description
J	Ingres Handler
O	Oracle Handler
X	Informix Handler
Y	Sybase Handler
B	Adabas D Handler

6 Database-Specific Maintenance

- Purge Database 38
- Changing Database Attributes 38

Purge Database

Predict objects of type Database are purged with the **Delete** command.

The **Delete** command applies to all database types apart from IMS. The following objects are deleted:

- the database object
- all links to parent and child objects.
- Rules which apply to the individual database types are given below.

Database-specific Rules

For Database Objects of Type Adabas

- A Purge operation is not executed if the database and files in the database are implemented.
- The Delete operation purges a database object and all links to related objects. All Adabas attributes for files which are contained in this database are purged or changed to default if the file is not contained in another database.
- File objects for which DDMs or table/cluster descriptions exist will not be purged.

For Database Objects of Type SQL

- A Purge operation is not executed if the database and files contained in the database are implemented.
- File objects for which DDMs or table/cluster descriptions exist will not be purged.

For Database Objects of Type IMS

- The Purge function will not be executed if UDFs exist for the IMS files.

Changing Database Attributes

The following topics are covered below:

- General Rules
- Changing the Database ID
- Linking the Database to another Virtual Machine
- Changing the Run mode Parameter
- Changing the Database Number
- Changing the Parameter Vista Access Only

- Messages

Note: The database type can only be changed with Predict on the server side.

General Rules

- Changes to database attributes are also applied to file objects if applicable. For example: if a database is linked to another virtual machine, existing Vista elements of files linked to the database are adapted accordingly.
- Special rules apply when renaming/renumbering databases that are connected to implemented databases. Connecting documentation and external objects is described in the section *Handling of External and Documentation Objects* in the *External Objects in Predict* documentation.

Changing the Database ID

The database ID can be changed with the **Rename** command.

The new database ID must not already exist in the dictionary.

Linking the Database to another Virtual Machine

Another virtual machine can be specified on the **Mandatory Association** tab.

The following rules apply to Adabas databases:

- Old virtual machine and new virtual machine are in the *same network*: the change is applied to the database and all files linked to the database.
- Old virtual machine and new virtual machine are in *different networks*: new Vista elements are created and/or existing Vista elements are purged. .
- If the Database is connected to an implemented database the new virtual machine must be in the same network as the old virtual machine.

Non-Adabas databases can be linked to another virtual machine without restrictions.

Changing the Run mode Parameter

The Run mode can be changed on the **Database Attributes** tab.

The following rules apply:

- Changing from Run mode parameter I (isolated) or L (local) to V (Vista) is not possible if a replicated or master file for Entire Transaction Propagator is linked to the database.
- Changing from Run mode parameter V (Vista) to I (isolated) or L (local) is not possible if:
  - files with Vista elements are linked to the database, or

- Vista elements for the database exist, or
- files with phys. distribution type P (partitioned) are linked to the database.

Changing the Database Number

The database number can be changed in the **Physical database number** text box of the **Database Attributes** tab.

The following rule applies:

If files contained in the database are connected to implemented files, the database number (DBnr) cannot be changed.

Changing the Parameter Vista Access Only

Vista access can be changed on the **Database Attributes** tab.

The Vista Access Only flag indicates whether Adabas Vista is required to access files in a database. When creating a database object, Predict sets this flag to Y or N (according to the attributes of the database). The following rule applies:

Setting the flag from N to Y is only possible for databases of type V (Vista). Vista elements for files will be created if they do not already exist.

Messages

If prerequisites for renaming/renumbering databases are not met, one of the following messages is issued.

Implemented file exists in the database
: If files contained in the database are connected to implemented files, the database number (DBnr) cannot be changed.

File with physical distribution type 'partitioned' found / File with Vista element found / Database Vista element found
: At least one file defined for use with Vista (phys. distribution type or Vista element is specified) is linked to the database. The Run mode parameter therefore must be V (Vista).

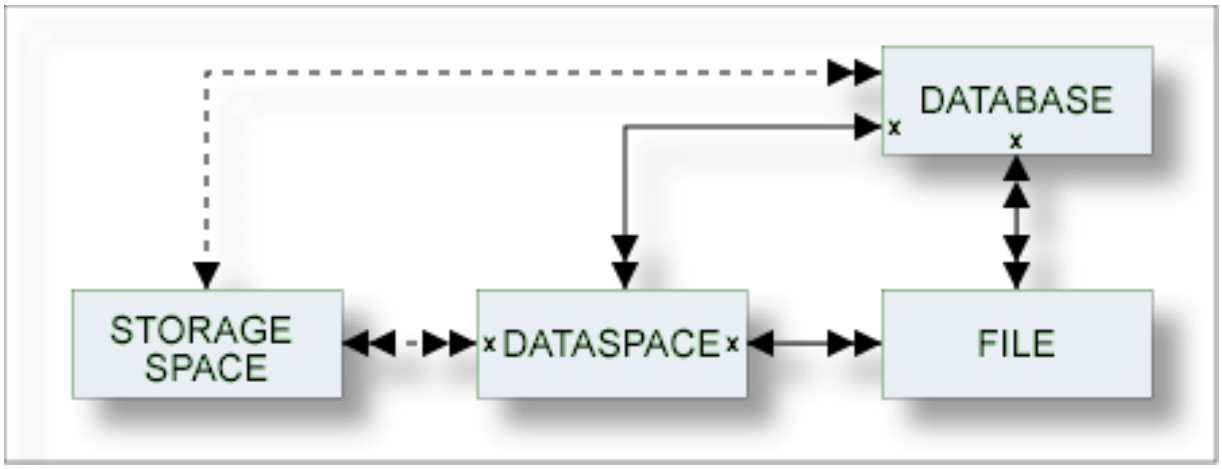
File with PROPAGATOR type 'master' or 'replicated' found
: At least one replicated or master file for ENTIRE TRANSACTION PROPAGATOR has been found. The Run mode parameter can therefore not be changed to V (Vista).

Different File types in the database
: Files of different types are linked to the database. Database therefore must be of type C (conceptual).

III Dataspace

DB2 table spaces or SQL/DS DBspaces are documented with objects of type Dataspace.

Note: DB2 storagegroups are documented with objects of type **Storagespace**.



The description of object type Dataspace is organized under the following headings:

Maintaining Objects of Type Dataspace

Dataspace-Specific Maintenance

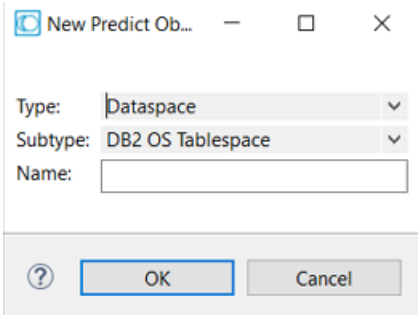
7 Maintaining Objects of Type Dataspace

- Adding a Dataspace 44
- Defining Basic Attributes of Dataspace - DB2 Mainframe 44
- Defining Basic Attributes of Dataspace - SQL/DS 50
- Defining Basic Attributes of Dataspace - DB2 Open Systems 51

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Adding a Dataspace

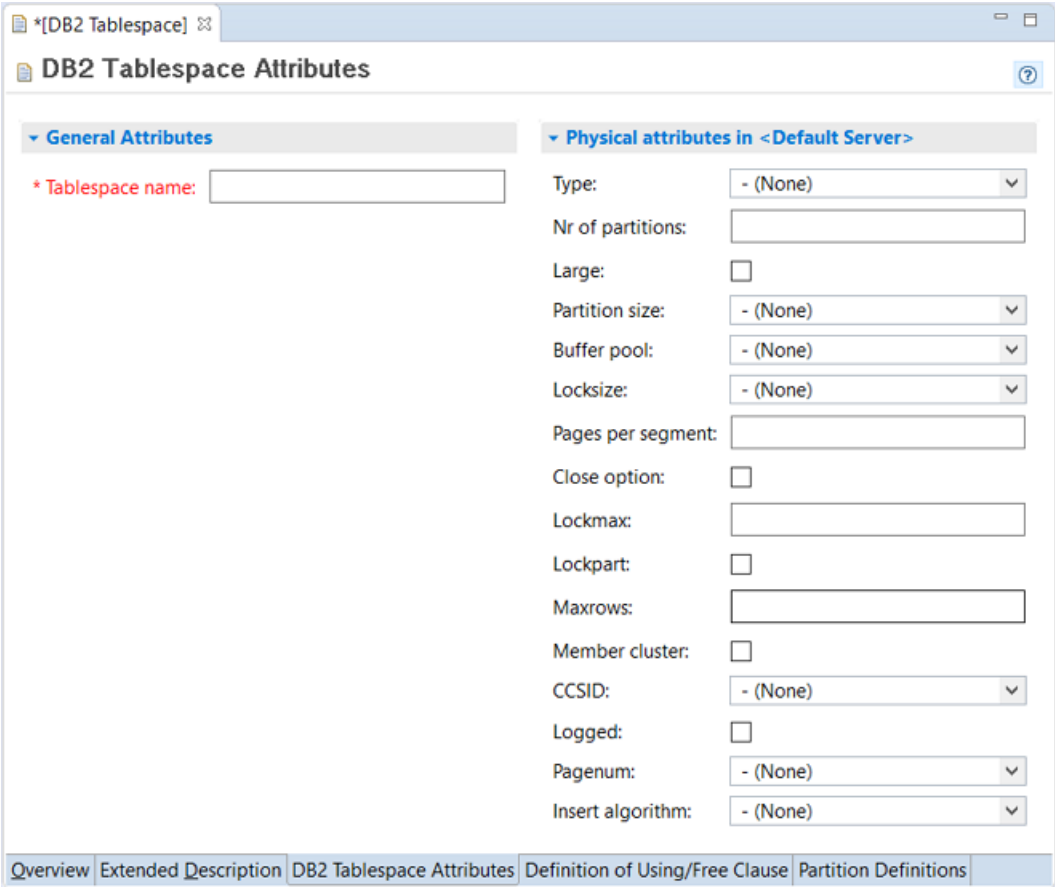
When you add a dataspace, you first have to specify the dataspace type in the **New Predict Object** dialog box.



When you choose the **OK** button, a dataspace type-specific window appears. The dataspace type is indicated in the title bar.

Defining Basic Attributes of Dataspace - DB2 Mainframe

The following window applies to DB2 mainframe dataspaces.



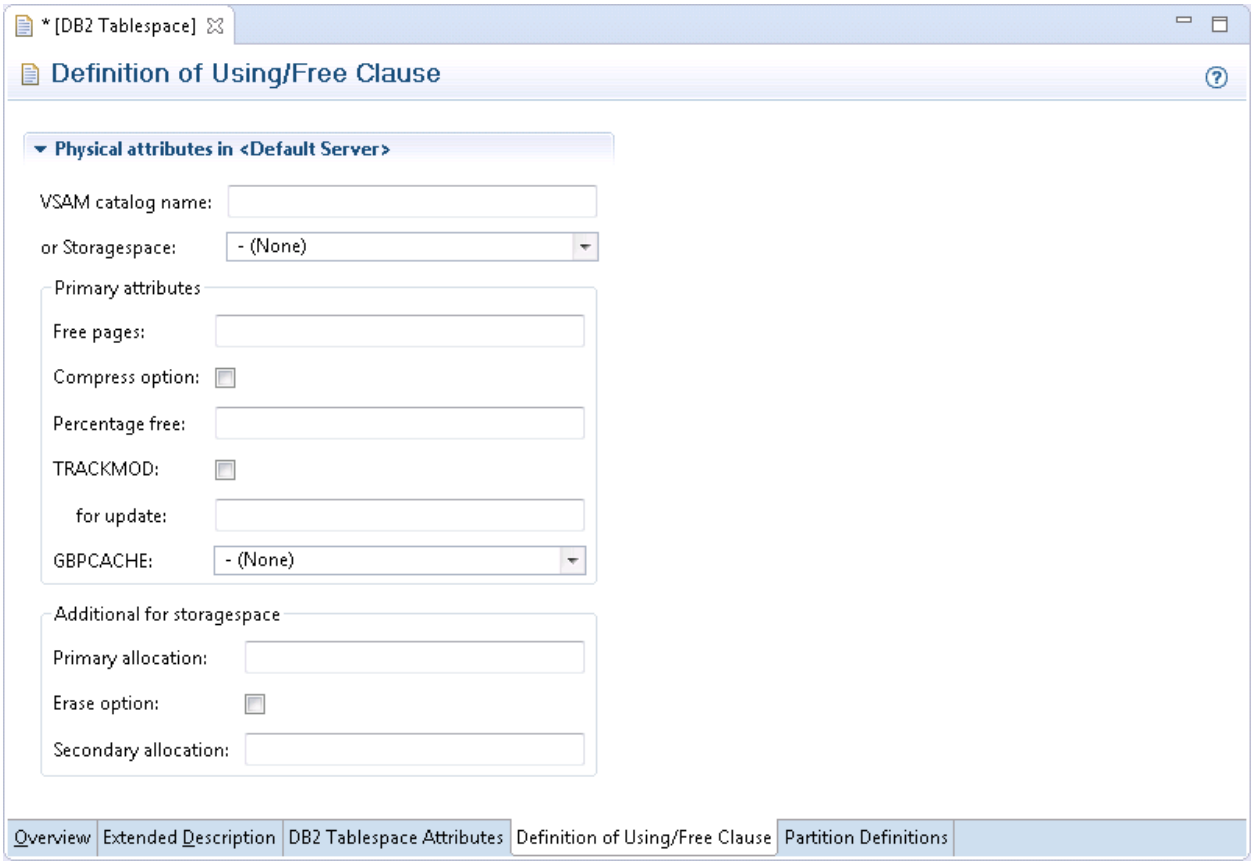
Parameters		
Tablespace name	Name of the table space in DB2.	
Type	Table space type.	
	S	Segmented.
	P	Partitioned.
	R	Range partitioned.
	G	Partition by growth.
	''	Simple. This is the default setting.
Nr of partitions	Number of partitions used by the table space (corresponding to the `NUMPARTS` parameter, max. 4096) for table space type P or R. If 0 is specified, the table space is not partitioned. Maximum number of partitions used by the table space (corresponding to the `MAXPARTITIONS` parameter, max. 4096) for table space type G. Partitions can be defined Partitions can be defined explicitly or with default values (see **Using/free clause** below) if table space type is either P or R. Partition definitions are used when generating table spaces from Predict dataspace objects.	

Parameters		
Large	Identifies a table space as large.	
	Y	Yes.
	N	No.
Buffer pool	Name of the buffer pool to be associated with the table space.	
Partition size	Only valid for partitioned tablespaces.	
Locksize	Locking level for the table space. Valid values:	
	A	Any level locking.
	P	Page level locking.
	R	Row level locking.
	S	Table space level locking.
	T	Table level locking (only valid for segmented DS).
Pages per segment	How many pages are to be assigned to each segment (parameter SEGSIZE) for table space type S, G or R. Pages per segment must be zero for table space type ' ' (Simple) or P.	
Logged	blank	Not specified. This is the default setting.
	Y	Yes.
	N	No.
Pagenum	Specifies the type of page numbering that is used when you alter a partition-by-range table space.	
	' '	None. This is the default setting.
	R	Relative.
	A	Absolute.
Insert algorithm	Specifies the algorithm that is used when rows are inserted into tables in this table space.	
	' '	None. This is the default setting.
	1	Level 1.
	2	Level 2.
Close option	Y	The data sets which support the table space are closed when nobody is using the table space.
Lockmax	The maximum number of pages or row locks an application can hold simultaneously in the table space. Valid values:	
	blank	
	SYSTEM	
	0 - 2,147,483,647.	
	If parameter **Locksize** is set to S or T, Lockmax must be set to 0.	
Lockpart	Partition locking. Valid values:	
	blank	not specified.
	Y	Yes.

Parameters		
	N	No.
Maxrows	The maximum number of rows.	
CCSID	Encoding scheme. Valid values:	
	blank	not specified.
	A	ASCII.
	E	EBCDIC.
	U	Unicode.
Member cluster	Specifies that data inserted by the INSERT statement is not clustered by the implicit clustering index (the first index) or the explicit clustering index. Instead, DB2 chooses where to locate the data in the table space based on available space. Valid values:	
	blank	not specified.
	Y	Yes.
	N	No.

Definition of Using/Free Clause

The values specified in the Definition of using/free clause tab are used as default values.

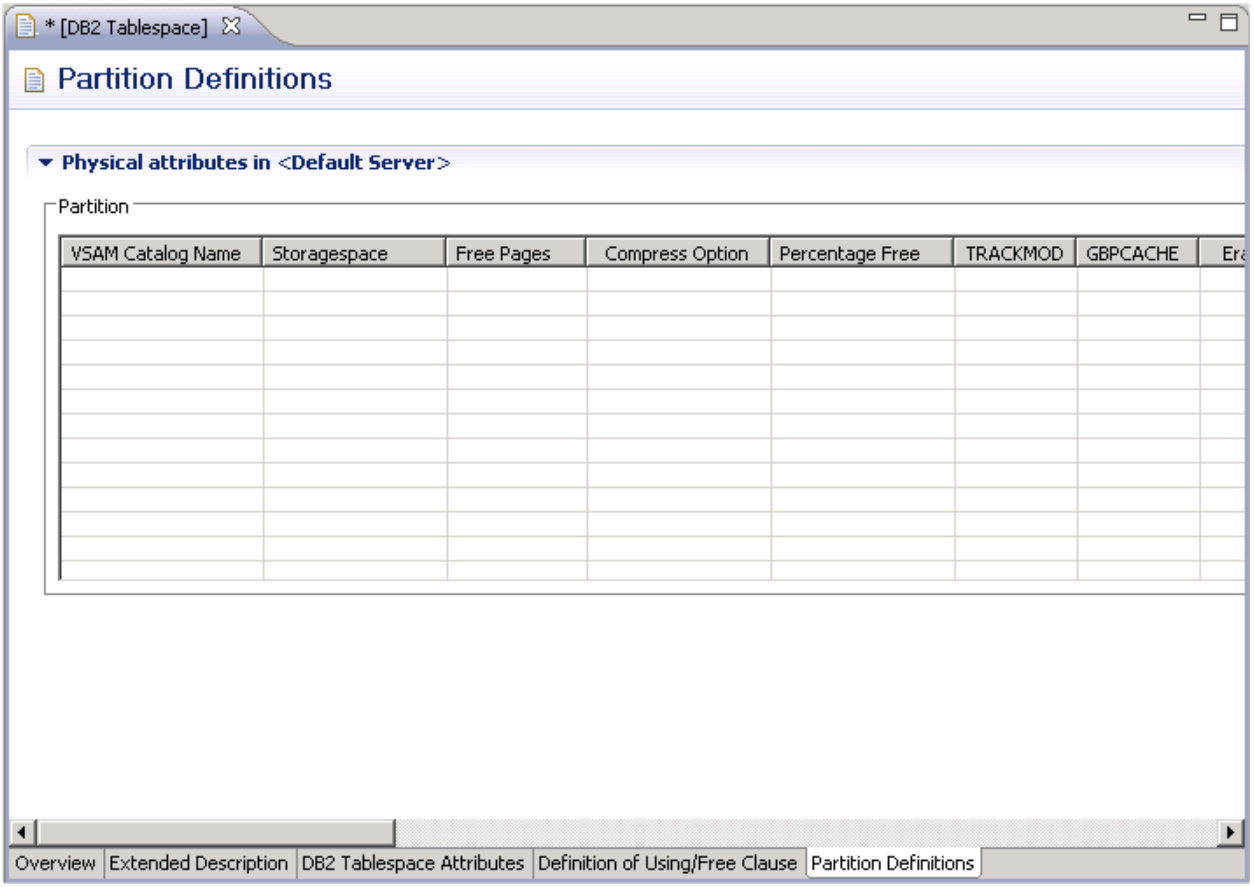


Parameters	
VSAM catalog name	Name of the VSAM catalog containing an entry for the data sets of the table space. Must not be specified if the parameter Storagespace is specified.
Storagespace	Name of the storagespace for the table space documented with the Predict Dataspace object. Must not be specified if the parameter VSAM catalog is specified.
Primary attributes	
Free pages	How often pages are to be left free when loading or reorganizing table spaces or partitions. Max. value is 255. Default is 0, leaving no free pages.
Percentage free	Percentage of each page to be left free. Equivalent in DB2 is the PCTFREE option.
for update	Percentage of each page that is reserved to be used only by future update operations. Equivalent in DB2 is the PCTFREE FOR UPDATE option. When you specify -1, DB2 uses real-time statistics to automatically calculate how much free space to reserve for updates. This is the default.
Compress option	Specifies whether data compression applies to the rows of the implicitly created tablespace.
	blank: Not specified. This is the default setting.
	N: No.
	Y: Yes.
	F: FIXEDLENGTH.
	H: HUFFMAN.
GBPCACHE	Only relevant in a data sharing environment. Specifies what pages of the table space or partition are written to the group buffer pool. Leave this field blank or enter:
	C: Changed. Only pages that have been changed are written to the group buffer pool.
	A: All pages are written.
	N: No pages are written to the group buffer pool.
TRACKMOD	Specifies whether DB2 tracks changed pages in the space map pages.
	Y: Changed pages are tracked in the space map pages.
	N: Changed pages are not tracked.
Additional for storagespace	
Primary allocation	Primary space allocation for DB2 defined data sets.
Secondary allocation	Secondary space allocation for DB2 defined data sets.
Erase option	Determines if DB2 defined data sets are to be erased when the table space is dropped:
	N: Do not erase data sets (default).
	Y: Erase data sets.

Defining Partitions

Each individual partition can be defined in the Definition of partitioned dataspace tab. The maximum number of partitions is 4096.

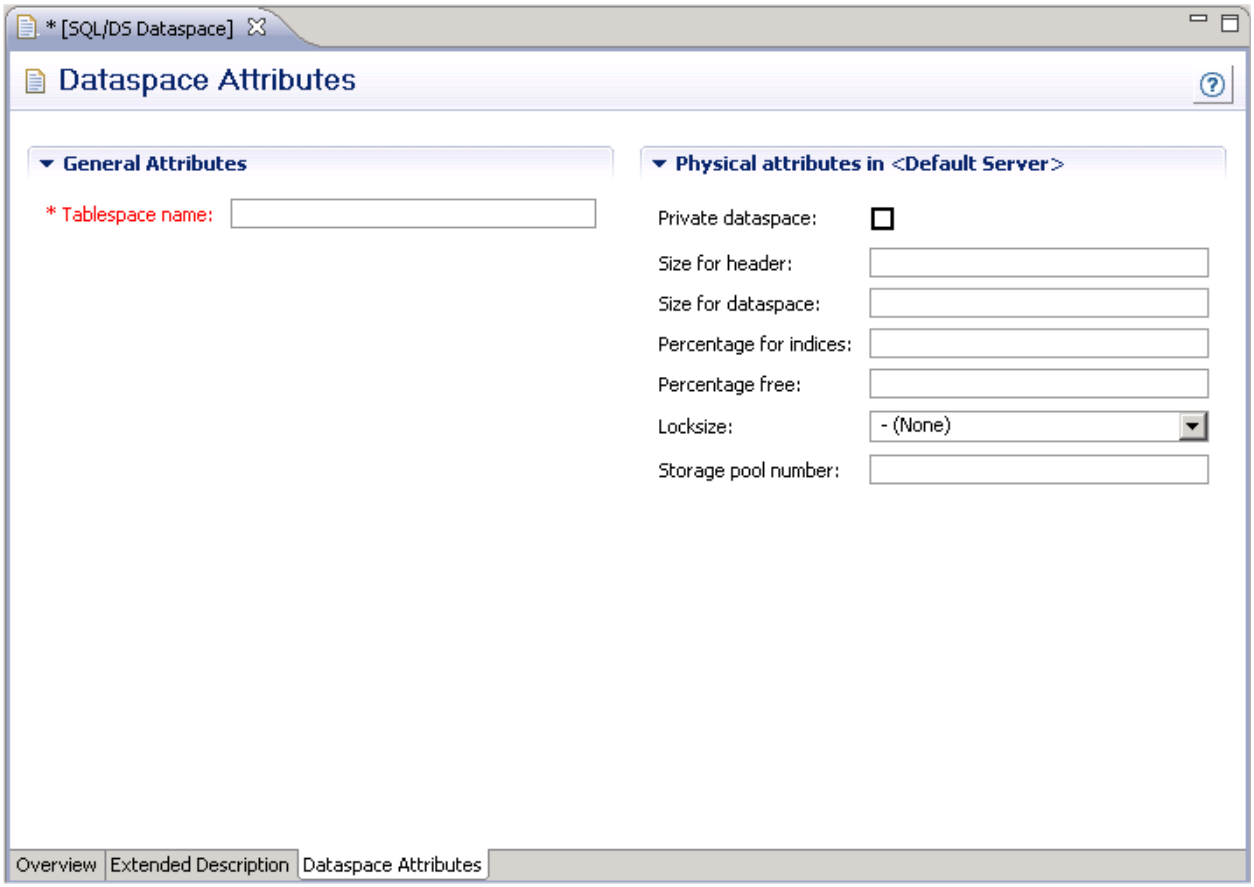
To define a new partition, use the **New** button. When you have defined more than one partition, a scroll bar is available. To go to a specific partition, use the scroll bar. To delete the currently displayed partition, choose the **Delete** button.



Parameters	
DSSIZE	Specifies the maximum size (in Gbyte) for each partition of a partitioned index.
For all other values/options see previous table above for a description of the parameters.	

Defining Basic Attributes of Dataspace - SQL/DS

The following window applies to SQL/DS dataspaces.

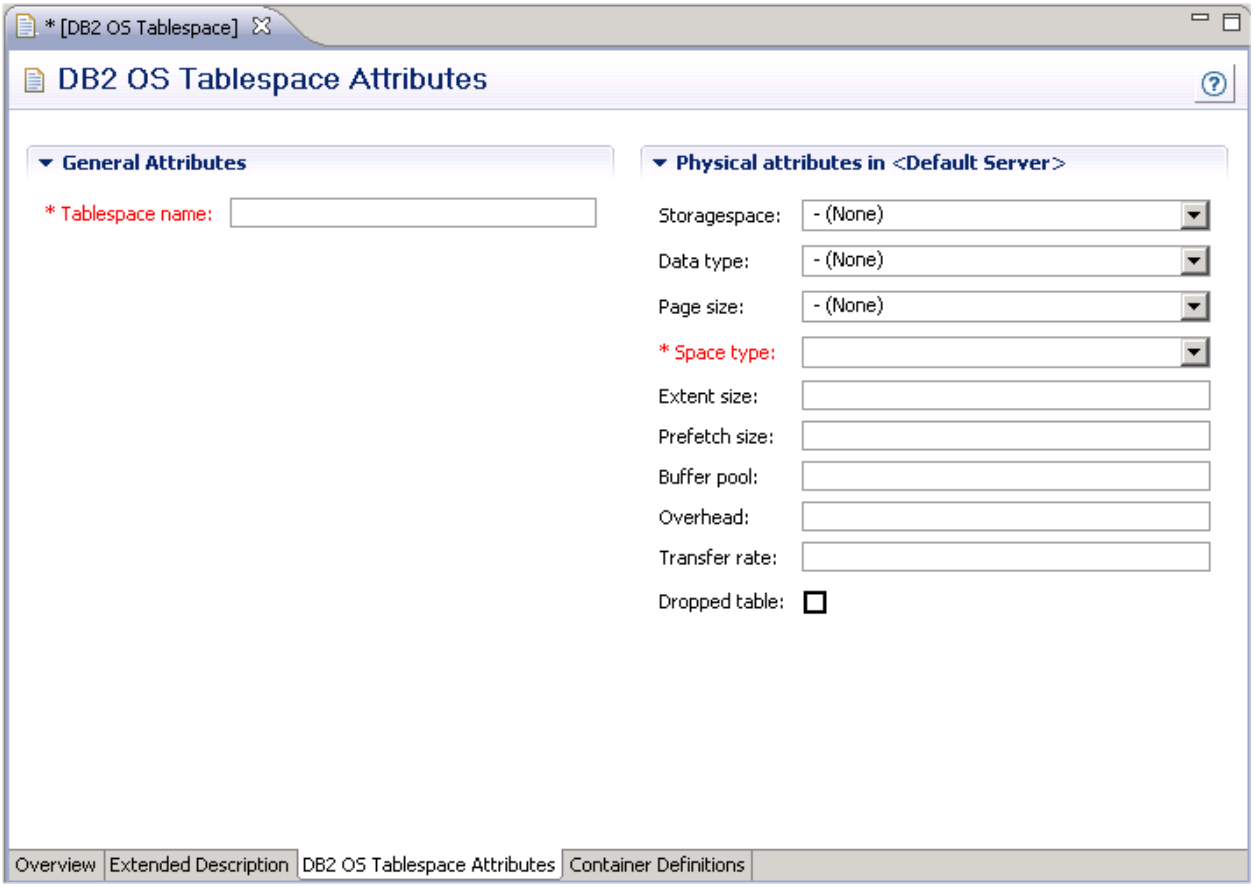


Parameters	
Tablespace name	Identifier of the table space and name of the DBspace in SQL/DS.
Private dataspace	Y: SQL/DS DBspace is private. N: Dataspace is public.
Size for header	Number of 4096-byte logical pages reserved for header.
Size for dataspace	Size reserved for the dataspace.
Percentage for indices	Percentage of the reserved space that can be used for indexes.
Percentage free	Percentage of reserved space to be kept free.
Locksize	Locking level for the dataspace. Valid values: A: Any level locking P: page

Parameters		
	S	dbspace
	R	row
	T	table
Storage pool number	Storage pool number. This parameter tells SQL/DS to acquire the dbspace from a specified storage pool.	

Defining Basic Attributes of Dataspace - DB2 Open Systems

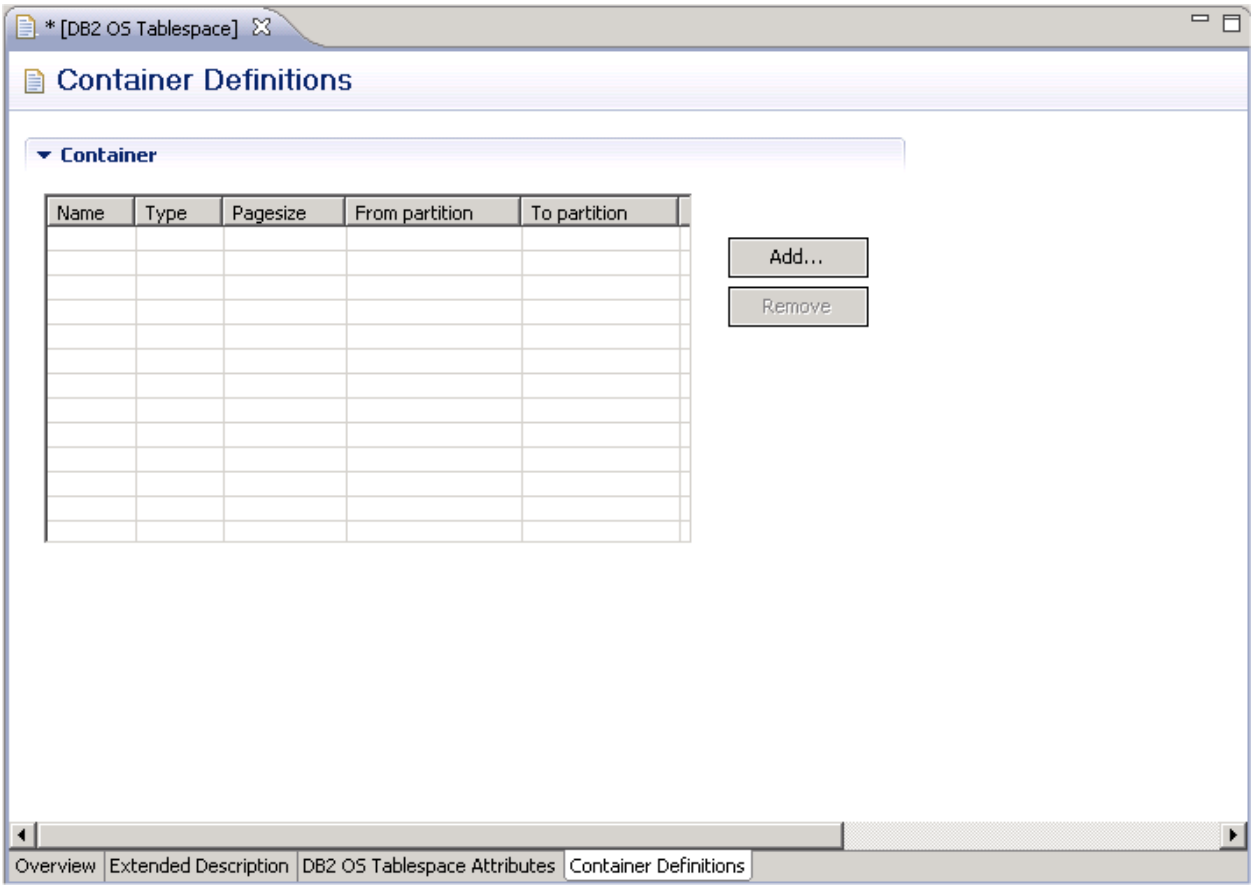
The following window applies to DB2 open systems dataspaces.



Parameters	
Tablespace name	Name of the tablespace in DB2.
Storagespace	Name of the storagespace for the tablespace documented with the Predict dataspace object.
Data type	Valid data types: R – Regular L – Large U – User temporary S – System temporary blank – not specified
Page size	Defines the size of pages used for the tablespace. Valid values: 4K, 8K, 16K, 32K, 4096, 8192, 16384, 32768 or not specified.
Space type	Specifies how the tablespace is to be managed: S – System managed D – Database managed
Extent size	Specifies the number of PAGESIZE pages that will be written to a container before skipping to the next container. Valid values are 0 or integer or integer with unit K, M or G.
Prefetch size	Specifies the number of PAGESIZE pages that will be written to a container before skipping to the next container. Valid values are 0 or integer or integer with unit K, M or G.
Buffer pool	Name of the buffer pool to be associated with the tablespace.
Overhead	Specifies the I/O controller overhead and disk seek and latency time (in milliseconds).
Transfer rate	Specifies the time to read one page into memory (in milliseconds).
Dropped table	Specifies if dropped tables in the tablespace may be recovered. Valid values: Y – Yes N – No blank – Not specified

Container Definitions

The following window applies when defining containers.



Parameters	
Container *number* (1, 2, etc.)	Specifies the container for a tablespace. Choose **Add** to enter a new container. Use the scrollbar to select an existing container (only applicable if more than one container exists).
Partition numbers	Specifies the partition numbers on which the container is created in a partitioned database.

8 Dataspace-Specific Maintenance

- Purge Dataspace 56

When maintaining dataspaces, only standard maintenance functions are needed. However, specific rules apply when purging objects of type dataspace. These rules are described below.

Purge Dataspace

Predict objects of type Dataspace are purged with the **Delete** command.

The following objects are deleted:

- the dataspace object
- all links to child objects and from parent objects
- the connection from the dataspace to the DB2 database is undone. All DB2 tables contained in this dataspace are removed from the file list of the corresponding DB2 database object.

IV Extract

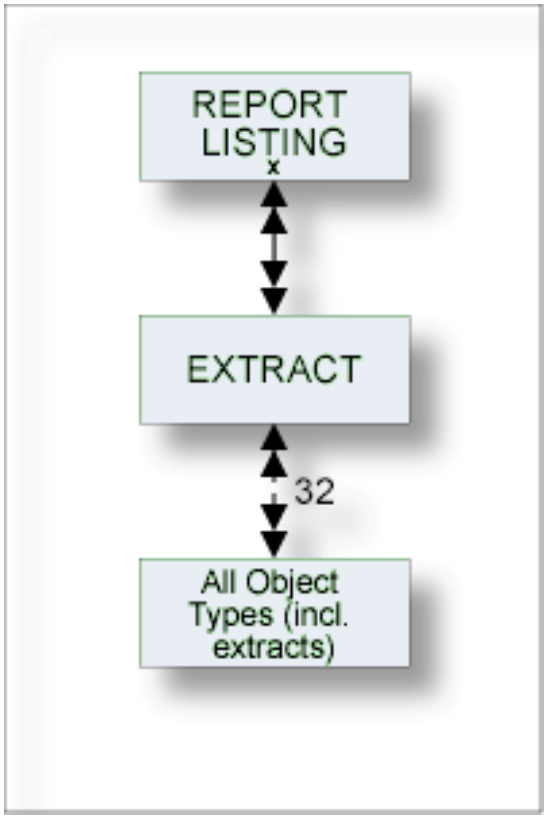
An object of type Extract in Predict fulfills two functions:

- to group objects logically
- to determine the objects to be transferred with the Predict Coordinator.

An object can be contained in a maximum of 32 extracts. The number of objects in an extract is virtually unlimited. An extract can contain other extracts - including itself.

Extracts #SAG-TRANSFER and #SAG-ERROR are created automatically by the Coordinator. See the Predict Coordinator documentation.

When you transfer objects with the Predict Coordinator, a report listing is created automatically and the extract containing the objects to be transferred is linked as a child object to this report listing. See the Predict Coordinator documentation.



The description of object type Extract is organized under the following headings:

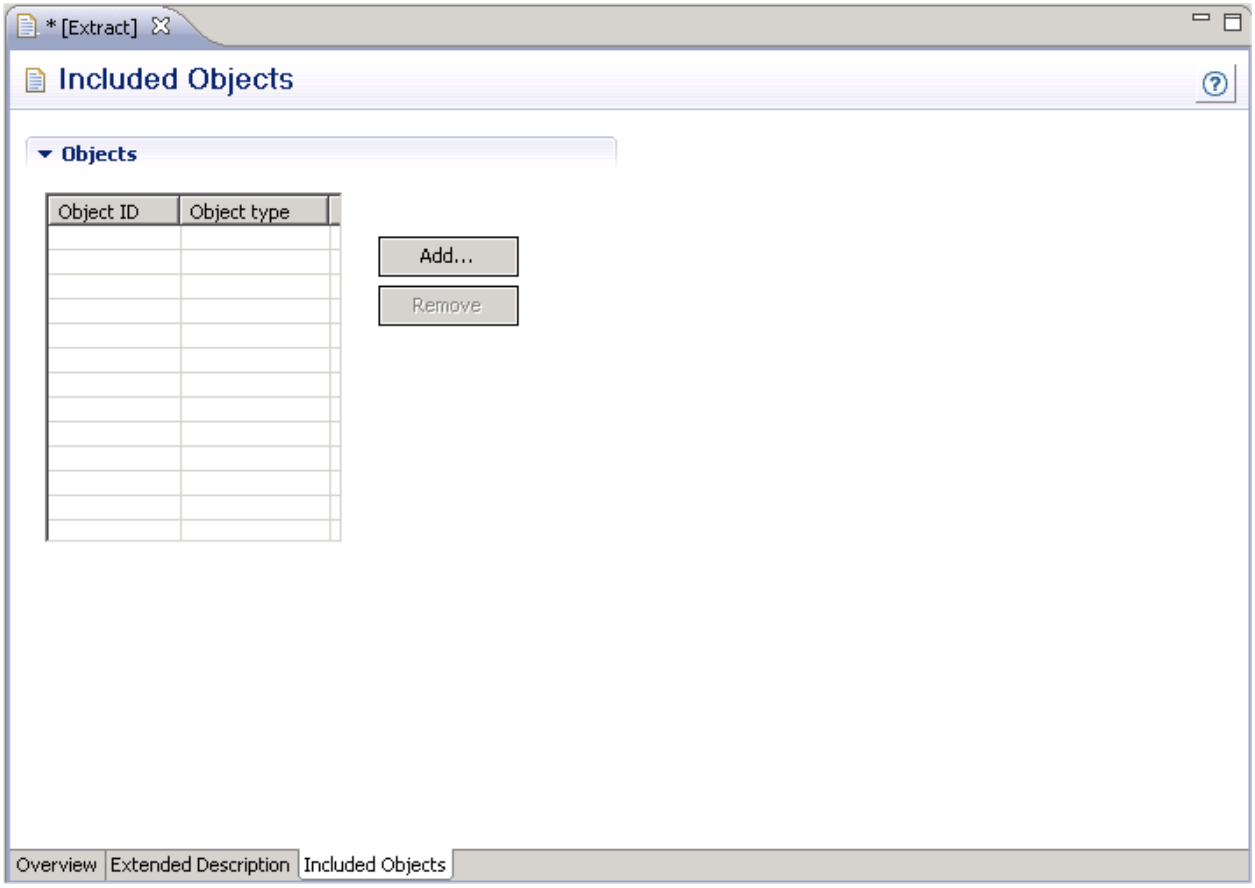
Maintaining Objects of Type Extract

9 Maintaining Objects of Type Extract

- Defining Basic Attributes of Extract 60

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

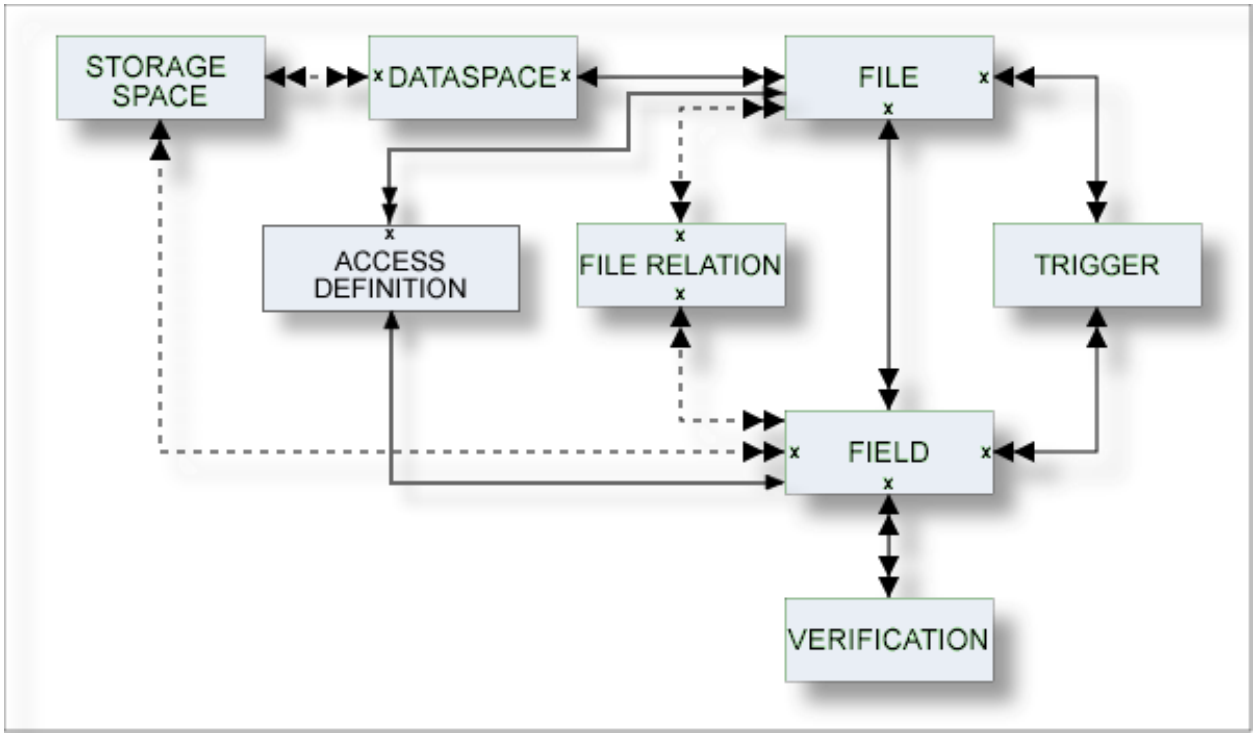
Defining Basic Attributes of Extract



Command Button	Description
Add	This creates a new row in the table. You have to specify an object ID and select an object type from a drop-down list box.
Remove	Delete the selected row.

Note: For parameters not listed here, see ***Global Attributes***.

V Field



The description of object type Field is organized under the following headings:

Defining Basic Attributes of Fields

Defining Derived Fields

Defining Additional Attributes of Fields

10 Defining Basic Attributes of Fields

- Field List Tab 64
- Field Type 65
- Level Number 66
- Field Format 67
- Character Set 68
- Character Set - Adabas 69
- Field Length 69
- Descriptor Type 78
- Descriptor Type - continued 79
- Maximum Number of Values / Occurrences 80
- Unique Option 81
- Field Short Name 81
- Suppression / Null Value Option 82
- Variable Length Option - IMS 83
- Null Default Option 83
- Natural Field Length 83
- Do Not Convert Option 84
- Related Standard File 84
- Check against standard 84
- Natural Attributes 84
- Hidden 86

Most attributes are applicable to fields of all file types.

Basic attributes applying to different field types are described below. Type-specific attributes are described in the section *Defining Derived Fields*.

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Field List Tab

Unlike all other predefined object types, a node for the object type Field is not shown at the top level of the tree view window containing the list of all documentation objects.

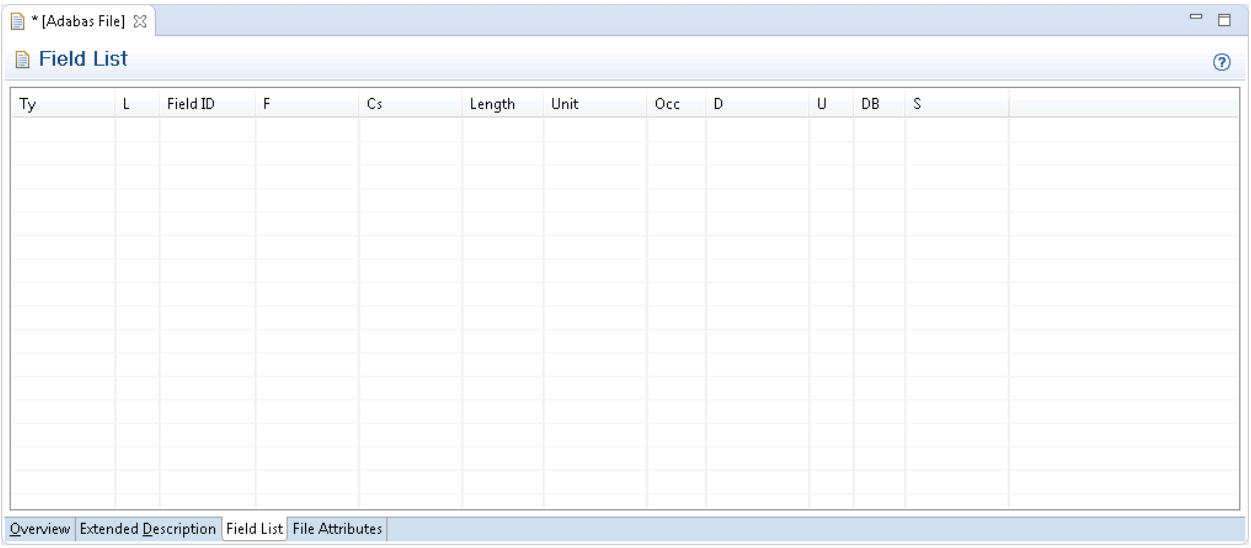
Fields can only be added via the **Field List** tab of the object type File.

Fields can be modified in two ways:

- via the **Field List** tab of the object type File, and
- via the dialog box which appears when you open a field (see *Defining Additional Attributes of Fields*).

Note: Field type, level number, field format, field length and the maximum number of values/occurrences can only be modified on the **Field List** tab since these attributes influence the record structure of a file.

The following information is displayed for fields:



Not all columns of the table on the **Field List** tab can be shown at the same time. Therefore, a horizontal scroll bar is provided. The columns are described below.

The following commands are available (in addition to the standard commands):

Command	Description
Insert	Add a new row. You can also use the Ins and Del keys here.
Delete	Delete the selected row.
Import	Invoke the **Find** dialog box. Specify all search criteria in this dialog box. When you choose the **Start Find** button, all found fields are automatically copied to the end of the list
Source	Only available if derived fields can be specified. Invokes a new window in which you can define derived fields. See *Defining Derived Fields*.
Standard	Invokes a new window in which you can couple the selected field and a standard field.

In addition, you can:

- Switch from the field list editor to the line editor with the F2 and ESC keys.
- Select all field entries and cut, copy and paste field entries using usual shortcut keys.
- Select/unselect multiple entries in any combination using the mouse and cut, copy and paste field entries using the usual shortcut keys.
- Drag and drop individual or multiple selected fields from one object to another.

Field Type

The field type is indicated in the column Ty of the **Field List** tab. The following types can be specified:

CM	Counter Field for multiple value field of type MU/MC
CP	Counter Field for periodic group of type PE/PC
DV	Derived field (SQL File types) see note below
GR	Group
HM	Hyperdescriptor as a multiple value field
HP	Hyperdescriptor as a field of a periodic group
HQ	Hyperdescriptor as a multiple value field of a periodic group
HY	Hyperdescriptor
MC	Multiple value field with automatic counter
MD	Indicator for union view. Can be set in the subquery editor only. MD indicates from which fields of the selected master file(s) the union, except or intersect view is created. See *Editing the Subquery of an SQL View*.
MU	Multiple value field
OD	Collation descriptor

PC	Periodic group with automatic counter
PE	Periodic group
PH	Phonetic descriptor
QN	SEQNO field
SB	Subfield/descriptor
SP	Superfield/descriptor
**,/*	Comment line
blank	None of the above. Normal field

Note: Derived field is also used in Predict as a generic term for hyperdescriptors, phonetic descriptors and sub/superfields and descriptors.

If HM, HP, HQ, HY, OD, PH, SB or SP is specified, an additional window can be invoked using the **Source** button. See *Defining Derived Fields*.

Defining Periodic Groups in Periodic Groups

- Within a redefinition, nested periodic groups (PE within a PE) can be defined in files of all types.
- Outside of a redefinition, nested periodic groups can only be defined in files of the following types:

S	Sequential file
C	Conceptual file
M	ISAM file
Z	Standard file
O	Other file

Level Number

The level number of the field is indicated in the column L of the **Field List** tab. The level number is used to define a group structure. Level numbers 1 to 9 can be used (except for Adabas files, see below).

- The level number must be increment by 1 immediately following a field of type RE, PE, PC or GR.
- For redefinitions, the level number must be at least one greater than the level number of the field being redefined.

Adabas Files

The following rules apply to level numbers for Adabas files:

- The PE/PC-groups, sub/superfields/descriptors, hyperdescriptors and phonetic descriptors must be at level 1.
- Level numbers of fields outside a redefinition must be in the range 1 - 7.

Field Format

The format of the field is indicated in the column F of the **Field List** tab. One of the following values can be specified (depending on the file type):

A	Alphanumeric	IV	Interval
AL	Long varchar	L	Logical
AV	Varchar	LO	Large object
B	Binary/char for bit data	LX	Bfile
BL	Long varchar bit data	MO	Money
BT	Bit	MS	Smallmoney
BV	Varchar for bit data	N/U	Numeric unpacked
D	Date	NS/US	Numeric unpacked with sign
DS	Smalldatetime	OK	Object key
DT	Datetime	P	Packed numeric
F	Floating point	PS	Packed numeric with sign
FD	Decimal floating point	S	Serial
G	Graphic	T	Time
GL	Long vargraphic	TK	Table key
GV	Vargraphic	TS	Timestamp
I	Integer	blank	Undefined

See tables in the section *Field Length* for valid combinations of format and length.

The following rules apply:

- Any format/length combination is allowed for the file types C (conceptual) or Z (standard).
- For groups, this attribute must be blank.
- For sub/superfields/descriptors in Adabas files, the appropriate format is provided by Predict based on the formats of the fields used. See *Rules Applying to Format Changes*.

- The following formats are valid for fields within a redefinition: A, B, D, F, I, L, N/U, NS/US, P, PS, T.

Character Set

The parameter Character set determines the format in which data is stored. It is indicated in column Cs of the **Field List** tab. The possible values depend on the file type and format.

File Type	Format		Character Set						
			ASCII	EBCDIC	Bitdata	Single Byte	Double Byte	Mixed Data	XML
Adabas	A, AV, LO					Y		Y	
Adabas D	A, AL, AV		Y	Y	Y				
DB2	A, AL, AV				Y	Y		Y	
	LO				Y	Y	Y	Y	Y
Oracle	A, AL				Y				
	AV							Y	
	LO				Y			Y	
Informix	A, AV							Y	
	AL				Y				
Ingres	A,AV				Y				
	AL	*			Y				
Sybase	A, AV	*			Y	Y	Y		
	AL				Y				

Note: A value must be specified for field types and formats marked with an asterisk (*).

Character Set	
ASCII	Data is stored in ASCII format.
EBCDIC	Data is stored in EBCDIC format.
Bitdata	Data is stored in binary form, no conversion is performed.
Single Byte	Data is stored in single-byte format. Double-byte characters are not possible.
Double Byte	Data is stored in double-byte format. String comparisons function differently to single-byte data.
Mixed Data	Data is stored in single and double-bytes. Data is subject to DB2 rules for multiple-byte character sets.
XML	Data is stored in XML format.

Character Set - Adabas

The following character set is used to define wide character fields.

Adabas		Predict	
Format	Option	Format	Character Set
A		A	blank or single
W		A	Mixed
A	LA	AV	blank or single
W	LA	AV	Mixed
A	LB, L4	LO	blank or single
W	LB, L4	LO	Mixed

Note: Option L4 applies to Adabas on Open Systems Version 5 or above only.

Field Length

The field length is indicated in column Length of the **Field List** tab. This length is independent of its internal representation. When generating external objects, the field length is adjusted according to the internal representation of data used by the data storage system. For example: a field which is documented with length P9 is implemented with length P5 by the Adabas LOADER utility and the Adabas nucleus.

The following additional rules apply:

- For files of type C (conceptual) or Z (standard): Any format/length combination is allowed, and field length zero is permitted for all field formats.
- For groups and phonetic descriptors: Field length must be set to zero.
- For sub/superfields/descriptors in Adabas files: The appropriate length is provided by Predict based on the definition.
- For large object fields: A field length greater than 99999 byte can be defined by entering an additional length unit in the Unit column.

  Valid values:

blank bytes
K kilobytes
M megabytes
G gigabytes

for example, the term 96 implies a field length of 96 bytes while the term 96 M implies a field length of 96 megabytes.

Table of Field Formats and Lengths

The table on the following pages contains the valid format/length combinations for fields of the following file types:

Column	File Type
A / U	Adabas file / userview
A(SQL) / AT / B	Adabas file with SQL usage, Adabas cluster table, Adabas SQL view
BT / BV	Adabas D table / view
D / E / IT / IV / MT	DB2 table / view / Intermediate table / view / DB2 query table
F	rdb file
I / J / K	IMS segment / segment layout / userview
JT / JV	Ingres table / view
L / R / V / W	Logical VSAM file / view / Physical VSAM file / view
M	ISAM file
O	Other
OT / OV	Oracle table / view
P / Q	Entire System Server file / userview
S	Sequential file
T	RMS file
X	General SQL file
XT / XV	Informix table / view
YT / YV	Sybase table / view

Note: The tables do not contain the file types C (conceptual) and Z (standard). For these file types, any format/length combinations are allowed.

Key for the following table

no length	Format is valid; length must not be specified.
no restr.	No restrictions: any length may be specified.
p.q (m/n)	p — number of places before the decimal point q — number of places after the decimal point Where: 0 <= p <= m 0 <= q <= n 1 <= p+q <= m
n.m - n2.m2	Range of places before and after the decimal point. For example, fields of format MO for Sybase tables and views can have up to 15 places before the decimal point and up to 4 places after the decimal point (1.0 - 15.04).
*1	0 means 2GB
*2	0 means 4GB

Field Format	A, U	A(SQL) AT, B	BT, BV	D, E, IT, IV, MT	F	I, J, K	JT, JV	L, R, V, W	M
A	1-253	1-253	1-4000	1-254	1-253	1-253	1-2000	1-253	no restr.
AL			0-99999 *1	1-99999			0-99999 *1		
AV	1-16381	1-32767	1-4000	1-32767			1-2000		
B	1-126	1-126		1-255	1-126	1-126	1-2000	1-126	no restr.
BL							0-99999 *1		
BT									
BV				1- 32704			1-2000		
D	no length	no length	no length	no length	no length	no length		no length	no length
DS									
DT							no length		
F	4 / 8	4 / 8	4 / 8	4 / 8		4 / 8	4 / 8	4 / 8	4 / 8
FD				9 / 17					
G				1-127					
GL				1-16383					
GV				1-16383					
I	1 / 2 / 4 / 8	1 / 2 / 4 / 8	2 / 4	2 / 4 / 8	1 / 2 / 4 / 8	1 / 2 / 4 / 8	1 / 2 / 4	1 / 2 / 4 / 8	1 / 2 / 4 / 8
ID									

Field Format	A, U	A(SQL) AT, B	BT, BV	D, E, IT, IV, MT	F	I, J, K	JT, JV	L, R, V, W	M
IV									
IY									
L	no length		no length			no length		no length	no length
LO	up to 2 GB			up to 2 GB					
LX									
MO							no length		
MS									
N	p.q (29/29)	p.q (29/29)			p.q (29/29)	p.q (29/29)		p.q (29/29)	p.q (29/29)
NS	p.q (29/29)	p.q (29/29)	p.q (18/18)	p.q (31/31)	p.q (29/29)	p.q (29/29)		p.q (29/29)	p.q (29/29)
OK							no length		
P	p.q (29/29)	p.q (29/29)			p.q (29/29)	p.q (29/29)		p.q (29/29)	p.q (29/29)
PS	p.q (29/29)	p.q (29/29)	p.q (18/18)	p.q (31/31)	p.q (29/29)	p.q (29/29)		p.q (29/29)	p.q (29/29)
S									
T	no length	no length	no length	no length	no length	no length		no length	no length
TK							no length		
TS			no length	no length					
U	p.q (29/29)	p.q (29/29)			p.q (29/29)	p.q (29/29)		p.q (29/29)	p.q (29/29)
US	p.q (29/29)	p.q (29/29)	p.q (18/18)	p.q (31/31)	p.q (29/29)	p.q (29/29)		p.q (29/29)	p.q (29/29)

Field Format	O	OT, OV	P, Q	S	T	X	XT, XV	YT, YV
A	no restr.	1-2000	no restr.	no restr.	1-253	1-253	1-32762	1-255
AL		0-99999 *1					0-99999 *1	0-99999 *1
AV		1-2000					1-32762	1-255
B								
BL	no restr.		1-126	no restr.	1-126			1
BT								
BV								no length
D	no length		no length	no length	no length		no length	
DS								no length

Field Format	O	OT, OV	P, Q	S	T	X	XT, XV	YT, YV
DT		no length					no length	no length
F	4 / 8	4 / 8	4 / 8	4 / 8		4 / 8	4 / 8	4 / 8
FD								
G								
GL								
GV								
I	1 / 2 / 4 / 8	2 / 4	1 / 2 / 4 / 8	1 / 2 / 4 / 8	1 / 2 / 4 / 8	1 / 2 / 4 / 8	2 / 4	1 / 2 / 4
ID								
IV							7 / 17	
IY								
L	no length		no length	no length				
LO		0-99999 *2						
LX		0-99999 *2						
MO							p.q (32767/99)	1.0 - 15.04
MS								1.0 - 6.04
N	p.q (29/29)		p.q (29/29)	p.q (29/29)	p.q (29/29)	p.q (32/32)		p.q (38/38)
NS	p.q (29/29)	p.q (29/29)	p.q (29/29)	p.q (29/29)	p.q (29/29)		p.q (32/32)	
OK								
P	p.q (29/29)		p.q (29/29)	p.q (29/29)	p.q (29/29)	p.q (32/32)		
PS	p.q (29/29)	p.q (29/29)	p.q (29/29)	p.q (29/29)	p.q (29/29)		p.q (32/32)	
S							no length	
T	no length		no length	no length	no length			
TK								
TS		no length						no length
U	p.q (29/29)		p.q (29/29)	p.q (29/29)	p.q (29/29)	p.q (32/32)		
US	p.q (29/29)	p.q (29/29)	p.q (29/29)	p.q (29/29)	p.q (29/29)		p.q (32/32)	

Field Format	1	2
A	1-253	1-253
AL		
AV		
B		
BL	1-126	1-126

Field Format	1	2
BT		
BV		
D	no length	no length
DS		
DT		
F	4 / 8	4 / 8
FD		
G		
GL		
GV		
I	1 / 2 / 4 / 8	1 / 2 / 4 / 8
ID		
IV		
IY		
L	no length	no length
LO		
LX		
MO		
MS		
N	p.q (29/29)	p.q (29/29)
NS	p.q (29/29)	p.q (29/29)
OK		
P	p.q (29/29)	p.q (29/29)
PS	p.q (29/29)	p.q (29/29)
S		
T	no length	no length
TK		
TS		
U	p.q (29/29)	p.q (29/29)
US	p.q (29/29)	p.q (29/29)

Note: For format LO the length can be specified in units of KB, MB or GB. Enter the desired value followed by the corresponding character (K for KB, M for MB and G for GB). For limitations refer to the above tables.

SQL: DBMS Format and Corresponding Predict Format

The table below indicates the DBMS format and the corresponding Predict format for fields in files of the following types:

BT, BV	Adabas D table/view
JT, JV	Ingres table/view
OT, OV	Oracle table/view
XT, XV	Informix table/view
YT, YV	Sybase table/view

Key for the following table

n	length	
p,q	p	total number of places
	q	number of places after the decimal point

File Type	DBMS Format	Predict Format	Character Set
BT, BV	BOOLEAN	L	
	CHAR(n)	A(n)	
	CHAR(n) ASCII	A(n)	ASCII
	CHAR(n) BYTE	A(n)	Bitdata
	CHAR(n) EBCDIC	A(n)	EBCDIC
	DATE	D	
	FIXED(p,q)	NU, US, or PS	
	FLOAT(15)	F4	
	FLOAT(18)	F8	
	INTEGER	I4	
	LONG	AL	
	LONG ASCII	AL	ASCII
	LONG BYTE	AL	Bitdata
	LONG EBCDIC	AL	EBCDIC
	SMALLINT	I2	
	TIME	T	
	TIMESTAMP	TS	
	VARCHAR(n)	AV(n)	
	VARCHAR(n) ASCII	AV(n)	ASCII

File Type	DBMS Format	Predict Format	Character Set
	VARCHAR(n) BYTE	AV(n)	Bitdata
	VARCHAR(n) EBCDIC	AV(n)	EBCDIC
JT, JV	BYTE(n)	B	
	BYTE VARYING	BV	
	C(n)	A(n)	
	CHAR(n)	A(n)	Bitdata
	DATE	DT	
	DECIMAL (p,q)	PS	
	DECIMAL (p,q)	NS	
	DOUBLE PRECISION	F8	
	LONG BYTE	BL	
	LONG VARCHAR	AL	Bitdata
	INTEGER	I4	
	INTEGER1	I1	
	MONEY	MO	
	OBJECT_KEY	OK	
	REAL	F4	
	SMALLINT	I2	
	TABLE_KEY	TK	
	TEXT(n)	AV(n)	
	VARCHAR(n)	AV(n)	Bitdata
OT, OV	BFILE	LX	
	BLOB	LO	Bitdata
	CHAR(n)	A(n)	
	CLOB	LO	
	DATE	DT	
	DECIMAL(p,q)	NS	
	DECIMAL(p,q)	PS	
	DOUBLE PRECISION	F8	
	INTEGER	I4	
	INTERVAL DAY	ID	
	INTERVAL YEAR	IY	
	LONG	AL	
	LONG RAW	AL	Bitdata
	NCLOB	LO	Mixed data

File Type	DBMS Format	Predict Format	Character Set
	NVARCHAR2(n)	AV(n)	Mixed data
	RAW(n)	A(n)	Bitdata
	REAL	F4	
	ROWID	A and type QN	
	SMALLINT	I2	
	TIMESTAMP	TS	
	VARCHAR2(n)	AV(n)	
XT, XV	BYTE	AL	Bitdata
	CHAR(n)	A(n)	
	DATE	D	
	DATETIME YEAR TO FRACTION(5)	DT	
	DECIMAL(p,q)	NS	
	DECIMAL(p,q)	PS	
	FLOAT	F8	
	INTEGER	I4	
	INTERVAL DAY TO FRACTION(5)	IV	
	MONEY	MO	
	NCHAR(n)	A(n)	Mixed data
	NVARCHAR(n)	AV(n)	Mixed data
	REAL	F4	
	SERIAL	S	
	SMALLINT	I2	
	TEXT	AL	
	VARCHAR(n)	AV(n)	
YT, YV	BINARY(N)	A(n)	Bitdata
	BIT	BT	
	CHAR(N)	A(n)	Single byte
	DATETIME	DT	
	FLOAT	F8	
	IMAGE	AL	Bitdata
	INT	I4	
	MONEY	MO	
	NCHAR(N)	A(n)	Double byte
	NUMERIC, DECIMAL (p,q)	NS	
	NUMERIC, DECIMAL (p,q)	PS	

File Type	DBMS Format	Predict Format	Character Set
	NVARCHAR(N)	AV(n)	Double byte
	REAL	F4	
	SMALLDATETIME	DS	
	SMALLINT	I2	
	SMALLMONEY	MS	
	TEXT	AL	
	TIMESTAMP	TS	
	TINYINT	I1 or B1	
	VARBINARY(N)	AV(n)	Bitdata
	VARCHAR(N)	AV(n)	Single byte

Descriptor Type

The descriptor type is indicated in column D of the **Field List** tab. The possible values are given in this and the following table.

Additional Information on Descriptor Fields

Descriptor fields can be of various types and control which fields can be used in search criteria (for example in Natural `FIND`, `READ` or `HISTOGRAM` statements). Natural programs can only read Adabas files using fields that are defined as a descriptor (usually type D). It should be part of the design to decide which fields are going to be descriptors, superdescriptors etc.

Adabas however, has a feature where a file can be read using any field - regardless of whether it is a descriptor or not. This is called a non-descriptor search. For large data volumes, this can be very slow. This is because there are no indexes built for non-descriptor fields and a physical scan of the data is required. Despite, if there are only a few records, and it is understood that a physical file scan is taking place, then it can be useful on some occasions.

By putting N into the descriptor type in a DDM, Natural programs are allowed to use this field as a non-descriptor search field. That way you can control in the DDM which fields are allowed to be searched on.

The various descriptor types are documented in more detail in *Columns of Field Attributes* in the section *Using the DDM Editor* of the *Natural Editors* documentation.

Code	Description	File Type										
		A, U	AT, B, A(SQL)	M	O	F	S	T	IV, D, E, MT	C	I, J, K	P, Q
D	Descriptor/Index	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y
	Disallow											
A	Alternate index									Y	Y	
N	Not inverted	Y	Y	Y	Y	Y	Y	Y	Y	Y		
	Search field										Y	
P	Primary Index		Y						Y	Y		
Q	Sequence									Y	Y	
E	Foreign key		Y						Y	Y		
F	Foreign index		Y						Y	Y		
	Force											
K	Common Key											
blank	No descriptor	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	None											

Descriptor Type - continued

Code	Description	File Type									
		Z	1	2	L, R, V, W	X	BT, BV	OT, OV	JT, JV	YT, YV	XT, XV
D	Descriptor/Index							Y			
	Disallow	Y									
A	Alternate index		Y	Y	Y						
N	Not inverted					Y	Y	Y	Y	Y	Y
	Search field										
P	Primary Index		Y	Y	Y	Y	Y	Y	Y	Y	Y
Q	Sequence										
E	Foreign key					Y	Y	Y	Y	Y	Y
F	Foreign index							Y			
	Force	Y									
K	Common Key									Y	
blank	No descriptor		Y	Y	Y	Y	Y	Y	Y	Y	Y
	None	Y									

The following rules apply:

- In an Adabas file, the descriptor must be D if type HM, HP, HQ, HY (all hyperdescriptors), OD (collation descriptor) or PH (phonetic descriptor) is specified.
- For a subdescriptor in an Adabas file, descriptor D and type SB (subfield) must be specified.
- For a superdescriptor in an Adabas file, descriptor D and type SP (superfield) must be specified.
- In a DB2 table, DB2 query table or Oracle table, if a key, partitioning key, cluster column or index (descriptor D, E, F or P) includes more than one field, the type SP (superfield) must be specified
- In a VSAM file or userview (type L, R, V or W), the descriptor must be either P or A if type SP (superfield) is specified.
- If A is specified for a field of a VSAM file (type L or V), an additional screen is displayed for entering the required definitions (see below).
- Descriptor type must be blank for fields within a redefinition.

Maximum Number of Values / Occurrences

Maximum number of values for a multiple value field or occurrences of a periodic group is indicated in the Occ column of the **Field List** tab. This parameter must be specified for multiple value fields and for periodic groups in a redefinition.

Field	Occurrences in range
Within a redefinition	1- 99999
Outside a redefinition	1 - 65535

When generating Copy Code, the value specified is used as the default for generating the specifications of MU/MC or PE/PC fields in a format buffer and/or record buffer.

When generating ADACMP/ADAFDU definitions, the Occ parameter is evaluated. If Occ is specified, the number of occurrences of each input data record is constant.

If Occ is not specified, the number of occurrences is taken from a counter field preceding a MU/MC or PE/PC field.

Note: For fields of type QN, the Occ column is used to identify either the table level or an individual occurrence of a multiple value field or periodic group.

Unique Option

The unique option is indicated in column U of the **Field List** tab. For groups, this attribute must be blank; for other fields, one of the following values can be specified:

U	Unique.
X	Used for unique descriptors in PE to exclude the occurrence (index) number from the definition of uniqueness.
blank	Not unique.

Unique option must be blank for fields within a redefinition.

Field Short Name

For file types listed below, the field short name is indicated in the column DB of the **Field List** tab. This two-character short name must be defined for the following file types:

A	Adabas file	L	Logical VSAM file
AT	Adabas cluster table	R	Logical VSAM view
I	IMS segment	U	Adabas userview
J	IMS segment layout	V	VSAM file (physical)
K	IMS userview	W	Physical VSAM view

A field short name must conform to the rules for coding Adabas field names.

Field short names for userviews of Adabas, IMS and VSAM files need not be unique.

For fields within a redefinition, parameter Field short name must be blank.

Field short names for SQL tables and views are maintained internally by Predict and cannot be modified by users.

Rotated fields of files of type A (with SQL usage), type AT and B have the same short name and are identified uniquely by an occurrence number (column Occ).

Suppression / Null Value Option

- For fields of Adabas files, the suppression option is indicated in column S of the **Field List** tab.
- For fields of SQL files, the null value option is indicated in column N of the **Field List** tab.

For groups and for fields within a redefinition, this attribute must be blank. For other fields, one of the following values can be specified:

F	Fixed length
N	Null value suppression
R	Not null
U	Null counted
blank	Normal suppression

Parameter	SQL File Types	Other File Types
Null value suppression		N
Fixed Length		F
Null allowed	U	U
Not null	R	R
Normal suppression		blank

Profile Parameter Automatic Null Value

With the profile parameter Automatic null value you can determine an automatic Suppression/Null Value option when fields are added in Predict. See *Customizing Predict with Profiles* in the section *Predict User Interface* in the *Introduction to Predict* documentation.

The value depends on the file type. See table below.

Dictionary View by default uses the profile SYSTEM.

Parameter	All SQL File Types except X	File Type X	Other File Types
Unique option = Unique, or Descriptor type = Primary, or Field format = serial	R	R	N
Others	U	blank	N

Note: SQL file types include files of type A with parameter Adabas SQL usage set to Y.

For DB2 fields with Unique option = unique, values R and U are possible.

Variable Length Option - IMS

The variable length option for IMS fields is indicated in column S of the **Field List** tab. The following values are valid:

Y	Variable length
blank	Fixed length

Null Default Option

The NULL default option for fields of SQL tables/views is indicated in the column Df of the **Field List** tab. Possible values:

N	No default
Y	With default
blank	none

For Ingres fields with format OK or TK, the following additional values are possible:

S	SYSTEM_MAINTAINED
T	not SYSTEM_MAINTAINED
U	with default SYSTEM_MAINTAINED
V	with default not SYSTEM_MAINTAINED
W	not default not SYSTEM_MAINTAINED

This parameter must be blank for fields within a redefinition.

Natural Field Length

The Natural field length is shown in the dialog box which appears when you open a field (see *Defining Additional Attributes of Fields*). You can define it on the **Base Extensions** tab. The following rules apply:

- The parameter has to be specified if the field can be:
  - alphanumeric and greater than 253
  - graphic and greater than 126

- numeric p.q (m/n) where p+q > 29 or q > 7.

  See table of valid formats and lengths in the section *Field Length*.
- The value specified here is the length that Natural can use for the field as defined in the DDM.

Do Not Convert Option

The "do not convert" option is allowed for A and AV format fields of the following file types:

- Adabas file/userview
- Conceptual file

This option is shown in the dialog box which appears when you open a field (see *Defining Additional Attributes of Fields*). You can define it on the **Base Extensions** tab.

Related Standard File

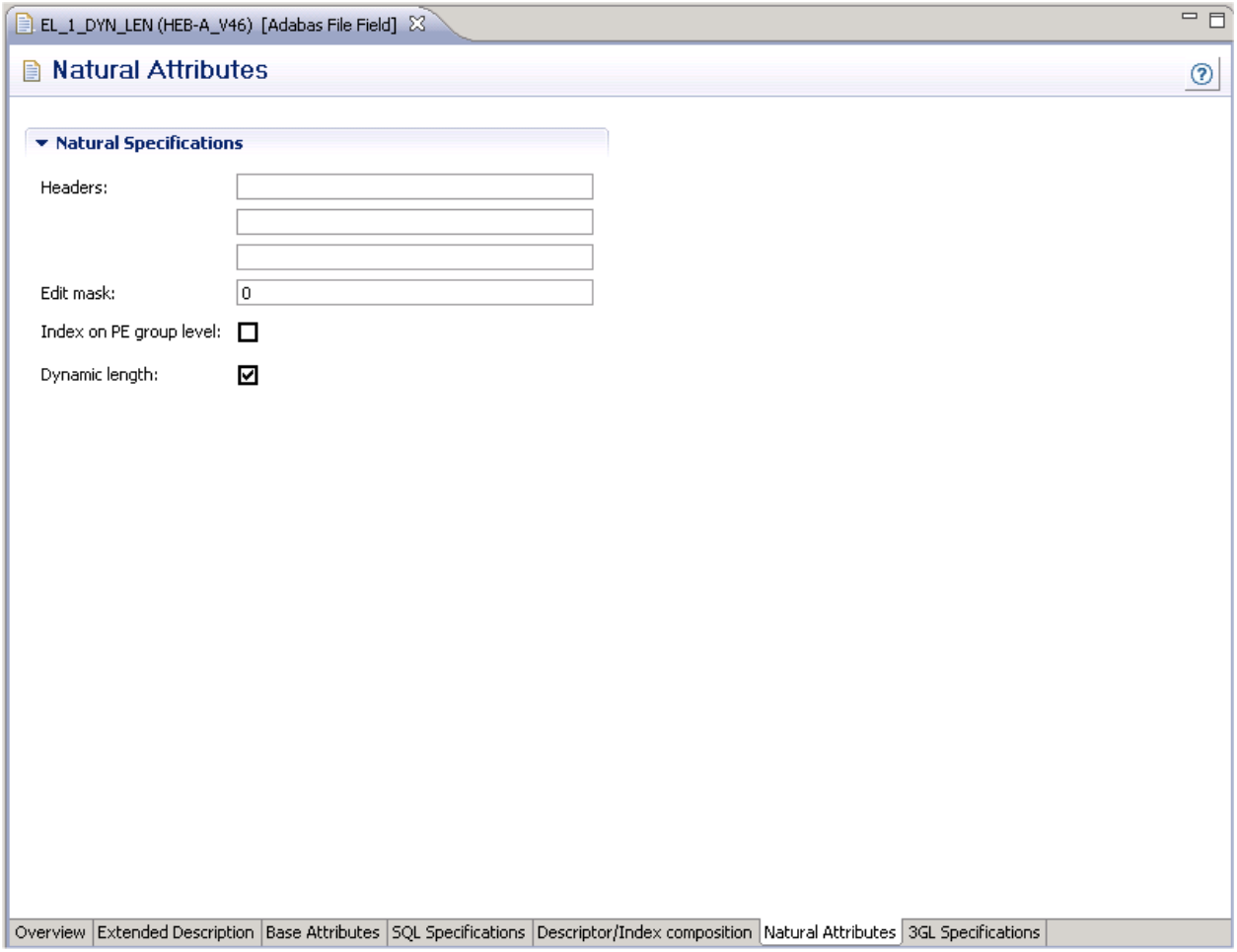
This parameter is described in the section *Rippling*.

Check against standard

This parameter is described in the section *Rippling*.

Natural Attributes

The following attributes can be defined in the **Natural Attributes** tab:



- Headers
- Index on PE Group Level
- Edit mask
- Dynamic length

Headers

The Natural headers are shown in the dialog box which appears when you open a field (see *Defining Additional Attributes of Fields*).

The Natural headers 1 - 3 are included in DDMs generated from the file containing the field.

Alphabetic characters in Natural headers are converted to upper-case if the Predict parameter Upper/lower case has been set to Y. See the section *Defaults* in the *Predict Administration* documentation.

Index on PE Group Level

This parameter is shown in the dialog box which appears when you open a field (see ***Defining Additional Attributes of Fields***).

If this parameter is left blank (default), the maximum occurrences is generated for each element in the group.

Edit mask

The Natural edit mask. See the description of the DISPLAY statement in the *Natural Statements* documentation for further details.

This parameter is shown in the dialog box which appears when you open a field (see ***Defining Additional Attributes of Fields***).

Alphabetic characters in the Natural edit mask are converted to upper-case if the Predict parameter Upper/lower case has been set to Y. See the section *Defaults* in the *Predict Administration* documentation.

Dynamic length

This parameter is shown in the dialog box which appears when you open a field (see ***Defining Additional Attributes of Fields***).

Hidden

Specifies that the column is not visible in the result for SQL statements unless you explicitly refer to the column by name.

If you check the box, columns are not visible.

EL_A (HEB-D_BASE_HIST) [DB2 Table Field] (Read Only)

Base Attributes

General Attributes

Type:	- (None)
Level:	1
Format:	A - Alphanumeric / Char
Length:	6.00
Occurrences:	0
Descriptor:	- (None)
Unique Option:	- (None)
Character Set:	- (None)
Null value:	U
Null default option:	- (None)
Natural length:	0.0
Hidden:	☐
Time zone:	☐
Fractional of seconds:	- (None)
Inline length:	0
Generated expression:	- (None)

Default Value

Default expression:	

Field Procedure

Physical attributes in <Default Server>

Procedure name:	
Procedure parameter:	

Constraint Name

Check constraint:	
Primary key:	
Unique:	
Not null:	

Field Name Synonyms

Natural:	
COBOL:	
PL/1:	
BAL/Assembler:	
FORTRAN:	
Pascal:	
Language ADA:	
Language C:	
User defined:	

DBMS Extensions

Physical attributes in <Default Server>

Use as security label:	☐
Partitioning:	- (None)
Use as hash key:	☐
Exclude NULL keys:	☐

Overview | Extended Descrip... | Base Attributes | Definition of Index | Defaults of Using... | Partition definit... | Identity/Change log | »3

11 Defining Derived Fields

- General Rules for Defining Derived Fields 90
- Defining Derived Fields of Special Types 93
- Rules Applying to Format Changes 96
- Rules Applying to Suppression/Length Changes 98
- Validation of Derived Field Definitions 98

Note: Derived field is a generic term in Predict for fields and descriptors defined on the basis of one or more source fields. This term should not be confused with field type DV applicable to SQL views (see *Field Type*).

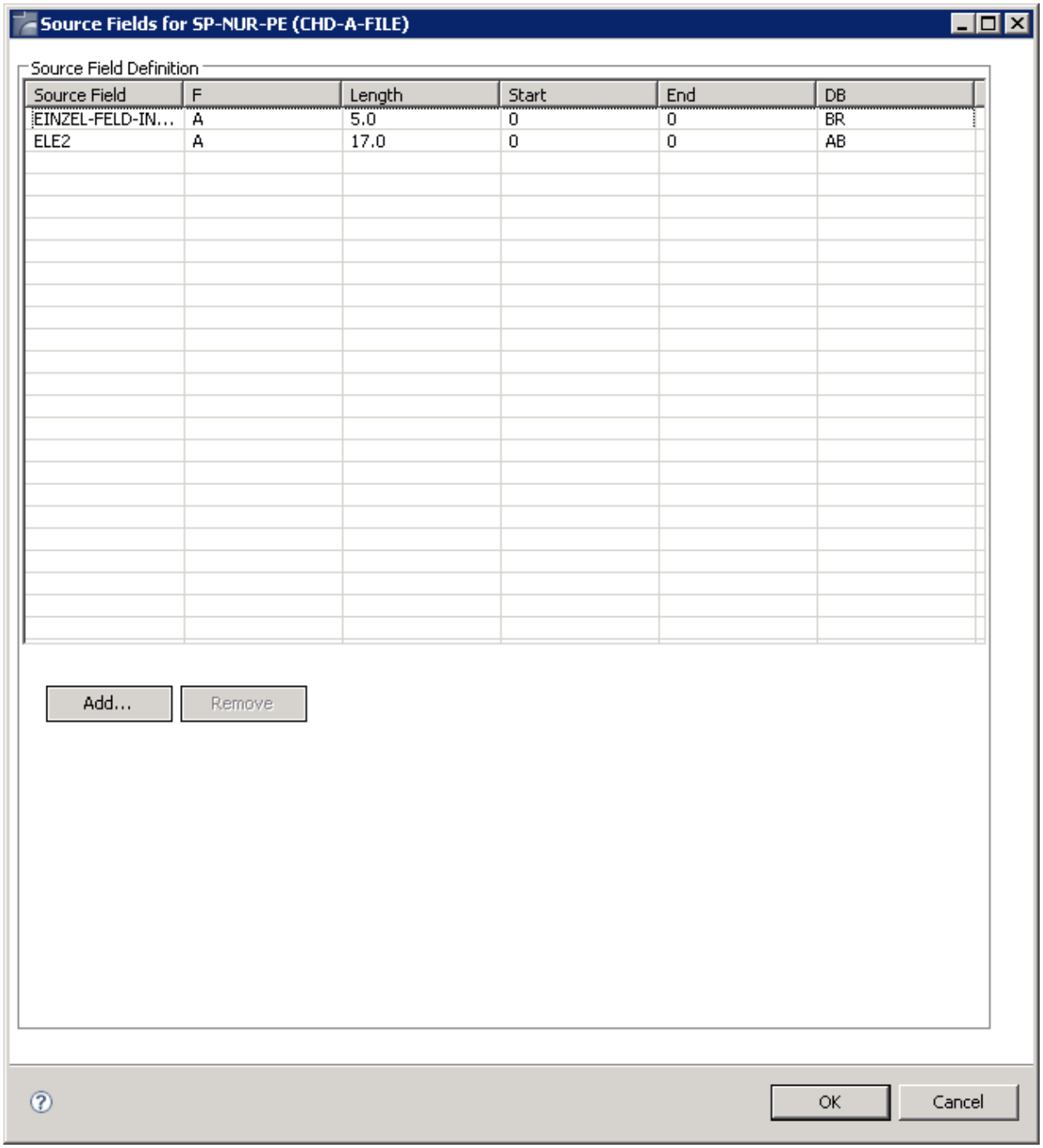
Defining derived fields and keeping the definitions consistent is a complex task. Predict offers a variety of functions to help with it.

General rules applying to the definition of derived fields are described in the following sections.

General Rules for Defining Derived Fields

To define derived fields, select the field on the **Field List** tab and choose the **Source** button. A new window appears, displaying a table in which you can define derived fields. The size and format of this table varies with the type of the derived field.

The window below only applies to files of type D or MT with format SP (superfield). Other SQL file types lack the Random option and have no Expression option either. Refer to **Key or Index Fields in SQL Files - Superfields** for details on these options.



The following commands are available (in addition to the standard commands):

Command	Description
Add	Create a new source field. The source field can be selected from a list of all fields contained in the file.
Remove	Delete the selected source field.

General Attributes of Definitions of Derived Fields

The following attributes are contained in most definitions of derived fields. Attributes specific to certain types of derived fields are described in the respective sections.

Attributes	
Source field	Name of the fields used by derived fields. Each cell in this column contains a drop-down list box. This drop-down list box provides for selection all fields of the field list which are valid in the current context.
F, Length	Format and length of the source field. These columns are read-only. Section *Rules Applying to Format Changes* describes how the format of the derived field is determined by Predict.
Start	The relative byte position where the part of the source field to be used by the derived field starts (not applicable to phonetic descriptors). See also *Specifying the Start and End Position* below.
End	The relative byte position where the part of the source field to be used by the derived field ends (not applicable to phonetic descriptors and VSAM Primary Superdescriptors or Alternate Indices). See also *Specifying the Start and End Position* below.
DB	Field short name of the source field. This column is read-only.

Specifying the Start and End Position

The start and end values given in the definition of a derived field are always byte positions within the source fields (beginning with 1 and counting from left to right for alphanumeric fields and binary fields and from right to left for numeric fields).

The full length is used if no start and end values are specified. In Adabas it is possible to address byte positions outside of the length of field. If this feature is used and a start byte outside of the source field specified, an end byte must be specified.

Note: Special rules apply when specifying the length of subfields/descriptors. See *Specifying the Length of Subfields*.

The following rules apply:

- Superfields/descriptor definitions can be based on up to twenty source fields.
- Only formats A, B and N are possible for superfields/descriptors.
- Format N can be useful for Natural, but is not recommended because an alphanumeric or binary value cannot be converted to a numeric field.

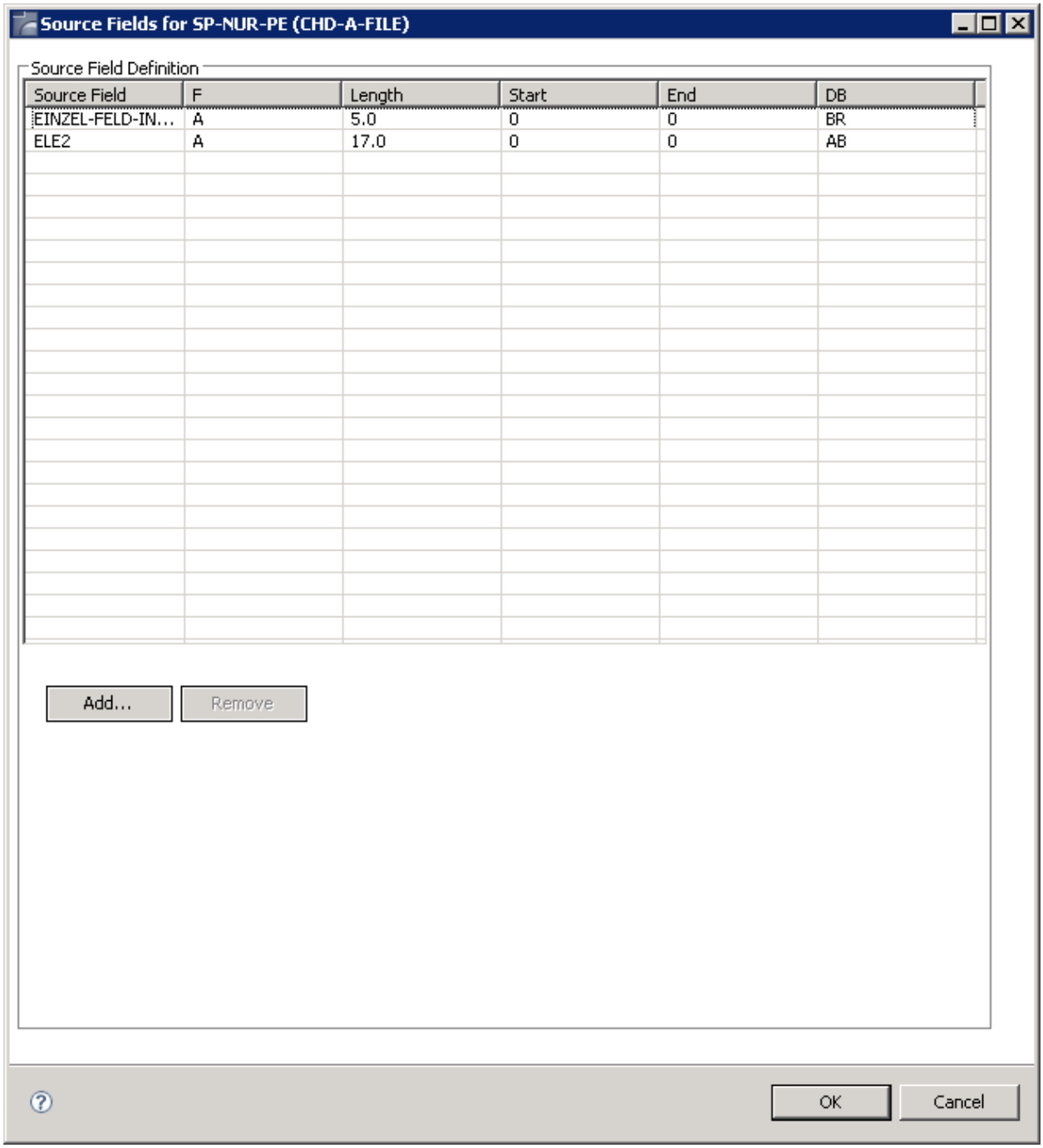
Defining Derived Fields of Special Types

The following topics are covered below:

- Superfields/Descriptors for Files of Type A, C and Z
- Subfields/Descriptors for Files of Type A, C and Z

Superfields/Descriptors for Files of Type A, C and Z

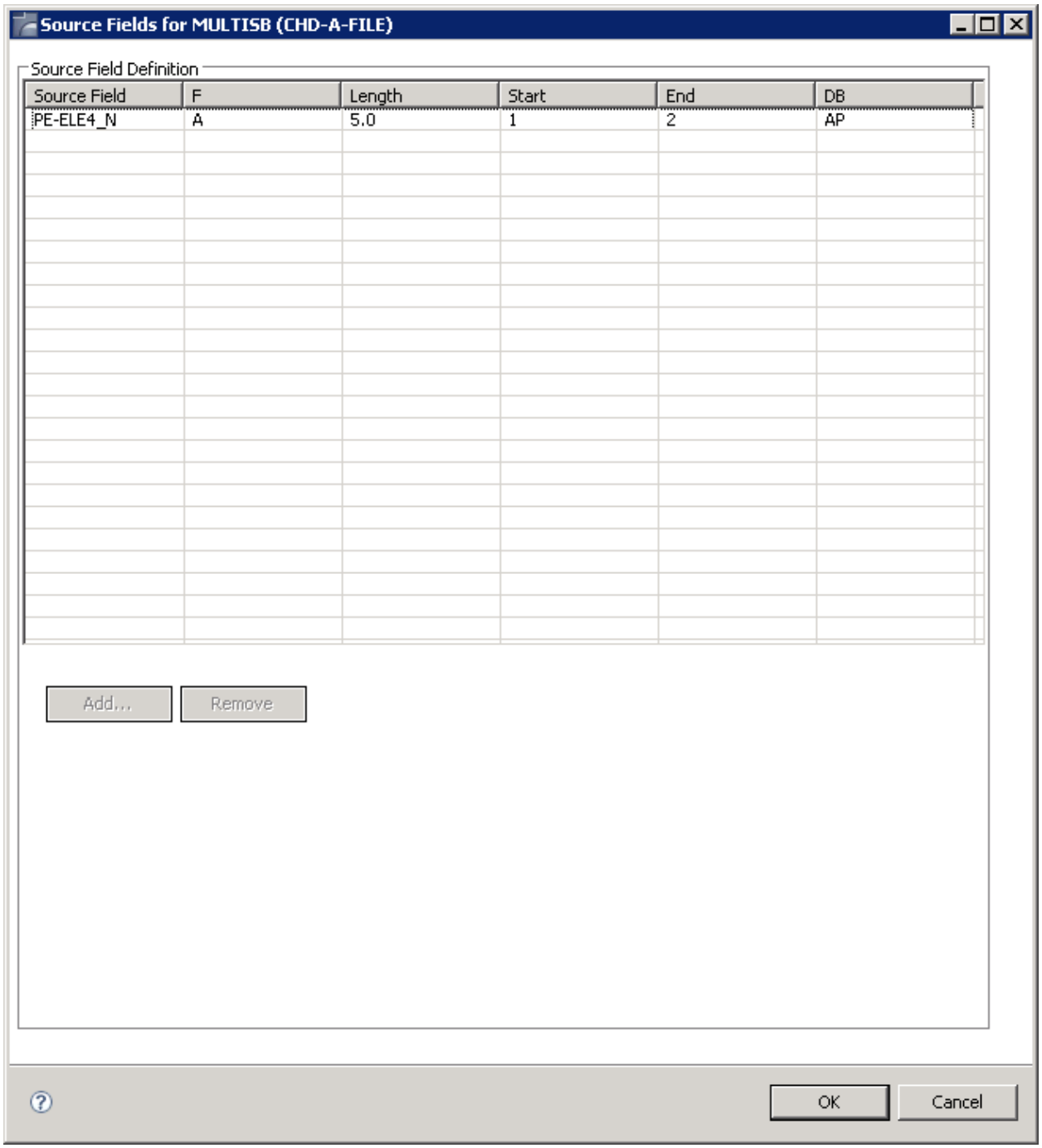
The window for the definition of superfields/descriptors for files of type Adabas, Conceptual and Standard looks as follows.



See also *General Rules for Defining Derived Fields*.

Subfields/Descriptors for Files of Type A, C and Z

Subfields/Descriptors for files of type Adabas, Conceptual and Standard are defined in the window below.



With subfields/descriptors, only one source field can be entered in the window.

Specifying the Length of Subfields

If the source field of a subfield/descriptor has format P and the start byte is greater than 1, the length of the subfield/descriptor is normal length+1, because the sign of the source field is always included in the subfield/descriptor field (see *Adabas Utilities* documentation).

Example:

Given that

- the source field has format P and length 5,
- the subfield/descriptor definition is source field from 2 to 3,
- the length of the subfield is 3 bytes (2 bytes + 1 byte for sign),
- the 3 bytes packed are 5 digits,

then the subfield/descriptor has format P and length 5.

Note: See also ***General Rules for Defining Derived Fields***.

Rules Applying to Format Changes

The format of derived fields is determined by Predict or can be defined manually.

Note: To understand the following, some knowledge of the hierarchical data structures of Predict and the process of rippling is required. See ***Rippling*** in the section *File* for more information.

The following topics are covered below:

- Determining the Format of Sub/Superfields/Descriptors
- Changing the Format of Superfield/Descriptors Manually
- Impact of Changes to Standard Fields - Rippling

Determining the Format of Sub/Superfields/Descriptors

The format of sub/superfields/descriptors in files of type Adabas, Conceptual and Standard (codes A, C, Z) is generated automatically by Predict. A format of a derived field that has been determined by Predict can, however, be overwritten manually. The following sections describe the rules applying.

Subfield/Descriptor

Subfield/descriptors always have the same format as the source fields they are derived from. If the format of a source field is changed, the format of the subfield/descriptor is changed accordingly.

Superfield/Descriptor without Format

If a superfield/descriptor is defined without a format, Predict assigns the format as follows:

- **Format=A**
  if at least one source field of the SP field is defined with format A, or if one of the source fields specified in the definition does not yet exist in the file.
- **Format=B**
  if no source field is defined with format A.

Superfield/Descriptor with Format

If the format of source fields has been changed, Predict checks if the new and the old format of the source field are compatible. If they are compatible, the change does not have any impact on the format of the superfield/descriptor.

The formats NS, US, N and U and the formats P and PS are compatible. So, if the format is changed from N to US, for example, the format of the superfield/descriptor will not change.

If the new and the old format of the source field are *not* compatible, a window appears in which a format change proposed by Predict can be confirmed or a new format can explicitly be assigned to the superfield/descriptor.

Changing the Format of Superfield/Descriptors Manually

The format of a superfield/descriptor can be changed manually. If a source field of the superfield/descriptor is then changed again, Predict checks if the change affects the format of the superfield/descriptor.

Impact of Changes to Standard Fields - Rippling

Changes to sub/superfield/descriptors and fields used in sub/superfield/descriptors (source fields) are rippled as described in the sections below.

Changes to Sub/Superfield/Descriptors

It is not recommended to define sub/superfield/descriptors in standard files and to use these in real files. It is however possible to do it. The following rule then applies:

Changes to the format and length and changes to the definition of derived fields in standard files are not rippled from standard files to real files and userviews. This is because the definition of derived fields is not coupled, and rippling format and length alone could lead to inconsistent data definitions in real files and userviews.

Changes to Source fields

Changes to the format of a standard field are rippled as normal to all fields in a file connected to this standard field.

If a field in an Adabas file is used in the definition of a sub/superfield/descriptor, the format of the sub/superfield/descriptor is also changed if one of the following conditions is met:

- the resulting format is A, or
- the resulting format is B and the old format was A.

Note: In the case of superdescriptors, if the format in the Adabas file is set (manually) to N and the correct format were B, no change is made (unless the field length is greater than 29).

Rules Applying to Suppression/Length Changes

The suppression and length of derived fields is determined by Predict.

The suppression and length of sub/superfields/descriptors in files of type Adabas, Conceptual and Standard (codes A, C, Z) is generated automatically by Predict. A derived field gets:

- null suppression if at least one source field has null suppression.
- the total length of the source field(s).

Note: The full source field length is used if no start and end values are specified.

If the suppression or length of a source field is changed, suppression and length of the sub/superfields/descriptors are changed accordingly.

Validation of Derived Field Definitions

If the format of derived fields is changed manually, Predict performs validation checks. These checks are described in this section.

Predict performs the following validations for derived fields:

- A superfield/descriptor can have only one source field which is a multiple-value field.
- Source fields with format D, T, or L must not have a start or end character.

The following rule applies for all file types except Conceptual and Standard: All source fields must exist in the file. This check is performed when you choose the **OK** or **Apply** button.

The following topics are covered below:

- Phonetic Descriptors for Files of Type A, C and Z
- Hyperdescriptors for Files of Type A, C and Z
- Collation Descriptors for Files of Type A, C and Z
- Key or Index Fields in SQL Files - Superfields
- VSAM Primary Superindex or Alternate Superindex

Phonetic Descriptors for Files of Type A, C and Z

The window for defining phonetic descriptors for files of type Adabas, Conceptual and Standard is identical to that for subfields/descriptors. See ***Subfields/Descriptors for Files of Type A, C and Z***.

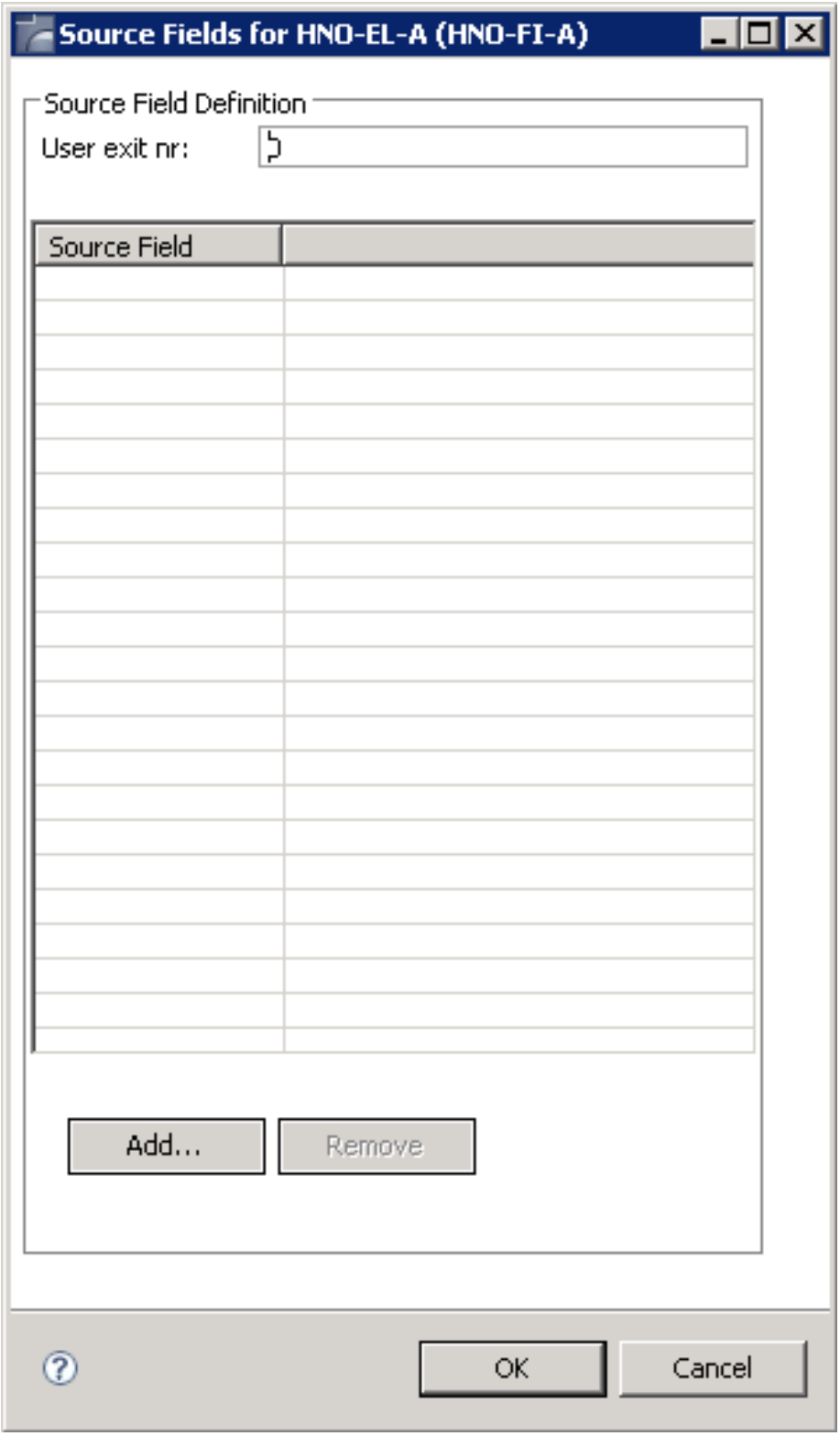
With phonetic descriptors, only one source field can be entered in the window.

The Start and End attributes do not apply to phonetic descriptors: Adabas always uses the first 20 bytes of this field to build a phonetic descriptor.

Note: See also ***General Rules for Defining Derived Fields***.

Hyperdescriptors for Files of Type A, C and Z

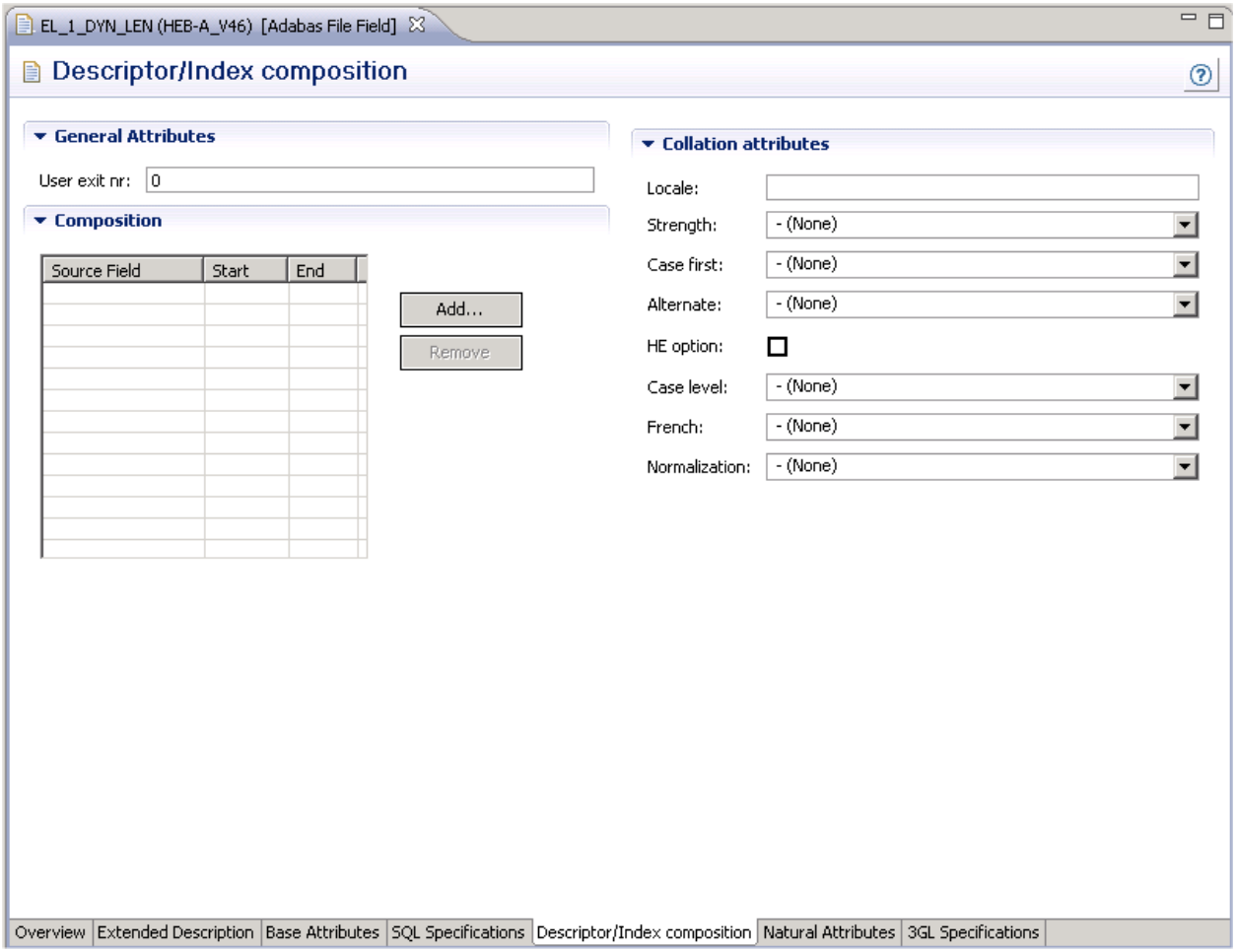
The window for defining hyperdescriptors looks as follows:


Source Fields for HNO-EL-A (HNO-FI-A)
Source Field Definition
User exit nr:
Source Field
Add...
Remove
OK
Cancel

Attributes	
User exit nr	A number between 1 and 31 identifying the user exit that defines the hyperdescriptor. See the *Adabas User Exits* documentation.

Collation Descriptors for Files of Type A, C and Z

The window for defining collation descriptors looks as follows:



Attributes	
User exit nr	A number between 1 and 8 identifying the user exit that defines the collation descriptor. See the *Administration* section of the *Adabas on Open Systems* documentation for further information.
HE option	If you specify this option, you must specify the corresponding parent field value in the value buffer for search operations, rather than the internal collation key. See the *Administration* section of the *Adabas on Open Systems* documentation for further information.

Attributes	
Locale	One of the locales supported by ICU. See the *Administration* section of the *Adabas on Open Systems* documentation for further information.
Strength	You can specify one of the following values: P (Primary), S (Secondary), T (Tertiary), Q (Quarternary), I (Identical) or blank (none). The value specified represents the comparison levels. See the *Administration* section of the *Adabas on Open Systems* documentation for further information.
Case first	Specifies whether uppercase letters will be sorted before lowercase letters or vice versa. Valid values: U (Upper), L (Lower) or blank (none). See the *Administration* section of the *Adabas on Open Systems* documentation for further information.
Alternate	Specifies the sorting sequence for punctuation characters such as space or hyphen. Valid values: S (Shifted), N (Non ignorable) or blank (none). See the *Administration* section of the *Adabas on Open Systems* documentation for further information.
Case level	If specified, an additional case level is formed between secondary and tertiary. Valid values: C (Caselevel), N (No caselevel) or blank (none). See the *Administration* section of the *Adabas on Open Systems* documentation for further information.
French	Specifies whether or not diacritics will be sorted as in French. Valid values: F (French), N (No french) or blank (none). See the *Administration* section of the *Adabas on Open Systems* documentation for further information.
Normalization	Specifies whether or not Unicode canonical equivalence is to be taken into account. Valid values: O (Normalization), N (No Normalization) or blank (none). See the *Administration* section of the *Adabas on Open Systems* documentation for further information.

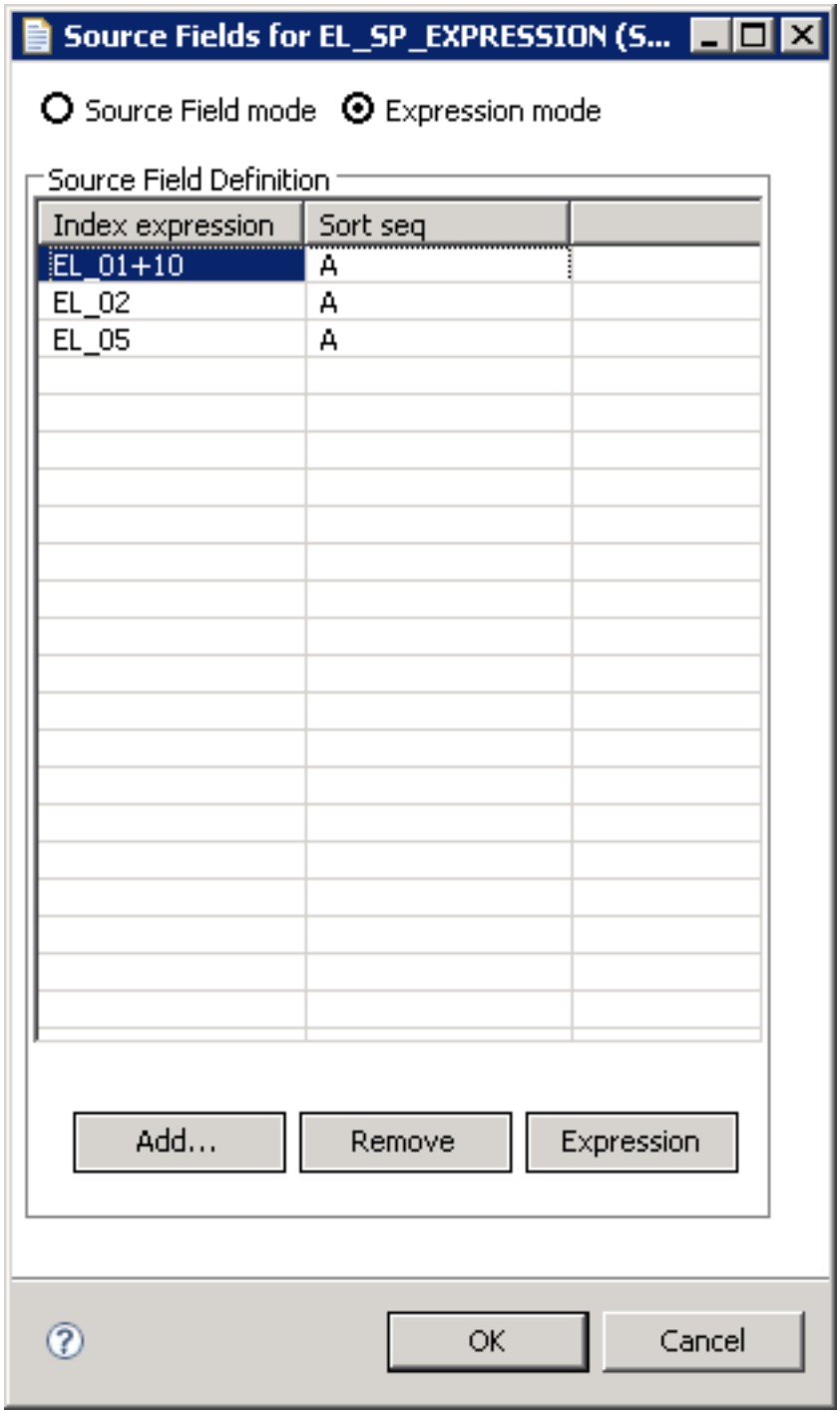
Key or Index Fields in SQL Files - Superfields

The window below is used for defining Keys or Indexes in fields of the following file types:

A	Adabas file (with parameter Adabas SQL usage set to Y)
BT	Adabas D table
D	DB2 table
MT	DB2 query table
JT	Ingres table
OT	Oracle table
X	General SQL
XT	Informix table
YT	Sybase table

The following rules apply:

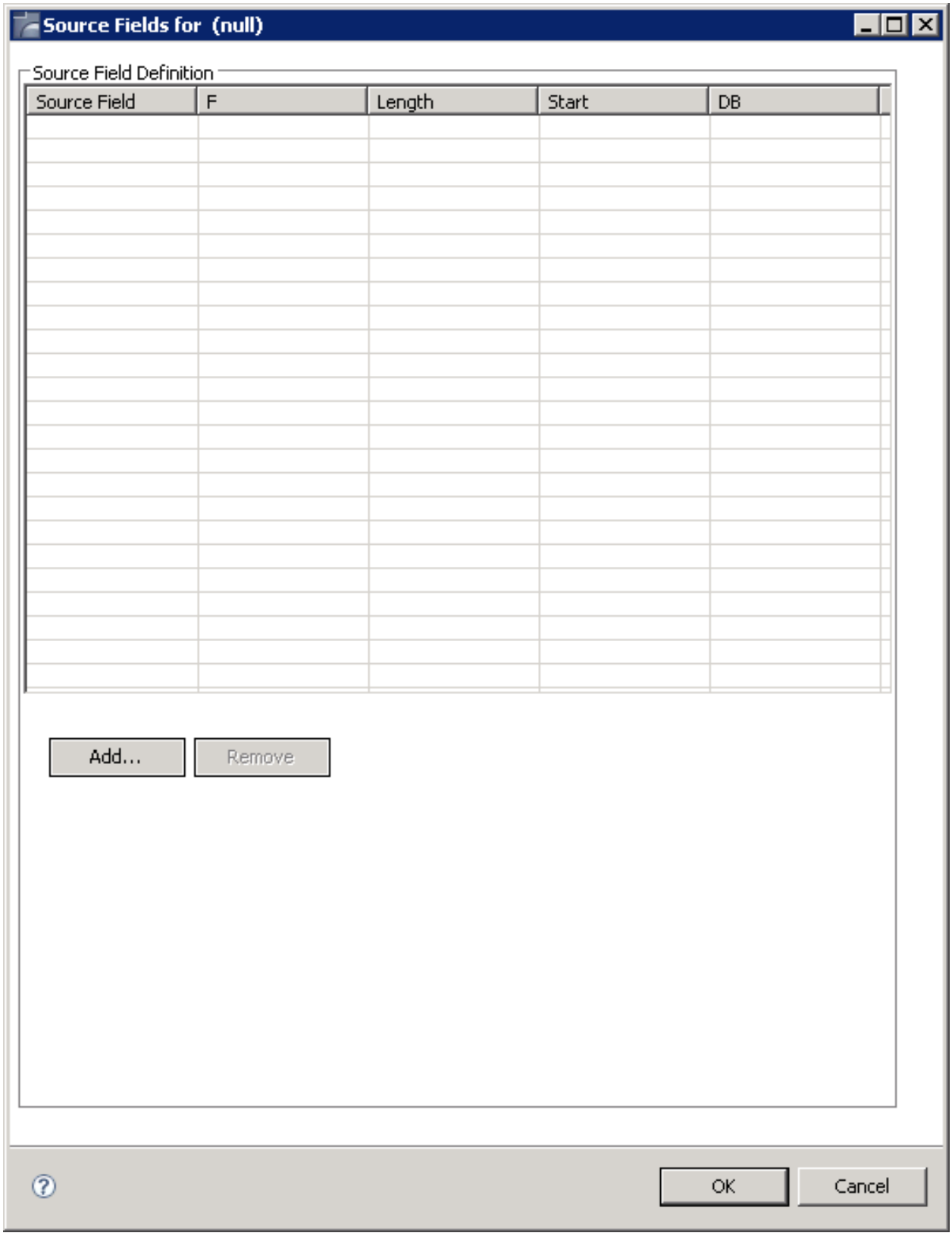
- If the field type is blank (normal field), the key or index is based on one field.
- If the field type is SP (superfield), the key or index includes more than one field.
- If the field type is SP (superfield), the index includes one or more fields if the file is D or MT.


Source Fields for EL_SP_EXPRESSION (S...
Source Field mode
Expression mode
Source Field Definition
Index expression
Sort seq
EL_01+10
A
EL_02
A
EL_05
A
Add...
Remove
Expression
OK
Cancel

Attributes	
Index name	The name of the key or index. Must be entered in qualified form: creator/schema name followed by key or index name, separated by a hyphen. The creator/schema and key or index name are subject to SQL naming conventions. Creator name and field name are concatenated and proposed as index name.
No Overlaps	Only applies when documenting DB2 temporal tables that record the application period (BUSINESS_TIME). Specifies whether (Y) or not (N) the `BUSINESS_TIME WITHOUT OVERLAPS` clause applies. If BUSINESS_TIME WITHOUT OVERLAPS is specified, the BUSINESS_TIME period will not overlap in time periods for the same column-name values.
Source field	The name of a column (source field) from which the key or index is derived. If the key or index is based on one field (field type blank), the name of that field is displayed and cannot be changed. If the key or index includes more than one field (Field type SP), up to 64 column names can be entered. Each must name a column of the table. **Note:** For fields in files of type X (General SQL), you can enter up to 16 column names.
Sort seq	Puts key or index entries in ascending order by source fields (column). Default. Puts key or index entries in descending order by source fields (column). Puts key or index entries in random order by source fields (column). Only applies to files of type D or MT. Specifies additional columns to append to the set of index key columns of a unique index. Only applies to files of type D or MT.
Expression	Only applies when an index and if files of type D or MT and when in Expression mode. Select the desired field and click on the **Expression** button to display an editor for the expression. Up to 64 expressions can be maintained. An asterisk indicates that an expression exists.

VSAM Primary Superindex or Alternate Superindex

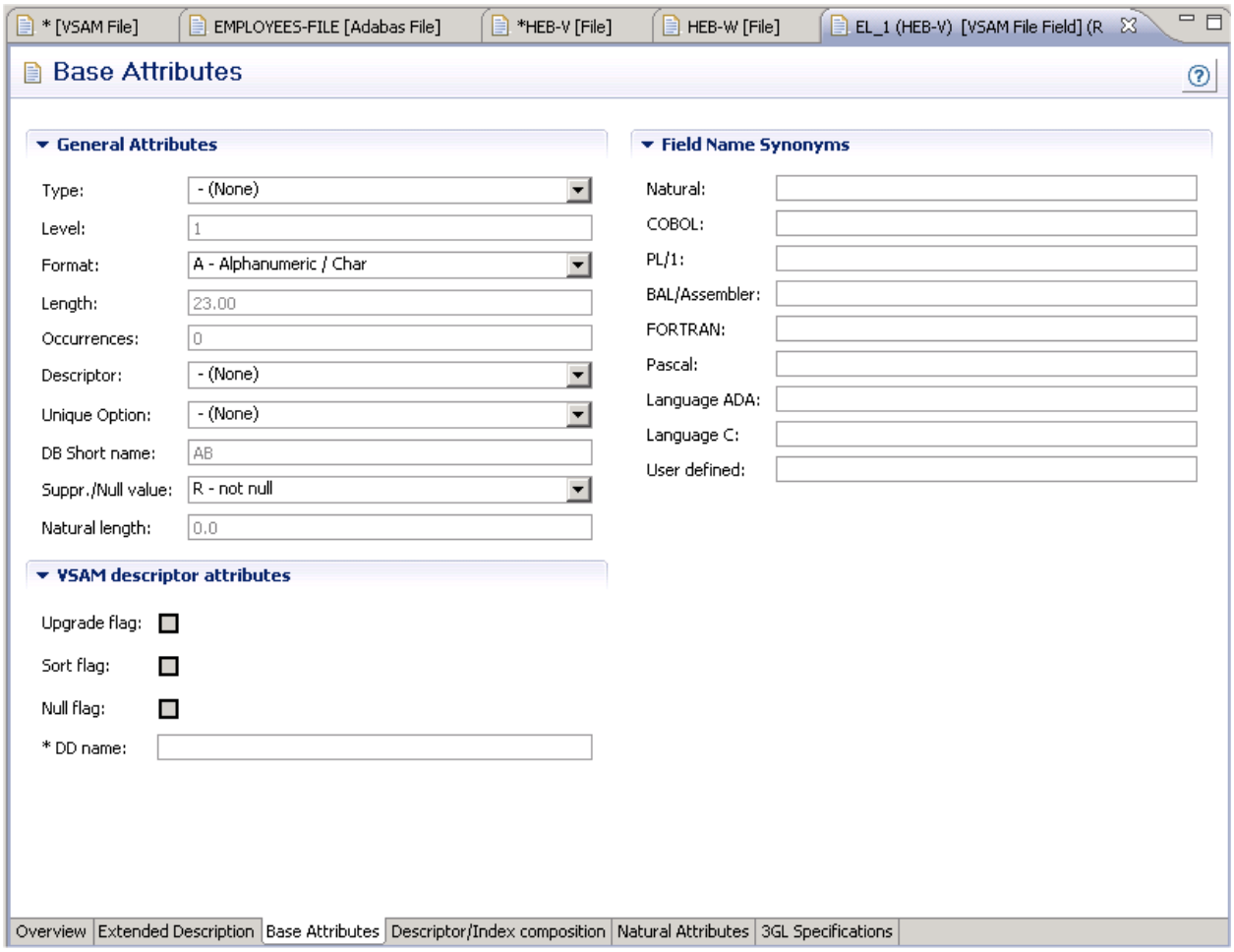
VSAM superdescriptors (Field type SP) in a file of type V (physical VSAM) and L (logical VSAM) are defined in the following window.


Source Fields for (null)
Source Field Definition
Source Field
F
Length
Start
DB
Add...
Remove
OK
Cancel

Attributes	
Start	The starting position (offset plus one) of the superdescriptor within the source field. An end position cannot be specified.

If a VSAM field on an alternate index (descriptor A) in a file of type V (physical VSAM), L (logical VSAM), or C (conceptual) is defined (the descriptor type is A), the following additional attributes can be defined when you choose **Open** from the context menu of a field: upgrade flag, sort flag, null flag and DD name (see below).

Additional Attributes for VSAM Alternate Fields



Additional Descriptor Attributes		
Upgrade flag	Y	Alternate index is updated by Natural.
	N	Alternate index is updated by VSAM.
Sort flag	Y	If the upgrade flag is also Y, the alternate index is read in ascending order. Otherwise, the alternate index is read in the order in which the values were entered during field update.
Null flag	Y	Records with a null value in this index field are suppressed.
DD name	The DD name associated with this alternate index file. In CICS, the FCT name of the VSAM file.	

12 Defining Additional Attributes of Fields

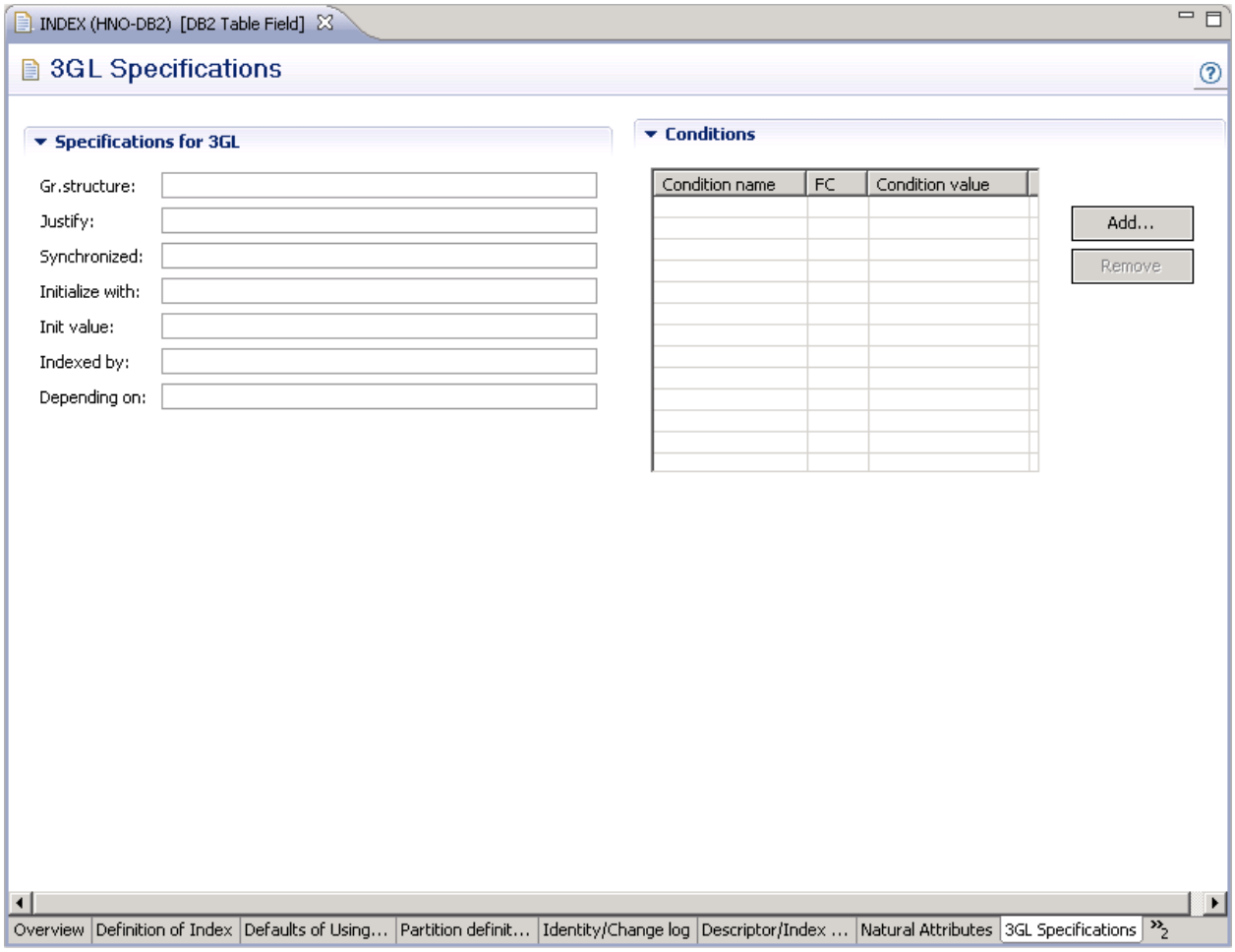
- 3GL Specification 111
- Conditions 113
- Field Name Synonyms 114
- Adabas Security and Edit mask 115
- DBMS Extensions 117
- Field Procedure 121
- Derived Field Expression 122
- Index Definition - DB2 123
- Index Definition - Oracle 128
- Default value 134
- Constraint name 135
- Identity definition / Change log 136
- Platform Compatibilities 138
- Base Extensions 139

If you choose **Open** from the context menu of a field, a window appears in which you can define additional attributes.

Only those types of additional attributes appear in the window that apply to the type of field. For example: the option Adabas security & Edit mask is not contained in the list when a DB2 index field is processed.

The additional attributes are described in the following sections.

3GL Specification



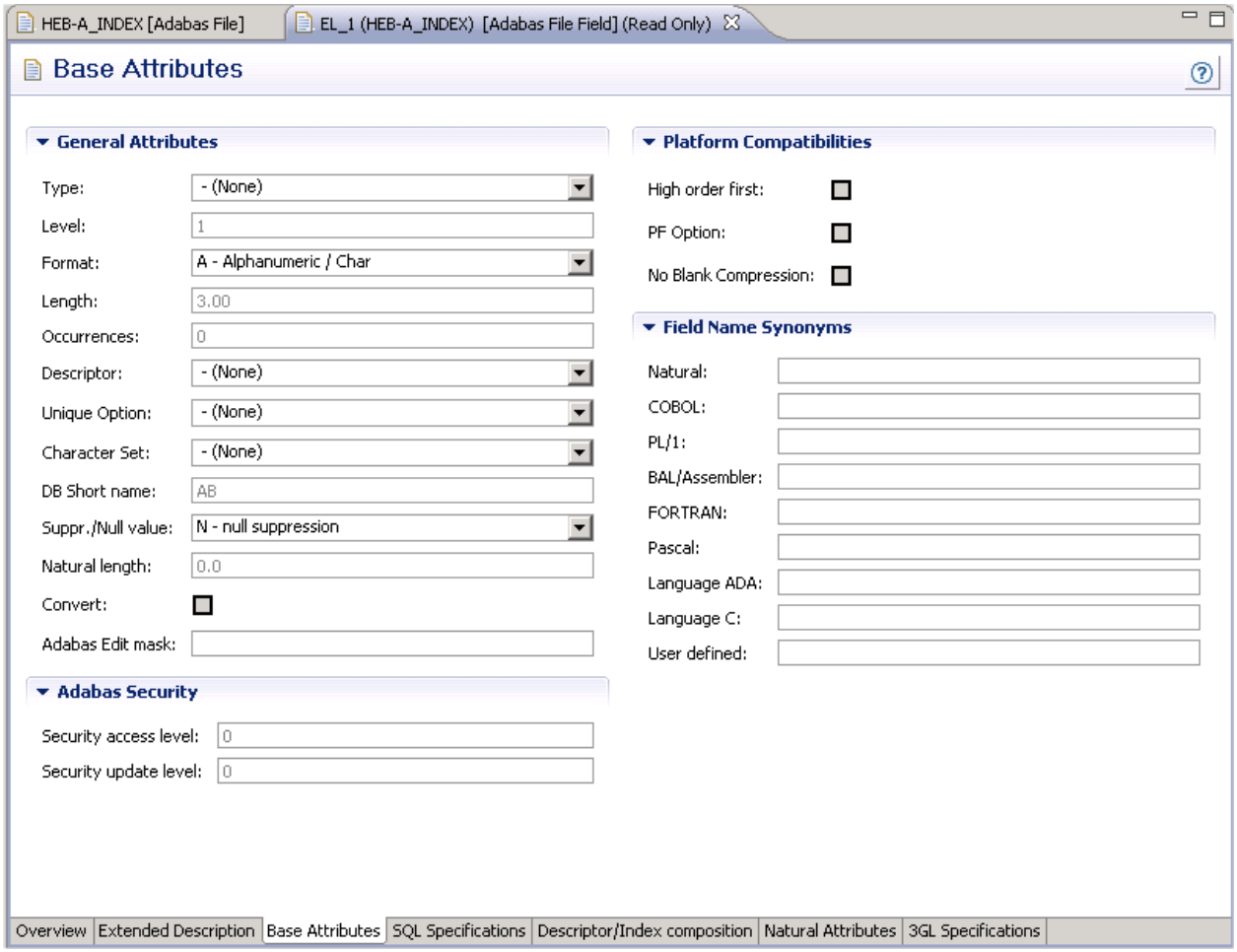
Attributes	
Gr.structure	The field attribute Gr.structure is used to change the record layout generated from a PE/PC field. If Gr.structure is set to N, all fields within a PE group are treated as multiple value fields. Setting Gr.structure to N prevents the format buffer for Adabas from becoming very large. Gr.structure = N can only be specified for real fields in the deepest PE group (highest level number). For example: if there are 3 PE groups in the file on level 1, 4 and 6, only the PE groups on level 6 can be marked with Gr.structure = N. If Gr.structure is set to blank, PE/PC groups are to be generated as groups which occur n times as a whole.
Justify	R When COBOL copy code is generated, the statement JUSTIFIED RIGHT is added for this field. Any data written to this field is then right-justified.

Attributes		
	L	Data will be left-justified. Default.
Synchronized	Applicable to fields of format I, F or B and length 1, 2, 4 or 8.	
	S	When Assembler, COBOL or PL/I copy/include code or a record layout is generated, this field can be aligned on a half-word, word, or double-word boundary (speeding up arithmetic operations). This affects format buffer generation and the offsets of the fields in the record buffer. Slack-bytes are inserted into the record buffer by the assembler or compiler but they are built into any format buffer by Predict using space characters X.
Initialize with	Determines the initial value for generation. To be used instead of the standard value (zeros for a numeric field, blanks for an alphanumeric field).	
	S	blank
	L	low value
	H	high value
	Z	zero
	Q	quote
	F	Fill with string specified in the parameter Init. value (mandatory). For example: if X is specified and the field length is 4, XXXX will be used for initialization.
	blank	Field will be initialized with the string specified in the field Init. value. If no Init. value is specified, no initialization is performed.
Init. value	If Initialize with is either F or blank a value used for initialization of a field must/can be specified. Length and format of the Init value must be valid for the field. For binary fields hexadecimal constants such as FB0A are valid. Hexadecimal values can be specified in two ways: ■ if Format=B, hexadecimal values can be specified directly. Example: F0 ■ if Format=A, hexadecimal values must be preceded by uppercase X or H and be enclosed in single quotes. Example: X'F0' or H'F0' See also ***Condition Name and Value***.	
Indexed by	String that is used when generating the COBOL INDEXED BY clause (only valid for fields of type MU/MC or PE/PC).	
Depending on	String used when generating the COBOL DEPENDING ON clause (only valid for fields of type MU/MC or PE/PC).	

Conditions

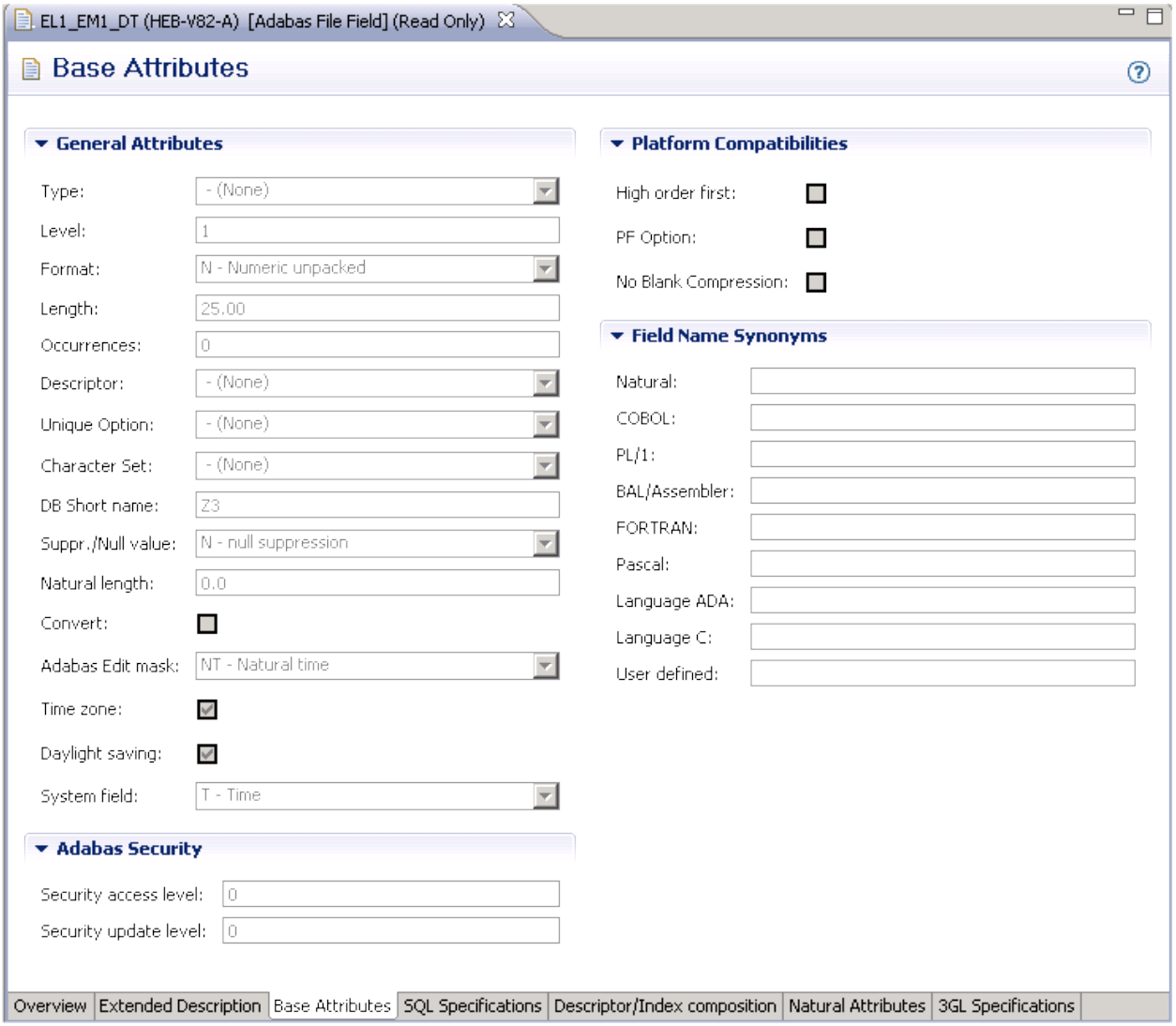
Attributes	
Condition name	A value to be used when generating either equate data in Assembler copy code or a level 88 entry in COBOL copy code. Up to 29,970 condition names can be entered. Each name needs at least one corresponding condition value. Using condition names can make logical conditions and assignments easier to handle.
FC	Figurative constant. Valid values: S — blank L — low value H — high value Z — zero Q — quote F — Fill with string specified in the parameter Condition. value. For example: if X is specified an the field length is 4, XXXX is used as condition value. blank — The string specified in the field Condition value is used.
Condition value	The length and format of this value must be valid for this field. This value must have a corresponding condition name. Up to 29,970 condition values can be entered. If several values correspond to the same name, put the name before the first value and leave the name field blank before later values. THRU in the name field indicates a range of values ending with the value on that line and beginning with the value on the previous line.

Field Name Synonyms



Attributes	
Field name synonyms	Synonyms to be assigned to the field when definitions in the following programing languages are generated: Natural, COBOL, PL/I, BAL (Assembler), FORTRAN, PASCAL, ADA or C.

Adabas Security and Edit mask



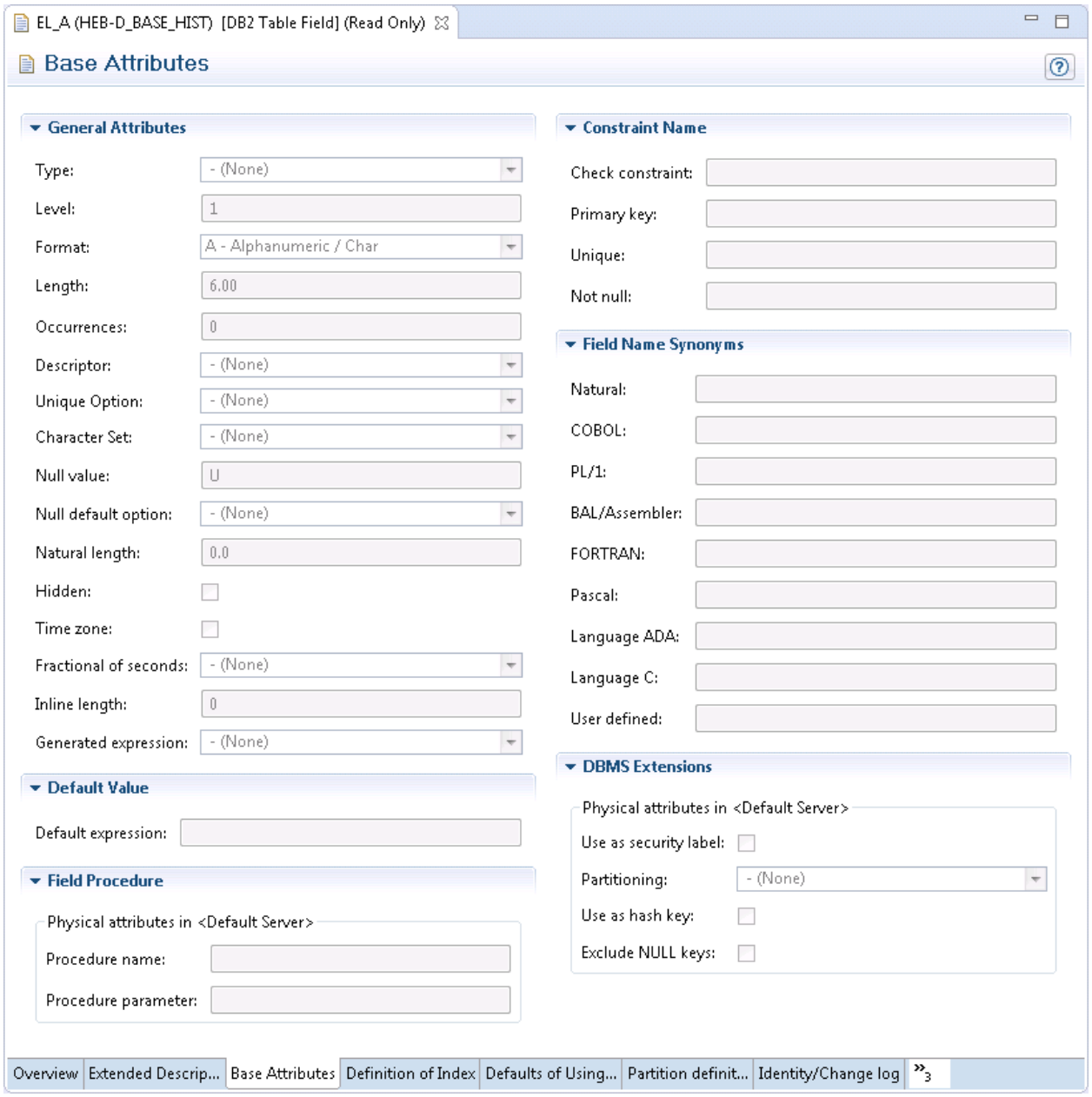
Attributes	
Adabas Edit mask	The Adabas edit mask to be used for the field. Determines how numeric fields are to be edited.
	Valid values for Edit mask rules used in the COBOL programming language:
	E1...E15. Edit mask is supported for compatibility reasons and for documentation purposes only. See the section *Format Buffer Syntax* in the *Adabas Command Reference* documentation for more information.

Attributes	
	Valid values for Date-Time Edit masks:
	D — Date.
	T — Time.
	DT — Datetime.
	TS — Timestamp.
	NT — Natural time.
	ND — Natural date.
	UT — Linux time.
	XTS — Linux timestamp.
	blank — None.
	In Predict, fields with format D by default use Natural date (NATDATE) and fields with format T by default use Natural time (NATTIME) Edit masks. The different Date-Time Edit masks require different minimum numeric field lengths. Detailed information is given in section *Date-Time Edit Mask Reference* of the *Adabas Mainframe* documentation.
Time zone	The time zone in which a timestamp is applied. Only available for fields using one of the following Adabas edit masks described above: DT, NT, TS, UT or XTS.
Daylight saving	The daylight savings indicator can only be specified for date-time fields defined with the TZ option. Indicates whether daylight saving applies (Y) or not (N).
Security access level	The Adabas access security level of the field.
Security update level	The Adabas update security level of the field.
System field	Type of the system-generated field in Adabas. Only available for Adabas fields of type A.
	Valid values:
	JN — Job name. For alphanumeric fields only. Length must be 8 or greater.
	OU — Open command user. For alphanumeric fields only. Length must be 8 or greater.
	SI — Session ID. For alphanumeric fields only. Length must be 28 or greater.
	SU — Session user. For alphanumeric fields only. Length must be 8 or greater.
	T — Time. For numeric fields only.

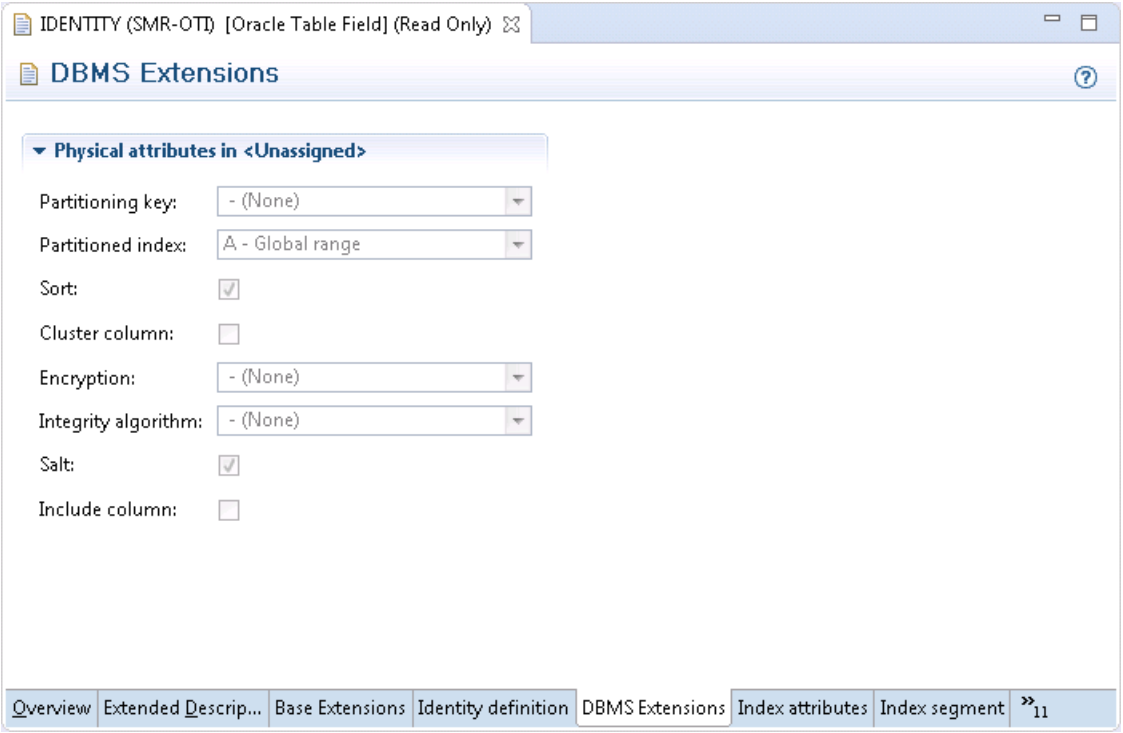
DBMS Extensions

Applicable only to DB2 fields of type D or MT, Oracle fields of type OT and to Adabas fields of type A.

The following screen shows the options for DB2 fields of type D and MT.



The following screen shows the options for Oracle fields of type OT.



Attributes		
Use as security label	Specifies that the table column will contain security label values. A table can have only one security label column. To define a table with a security label column, the primary authorization ID of the statement must have a valid security label, and the RACF SECLABEL class must be active. In addition, the following conditions are also required: ■ The data type of the column must be CHAR(8). ■ The subtype of the column must be SBCS. ■ The column must be defined with the NOT NULL and WITH DEFAULT clauses. ■ No field procedures, check constraints, or referential constraints are defined on the column. ■ An edit procedure is not defined on the table. Refer to your *DB2* documentation for further details.	
Partitioning	Indicates for each implementation of a table in an SQL server, which field controls the partitioning. Valid values:	
	blank	Non-partitioning. The current field is not a partitioning key and has no index.
	T	Table partitioning key.

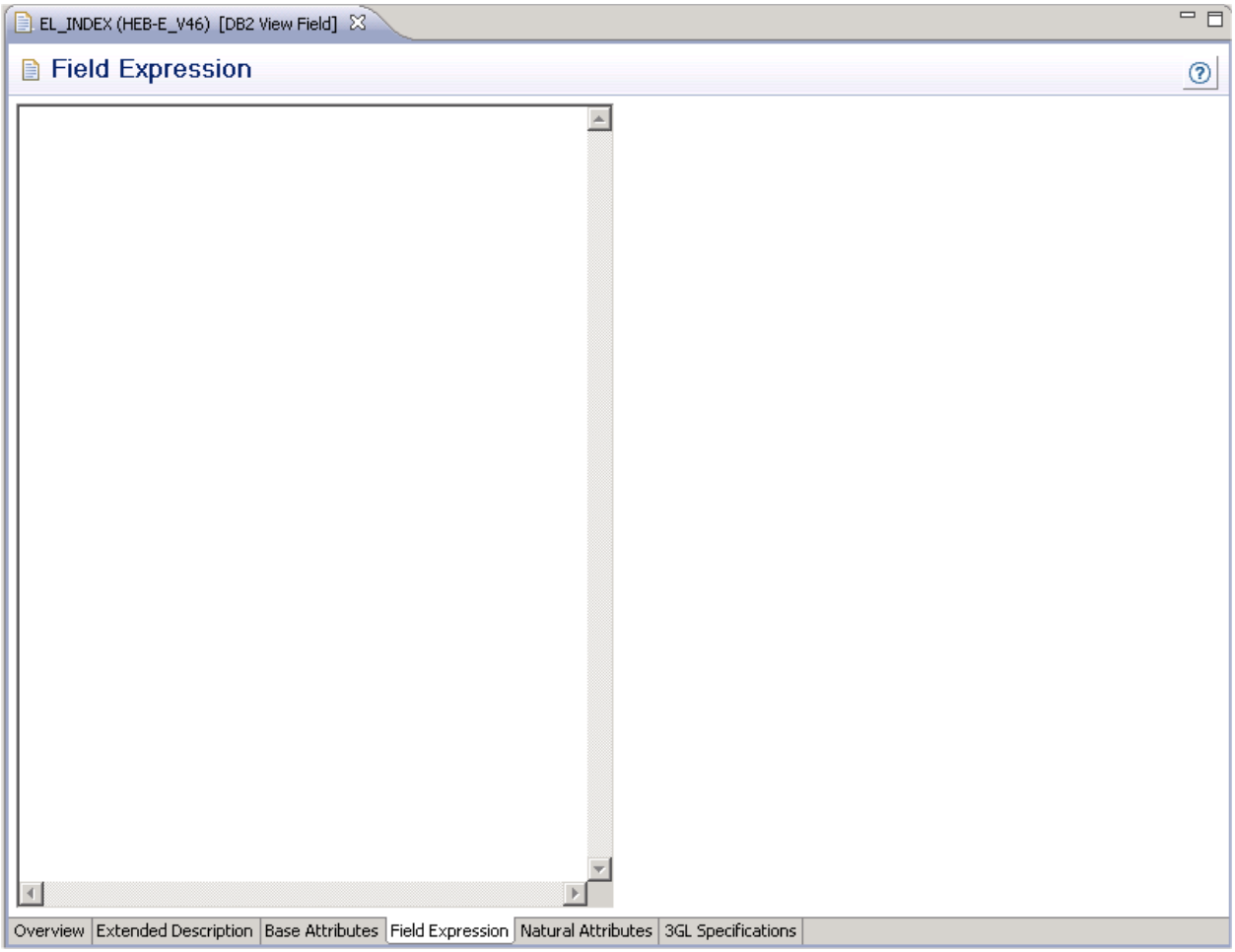
Attributes		
		The table is data partitioned and the current field is the partitioning key. Additionally, it may have a non-partitioned index.
	I	Index controlled key. The table partitioning is controlled by a partitioning index. The current field must have an index defined; this is indicated by descriptor type D (index), P (primary key) or F (foreign index).
	N	Non-partitioned secondary index. The field has a non-partitioned index, but it is not the partitioning key.
	P	Data partitioned secondary index (DPSI). The field has a partitioned index, but it is not the partitioning key. Within the file a different field must be defined as partitioning key.
	Q	Table partitioning key and DPSI. The table is data partitioned and the current field is the partitioning key. In addition, a partitioned index is defined for the field.
Cluster index	Only available for descriptor type D (index), P (primary key) or F (foreign index).	
	Y	The records (rows) in the DB2 table are stored in the sequence of this index. Valid for max. one index per table. A table contained in a partitioned tablespace must have one index marked as a clustered index, if the partitioning option Index controlled key is to be used.
Exclude NULL keys	Only available for DB2 fields of type D and MT. Indicates whether NULL keys are excluded.	
	blank	Not specified.
	Y	Yes.
	N	No.
Partitioning key	Only available for Oracle fields of type OT. Indicates how the partitioning key is organized:	
	R	Range partitioned.
	H	Hash partitioned.
	blank	None.
Partitioned index	Only available for Oracle fields of type OT. Indicates how the partitioned index is organized:	

Attributes		
	A	Global range.
	B	Global hash.
	C	Local range.
	D	Local hash.
	blank	None.
Sort	Only available for Oracle fields of type OT. Specifies whether the rows of the cluster on this column are sorted after applying the hash function	
Cluster column	Only available for Oracle fields of type OT. Specifies whether or not this field (or combination of fields in case of an SP field) is used as cluster column	
Encryption	Only available for Oracle fields of type OT. Specifies the encryption method to be used:	
	1	AES256
	2	AES192
	3	AES128
	4	3DES168
	blank	None.
Integrity algorithm	Only available for Oracle fields of type OT. Specifies the integrity algorithm to be used:	
	S	SHA1
	N	NOMAC
	blank	None.
Salt	Only available for Oracle fields of type OT. Specify SALT to instruct the database to append a random string to the clear text of the column before encrypting it.	
Include column	Only available for Oracle fields of type OT. If this column is part of the primary key, this column determines where to divide an index-organized table row into index and overflow portions.	

Field Procedure

Attributes	
Procedure name	Name of a field procedure (DB2 parameter FIELDPROC). See the *Natural for DB2* documentation for more details.
Procedure parameter	Parameters passed to the field procedure. See the *Natural for DB2* documentation for more details.

Derived Field Expression



Applicable only to fields of type DV in files of the following types:

B	Adabas SQL views
E	DB2 views
IV	Intermediate view
JV	Ingres view
OV	Oracle view
X	General SQL
XV	Informix view
YV	Sybase view

The subquery of the file that contains the current field can specify a correlation name for any file whose fields it uses. The name of each field referenced in the expression must be qualified (pre-

ceded) by the correlation name of the file from which the field is taken, if a correlation namehas been specified for that file, or the ID of the file from which the field is taken, if no correlation name has been specified for it. The expression can include both comment lines (with /*, * or ** in the first two columns) and line comments (preceded by /*).

Example: A field which contains the annual salary:

```
*
* Field expression of a derived field
*
USER-TABLE1-SALARY * 12     /* SALARY FOR 12 MONTHS
+ CORRELATOR2-BONUS
↩
```

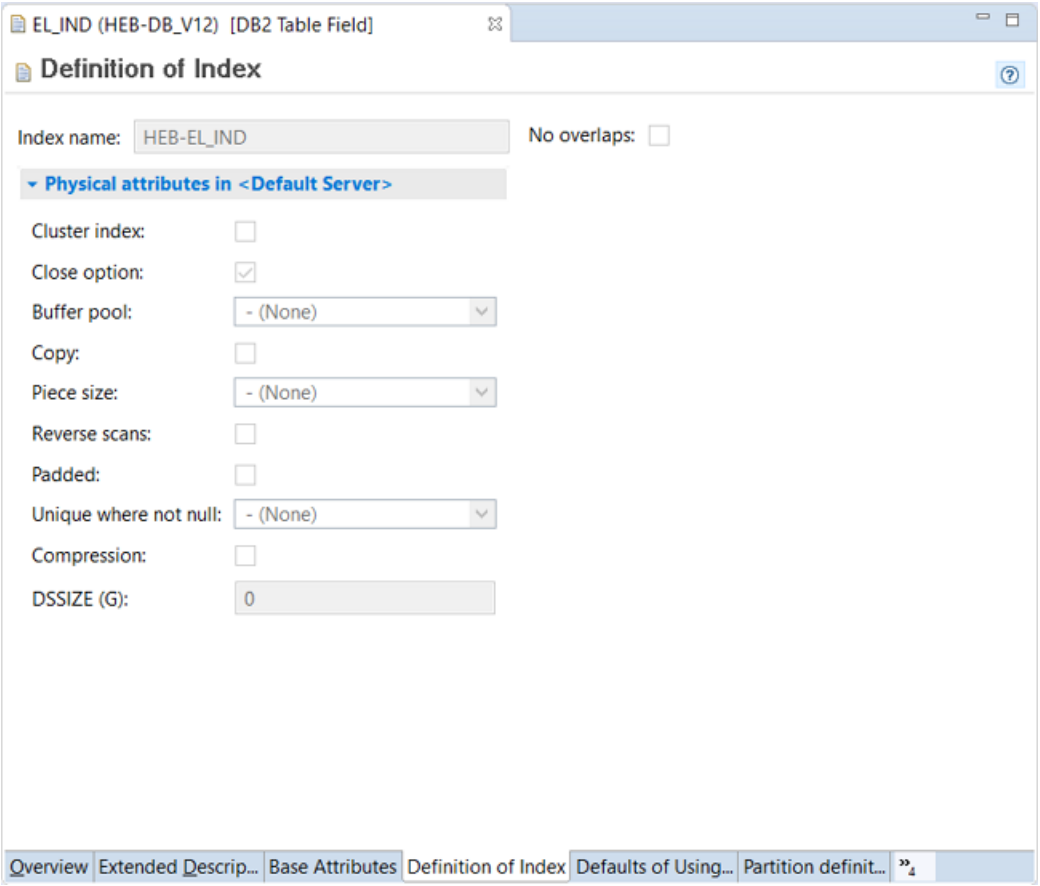
Index Definition - DB2

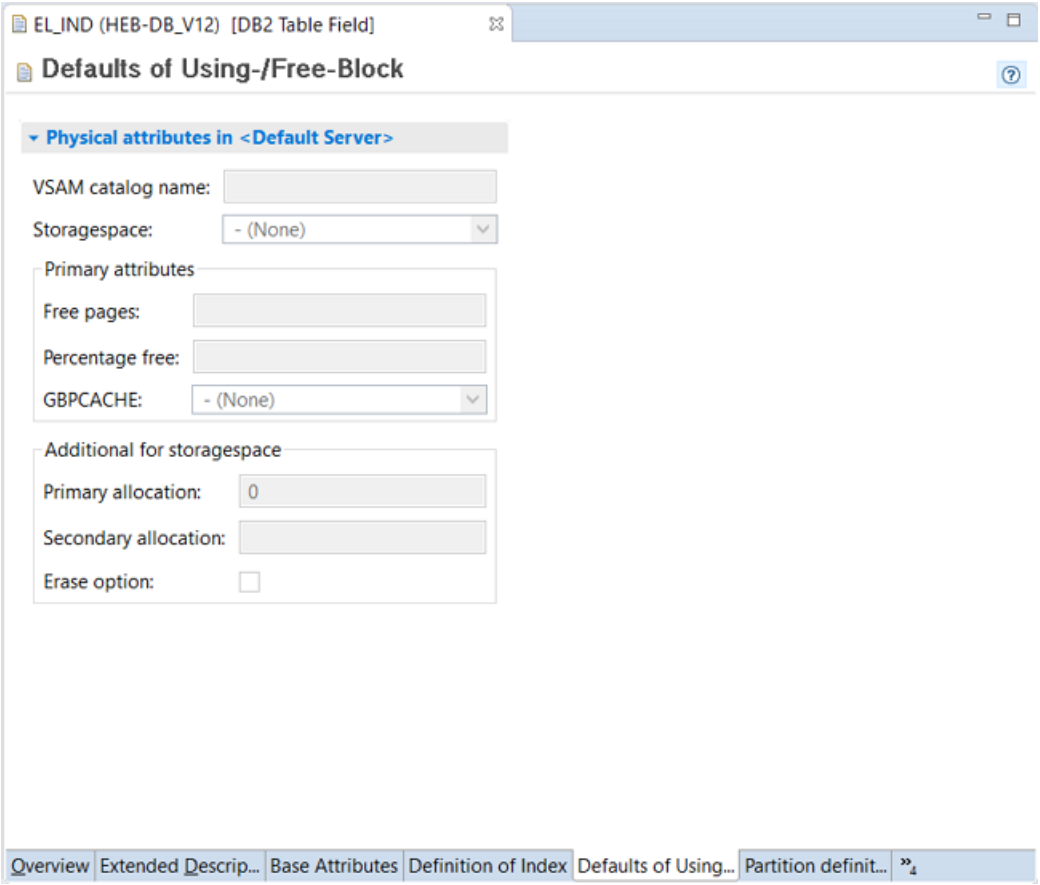
Index fields (descriptor type D, F or P) in a file of type D (DB2 table), are defined in the tabs below.

- Definition of Index / Defaults of Using-/Free-Block

- Partition definitions

Definition of Index / Defaults of Using-/Free-Block





Attributes	
Definition of Index	
Index name	The name of the DB2 index. See *Key or Index Fields in SQL Files - Superfields*.
Cluster index	Y The records (rows) in the DB2 table are stored in the sequence of this index. Valid for max. one index per table. A table contained in a partitioned tablespace must have one index marked as a clustered index, if the partitioning option Index controlled key is to be used..
Bufferpool	The buffer pool associated with the index.
Close option	Y The data sets supporting this index are closed when nobody uses the index.
Copy	Indicates whether the copy utility is allowed for the index. Y — Full image or concurrent copies allowed. N — Full image or concurrent copies not allowed.
Piece size	The maximum piece size for a non-partitioned index. Valid values: 0, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576, 2097152, 4194304.
Reverse scans	Specifies whether an index supports forward scans only or reverse scans of the index also.

Attributes		
Definition of Index		
	Y	Reverse scans allowed.
	N	Reverse scans not allowed. This is the default.
Padded	Specifies how varying-length string columns are to be stored in the index.	
	Y	Padded.
	N	Not padded.
	blank	Not specified.
Unique where not null	Null value option. Specifies whether a null value is allowed for a unique index. Valid values:	
	U	Null allowed. Any two null values are taken to be unequal. If the key is a single column, that column can contain any number of null values, though its other values must be unique.
	R	Not null. Any two null values are taken to be equal. For example, if the key is a single column, that column can contain no more than one null value.
	blank	Determined by gener. Null value option is set via default server settings in Predict.
Compression	Specifies whether compression for index data will be used.	
	Y	Compression is used.
	N	No compression is used. This is the default.
DSSIZE	Specifies the maximum size (in Gbyte) for each partition of a partitioned index.	
Defaults of Using- and Free-Block		
VSAM catalog name	Name of the VSAM catalog for the index. The parameter VSAM catalog name must be specified if data sets are already defined for the index.	
Storagespace	Storagespace where DB2 defines the data sets for the index (optional). If no storagespace is specified, DB2 uses the default storagespace.	
Primary allocation	Minimum primary space allocation (in Kbyte) for DB2-defined index data sets. A value specified in this field is stored only if the attribute storage group has been specified.	
Secondary allocation	Minimum secondary space allocation (in Kbyte) for DB2-defined index data sets. A value specified in this field is stored only if the attribute storage group has been specified.	
Erase option	Y	The DB2-defined data sets are to be erased (filled with nulls) when the index is dropped.
	A value specified in this field is stored only if the attribute Storage group has been specified.	

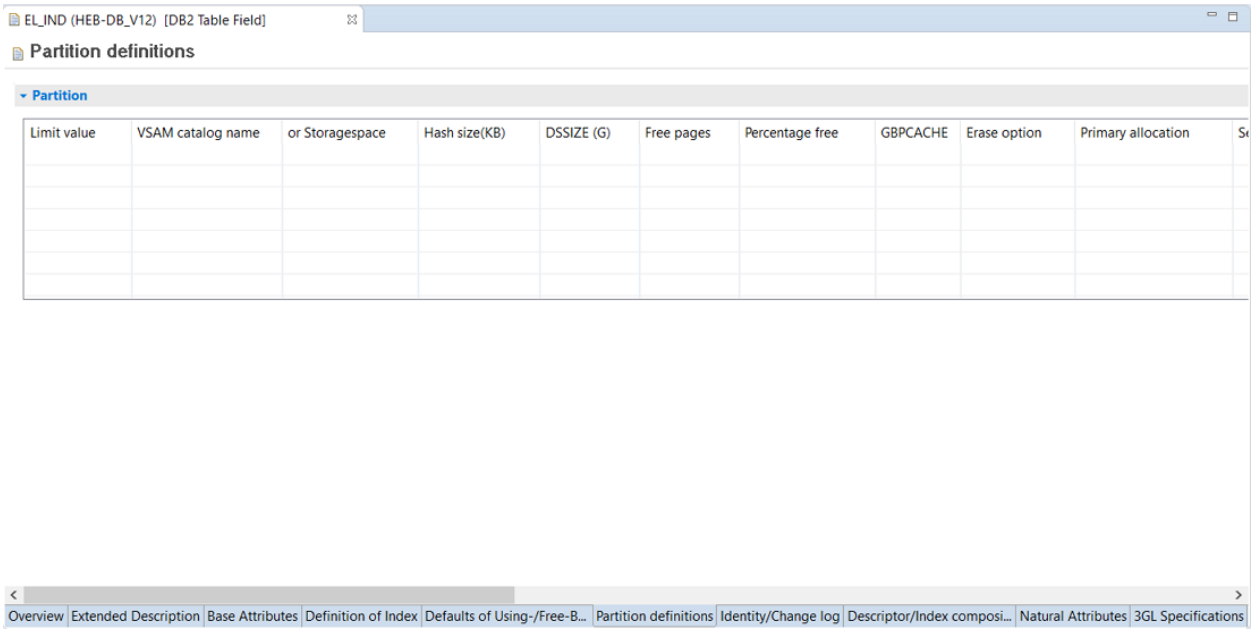
Attributes	
Definition of Index	
Free pages	A number from 0 to 255 which indicates that one page is to be left free each time this number of pages is used when the load operation creates index entries or when the index is reorganized. Zero indicates that no pages are to be left free.
Percentage free	A number from 0 to 99: the percentage of each page to be left as free space when index entries are created by a load operation or when the index is reorganized.
GBPCACHE	Only relevant in a data sharing environment. Specifies what pages of the table space or partition are written to the group buffer pool. Leave this field blank or enter: C — Changed. Only pages that have been changed are written to the group buffer pool. A — All pages are written. N — No pages are written to the group buffer pool.

Partition definitions

For a partitioned index (a cluster index for a table in a partitioned table space), the following tab is displayed. Each partition can then be defined in accordance with the Default values of using- and free-block (see description above).

Note: The data displayed in this screen depend on the values selected in the Partitioning field of the **DBMS extensions** screen.

To define a new partition, use the **New** button. When you have defined more than one partition, a scroll bar is available. To go to a specific partition, use the scroll bar. To delete the currently displayed partition, choose the **Delete** button.



Attributes	
Limit value	The highest value of the index key in this partition. At least one constant must be used and as many constants as there are columns in the key can be specified. The concatenation of all the constants is the highest value of the key in this partition of the index. **Note:** No checking is performed here.
DSSIZE	The value of DSSIZE given in the PARTITION clause for that partition.

All other attributes are described above.

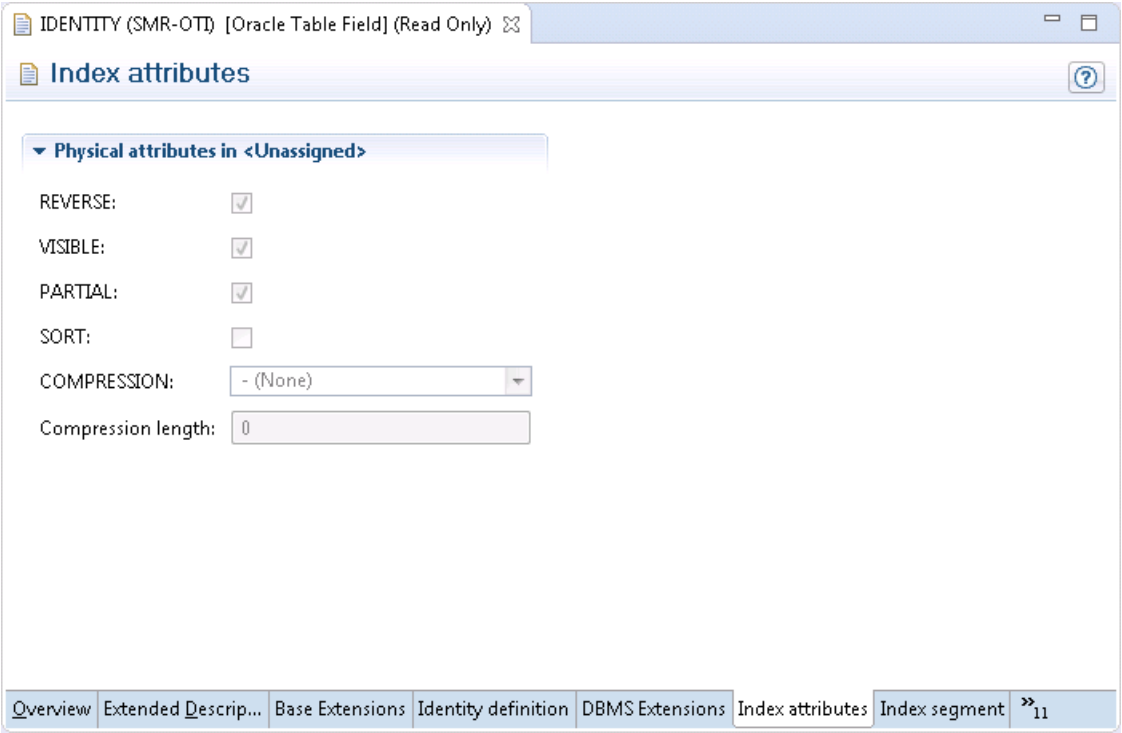
Index Definition - Oracle

Index fields (descriptor type D, F or P) in a file of type OT (Oracle table), are defined in the tabs below.

- Index Attributes
- Index Segment
- Index Storage

- Partitioning Key and Partitioned Index

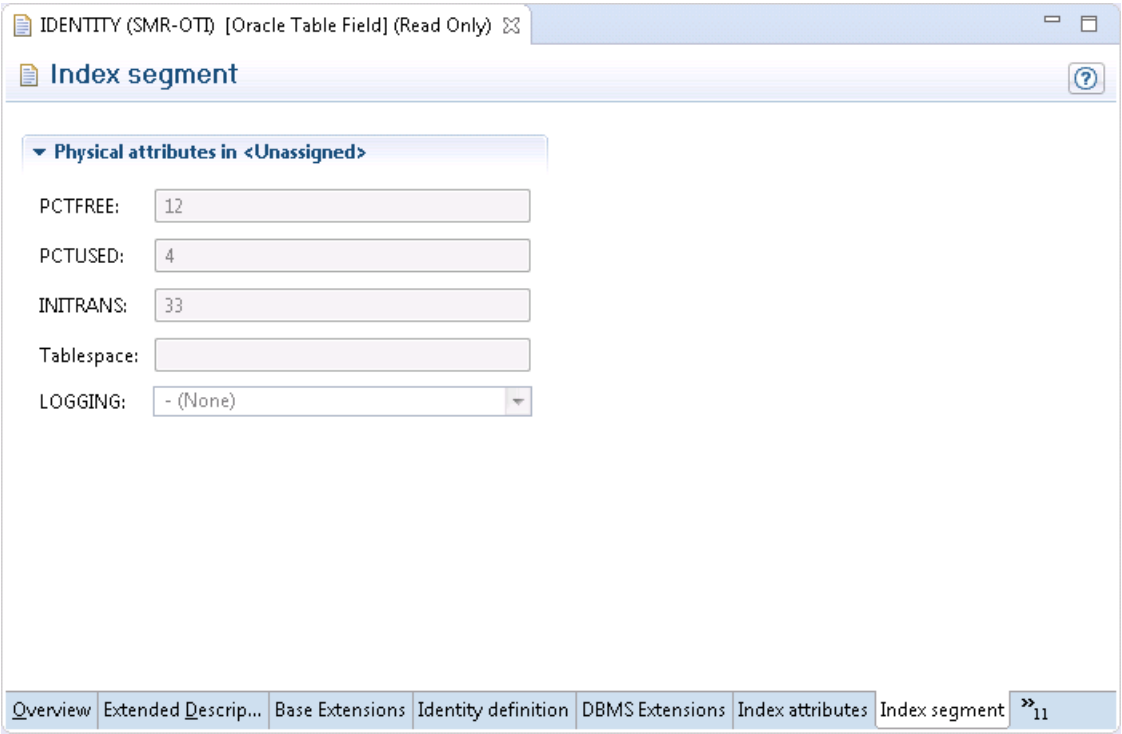
Index Attributes



Parameters		
Index attributes		
REVERSE	Specifies whether the index is a reverse key index.	
	Y	Yes.
	N	No.
VISIBLE	Specifies whether the index is visible.	
	Y	Yes.
	N	No.
PARTIAL	Specifies whether the index is a partial index in a partitioned table.	
	Y	Yes.
	N	No.
SORT	Specifies whether the index is sorted in ascending order.	
	Y	Yes.
	N	No.
COMPRESSION	Specify whether or not to use compression.	
	Y	Yes.

Parameters		
	N	No.
	A	Advanced.
	blank	Not specified. This is the default.
Compression length	Specify the compression length.	

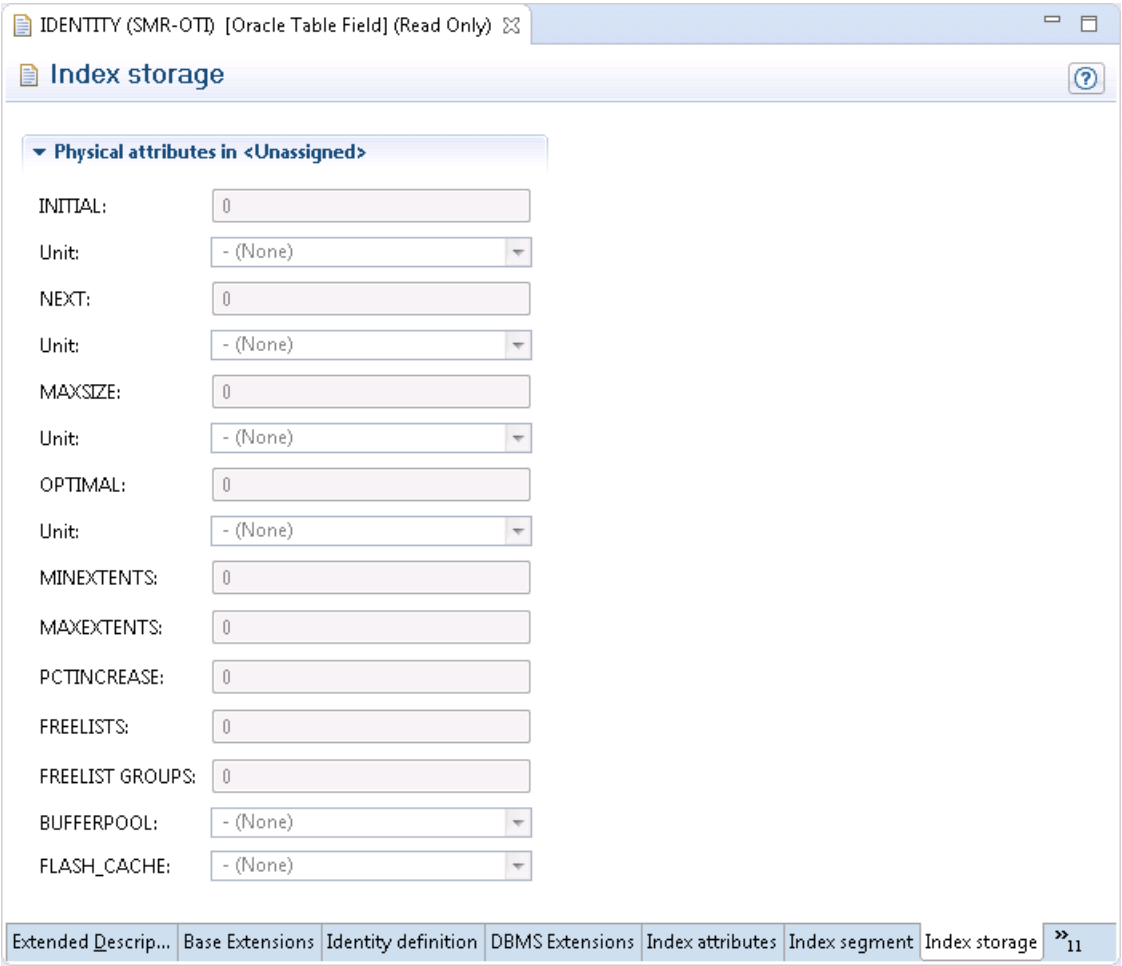
Index Segment



Index segment		
PCTFREE	If an integer from 1 - 99 is specified here, the clause `PCTFREE` *n* is generated in the `CREATE TABLE` statement. `PCTFREE` reserves a set amount of room in every block allocated to a table for future updates to that table's data.	
PCTUSED	If an integer from 1 - 99 is specified here, the clause `PCTUSED` *n* is generated in the `CREATE TABLE` statement. `PCTUSED` specifies the minimum level of space usage that Oracle will maintain for each block of the table.	
INITRANS	If a value from 1 - 255 is entered here, the clause `INITRANS` *n* is generated in the `CREATE TABLE` statement. `INITRANS` is the initial number of transaction entries that are allocated within each block.	
Tablespace	If a tablespace name is entered here, the clause `TABLESPACE` name is generated in the `CREATE TABLE` statement. This name represents the tablespace in which the table will be created.	
LOGGING	Specify whether or not to use the `LOGGING` clause in a `CREATE TABLE` or `ALTER TABLE` statement.	
	Y	Yes.
	N	No.
	F	File system like.

	blank	Not specified. This is the default.

Index Storage



Index storage		
If specified, the values below are used in the `STORAGE` clause generated with the `CREATE TABLE` statement. All of the values below must be specified as integers.		
INITIAL	The size of the first extent allocated when the object is created - the original amount of space allocated to the object. A value for Unit has to be applied in addition:	
	K	Kilobyte.
	M	Megabyte.
	G	Gigabyte.
	T	Terabyte.
	P	Petabyte.
	E	Exabyte.

Index storage		
NEXT	The size of every subsequent extent to be allocated. A value for Unit has to be applied in addition. Possible values for Unit are described under INITIAL.	
MAXSIZE	The `MAXSIZE` clause lets you specify the maximum size of the storage element.	
OPTIMAL	Specifies an optimal size in bytes for a rollback segment.	
MINEXTENTS	The total number of extents to be allocated when the segment is created.	
MAXEXTENTS	The total number of extents, including the first, which can ever be allocated.	
PCTINCREASE	The percent by which each NEXT extent will grow over the last extent allocated.	
FREELISTS	The number of process free lists used to administer the free data blocks.	
FREELISTS GROUPS	Magnitude of the set of free lists.	
BUFFERPOOL	Determines the configuration of the buffer cache.	
	D	Default
	K	Keep
	R	Recycle
	blank	not specified
FLASH_CACHE	Defines the configuration of a second tier of buffer cache on flash disks.	
	D	Default
	K	Keep
	N	None
	blank	not specified

Partitioning Key and Partitioned Index

For a partitioning key and partitioned index, the following can be defined depending on whether they are hash- or range-partitioned.

To define a new partition, use the **New** button. When you have defined more than one partition, a scroll bar is available. To go to a specific partition, use the scroll bar. To delete the currently displayed partition, choose the **Delete** button.

Range-Partitioning Key Definitions

Attributes	
Name	The partition name.
Literal	The literal name.
Segment	Specifies the Segment attributes of the partition.
Storage	Specifies the Storage attributes of the partition.

Hash-Partitioning Key Definitions

Parameter	Description	
Name	The partition name.	
Tablespace	The tablespace name.	
COMPRESSION	Specify whether or not to use compression.	
	Y	Yes.
	N	No.
	A	Advanced.
	blank	Not specified. This is the default.
Compression length	Specify the compression length.	

Default value

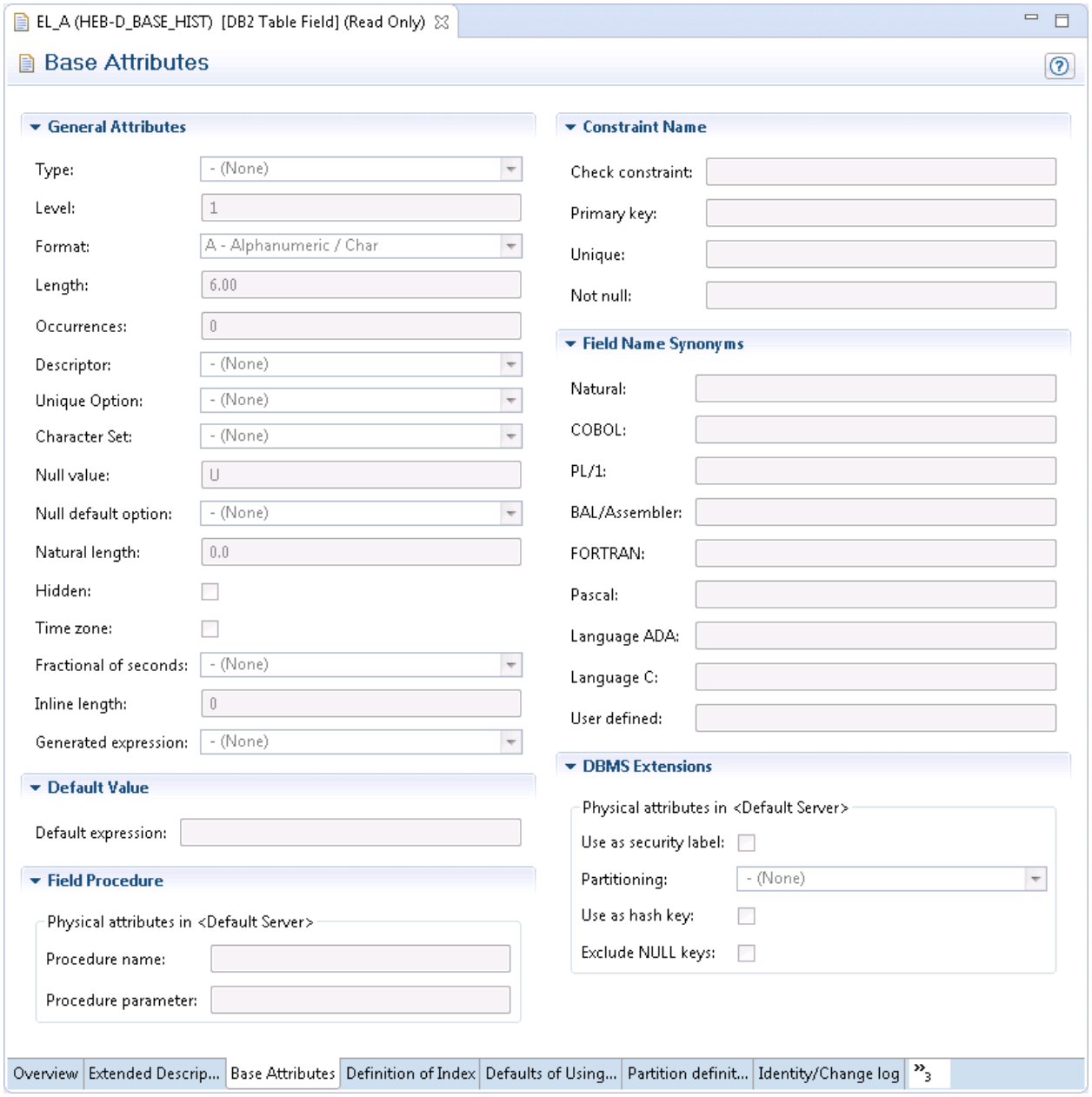
This additional attribute is only applicable for fields in

- Sybase tables with Null value option set to R and Null default option set to Y.
- Adabas D tables, DB2 tables, Informix and Oracle tables with Null value option set to R or U and Null default option set to Y.

Attribute	
Default name	The default specified here is used in the CREATE TABLE statement. Sybase naming conventions apply. See *Naming Conventions for SQL Objects*. **Note:** For Sybase, a default is an object in its own right. For other SQL systems, a default value is specified in the CREATE TABLE statement (not null with default default_expression). For Informix no default name is allowed.
Default expression	An SQL expression can be specified between the angled brackets. This expression determines the default value, for example a constant or function. If specified, this value is always used by the function Generate CREATE statement.

Constraint name

Depending on the field definition, up to four constraint names can be specified.



Attributes	
Check constraint	Constraint name in the respective SQL system for the fact that a linked verification of status S exists.
Primary key	Constraint name for the fact that the field is a primary key.
Unique	Constraint name for the fact that a unique constraint exists (indicated with U in column Unique option of the field object in Predict).
Null/Not null	Constraint name for the fact that the Null or Not null default option is set to Y.

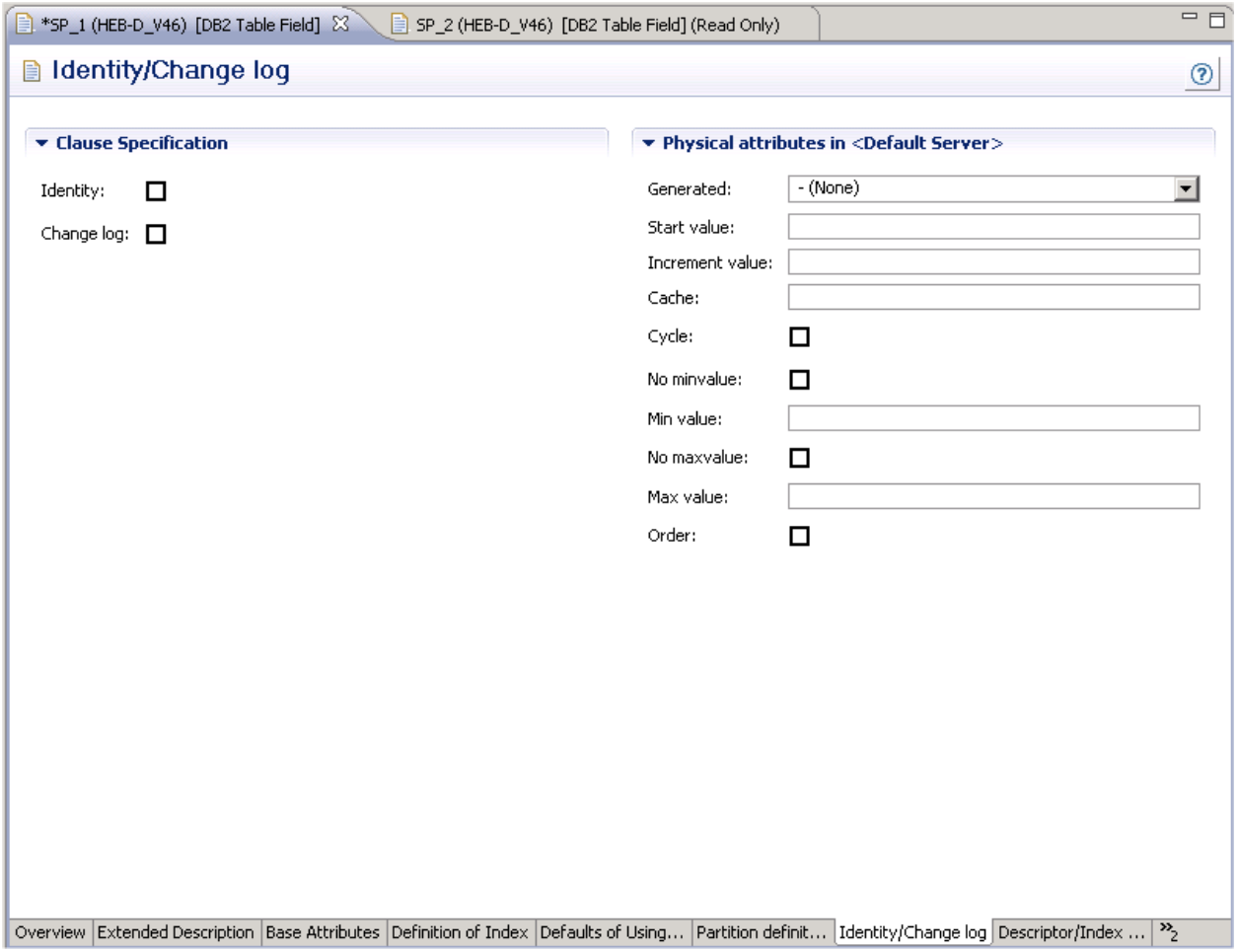
Identity definition / Change log

These additional attributes are only applicable for fields in DB2 or Oracle tables.

Identity definition

The following additional attributes are only applicable for fields of type

- QN (Row ID) or
- a numeric field.



Attributes		
Identity	Specifies whether the column is an identity column for the table.	
As transaction ID		
Generated	Indicates whether DB2 or Oracle generates values for the column. Valid values:	
	A	Always
	D	By default
	blank	None
Start with limit	The first generated value will be the highest/lowest existing value +/- increment value	
Start value	Specifies the first value for the identity column.	
Increment value	Specifies the interval between consecutive values of the identity column.	
Cache	Specifies whether to keep preallocated values in memory.	
Cycle	Specifies that values continue to be generated for this column after the maximum or minimum value has been reached.	
Min value	Specifies the numeric constant that is the minimum value that is generated for this identity column.	

Attributes		
No minvalue	Specifies for DB2 tables whether or not a minimum end point of the range of values for the identity column has been set.	
Max value	Specifies the numeric constant that is the maximum value that is generated for this identity column.	
No maxvalue	Specifies for DB2 tables whether or not a maximum end point of the range of values for the identity column has been set.	
Order	Specifies whether the sequence numbers must be generated in order of request. Valid values:	
	Y	Order
	N	No order
	blank	Not specified

Change log

The following additional attributes are only applicable for fields of type TS (Timestamp).

Attributes		
Change log	Specifies whether the column is a change log column for the table.	
Generated	Indicates whether DB2 generates values for the column. Valid values:	
	A	Always
	D	By default
	blank	None

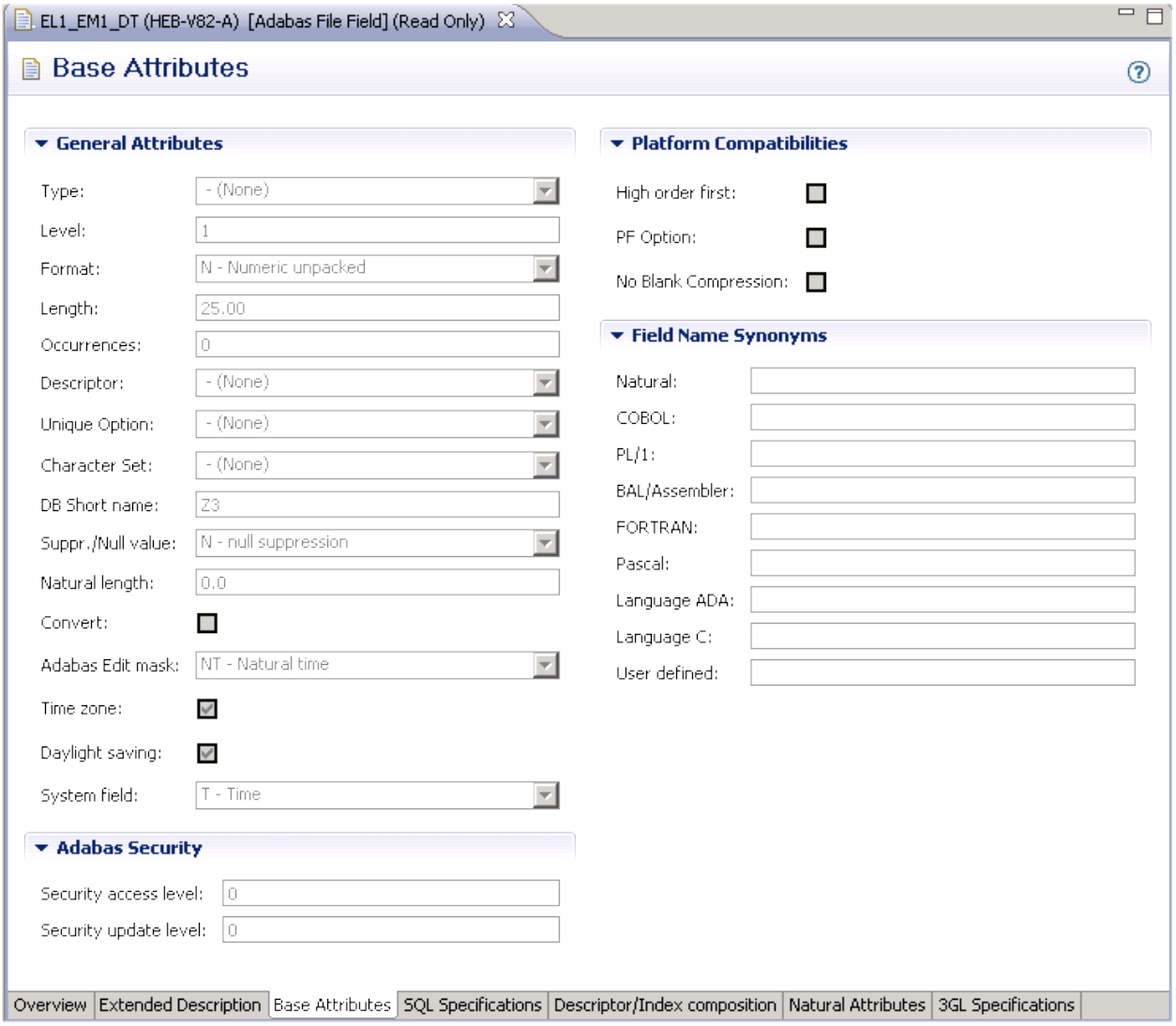
Platform Compatibilities

The following topics are covered:

- No Blank Compression

No Blank Compression

This additional attribute is only applicable for Adabas fields with format A, AV or LO. The option No Blank Compression controls that trailing blanks are suppressed when a value is stored.



Base Extensions

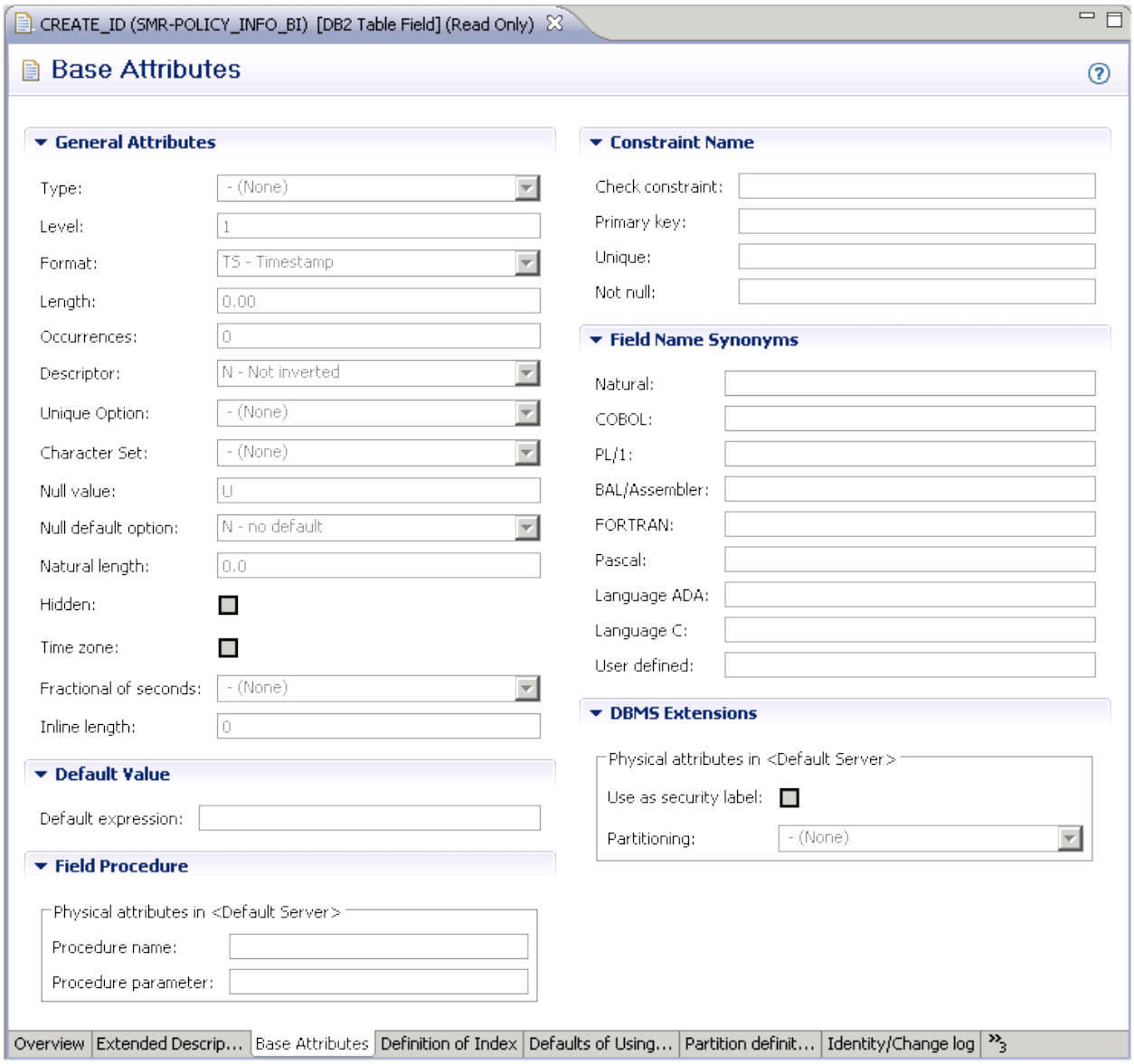
The following topics are covered:

- Fractional of Seconds and Time Zone
- Inline Length
- Generated Expression

- Period

Fractional of Seconds and Time Zone

These additional attributes are applicable for DB2 and Oracle table fields with format TS.



Attributes	
Fractional of seconds	A timestamp value can include a precision of fractional seconds in the range from 0 to 12 (Oracle 0-9). For DB2, if no value is entered, the default value 6 is taken.
Time zone	Provides information on the time difference in hours and minutes between the local time and Coordinated Universal Time (UTC). Adds an additional time zone offset value to the local timestamp value in the format +HH:MM (ahead of UTC) or -HH:MM (behind UTC).

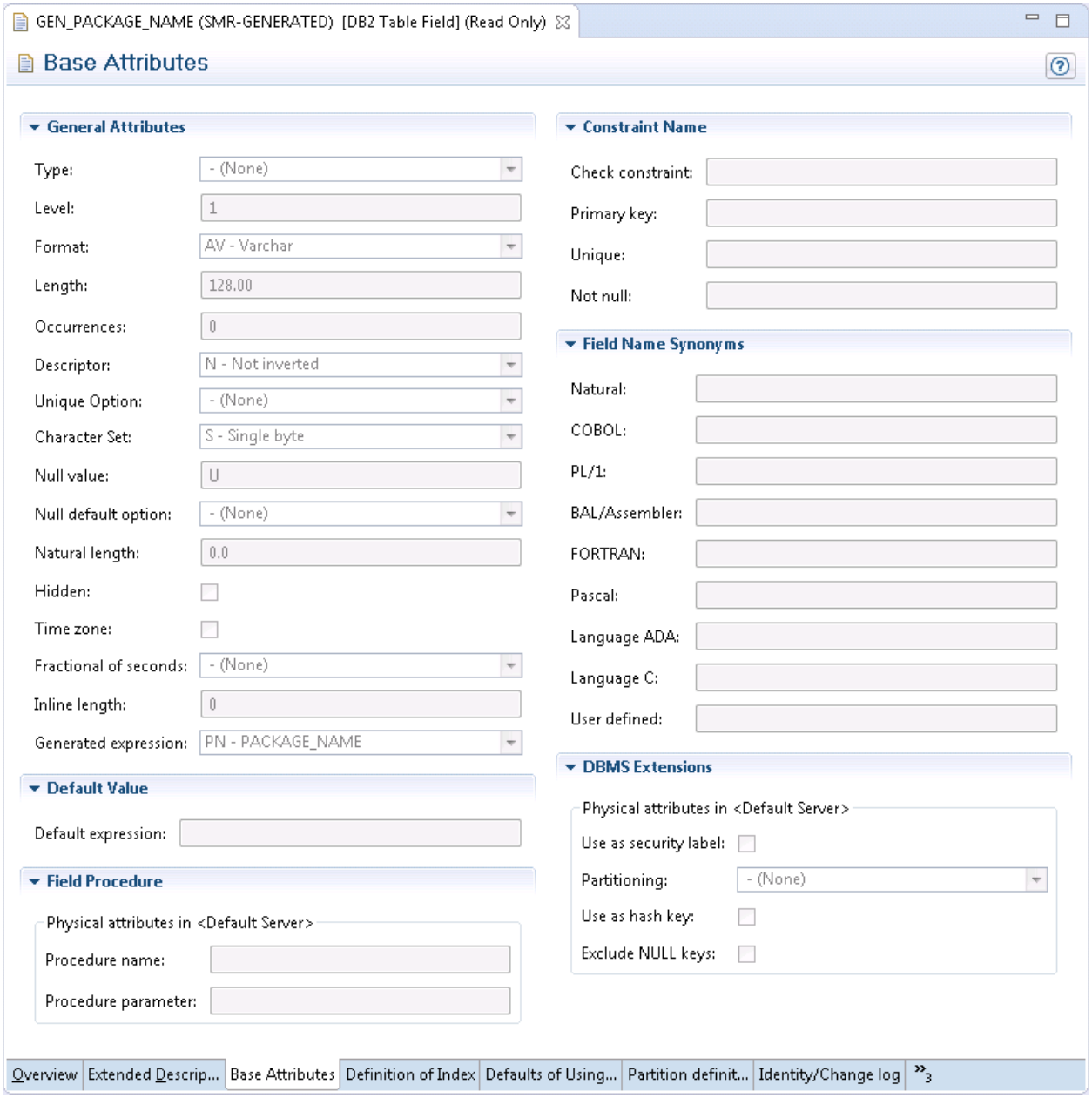
Inline Length

This additional attribute is applicable for DB2 LOB fields.

Attributes	
Inline length	For BLOB, CLOB and DBCLOB columns. Specifies the maximum number of bytes that are stored in the base table space for the column. Must be between 0 and 32680 for a BLOB or CLOB column and between 0 and 16340 for a DBCLOB column..

Generated Expression

These additional attributes are applicable for DB2 table fields with formats A, AL and AV.



Attributes		
Generated expression	For fields of DB2 tables (field types A, AL and AV) you can specify an *as-generated-expression-clause*. The value provided in Generated expression is the expression that is evaluated by DB2 when generating the value of the field. Valid values:	
	DC	Data change operation. Equivalent DB2 expression: `DATA CHANGE OPERATION`
	CC	Client accounting. Equivalent DB2 expression: `CURRENT CLIENT_ACCTNG`

Attributes		
	CN	Client application name. Equivalent DB2 expression: `CURRENT CLIENT_APPLNAM`
	CT	Client correlation token. Equivalent DB2 expression: `CURRENT CLIENT_CORR_TOKEN`
	CU	Client user ID. Equivalent DB2 expression: `CURRENT CLIENT_USERID`
	CW	Client workstation name. Equivalent DB2 expression: `CURRENT CLIENT_WRKSTNNAME`
	CS	Client server. Equivalent DB2 expression: `CURRENT SERVER`
	CI	SQL ID. Equivalent DB2 expression: `CURRENT SQLID`
	SU	Session user. Equivalent DB2 expression: `SESSION_USER`
	PN	Package name. Equivalent DB2 expression: `SYSIBM.PACKAGE_NAME`
	PS	Package schema. Equivalent DB2 expression: `SYSIBM.PACKAGE_SCHEMA`
	PV	Package version. Equivalent DB2 expression: `SYSIBM.PACKAGE_VERSION`
	blank	None.

Period

The field option Period specifies whether the value of the end column is included or excluded in the period BUSINESS_TIME.

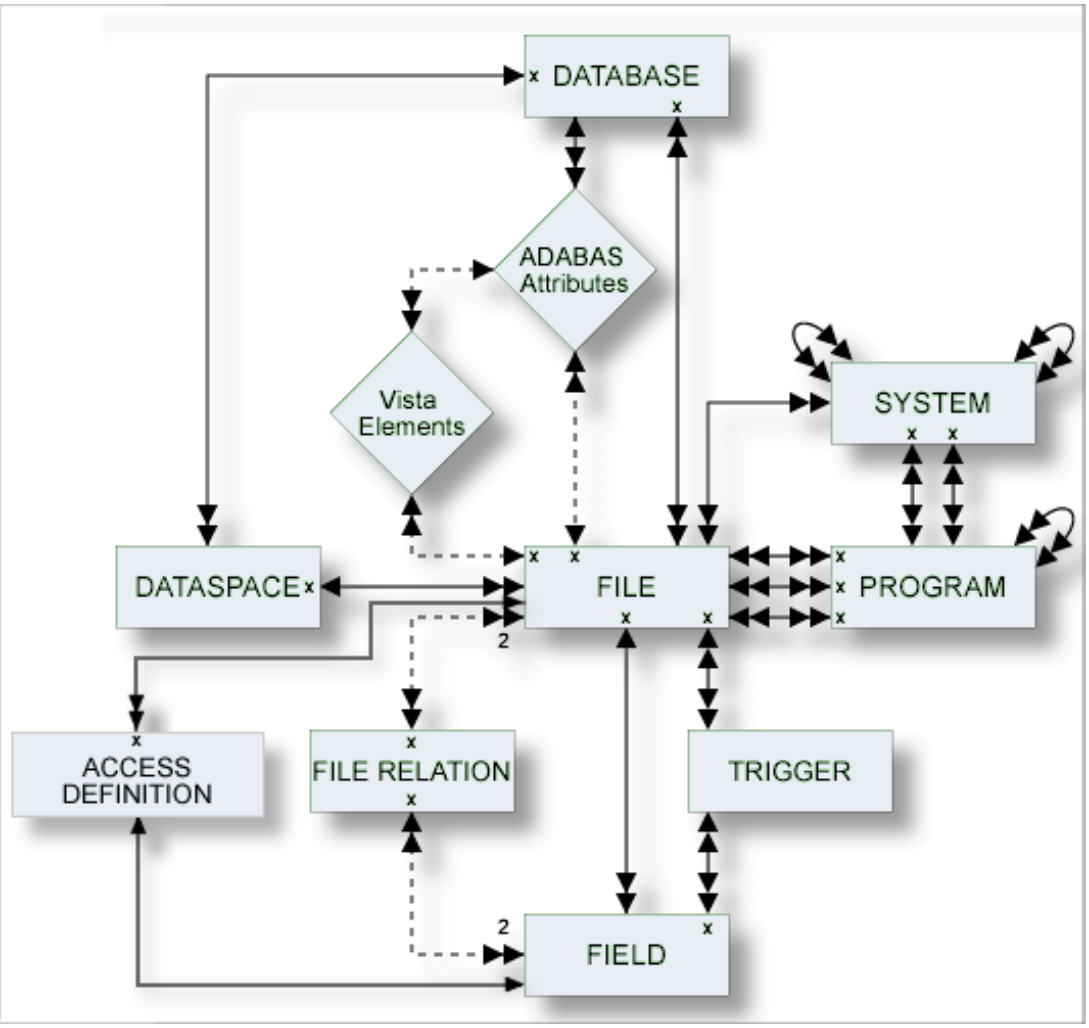
This option can be used only for fields named BUSINESS_TIME in files of type *D* and *MT*.

Attributes		
Period	Specifies whether the value of the end column is included or excluded in the period BUSINESS_TIME.	
	I	Inclusive
	E	Exclusive

VI File

With Predict objects of type File, file structures can be defined for a wide variety of data storage systems and for use with different programming languages.

In addition, Adabas attributes can be defined for Adabas files with function "Attributes of Link" of mode "Contains FI".



Where to Find Detailed Information on Defining Distributed Data Structures

Basic information on attributes of files and how to execute file-specific functions is given in the sections below. If you wish to define data using simple files in a database not accessible via a network, you will find all the required information in the sections below. Additional information needed when defining complex data distribution structures using Adabas Vista or Entire Transaction can be found in the respective sections of the *Predict and Other Systems* documentation.

The description of object type File is organized under the following headings:

Maintaining Objects of Type File

Adabas Files, File Type A

File Types Conceptual, Standard and Other

SQL File Types

Adabas SQL Server

Adabas D

DB2

Informix

Ingres

Oracle

Sybase

General SQL File, File Type X

RDB

IMS

VSAM

ISAM

Entire System Server

File-Specific Maintenance

Rippling - Ensuring Consistent Data Definitions

13 Maintaining Objects of Type File

- Common File Attributes 150
- Defining Basic File Attributes 153

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Common File Attributes

The following attributes are applicable to all or most file types.

- File ID
- File Type
- File number
- Natural Construct Parameters

File ID

For naming conventions valid for all object types see ***Naming Conventions***.

Special naming conventions apply to SQL file types. See overview in the section ***Naming Conventions for SQL Objects***.

File Type

A file object has one of the following types. The file type must be compatible with the database in which it is contained.

File Type	Description
A	Adabas File
AT	Adabas Cluster Table
B	Adabas SQL view
C	Conceptual File
D	DB2 table
E	DB2 view
F	rdb file
I	IMS segment
J	IMS segment layout
K	IMS userview
L	Logical VSAM file
M	ISAM file
O	Other file
P	Entire System Server file
Q	Entire System Server userview

File Type	Description
R	Logical VSAM view
S	Sequential file
T	RMS file
U	Adabas userview
V	VSAM file (physical)
W	VSAM userview
X	General SQL file
Z	Standard file
1	LEASY
2	ISAM BS 2000
OT	Oracle table
OV	Oracle view
BT	Adabas D table
BV	Adabas D view
JT	Ingres table
JV	Ingres view
YT	Sybase table
YV	Sybase view
XT	Informix table
XV	Informix view
IT	Intermediate table
IV	Intermediate view
MT	DB2 query table

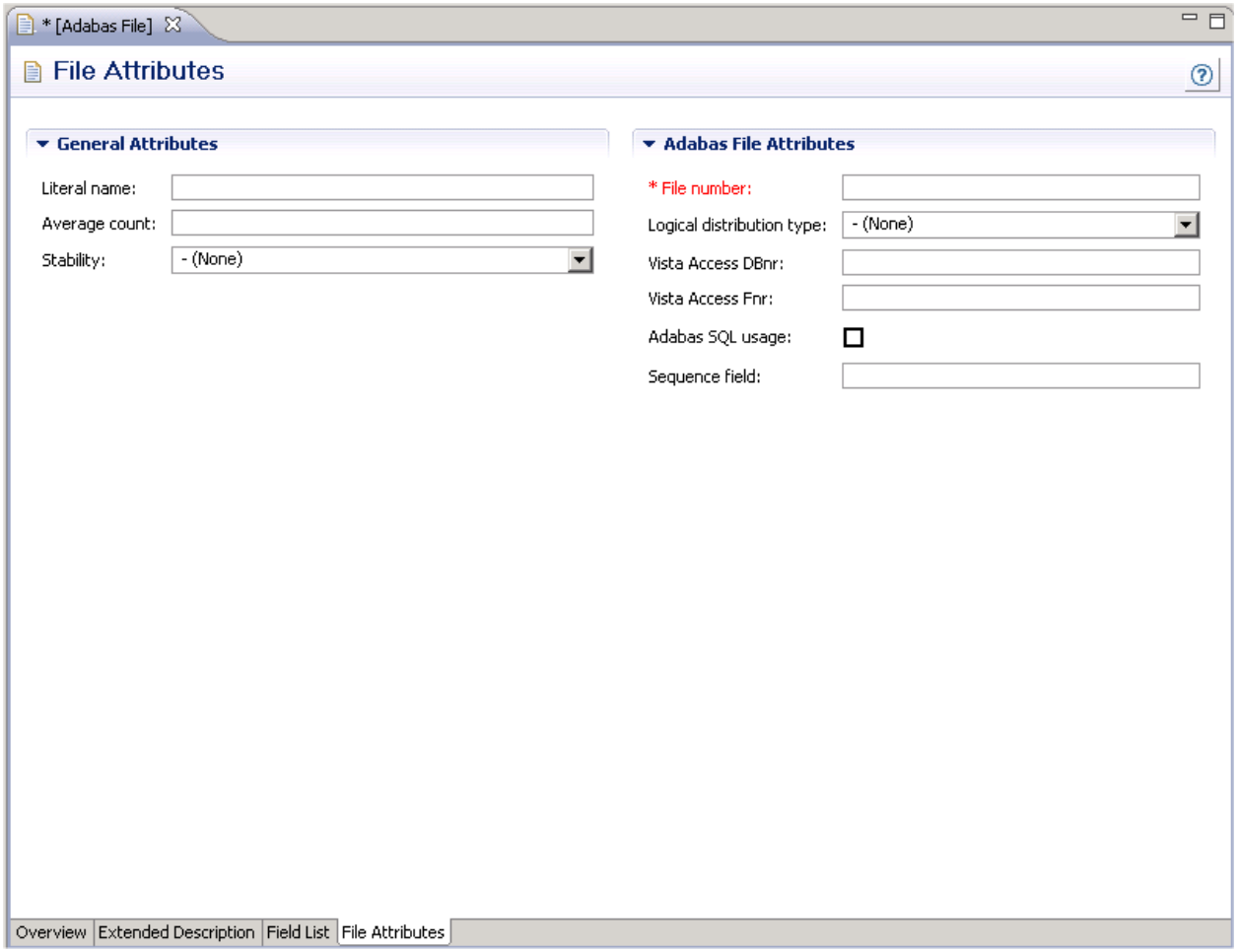
File number

The number of the file. The possible value depends on the file type:

File Type	File Number
AT, J, K, Q, R, U	File number is taken from the specified master file
B, D, E, I, X, BT, BV, IT, IV, JT, JV, OT, OV, XT, XV, YT, YV, MT	not applicable
A, V, L, T, P, 1, 2	1 - 32767
Other file types	1 - 99999

Natural Construct Parameters

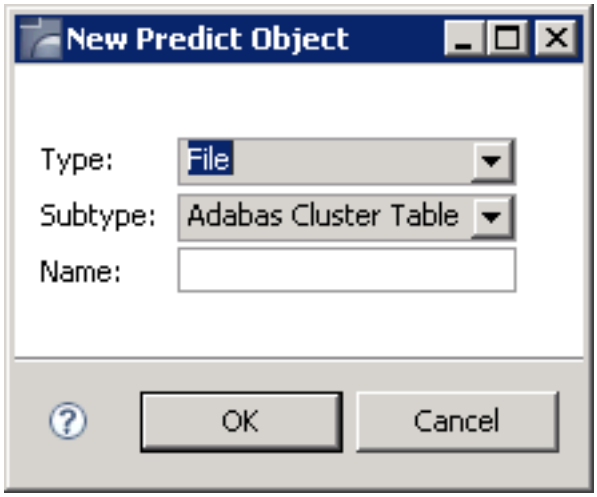
The following parameters are only relevant if you are using Natural Construct.



Parameter	Description	
Literal name	String to be used by Natural Construct in messages issued to confirm (un)successful access of a file via a DDM generated from the Predict file object.	
Average count	The average number of records contained in the file.	
Stability	Indicates how permanent the data contained in the file is.	
	F	Fixed. The file contains information which will always be valid, for example days of the week.
	S	Stable. The file contains information which does not change very often, for example file EMPLOYEES.
	V	Volatile. The file contains information which is constantly being updated, for example an invoice file.
	blank	Not specified (default value).

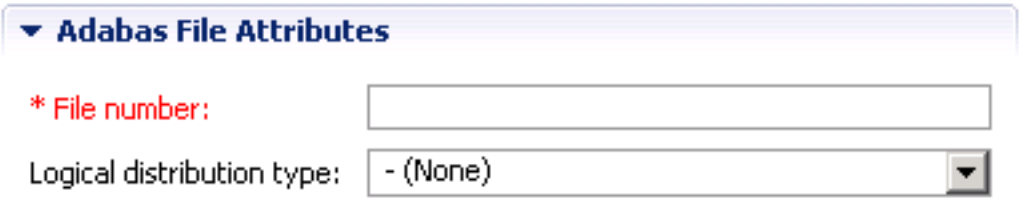
Defining Basic File Attributes

When you add a file, you first have to specify the file type in the **New Predict Object** dialog box.



When you choose the **OK** button, a file type-specific window appears. The file type is indicated in the title bar.

The following sample window shows the parameters which apply to all types of files.



General Parameters	
Has Master Files	This parameter can be found in the **Mandatory Association** section. This section is only shown for the file types listed below. For the file types listed below, enter the ID of the related file. The type of related file is given below:

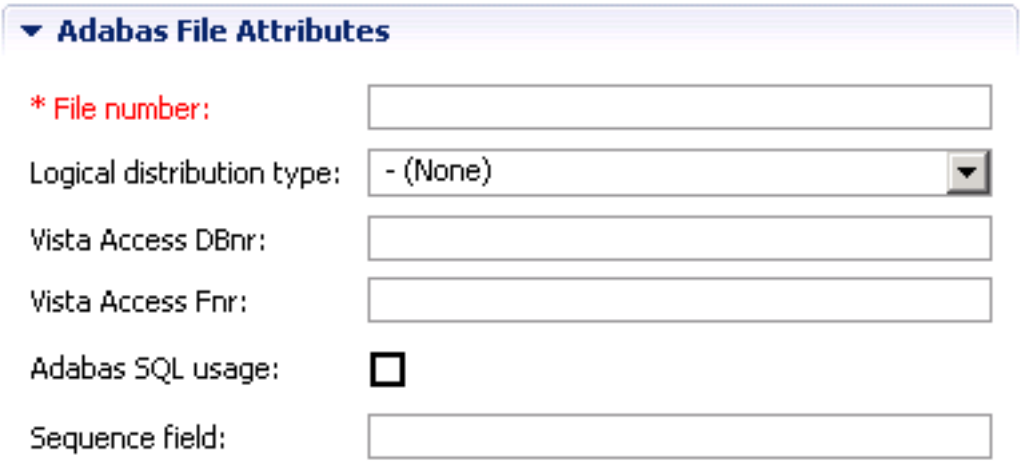
File Type	Type of Master File	
AT	A	Adabas File
J and K	I	IMS segment
L and W	V	Physical VSAM File
Q	P	Entire System Server File
R	L	Logical VSAM File
U	A	Adabas File

General Parameters		
File number	See the table of possible values in the section *File number*.	
Logical distribution type	How the logical file is to be stored:	
	E	Expanded
	P	Partitioned
	N	Propagator file. Not applicable when defining data distribution for Adabas Vista.
	blank	Simple file (default).
	Note: This parameter is only applicable to files of type Adabas. For files of other types, this parameter must be blank.	

14 Adabas Files, File Type A

- Add/Modify a File 156
- Modifying Adabas Attributes 157
- Phys. distribution attr. 159
- Adabas Security Definition 159
- Extent Allocation - Size Specifications For More Than One Extent 161
- Modifying ADAM Descriptor Definition 162
- Encodings 163

Add/Modify a File



Note: Parameters common to all object types are described under ***Global Attributes***. For parameters common to all file types, see ***Common File Attributes***.

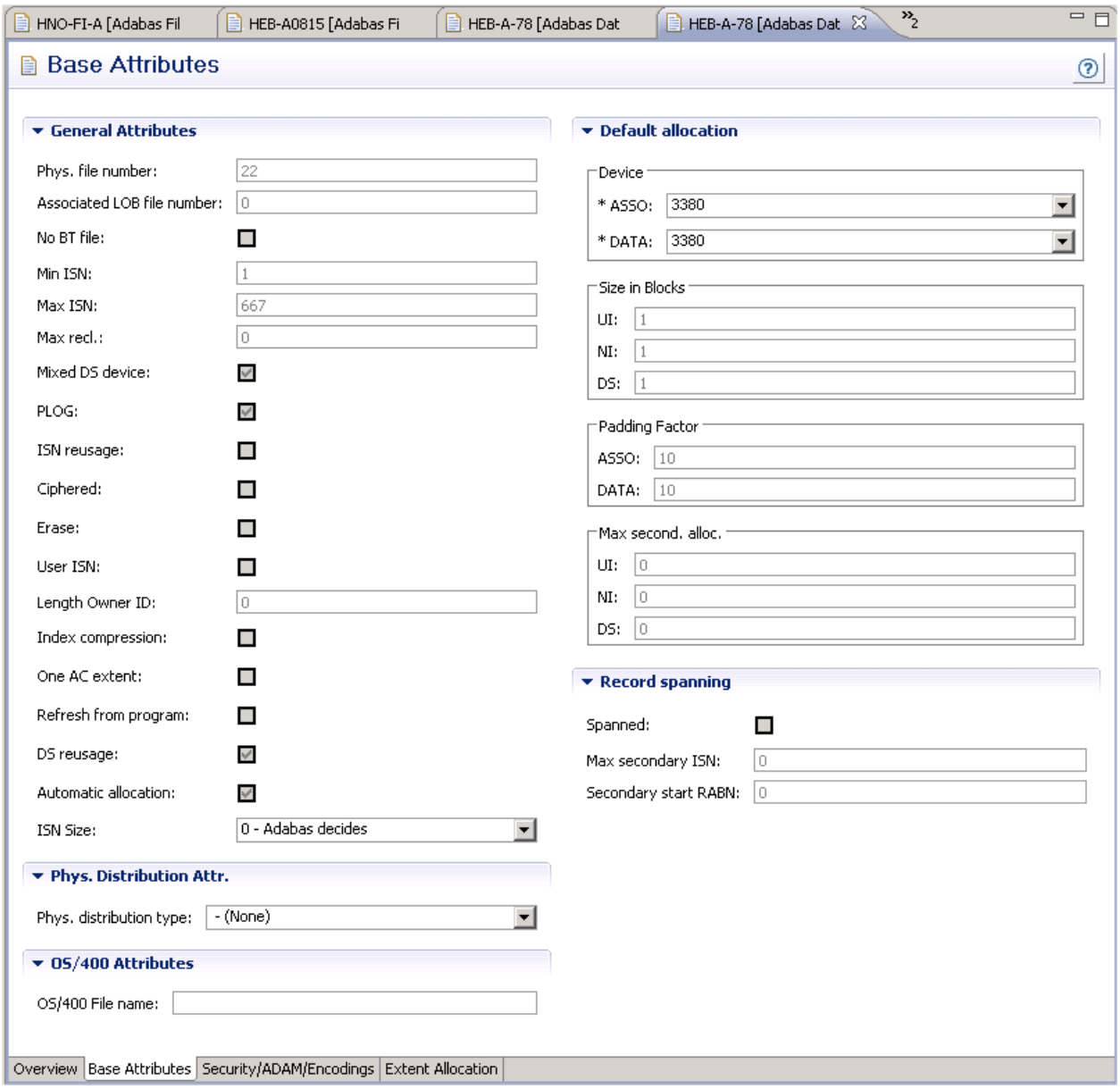
Parameters	
Sequence field	The descriptor to be used by Natural for logical sequential reading. Determines the sequence in which records are delivered by the READ LOGICAL statement. The GENERATE DDM function will use this field as the default READ LOGICAL field in the Natural data definition module.
Vista Access DBnr, Vista Access Fnr	The L-DBnr and L-Fnr are used as database and file number for function Generate DDM if the parameter Use Vista access-nr is set to Y or T in the Generate DDM menu. Valid values are 0 to 65535 for DBnr and Fnr. No check for uniqueness is performed. **Note:** This parameter should not be confused with the Vista parameter Vista number, which is used to identify a file uniquely within a network. See *Including the Definition in the Vista Table* in the section *Adabas Vista* in the *Predict and Other Systems* documentation
Adabas SQL usage	File is accessible via Adabas SQL Server.

Modifying Adabas Attributes

This can only be defined via the "Contains FI" association of the object type Database.

When the node for such a type of association is expanded in the list of all documentation objects, you can select a field and choose **Attributes of Link** from the context menu.

Note: If you do not modify the values in this window, the default values set in the Default Adabas Attributes screen are taken. See the *Predict Administration* documentation.



Parameters	
Required attributes	
Phys. file number	If a database is specified, the file number is taken as a physical file number automatically if this is possible.
AssociatedLOB file number	An associated LOB file can be specified for a base file containing fields of type LO. 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.
Min ISN	ADALOD LOAD parameter MINISN.
Max ISN	ADALOD LOAD parameter MAXISN.

Device and Size Specification for Adabas Files

The device type and the size of the Upper Index (UI), Normal Index (NI) and Data Storage (DS) can be specified.

Four characters specify the type of device used to store this part of the file. This device type must already be defined in the Predict database object containing this file. When this device type is changed in the database, the same change should be made in every file object contained in the database.

DATA padding factor	ADALOD LOAD parameter DATAPFAC.
ASSO padding factor	ADALOD LOAD parameter ASSOPFAC.
Device	The device type of the Upper Index (UI), Normal Index (NI) and Data Storage (DS). The device type for Data Storage is ADALOD LOAD parameter DSDEV.
Size (Cylinders/Blocks)	ADALOD LOAD parameters UISIZE (Upper Index), NISIZE (Normal Index) and DSSIZE (Data Storage).

Note: See also *Extent Allocation*.

Loading attributes	
Max recl.	ADALOD LOAD parameter MAXRECL.
ISN reusage	ADALOD LOAD parameter ISNREUSE.
User ISN	ADALOD LOAD parameter USERISN.
One AC extent	ADALOD LOAD parameter NOACEXTENSION.
DS reusage	ADALOD LOAD parameter DSREUSE.
Maximum secondary allocation	ADALOD LOAD parameters MAXUI (Upper Index), MAXNI (Normal Index) and MAXDS (Data Storage).

For attributes on the **Base Attibutes** tab not described in this section, see *Miscellaneous Attributes*.

Phys. distribution attr.

This can only be defined via the "Contains FI" association of the object type Database.

When the node for such a type of association is expanded in the list of all documentation objects, you can select a field and choose **Attributes of Link** from the context menu.

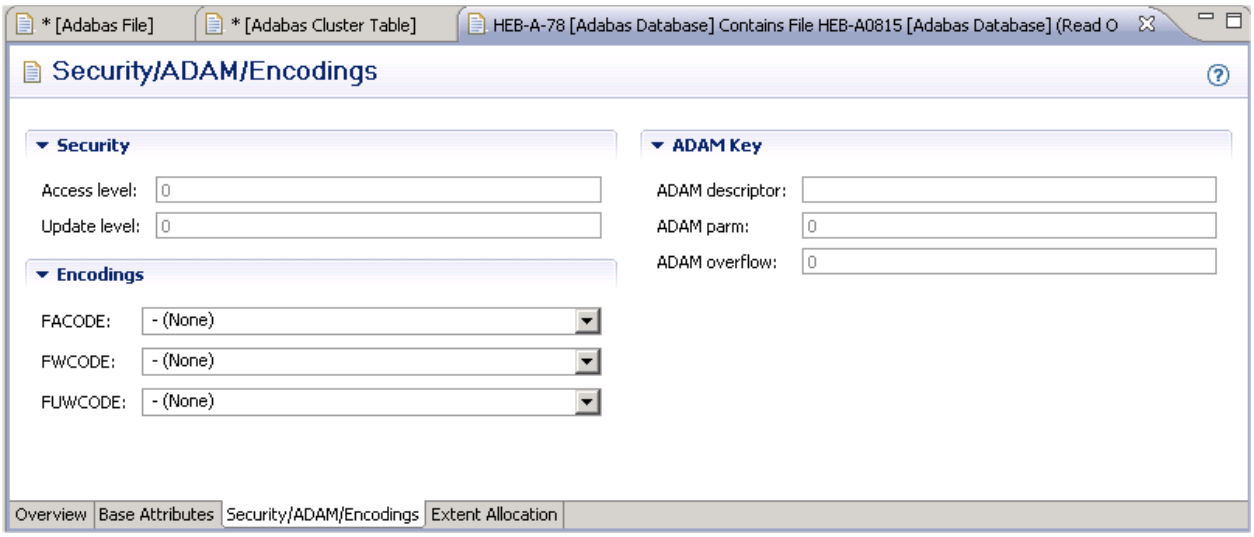
Parameters	
Phys. distribution type	The types for the physical file are limited by the logical distribution type, as shown below:

Physical distribution Type		Logical distribution Type	
E	expanded	E	expanded
P	partitioned	P	partitioned
blank	simple File		any

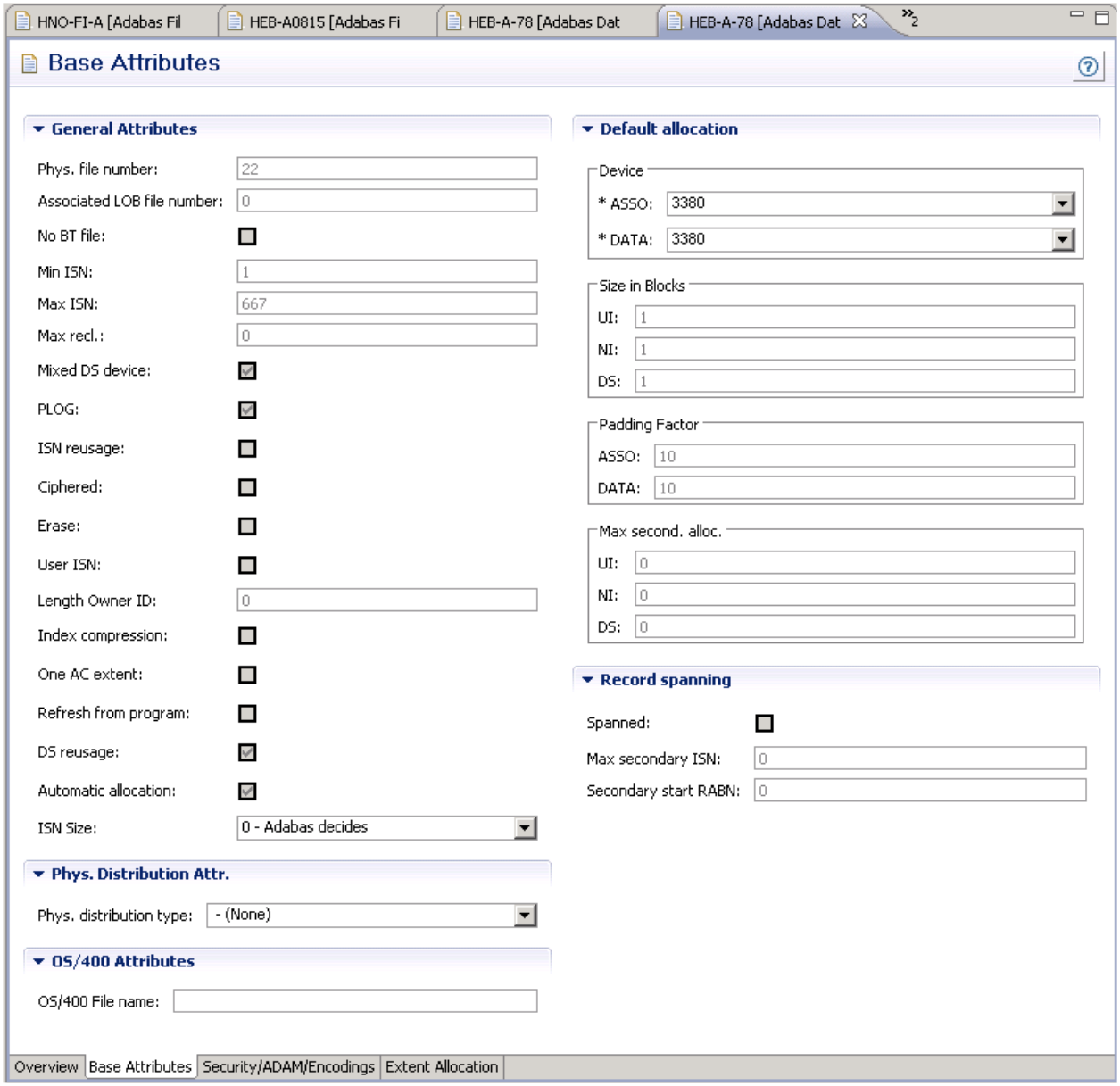
Adabas Security Definition

This can only be defined via the "Contains FI" association of the object type Database.

When the node for such a type of association is expanded in the list of all documentation objects, you can select a field and choose **Attributes of Link** from the context menu.



Parameters	
Access level	The Adabas access security level of the file.
Update level	The Adabas update security level of the file.



Parameters		
Ciphered	Y	The file is a ciphered file.
Length Owner ID	Length of internal Owner ID of a multi-client file.	
Refresh from program	Adabas parameter PGMREFRESH. See the Adabas DBA documentation.	
Automatic allocation	Y	Adabas will automatically allocate and deallocate extents. See the *Adabas Reference* documentation.
PLOG	Y	Database runs with protection log. Linux only.
ISN Size	Length of ISN. Valid values: 0, 2, 3 and 4. For Adabas/Linux: 0, 2 and 4 are valid. For mainframes: 0, 3 and 4 are valid.	
Erase	Y	For Adabas/Linux. All index and data storage blocks are overwritten with zeroes when they are returned to the free space table.
Index compression	Y	Adabas reduces space requirements for the index and for data storage by removing redundant information on an individual descriptor basis.
No BT file	Y	Exclude file from BACKOUT TRANSACTION processing.
Record spanning	Spanned	When record spanning is enabled, the size of compressed records in a file may exceed the maximum data storage block size. Default is Y.
	Max secondary ISN	Defines the initial size of secondary ISNs.
	Secondary start RABN	If spanned records are used, a secondary address converter is used to map the secondary ISNs to the RABNs of the Data Storage blocks where the secondary records are stored.
Max occ system fields	This parameter specifies the maximum number of values generated for a system-generated multiple-value field.	

For attributes on the **Base Attibutes** tab not described in this section, see ***Modifying Adabas Attributes***.

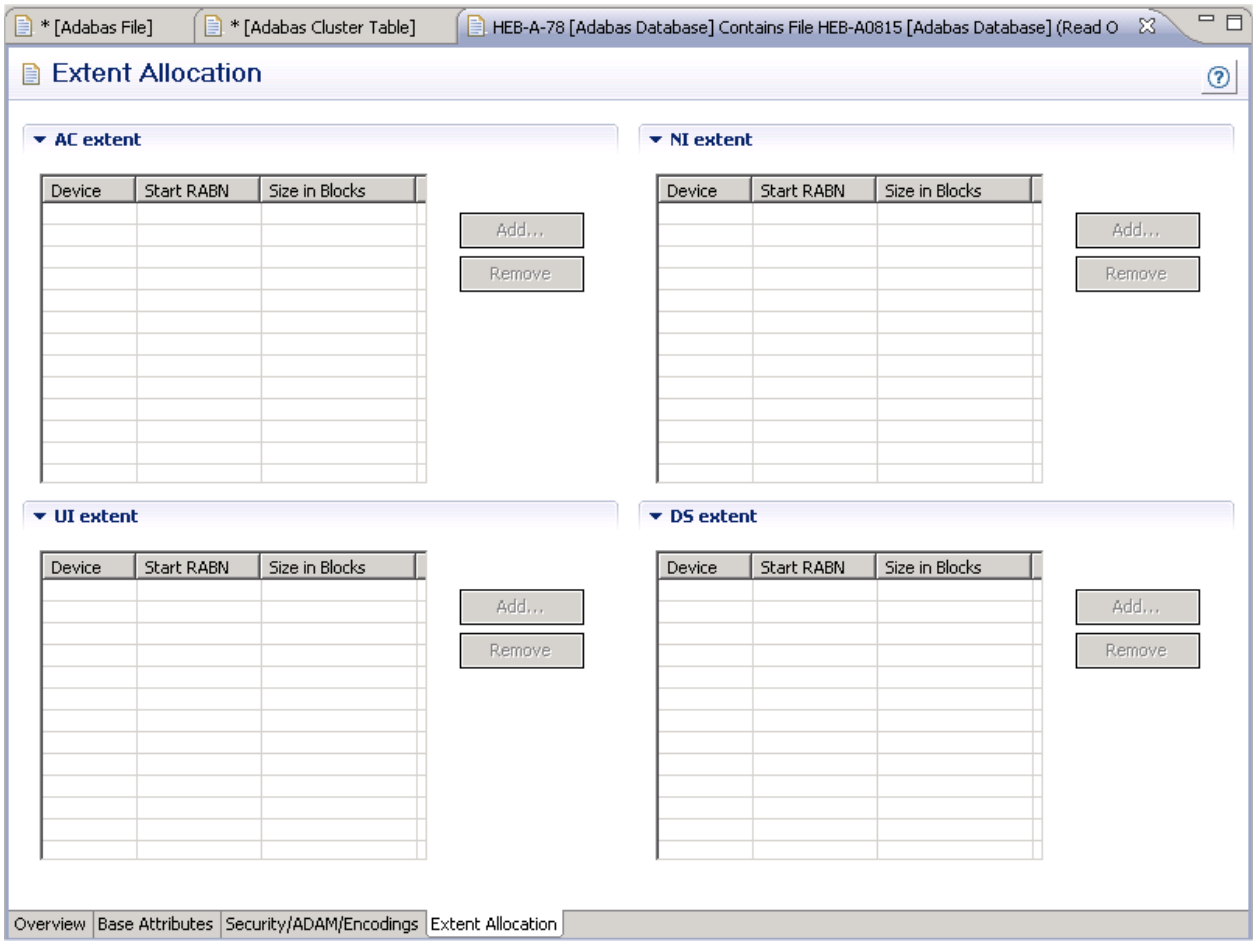
Extent Allocation - Size Specifications For More Than One Extent

This can only be defined via the "Contains FI" association of the object type Database.

When the node for such a type of association is expanded in the list of all documentation objects, you can select a field and choose **Attributes of Link** from the context menu.

More than one extent can be specified.

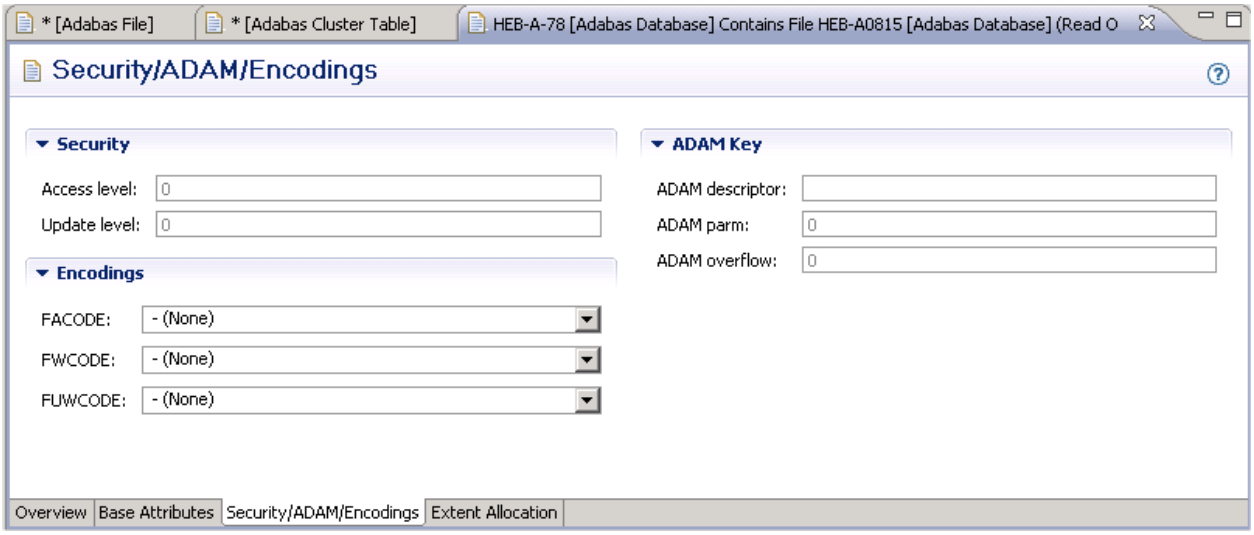
The size and first RABN (Start) of the Address Converter (AC), Upper Index (UI), Normal Index (NI) and Data Storage (DS) can be specified for up to 99 extents. The scroll bars are activated when more than one extent has been specified.



Modifying ADAM Descriptor Definition

This can only be defined via the "Contains FI" association of the object type Database.

When the node for such a type of association is expanded in the list of all documentation objects, you can select a field and choose **Attributes of Link** from the context menu.



Parameters	
ADAM descriptor	Fields to be used as ADAM descriptor. ADALOD LOAD parameter ADAMDE.
ADAM parm	ADALOD LOAD parameter ADAMPARM.
ADAM overflow	ADALOD LOAD parameter ADAMOFLOW.

Encodings

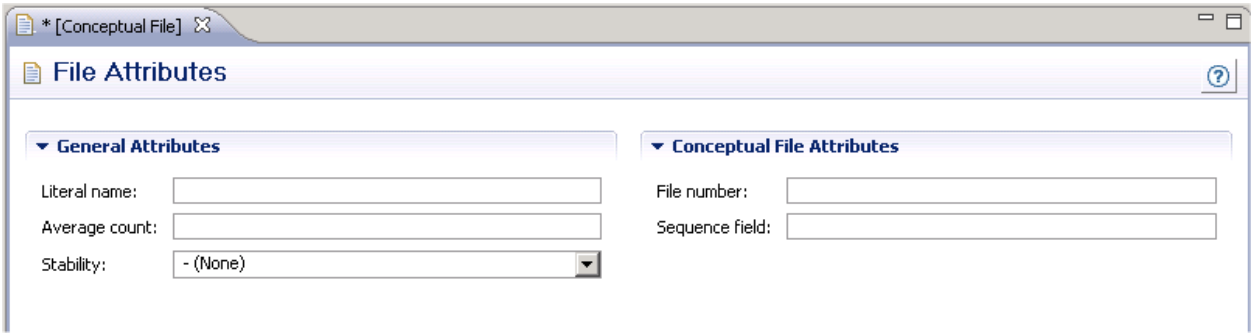
This can only be defined via the "Contains FI" association of the object type Database.

When the node for such a type of association is expanded in the list of all documentation objects, you can select a field and choose **Attributes of Link** from the context menu.

Universal encoding support of an Adabas file can be defined in the tab shown below.

Note: See the *Adabas Administration* documentation for further information on this topic.

15 File Types Conceptual, Standard and Other



Parameters	
File number	Files of these types can have a file number from 0 - 99999.
Sequence field	The function Generate DDM will use this field as the default READ LOGICAL field in the Natural data definition module. For conceptual files for documentation and later use.
Literal name, Average count, Stability	Only applicable if you are using Natural Construct. See *Natural Construct Parameters*.

16 SQL File Types

- Naming Conventions for SQL Objects 168
- Common Parameters for SQL File Types 169
- Field Lists of SQL Views 175
- Editing the Subquery of an SQL View 176

Predict offers various file types for documenting tables and views of the SQL systems listed below. The file objects which document the SQL tables and views can be used to generate SQL CREATE statements, DDMs and copy code members for 3GLs. The CREATE statements are stored as Natural members in file FDIC.

Naming Conventions for SQL Objects

Special naming conventions apply to the following objects in Predict

- SQL file types. See table below.
- Fields linked as children to these file types
- Constraint names
- Correlation names
- Tablespace for Oracle
- The file IDs must be fully qualified. A fully qualified ID consists of three parts:
  - Hyphen to separate creator/schema from table/view name
  - Table/view name. The maximum length depends on the SQL system. See table below.
- Fully qualified IDs may not exceed 32 characters.
- The permitted characters listed in the table below apply to creator/schema and table/view name.

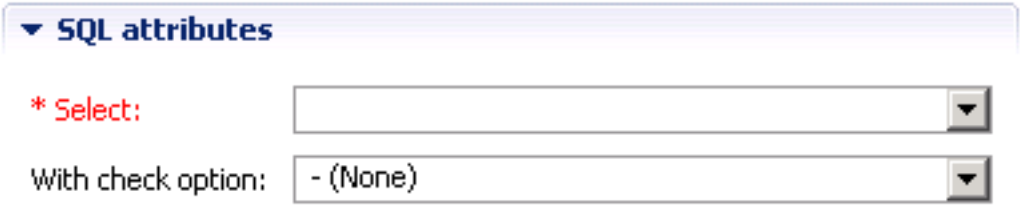
	Filetype								
		AT,B, A(SQL)	BT, BV	D, E, IV, IT	JT, JV	OT, OV	X	XT, XV	YT, YV
Convention	Maximum length of table/view name	32	18	18	24	30	18	18	30
	Upper case			Y		Y	Y		
	Upper/lower case	Y	Y		Y			Y	Y
	'_' allowed at first pos.			Y	Y				Y
	'#' allowed at first pos.		Y	Y					
	'$' allowed at first pos.		Y	Y					
	'@' allowed at first pos.		Y	Y					
	'_' allowed from second pos.	Y	Y	Y	Y	Y	Y	Y	Y
	'#' allowed from second pos.		Y	Y	Y	Y	Y		Y
	'$' allowed from second pos.		Y	Y	Y	Y	Y		Y
	'@' allowed from sec. pos.		Y	Y	Y				Y
	Numbers allowed from second pos.	Y	Y	Y	Y	Y	Y	Y	Y

Type-specific rules are also given in the respective parts of this section.

Common Parameters for SQL File Types

The following parameters are valid for all or most SQL file types.

SQL Attributes



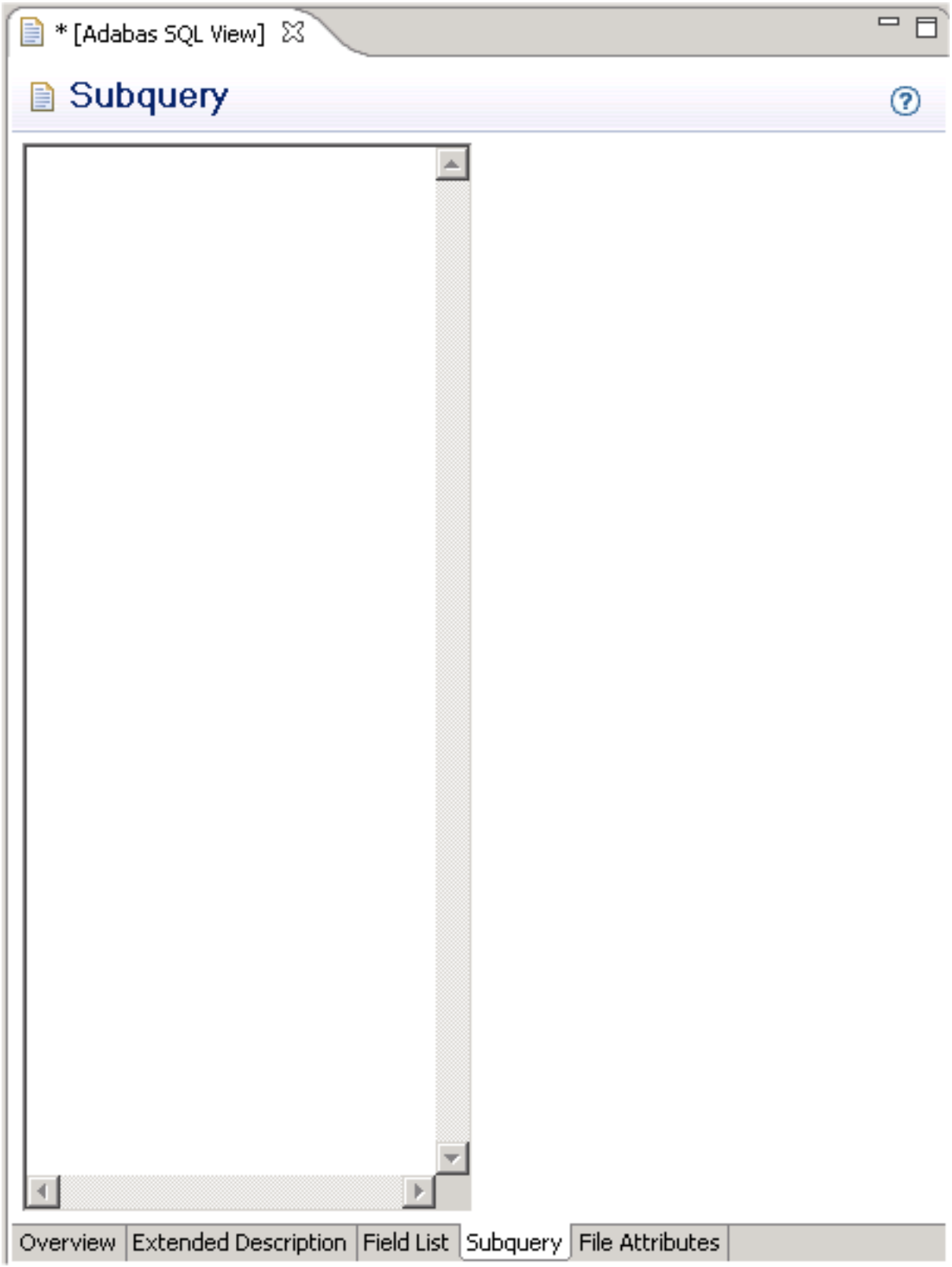
These parameters apply to all SQL views.

Select	A	Select all: Redundant duplicates are not eliminated.
	D	Select distinct: Redundant duplicates are eliminated.
With check option	Y	All inserts and updates to the view are checked against the view definition.

Additional attributes / Associations

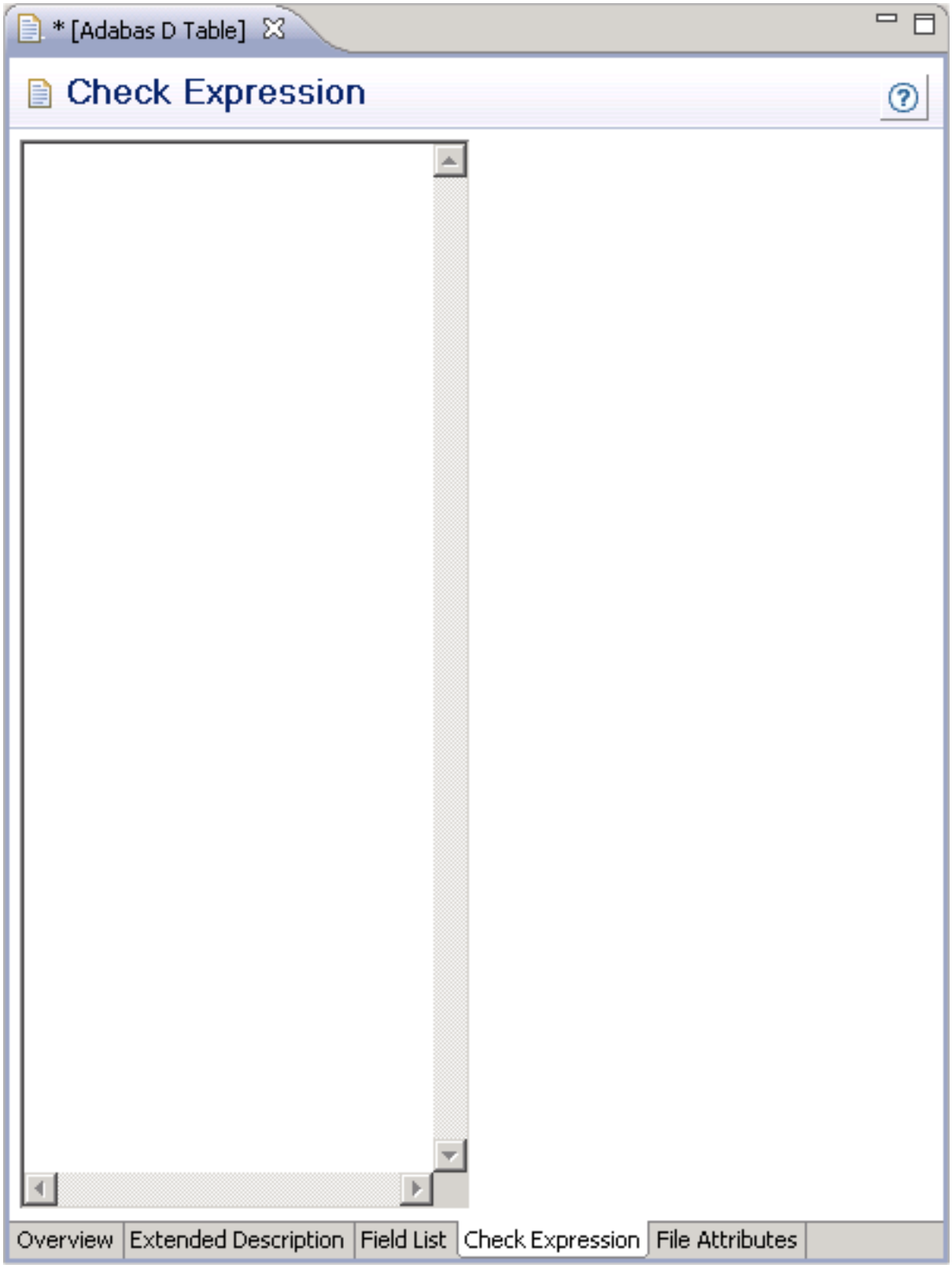
Profile options are described in the section *Defaults* in the *Predict Administration* documentation. The editors are described in the section *Editors in Predict* in the *Predict Reference* documentation.

Subquery


* [Adabas SQL View]
Subquery
Overview
Extended Description
Field List
Subquery
File Attributes

This option is available for all SQL views.

Check Expression



This option is available for the following SQL tables:

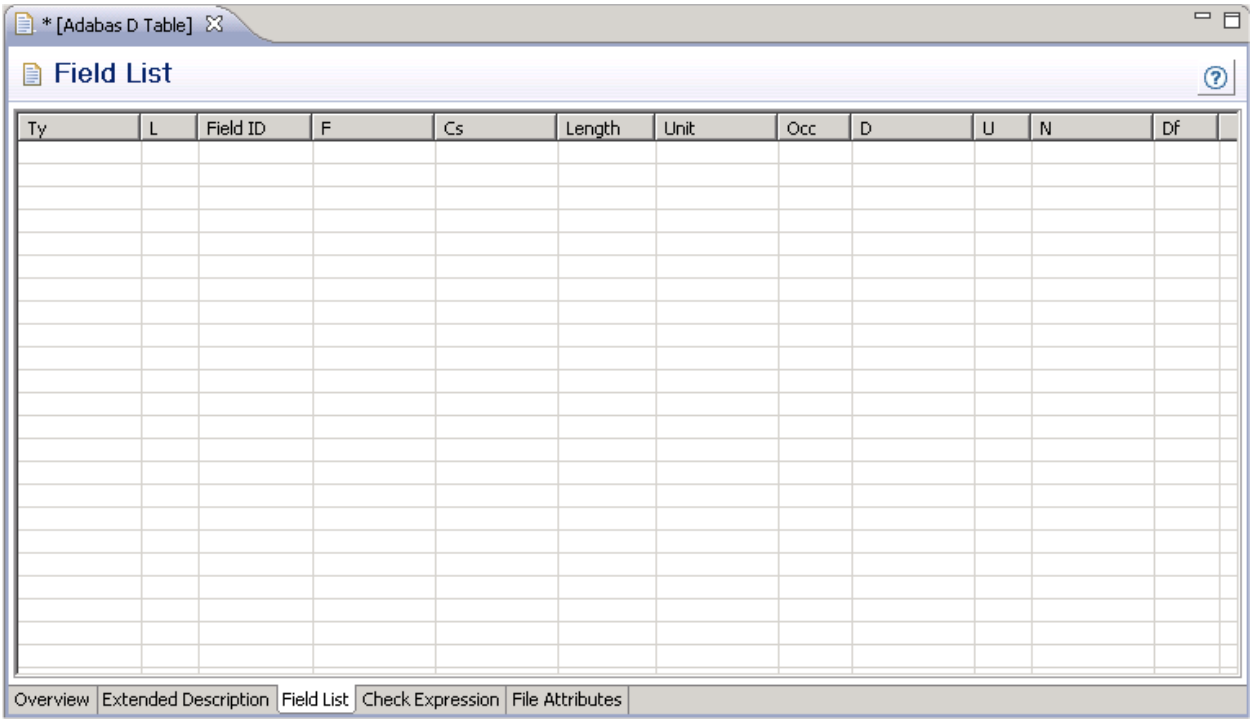
- Adabas D
- DB2
- Oracle
- Informix
- Ingres
- Sybase

It is also available for the following file type:

General SQL file

No special checks are performed when check expression is saved.

Field Lists of SQL Views



Column	Meaning
Ty	Field type.
L	Field level.
Field ID	ID of field object documenting the SQL view. The ID of the field object in Predict documenting a field in a view can differ from the name of the field in the original table or view.

Adding new Fields to Field Lists of SQL Views

New fields can easily be inserted into the field list of an SQL view using one the following two methods:

Manually

Enter parameter Field ID described above. See the section *Naming Conventions for SQL Objects*.

Import

Choose Import to import fields from other SQL tables or views into the current field list.

Editing the Subquery of an SQL View

Structure of a Subquery Clause

The following rules apply:

- In the first part of the subquery clause, the related master files and their correlation names can be specified in SQL syntax.
- The file type of the related master files must be compatible with the file type:

File Type of View	Related Master File Type
B	A(SQL), AT, B
BV	BT, BV
E, IV	D, E, IV
JV	JT, JV
OV	OT, OV, IV
XV	XT, XV
YV	YT, YV

- Any correlation name that is specified must be used whenever the file is referred to. Type-dependent rules apply to the length of a correlation name and the characters permitted. See table in the section *Naming Conventions for SQL Objects*.
- The first part of the subquery is generated automatically if the fields of the file are defined in Predict before the subquery is edited.
- The second part of the subquery contains the selection criteria of the view: the WHERE clause, GROUP BY clause or HAVING clause or any combination of these. The name of each field referenced in the selection criteria must be qualified by the ID of the file from which the field is taken or - if a correlation name has been specified in the first part of the subquery - by the correlation name.
- If joined views are edited, the selection criteria and the type of join are displayed for each join.
- If union views are edited, placeholders are used instead of field selection lists, because field selection lists are maintained with the field list editor. The placeholder indicates where the selected field list will be added when generating the view. Placeholders are displayed in angle brackets.
- When generating a CREATE VIEW statement for a view, hyphens (-) are replaced by underscores (_) or points (.).
- The subquery can include comment lines (with /*, * or ** in the first two columns) and line comments (preceded by /*).

17 Adabas SQL Server

- Adabas SQL Server - Overview 178
- Naming Conventions for Adabas SQL Server 178
- Adabas Cluster Table 180
- Adabas SQL View 181

Adabas SQL Server - Overview

There are two methods of documenting Adabas tables:

- **Files of Type A (SQL)**
  If an Adabas table corresponds *exactly* to a base table in Adabas SQL Server, it can be documented as a file of type A (SQL). The Adabas file must not contain groups structures or multiple value fields. Rotated fields are not supported with this method. This method is retained for reasons of compatibility with earlier Predict versions.

  Note: This type is used in conjunction with our Data Virtualization & Movement product CONNX.

- **Files of Type AT**
  Tables can also be documented with files of type AT (Adabas cluster table). Files of this type can be understood as userviews to an Adabas file. See ***Adabas Cluster Table***.

Adabas SQL *views* are documented with files of type B. See ***Adabas SQL View***.

Naming Conventions for Adabas SQL Server

The following naming conventions apply to files documenting Adabas SQL Server tables and views (files of type AT, B).

Upper / lower case

If the Predict parameter General Defaults > Miscellaneous > Upper/lower case / Object ID is set to L, the following attributes are stored in upper and lower case as entered:

- File ID (object IDs containing lower case letters are not recommended)
- Derived field expressions
- SQL verifications
- Check expressions
- Constraint names

See also section *Defaults* in the *Predict Administration* documentation.

Length

Table/View names for Adabas SQL Server objects can have up to 32 characters.

Permitted characters

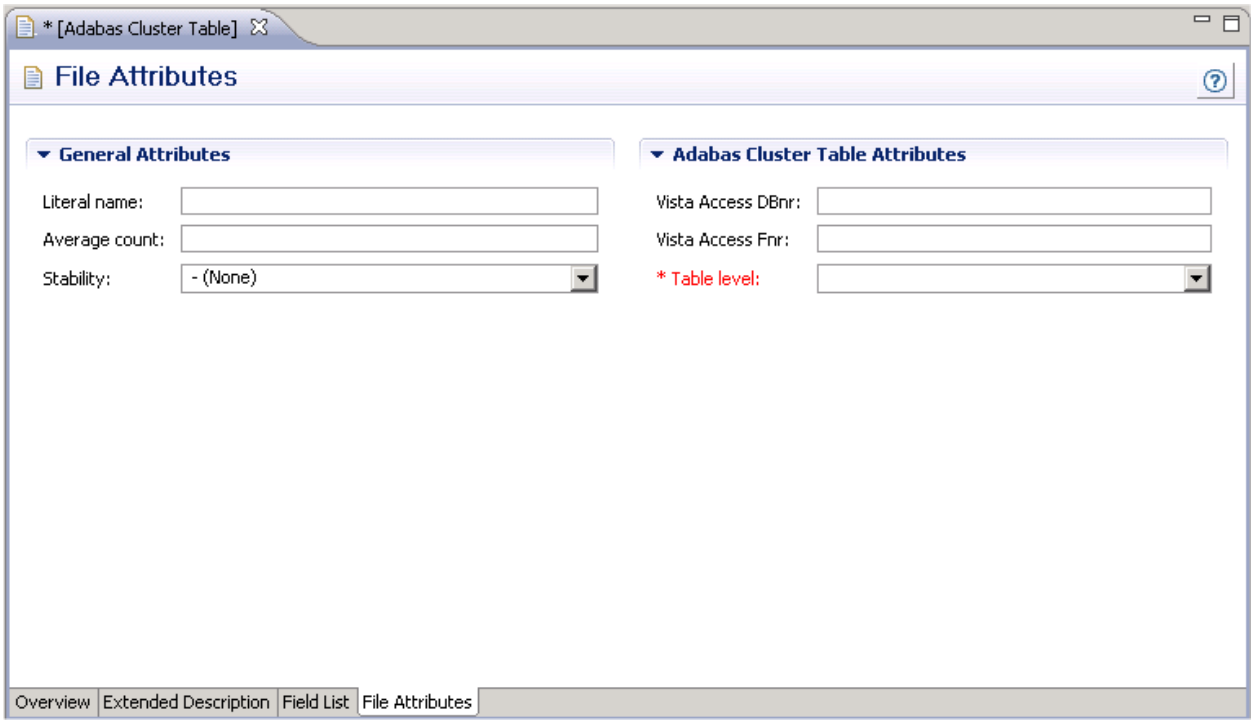
See overview of permitted characters in the section *Naming Conventions*.

Qualifier

The identifier of a table or view must be given in qualified form: the schema identifier, a delimiter and the table/view name. A hyphen is used as a delimiter (not a period as in SQL). An example: SYSSAG-SYSCOLUMNS. Hyphens in names are treated as follows:

- When a table/view is generated from a Predict file object, the hyphen is transformed into a period (.).
- Because hyphens are used as delimiters, only one hyphen can occur in the SQL identifier. Column names must not contain a hyphen.
- The hyphen can be used as a minus sign or negative sign in the field expression or the subselect clause and must then be preceded by a blank.

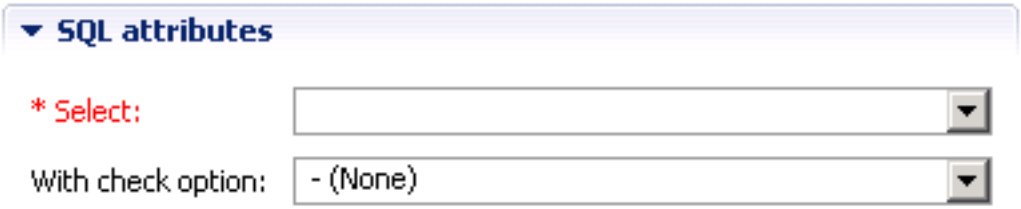
Adabas Cluster Table



Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters		
Table level	0	Only "flat" structures are permitted (no MU or PE fields).
	1	For defining multiple fields and periodic groups.
	2	For defining multiple fields within a periodic group.
	There are two methods of documenting periodic groups and multiple value fields in AT files: ■ If the occurrences of PE/MU fields are *fixed*, you can use rotated fields in the AT file. ■ If the occurrences of PE/MU fields are *variable*, use subtables (AT files at level 1 or level 2). For more information see the section *Adabas SQL Server* in the *Predict and Other Systems* documentation.	

Adabas SQL View



Note: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

18 Adabas D

- Naming Conventions for Adabas D 184
- Adabas D Table, File Type BT 185
- Adabas D View, File Type BV 186

Adabas D tables and views can be documented in Predict with file objects of type BT and BV respectively. These file objects can be used to generate DDMs or CREATE TABLE/VIEW statements.

Naming Conventions for Adabas D

The following naming conventions apply to files documenting Adabas D tables and views.

Upper / lower case

If the Predict parameter General Defaults > Miscellaneous > Upper/lower case / Object ID is set to L, the following attributes of Adabas D objects are stored in upper and lower case as entered:

- File ID (object IDs containing lower case letters are not recommended)
- Derived field expressions
- SQL verifications
- Check expressions
- Constraint names

See also section *Defaults* in the *Predict Administration* documentation.

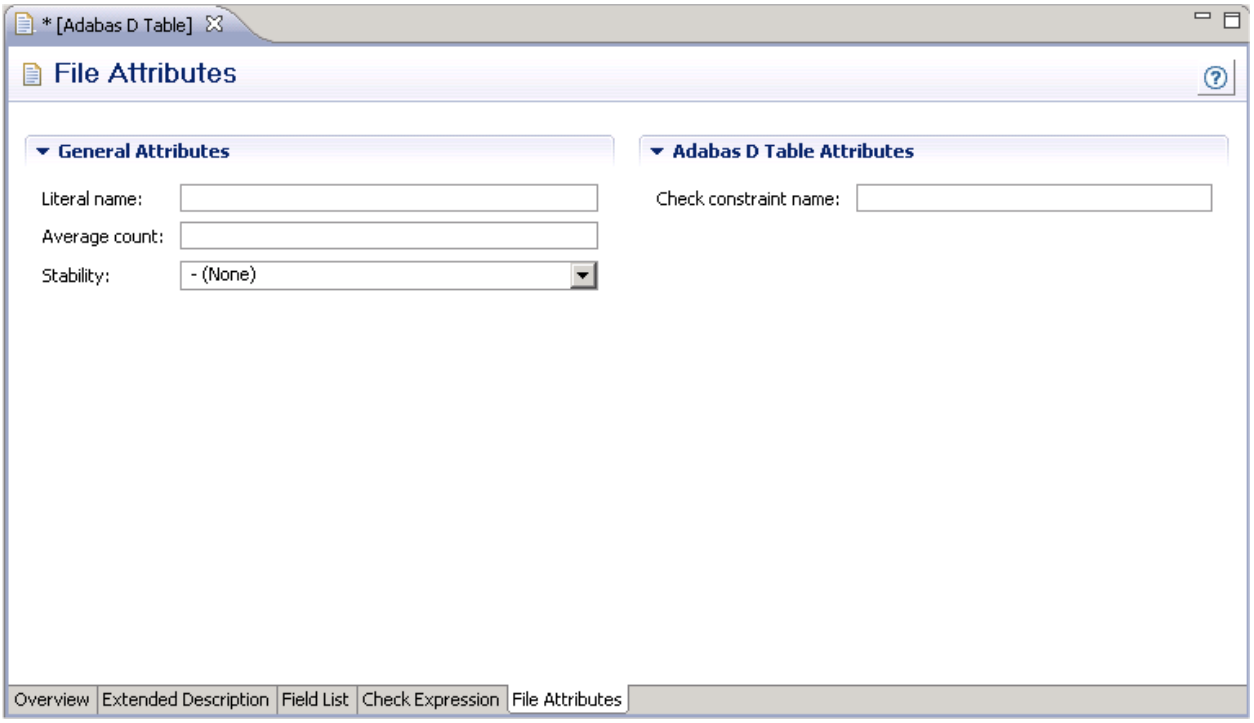
Length

- Table/View names for Adabas D objects can have up to 18 characters.
- A fully qualified ID (Creator + Hyphen + Table/View name) may not exceed 27 characters.

Permitted characters

See overview of permitted characters in the section ***Naming Conventions***.

Adabas D Table, File Type BT

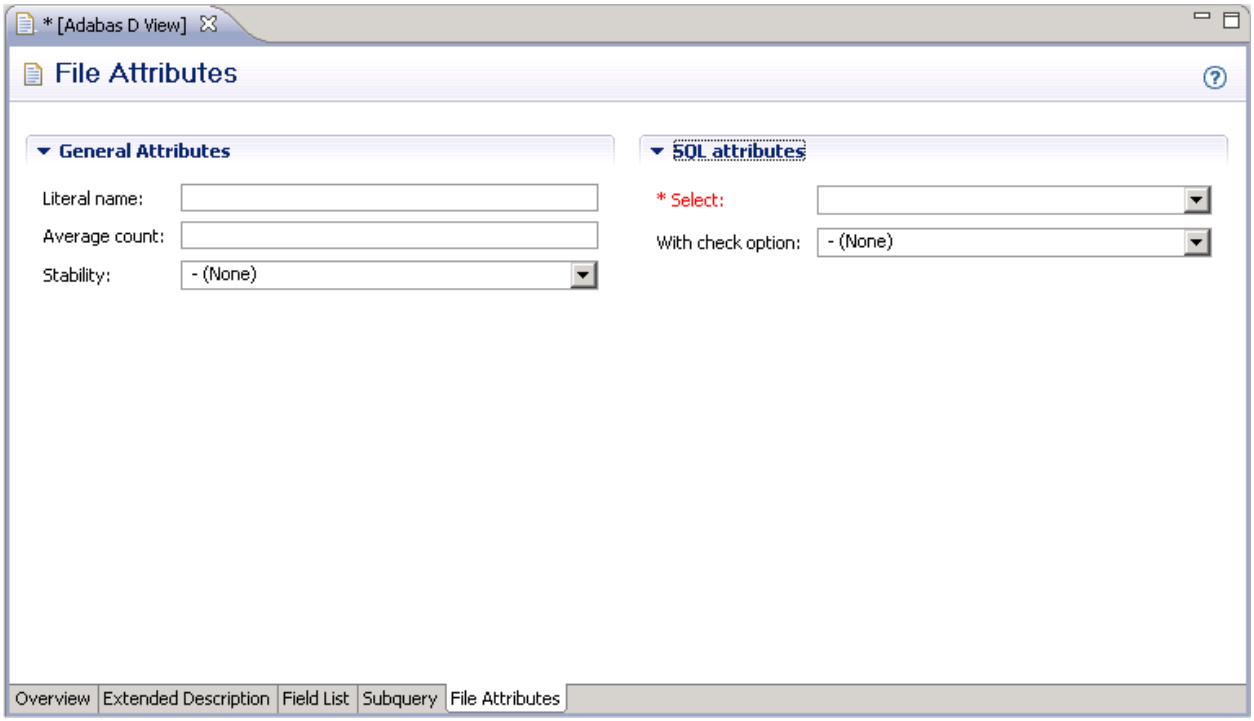


Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters	

Check constraint name	If a table check expression has been defined and the name of a check constraint is entered here, the following clause is generated in the `CREATE TABLE` statement:
	
	`CONSTRAINT constraint_name`
	`CHECK (check_expression)`

Adabas D View, File Type BV



Note: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

19 DB2

- Naming Conventions for DB2 188
- DB2 Table, File Type D 189
- DB2 View, File Type E 192
- Intermediate View, File Type IV 192
- Intermediate Table, File Type IT 193
- DB2 Query Table, File Type MT 194

DB2 tables and views can be documented in Predict with file objects of type D and E respectively. These file objects can be used to generate DDMs or `CREATE TABLE/VIEW` statements.

Naming Conventions for DB2

The following naming conventions apply to files documenting DB2 tables and views.

Upper / lower case

File IDs must be entered in upper case. If the Predict parameter General Defaults > Miscellaneous > Upper/lower case / Object ID is set to L, lower-case IDs are not converted to upper case and an error message is given.

Hyphens

- A hyphen is used to delimit the creator from the table/view name.
- Only one hyphen is permitted in the ID of a DB2 table/view object.
- When a table or view is generated from the Predict file object, the hyphen is converted to a period.

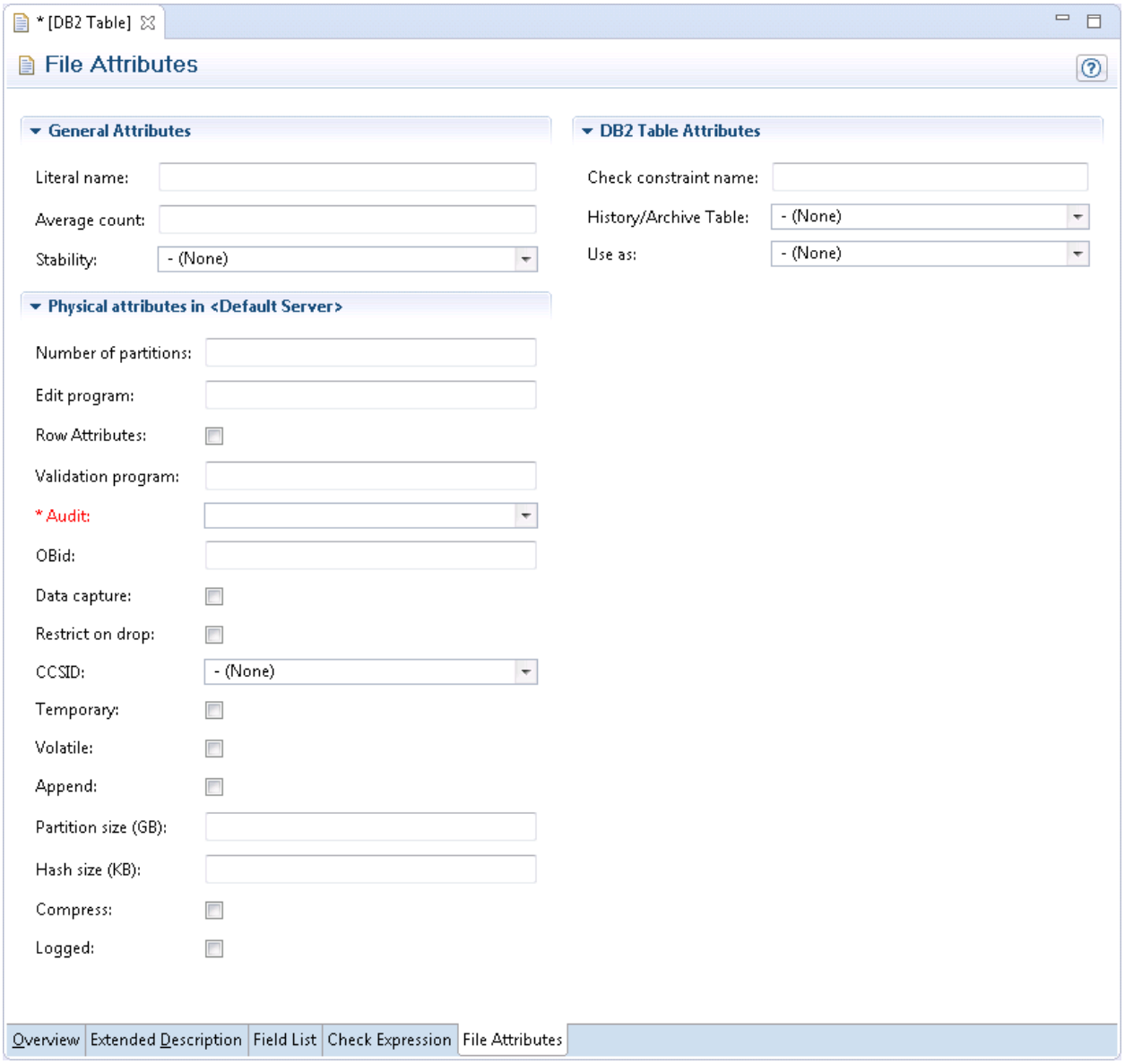
Length

- Table/View names for DB2 objects can have up to 18 characters.
- A fully qualified ID (Creator + Hyphen + Table/View name) must not exceed 27 characters.

Permitted characters

See overview of permitted characters in the section *Naming Conventions*.

DB2 Table, File Type D



Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

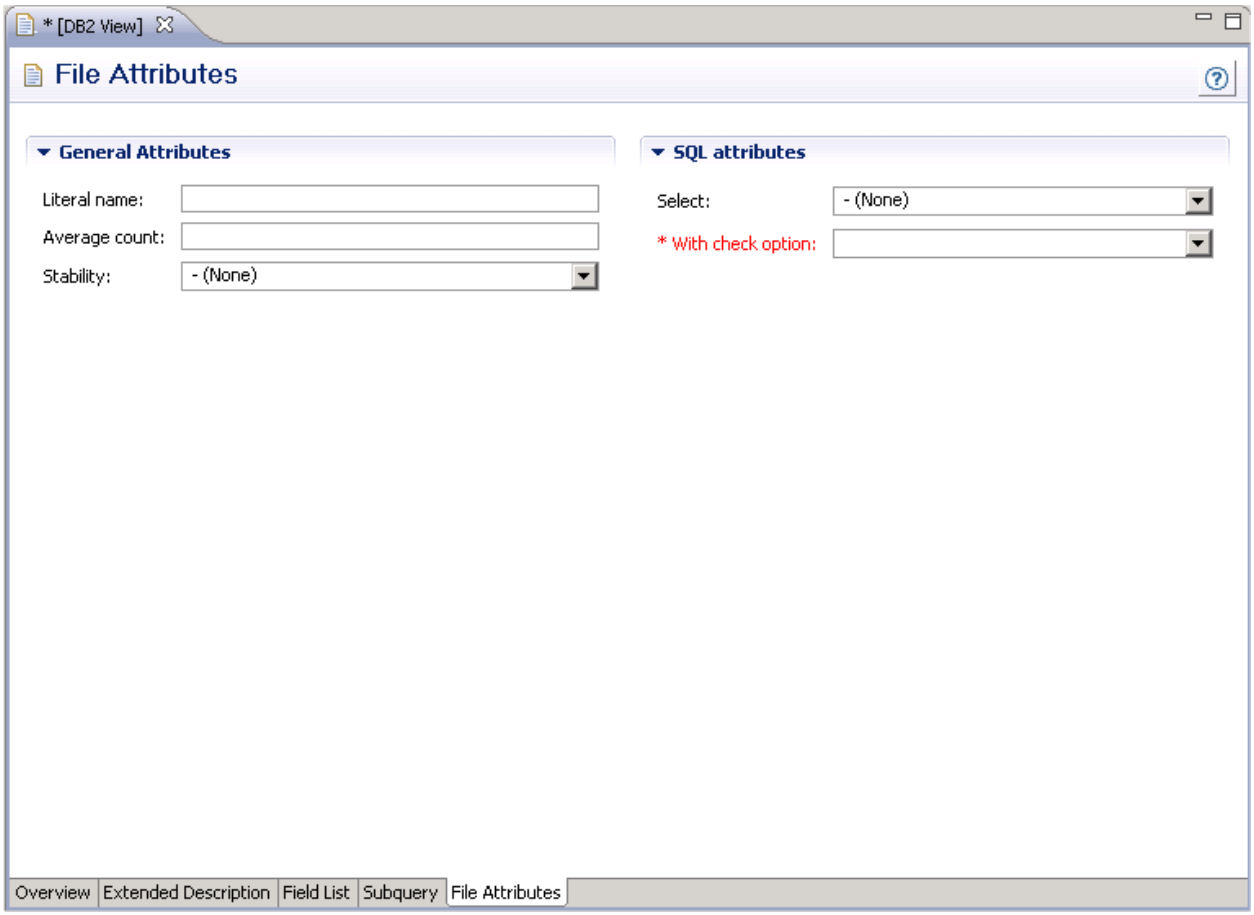
Parameters	
Number of partitions	The number of partitions of the table.
Edit program	The name of an edit routine for the table.
Row attributes	Specifies whether the edit procedure parameter list contains an address for the description of a row. Y: Yes. This is the default. N: No.
Validation program	The name of a validation routine for the table.
Audit	The type of access to this table that will cause auditing to be performed. Valid values: A: All C: Changes N: None
OBid	Identifies the OBID to be used for the table. An OBID is the identifier for an object's internal descriptor in DB2. **Note:** This parameter is required if parameter DB2 ROSHARE parm of the database object containing the table is set to R. See ***Database Type D - DB2***. See your *DB2* documentation for more information.
Data capture	Y: Data changes are passed to a user exit.
Compress	Specifies whether data compression applies to the rows of the implicitly created tablespace. blank: Not specified. This is the default setting. Y: Yes. N: No. F: Fixed length. H: Huffman.
Restrict on drop	Y: The DB2 table cannot be dropped. To drop a table with this setting, this parameter must be set explicitly to N.
Partition size	Specifies that the table is to be partitioned by growth, every *n* GBytes. Where *n* is to be replaced by the desired integer value.
Hash size	Specifies the amount of fixed hash space to preallocate for the partition that is associated. Hash size is *n* KBytes. Where *n* is to be replaced by the desired integer value.
Logged	Specifies whether changes that are made to the data in the implicitly created tablespace are recorded in the log. blank: Not specified. This is the default setting. Y: Yes. N: No.

Parameters	

Check constraint name	If a table check expression has been defined and the name of a check constraint is entered here, the following clause is generated in the `CREATE TABLE` statement:
	
	`CONSTRAINT constraint_name`
	`CHECK (check_expression)`

History/Archive Table	Only for system-period temporal tables. Name of the history or archive table linked to the DB2 base table. If this option is selected, the following values can be set: blank — Not specified. This is the default setting. A — Archive table. E — History add extra row. This is required for a history table using the `ON DELETE ADD EXTRA ROW` option. H — History table.
CCSID	Encoding scheme. Valid values: blank — not specified A — ASCII E — EBCDIC U — Unicode
Temporary	Y — Global temporary table N — not temporary.
Volatile	Specifies how DB2 is to choose access to the table. Valid values: Y — Specifies that index access should be used on this table whenever possible for SQL operations. N — Specifies that SQL access to this table should be based on the current statistics. This is the default.
Append	Specifies whether append processing is used for the table. Y — Yes. N — No.
Check expression	This is defined on the **Check Expression** tab.
Keylabel name	Specifies whether key label is specified at the table level for encryption.
Accelerator name	Identifies the accelerator in which the table will be defined.
Pagenum	Identifies the type of page numbering that is used when you create a partition-by-range table space.. Valid values: R — Relative. A — Absolute.

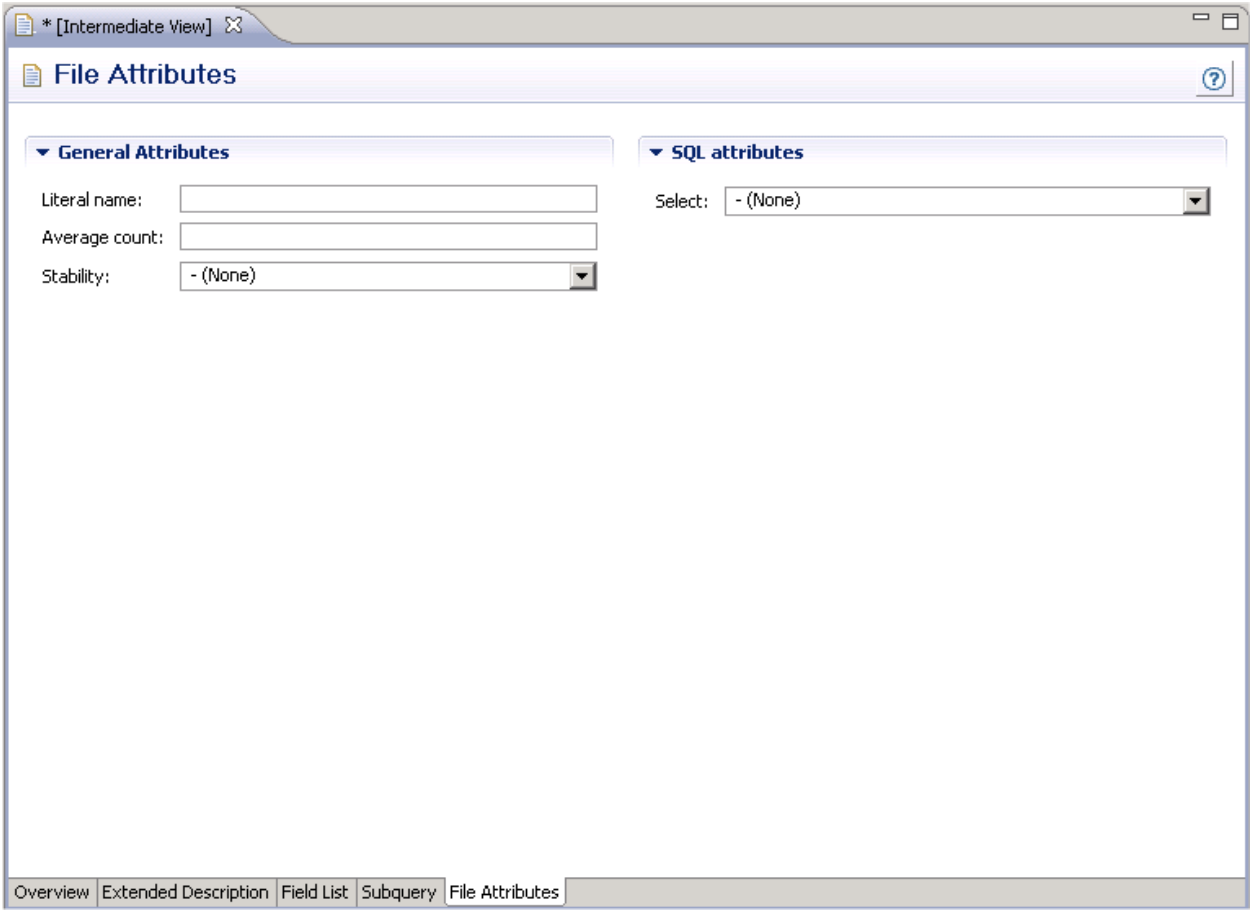
DB2 View, File Type E



Note: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Intermediate View, File Type IV

The intermediate view can be used to specify subselects, joined tables and table functions in the from clause of DB2 views. The intermediate view defines a temporary view that does not exist in the DB2 catalog.

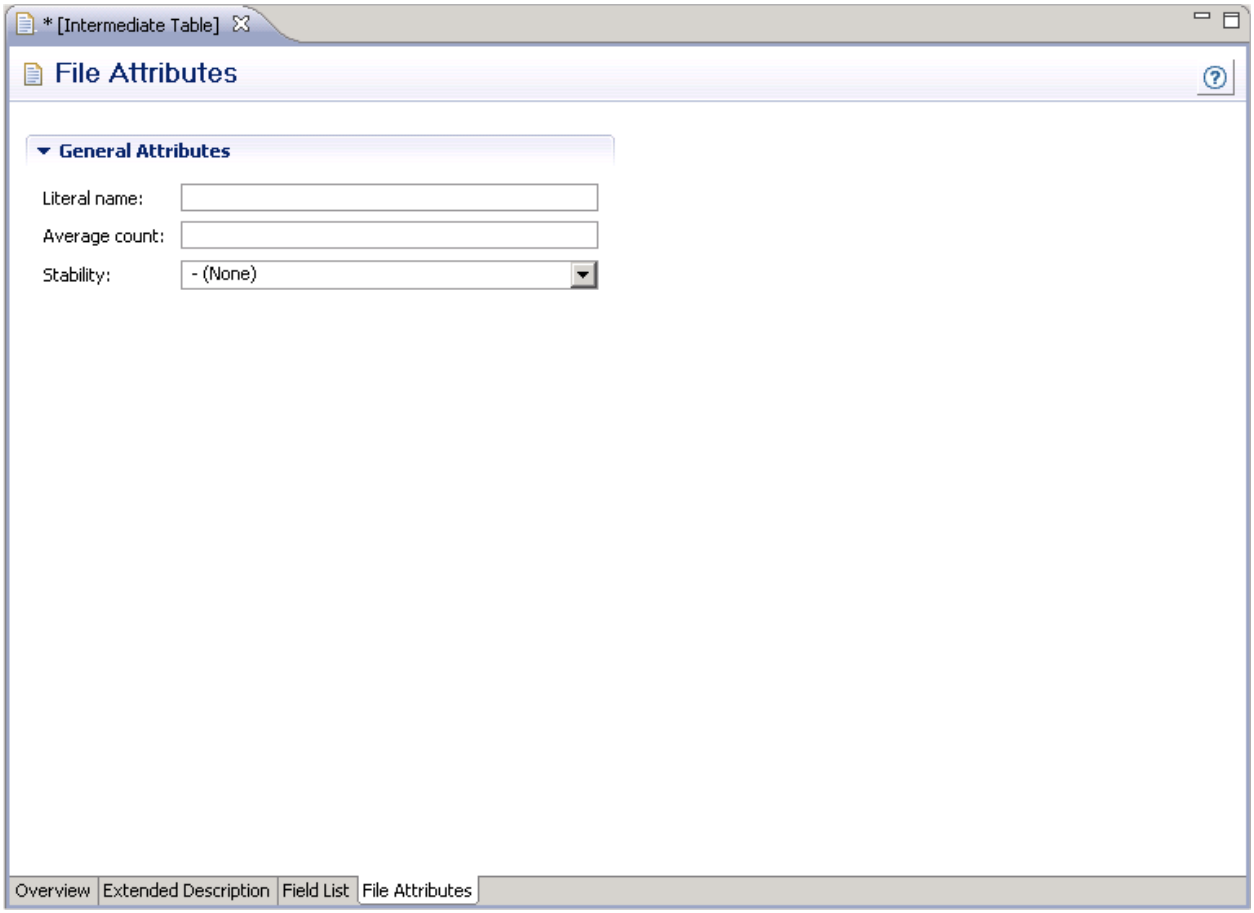


Note: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Intermediate Table, File Type IT

The field list of an intermediate table can be used to specify the parameters for:

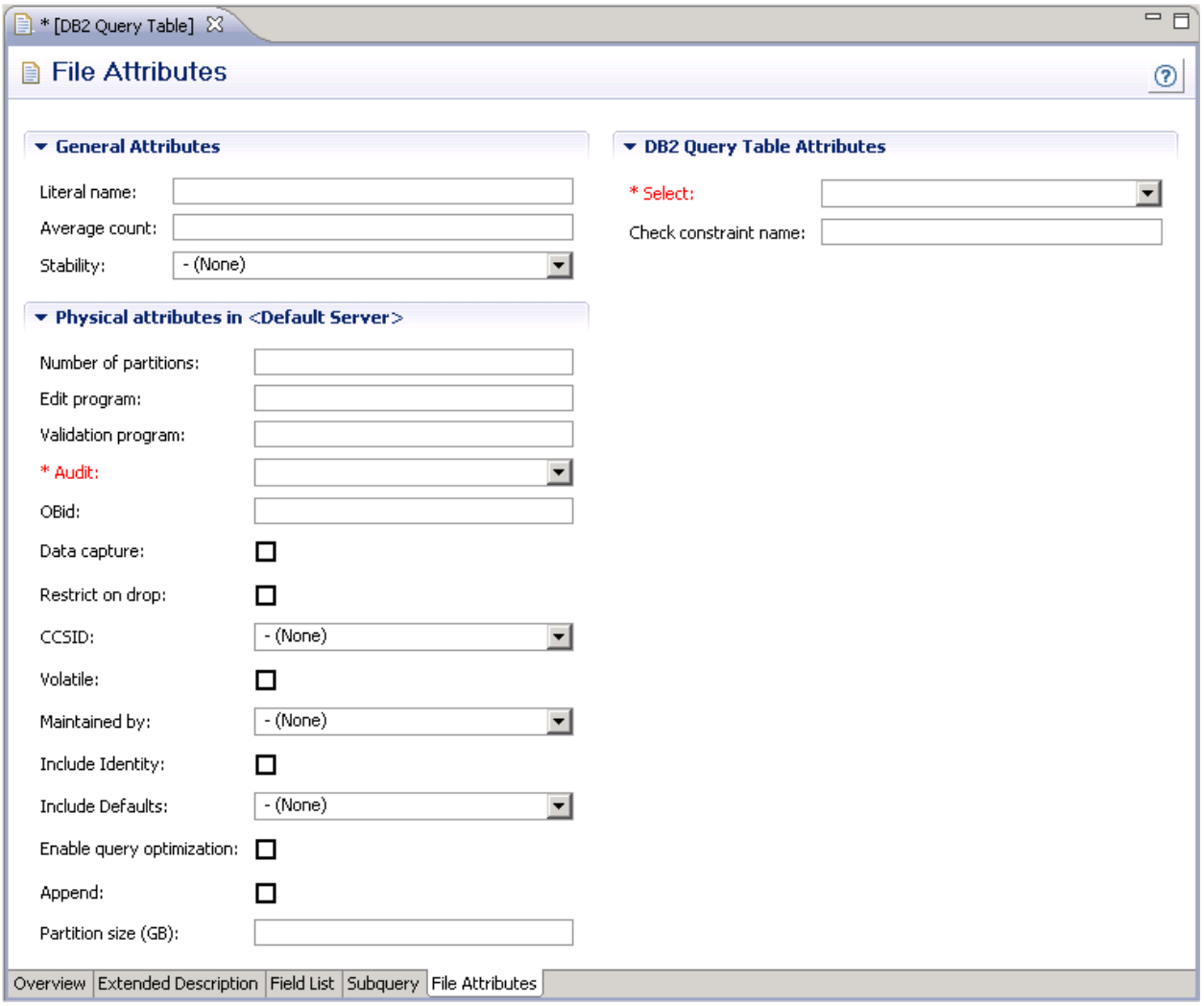
- a database function (object type PR subtype U) or
- an SQL procedure (object type PR subtype R).



Note: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes* . See also *Common Parameters for SQL File Types*.

DB2 Query Table, File Type MT

Materialized query tables in DB2 are represented in Predict as objects of type DB2 query table, file type MT.



Note: Parameters not listed below are described in **DB2 Table, File Type D** or in other sections of this documentation: Parameters common to all object types, for example Keys, are described under ***Global Attributes***. Parameters common to all file types, for example Literal name, are described under ***Common File Attributes***. See also ***Common Parameters for SQL File Types***.

Parameters		
Number of partitions	The number of partitions of the DB2 query table.	
Edit program	The name of an edit routine for the DB2 query table.	
Validation program	The name of a validation routine for the DB2 query table.	
Audit	The type of access to this table that will cause auditing to be performed. Valid values:	
	A	All
	C	Changes

Parameters	
	N: None
OBid	Identifies the OBID to be used for the table. An OBID is the identifier for an object's internal descriptor in DB2. **Note:** This parameter is required if parameter DB2 ROSHARE parm of the database object containing the table is set to R. See *Database Type D - DB2*. See your *DB2* documentation for more information.
Data capture	Y: Data changes are passed to a user exit.
Restrict on drop	Y: The DB2 query table cannot be dropped. To drop a table with this setting, this parameter must be set explicitly to N.
CCSID	Encoding scheme. Valid values: blank: not specified A: ASCII E: EBCDIC
Volatile	Specifies how DB2 is to choose access to the table. Valid values: Y: Specifies that index access should be used on this table whenever possible for SQL operations. N: Specifies that SQL access to this table should be based on the current statistics. This is the default.
Maintained by	Specifies how the data in the DB2 query table is maintained. Valid values: blank: Not specified. S: System. U: User.
Include Identity	Specifies that, if available, identity column attributes are inherited from the definition of the source table.
Include Defaults	Specifies that column defaults for each updatable column of the definition of the source table are inherited. Valid values: blank: Not specified. Y: Include. N: Exclude. U: Using type.
Enable query optimization	Specifies that the DB2 query table can be used for query optimization.
Keylabel name	Specifies whether key label is specified at the table level for encryption.
Accelerator name	Identifies the accelerator in which the table will be defined.
Pagenum	Identifies the type of page numbering that is used when you create a partition-by-range table space.. Valid values: R: Relative.

Parameters		
	A	Absolute.

20 Informix

- Naming Conventions for Informix 200
- Informix Table, File Type XT 201
- Informix View, File Type XV 203

Informix tables and views can be documented in Predict with file objects of type XT and XV respectively. These file objects can be used to generate DDMs or `CREATE TABLE/VIEW` statements.

Naming Conventions for Informix

The following naming conventions apply to files documenting Informix tables and views.

Upper / lower case

If the Predict parameter General Defaults > Miscellaneous > Upper/lower case / Object ID is set to L, the following attributes of Informix objects are stored in upper and lower case as entered:

- File ID (object IDs containing lower case letters are not recommended)
- DV field expressions
- SQL verifications
- Check expressions
- Constraint names

See also section *Defaults* in the *Predict Administration* documentation.

Length

- Table/View names for Informix objects can have up to 18 characters.
- A fully qualified ID (Creator + Hyphen + Table/View name) may not exceed 27 characters.

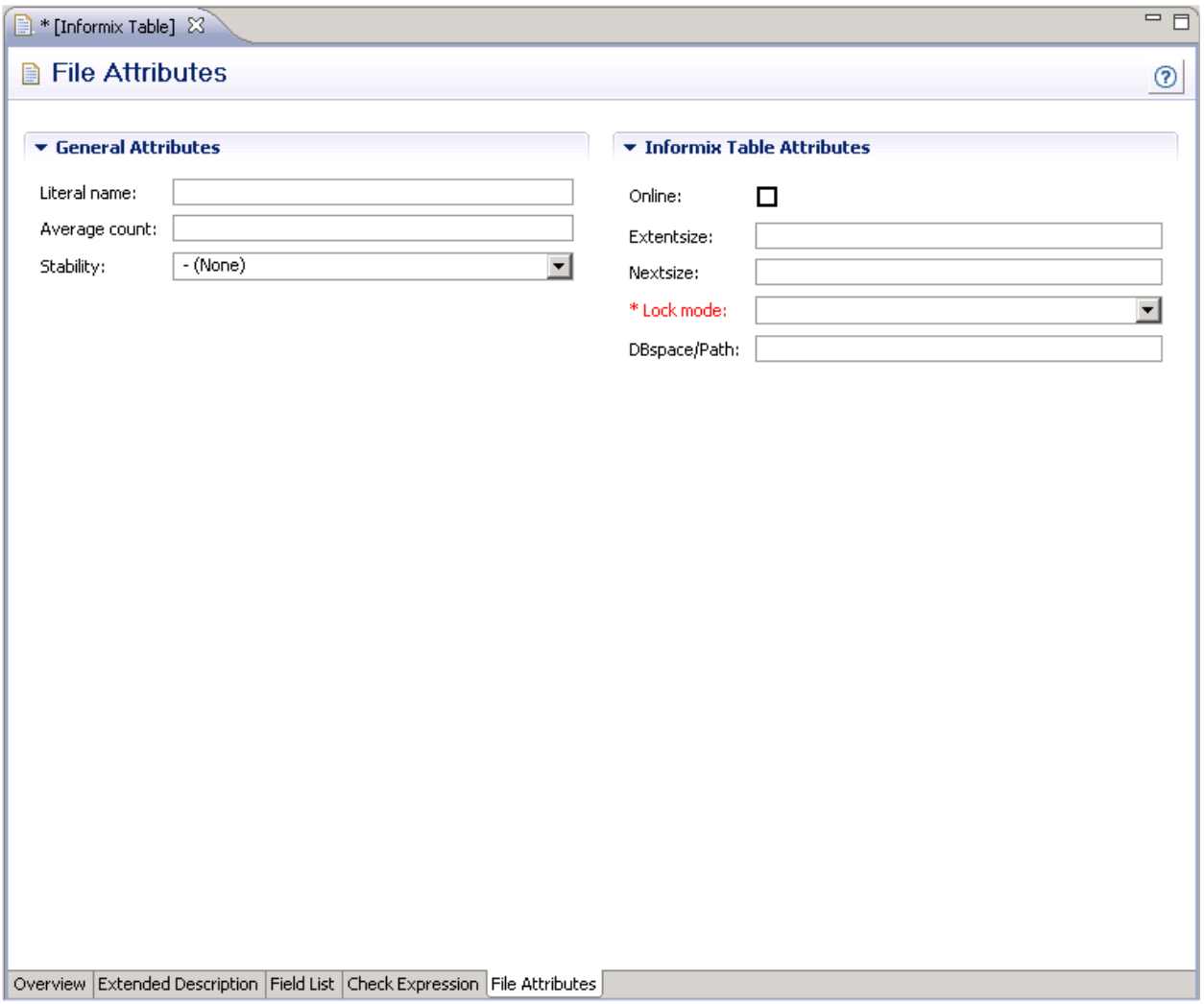
Permitted characters

- IDs containing special characters must be enclosed in double quotes, for example:

```
"USR1"-"FIL£ABC"
```

- See overview of permitted characters in ***Naming Conventions***.

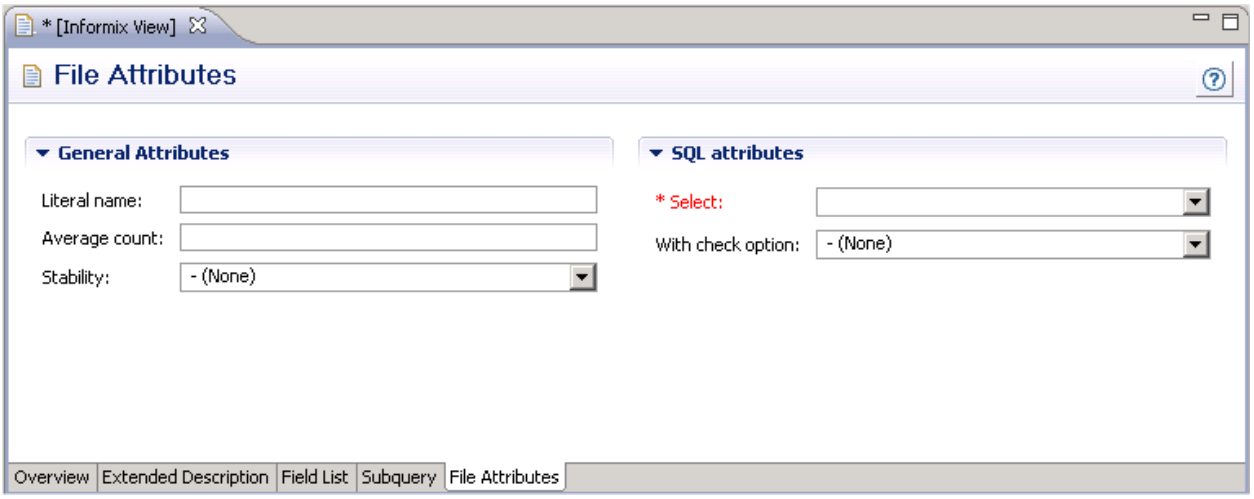
Informix Table, File Type XT



Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters	
Online	Y | An Informix ONLINE database is used.
Note: The following parameters are only applicable if Online is set to Y.	
Extentsize	Size of the initial extent for the table and its key.
Nextsize	Size of subsequent extents which are added if necessary.
Lock mode	Determines whether locking is set to page level or row level. P | Page level locking. R | Row level locking.
DBspace/Path	Name of the DBspace where Informix ONLINE is to store the table. If this parameter is not specified, the table is stored in the DBspace of the database entered under in database.

Informix View, File Type XV



Note: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

21 Ingres

- Naming Conventions for Ingres 206
- Ingres Table, File Type JT 207
- Ingres View, File Type JV 208

Ingres tables and views can be documented in Predict with file objects of type JT and JV respectively. These file objects can be used to generate DDMs or `CREATE TABLE/VIEW` statements.

Naming Conventions for Ingres

The following naming conventions apply to files documenting Ingres tables and views.

Upper / lower case

If the Predict parameter General Defaults > Miscellaneous > Upper/lower case / Object ID is set to L, the following attributes of Ingres objects are stored in upper and lower case as entered:

- File ID (object IDs containing lower case letters are not recommended)
- DV field expressions
- SQL verifications
- Check expressions
- Constraint names

See also section *Defaults* in the *Predict Administration* documentation.

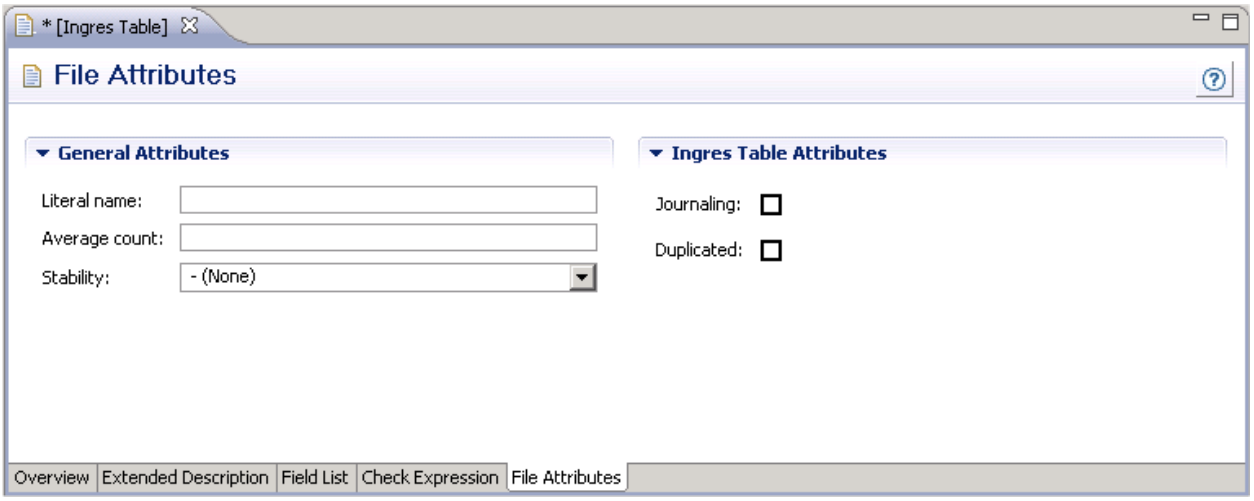
Length

- Table/View names for Ingres objects can have up to 24 characters.
- A fully qualified ID (Creator + Hyphen + Table/View name) may not exceed 32 characters.

Permitted characters

See overview of permitted characters in ***Naming Conventions***.

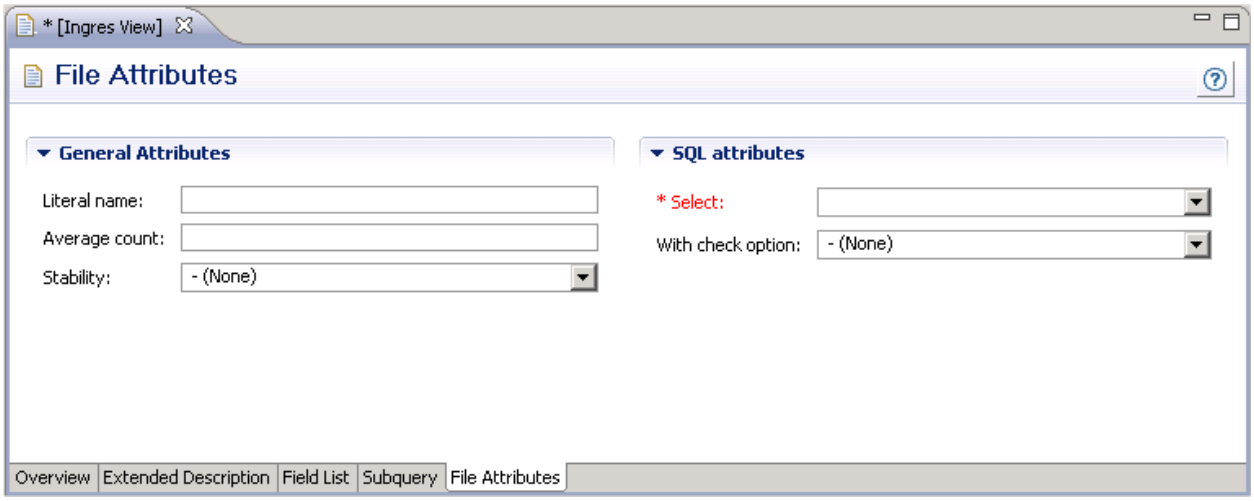
Ingres Table, File Type JT



Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters		
Journaling	Y	The clause `WITH JOURNALING` is entered in the `CREATE` statement.
	N	The clause `WITH NO JOURNALING` is entered in the `CREATE` statement.
Duplicated	Y	The clause `WITH DUPLICATES` is entered in the `CREATE` statement.
	N	The clause `WITH NO DUPLICATES` is entered in the `CREATE` statement.

Ingres View, File Type JV



Note: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

22 Oracle

- Naming Conventions for Oracle 210
- Oracle Table, File Type OT 211
- Oracle View, File Type OV 215

Oracle tables and views can be documented in Predict with file objects of type OT and OV respectively. These file objects can be used to generate DDMs or `CREATE TABLE/VIEW` statements.

Naming Conventions for Oracle

The following naming conventions apply for Oracle objects (Files of type OT and OV)

Upper / lower case

IDs must be entered in upper case. If the Predict parameter General Defaults > Miscellaneous > Upper/lower case / Object ID is set to L and you try and enter a file ID containing lower case letters, an error message is given.

See also section *Defaults* in the *Predict Administration* documentation.

Length

- Table/View names for Oracle objects can have up to 30 characters.
- A fully qualified ID (Creator + Hyphen + Table/View name) must not exceed 32 characters.

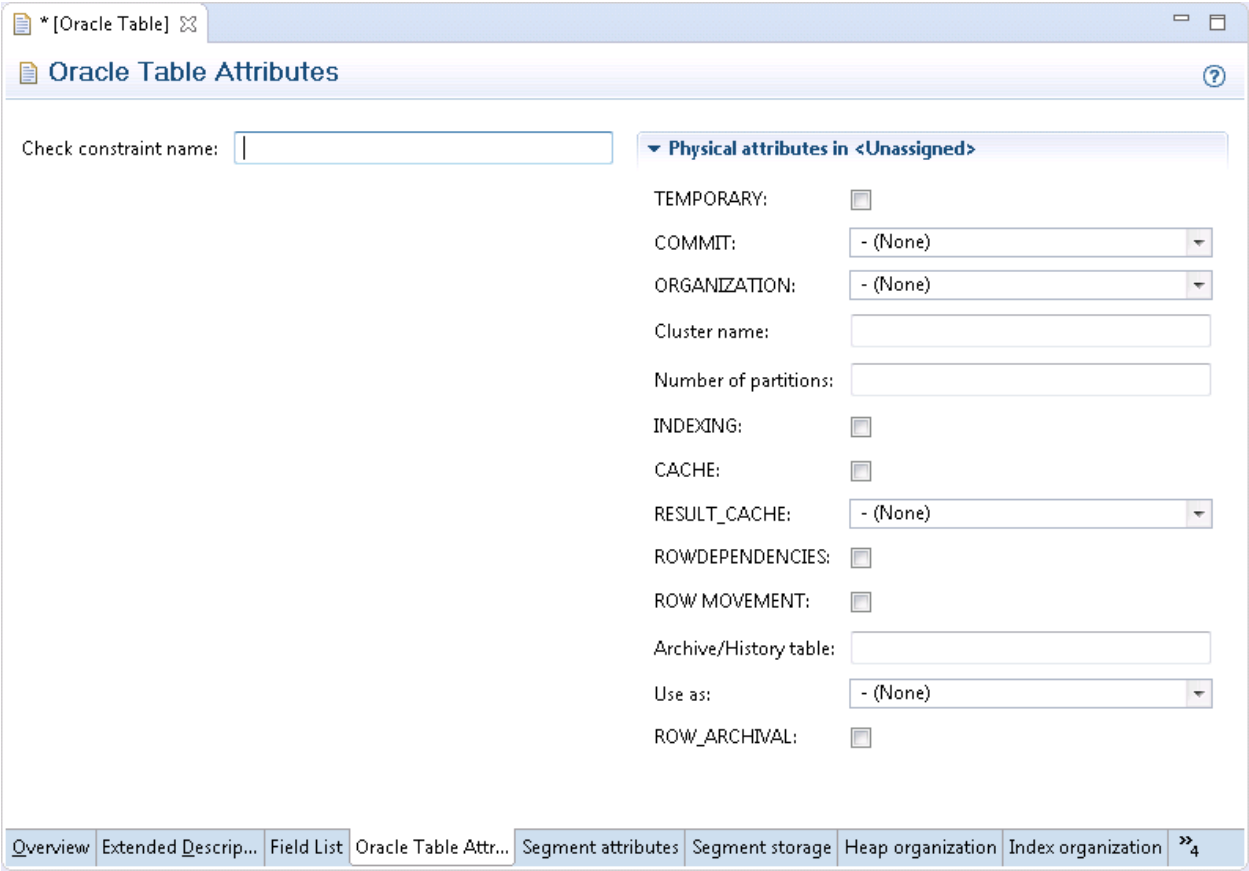
Permitted characters

- IDs containing special characters must be enclosed in double quotes, for example:

```
"USR1"-"FIL£ABC"
```

- See overview of permitted characters in ***Naming Conventions***.

Oracle Table, File Type OT



Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters		
TEMPORARY	Y	Global temporary table.
	N	Not temporary.
ORGANIZATION	Specify how the table is organized.	
	H	Heap.
	I	Indexed.
	C	Clustered.
Cluster name	If a cluster name is entered here, the clause `CLUSTER` *`name`* is generated in the `CREATE TABLE` statement. The table is to be included in the specified cluster.	
Number of partitions	The number of partitions of the table.	

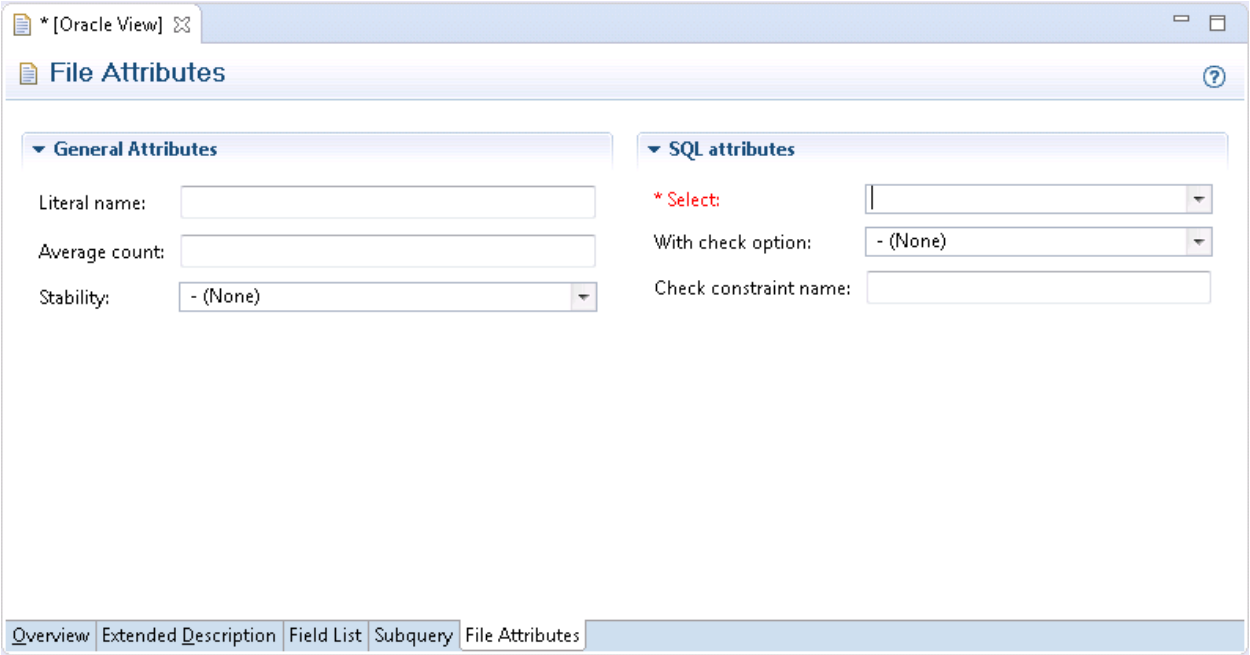
Parameters		
CACHE	Y	Yes.
	N	No.
ROWDEPENDENCIES	Y	Yes.
	N	No.
Archive/History table usage as	Name of the history or archive table linked to the Oracle table. If this option is selected, the following values can be set for "usage as":	
	A	Archive table.
	H	History table.
	blank	Not specified. This is the default.
COMMIT	D	Delete.
	P	Preserve.
	blank	Not specified. This is the default.
INDEXING	Specify whether or not this table is indexed.	
	Y	Yes. Table is indexed.
	N	No. Table is not indexed. This is the default.
RESULT_CACHE	Specify whether query results are stored in the result cache.	
	D	Default.
	F	Force.
	blank	Not specified. This is the default.
ROW MOVEMENT	Y	Yes.
	N	No.
ROW_ARCHIVAL	Y	Yes.
	N	No.
Segment attributes		
PCTFREE	If an integer from 1 - 99 is specified here, the clause `PCTFREE` *n* is generated in the `CREATE TABLE` statement. `PCTFREE` reserves a set amount of room in every block allocated to a table for future updates to that table's data.	
PCTUSED	If an integer from 1 - 99 is specified here, the clause `PCTUSED` *n* is generated in the `CREATE TABLE` statement. `PCTUSED` specifies the minimum level of space usage that Oracle will maintain for each block of the table.	
INITRANS	If a value from 1 - 255 is entered here, the clause `INITRANS` *n* is generated in the `CREATE TABLE` statement. `INITRANS` is the initial number of transaction entries that are allocated within each block.	

Parameters		
Tablespace	If a tablespace name is entered here, the clause `TABLESPACE` name is generated in the `CREATE TABLE` statement. This name represents the tablespace in which the table will be created.	
LOGGING	Specify whether or not to use the `LOGGING` clause in a `CREATE TABLE` or `ALTER TABLE` statement.	
	Y	Yes.
	N	No.
	F	File system like.
	blank	Not specified. This is the default.
Segment storage		
If specified, the values below are used in the `STORAGE` clause generated with the `CREATE TABLE` statement. All of the values below must be specified as integers.		
INITIAL	The size of the first extent allocated when the object is created - the original amount of space allocated to the object. A value for Unit has to be applied in addition:	
	K	Kilobyte.
	M	Megabyte.
	G	Gigabyte.
	T	Terabyte.
	P	Petabyte.
	E	Exabyte.
NEXT	The size of every subsequent extent to be allocated. A value for Unit has to be applied in addition. Possible values for Unit are described under INITIAL.	
MAXSIZE	The `MAXSIZE` clause lets you specify the maximum size of the storage element.	
OPTIMAL	Specifies an optimal size in bytes for a rollback segment.	
MINEXTENTS	The total number of extents to be allocated when the segment is created.	
MAXEXTENTS	The total number of extents, including the first, which can ever be allocated.	
PCTINCREASE	The percent by which each NEXT extent will grow over the last extent allocated.	
FREELISTS	The number of process free lists used to administer the free data blocks.	
FREELISTS GROUPS	Magnitude of the set of free lists.	
BUFFERPOOL	Determines the configuration of the buffer cache.	
	D	Default
	K	Keep
	R	Recycle
	blank	not specified
FLASH_CACHE	Defines the configuration of a second tier of buffer cache on flash disks.	
	D	Default
	K	Keep
	N	None

Parameters		
	blank	not specified
Heap organization		
Table compression	Y	Yes.
	B	Basic.
	A	Advanced.
	QL	Query low.
	QH	Query high.
	Q1	Query low locking.
	Q2	Query high locking.
	AL	Archive low.
	AH	Archive high.
	A1	Archive low locking.
	A2	Archive high locking.
	NL	No row level locking.
	N	No.
	blank	Not specified. This is the default.
Index organization		
MAPPING TABLE	Y	Yes.
	N	No.
PCTTHRESHOLD	Maximum size of the portion of the row that is stored in the index block, as a percentage of block size. Must be in the range of 1 to 50.	
COMPRESS	Activate index compression for index-organized tables.	
	Y	Yes.
	N	No.
	blank	Not specified. This is the default.
Compress length	Specify the compression length.	
Index overflow		
Refer to the descriptions given in **Segment attributes** above.		
Index overflow storage		
Refer to the descriptions given in **Segment storage** above.		

Refer to your Oracle documentation for more information on these Oracle-specific parameters.

Oracle View, File Type OV



Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters	
Check constraint name	Name of check option used if parameter With check option is set. See *SQL Attributes*.

23 Sybase

- Naming Conventions for Sybase 218
- Sybase Table, File Type YT 219
- Sybase View, File Type YV 220

Sybase tables and views can be documented in Predict with file objects of type YT and YV respectively. These file objects can be used to generate DDMs or `CREATE TABLE/VIEW` statements.

Naming Conventions for Sybase

The following naming conventions apply to files documenting Sybase tables and views.

Upper / lower case

If the Predict parameter General Defaults > Miscellaneous > Upper/lower case / Object ID is set to L, the following attributes of Sybase objects are stored in upper and lower case as entered:

- File ID (object IDs containing lower case letters are not recommended)
- DV field expressions
- SQL verifications
- Check expressions
- Constraint names

See also section *Defaults* in the *Predict Administration* documentation.

Length

- Table/View names for Sybase objects can have up to 30 characters.
- A fully qualified ID (Creator + Hyphen + Table/View name) must not exceed 32 characters.

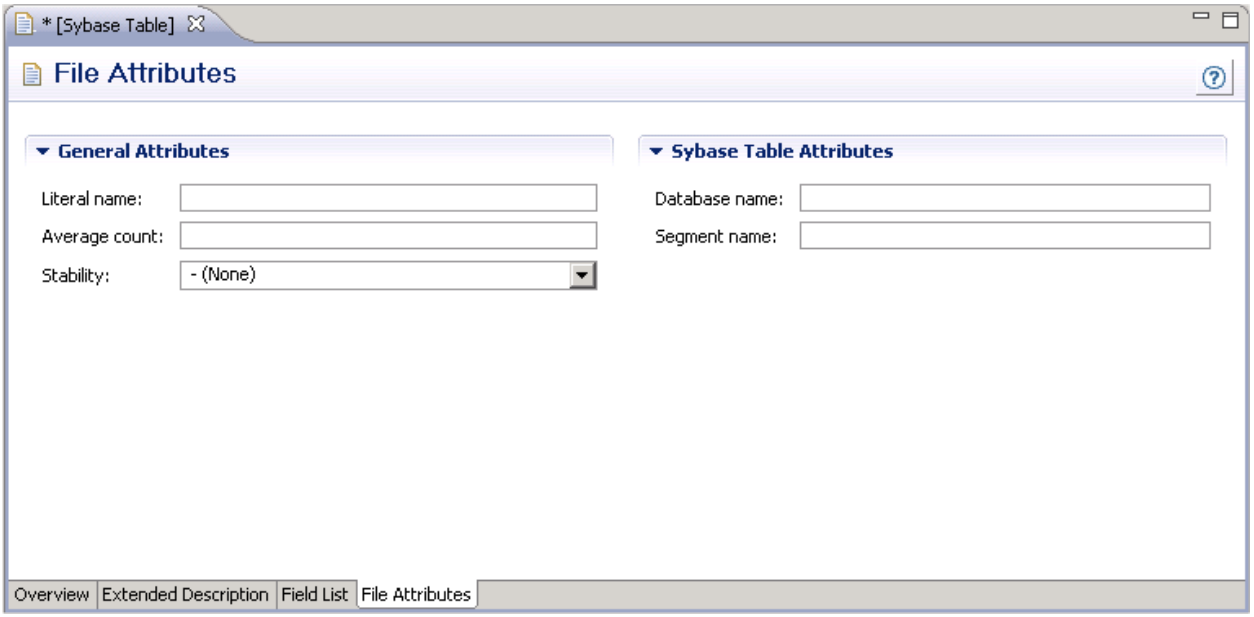
Permitted characters

- IDs containing special characters must be enclosed in double quotes, for example:

```
"USR1"-"FIL£ABC"
```

- See overview of permitted characters in ***Naming Conventions***.

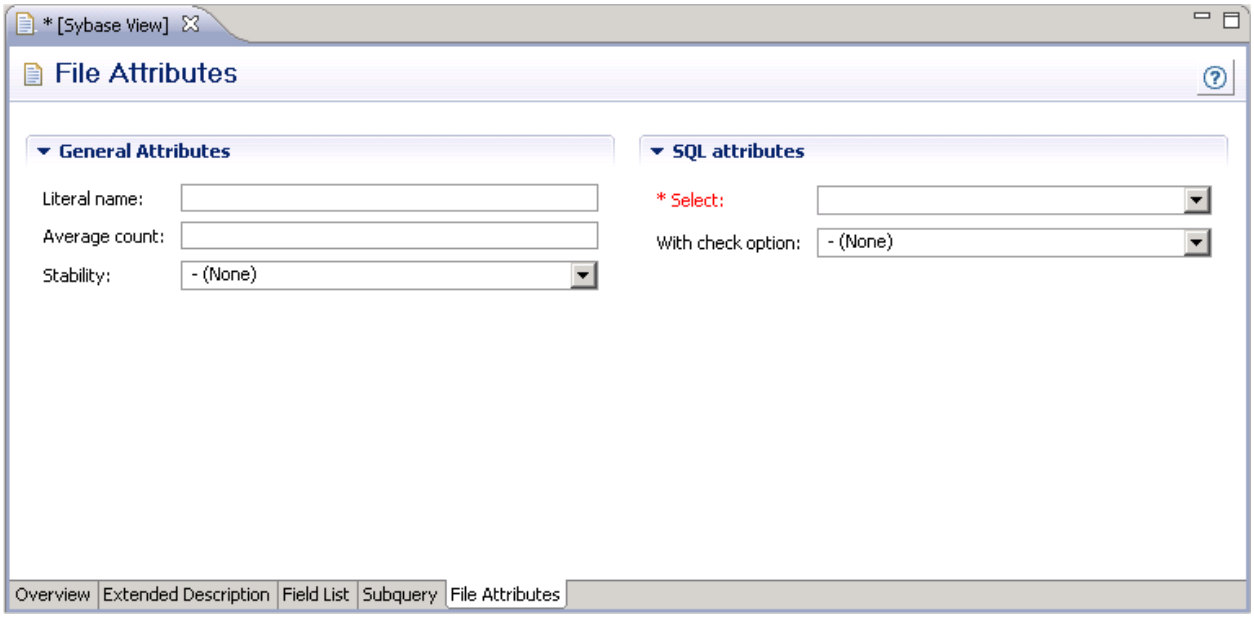
Sybase Table, File Type YT



Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters	
Database name	Name of the database in Sybase containing the table.
Segment name	Name of the segment where the table is to be placed

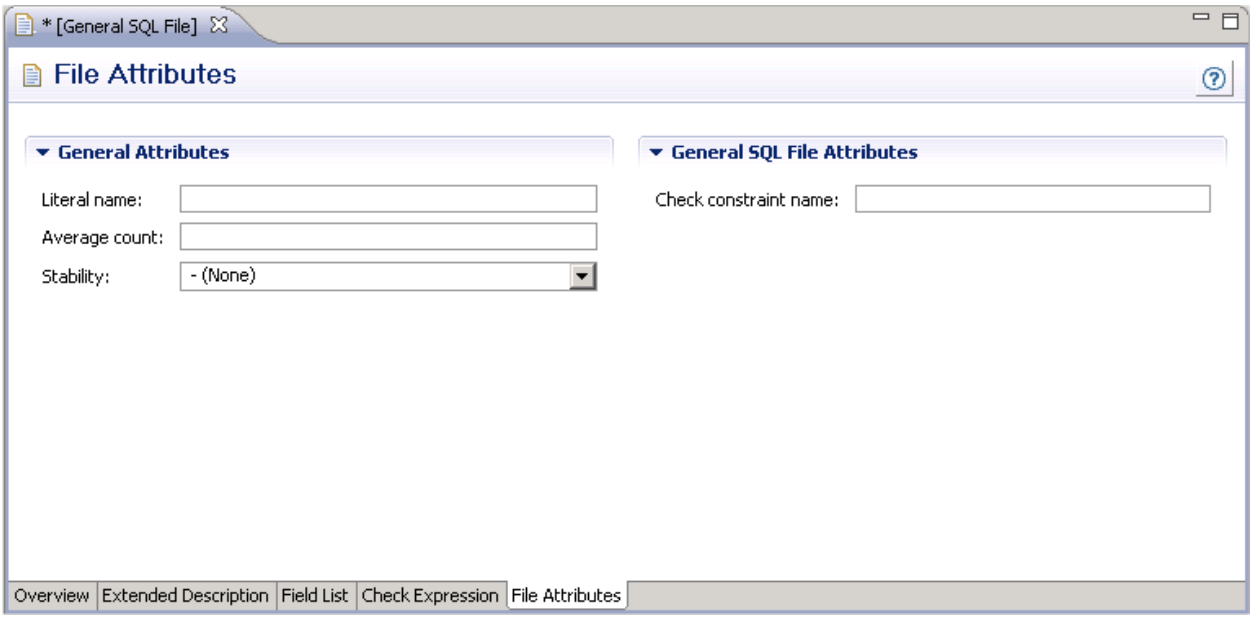
Sybase View, File Type YV



Note: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes* . See also *Common Parameters for SQL File Types*.

24 General SQL File, File Type X

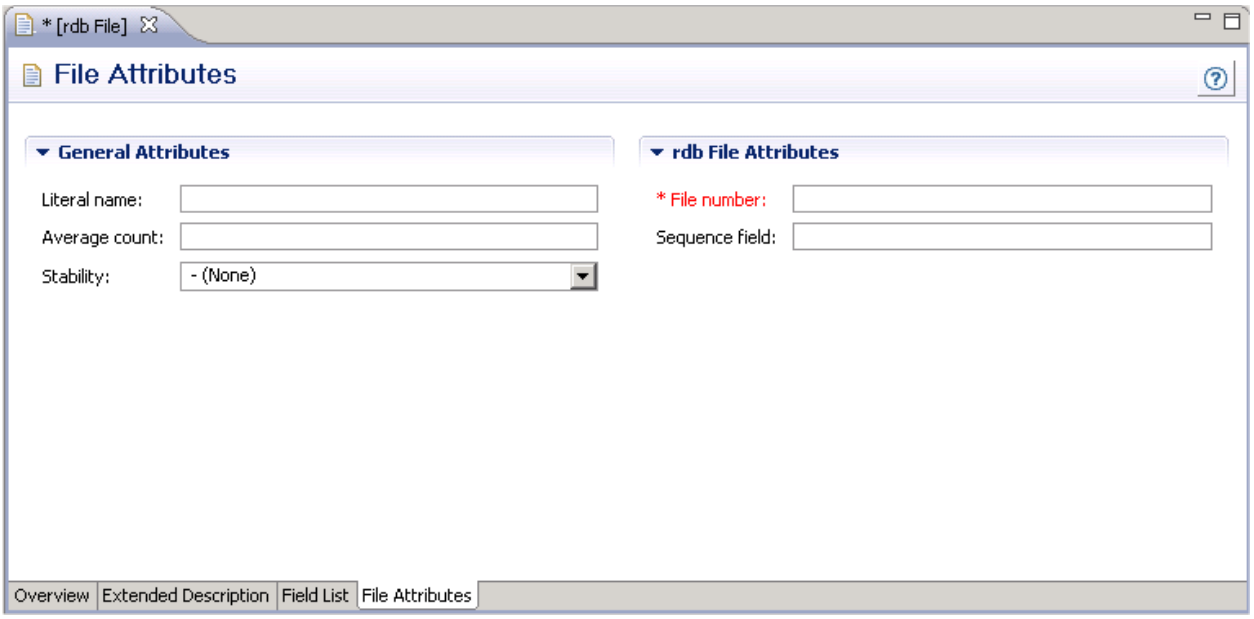
Files of type General SQL File are used to document all SQL systems not explicitly supported by Predict.



Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters	
Check constraint name	The name of a check constraint can be entered here.

25 RDB



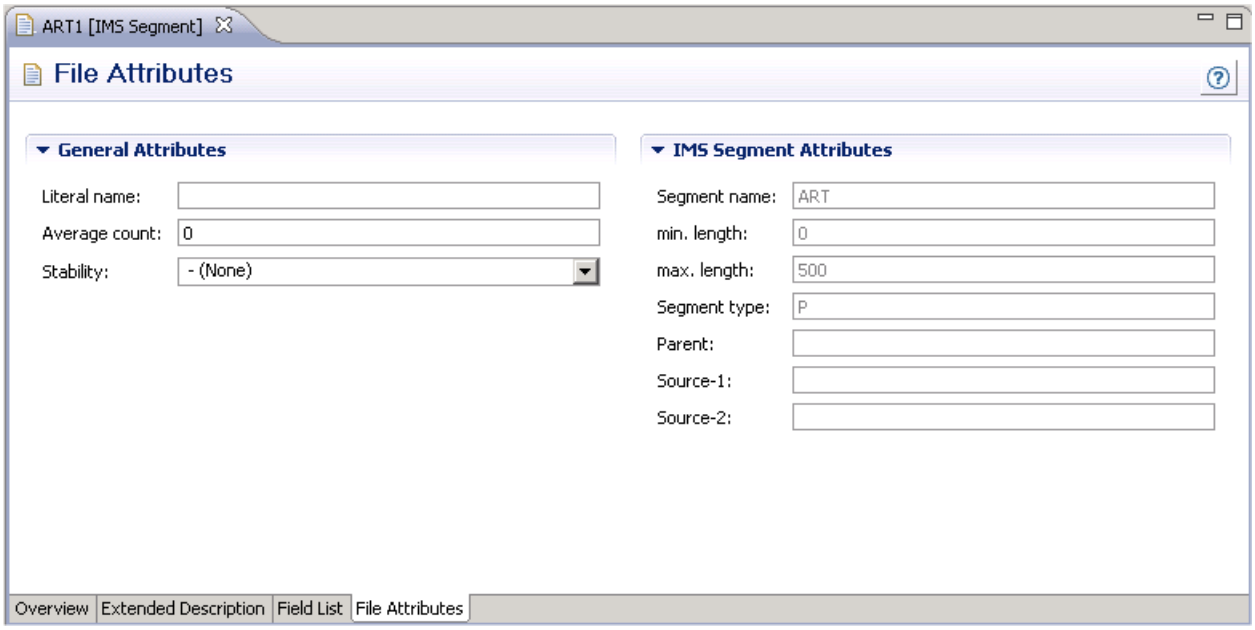
Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters	
Sequence field	The descriptor to be used by Natural for logical sequential reading. Determines the sequence in which records are delivered by the `READ LOGICAL` statement. The GENERATE DDM function will use this field as the default `READ LOGICAL` field in the Natural data definition module.

26 IMS

- IMS Segment Layouts and Userviews - File Types J and K 226
- Editing Field Lists of IMS Files 227

IMS Segment Layouts and Userviews - File Types J and K



The following attributes of an IMS segment (type I) are shown for that file and for the related files of types J and K.

Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters	
Segment name	The name of the IMS segment from which the related Predict file object of type I was incorporated.
min. length	The minimum length of the IMS segment (zero if the length is fixed).
max. length	The maximum length of the IMS segment (if it is fixed).
Segment type	The type of the IMS segment. Possible values: Logical child (C) Logical (L) Physical (P) Virtual (logical) child (V). Segments of type logical occur only in logical IMS databases. Segments of types child, physical and virtual occur only in physical IMS databases.

Parameters	
Parent	The ID of the Predict file object of type I incorporated from the parent segment of the IMS segment (the segment one level above it in the hierarchical structure of the IMS database). For a root segment, this field is left blank.
Source-1	The following rules apply: ■ For a segment of type V, the ID of the Predict file object of type I that was incorporated from the related segment of type C. ■ For a segment of type L, the ID of the Predict file object of type I that was incorporated from the segment of a physical database from which this segment of a logical database is derived. ■ For a segment of type CHILD or P, this field is left blank.
Source-2	The following rules apply: ■ For a segment of type LOGICAL derived from a segment of type C, the ID of the Predict file object of type I that was incorporated from the logical parent of the segment of type C. ■ For a segment of type LOGICAL derived from a segment of type V, the ID of the Predict file object of type I that was incorporated from the logical parent of the segment of type V (the physical parent of the related segment of type C). ■ For any other segment, this field is left blank.

Editing Field Lists of IMS Files

Restrictions that apply when editing a field list of an IMS file depend on the type of the IMS file and are described in the table below.

File Type	Restrictions
I (IMS Segment)	The following attributes can be maintained: ID, keywords, owners, abstract, format, NAT hdr1-3 (Natural headers), NAT editm (Natural edit mask), 3GL specification, Condition name & value and Field name synonyms. See ***Defining Basic Attributes of Fields*** and ***Defining Additional Attributes of fields*** in the section *Field* in this documentation. No fields can be added or deleted. Format changes are rippled across related files of type J or K. Only the following changes of format are allowed: ■ between P (packed) and PS (packed signed); ■ between P6 or P7 and D (date); ■ between P12 or P13 and T (time).

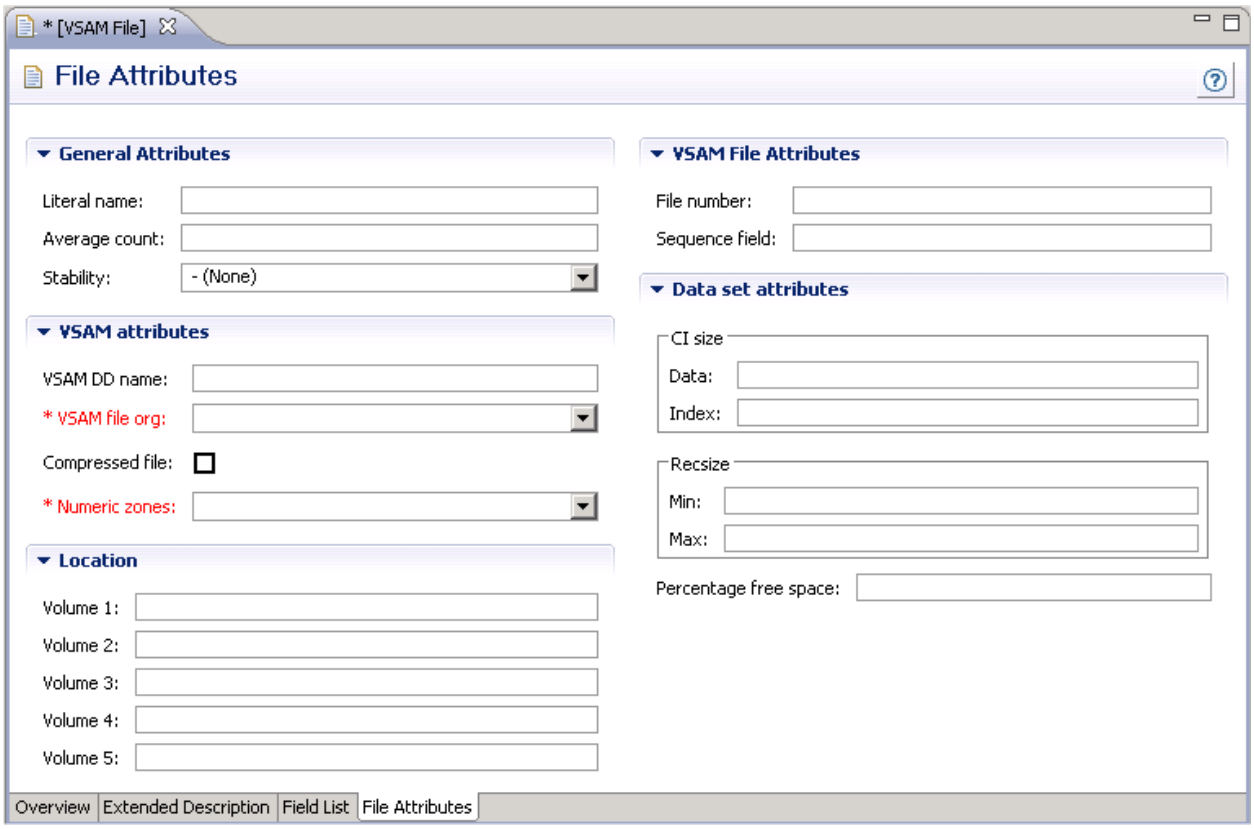
File Type	Restrictions
J (IMS Segment Layout)	The following rules apply: ■ A file of type J can contain user-defined fields and fields of the related file of type I. The two-character short names of the user-defined fields must fall within the range preceding the parameter Start in logical defined by the DDA in the Miscellaneous defaults of the Modify General Defaults function. Its value is normally HA. ■ Fields of the related file of type I that are included in a File of type J must have the same attributes in the File of type J as they have in the file of type I. ■ Their offset in the file of type J must be the same as their IMS-OFFSET in the file of type I. For a variable-length segment, only one field in one file of type J can be defined as variable length. ■ If it is a field, it must be the last field in the segment. ■ If it is a multiple value field or a periodic group, it can be anywhere in the segment. ■ However, if it is not the last field, its maximum occurrence must be specified. Predict checks that the above conditions are met when the field list of the file is cataloged. Changes to user-defined fields are rippled across related files of type J or K.
K (IMS Userview)	A file of type K can contain fields of the related file of type I and fields of all related files of type J. ID, keywords, owners, comments, format, NAT hdr1-3 (Natural headers) and NAT editm (Natural edit mask), 3GL specification, Condition name & value and Field name synonyms can be maintained.

27 VSAM

- Physical VSAM File - File Type V 230
- VSAM Logical Files, VSAM Userviews - File Types L, W and R 232

See also section *VSAM* in the *Predict and Other Systems* documentation.

Physical VSAM File - File Type V

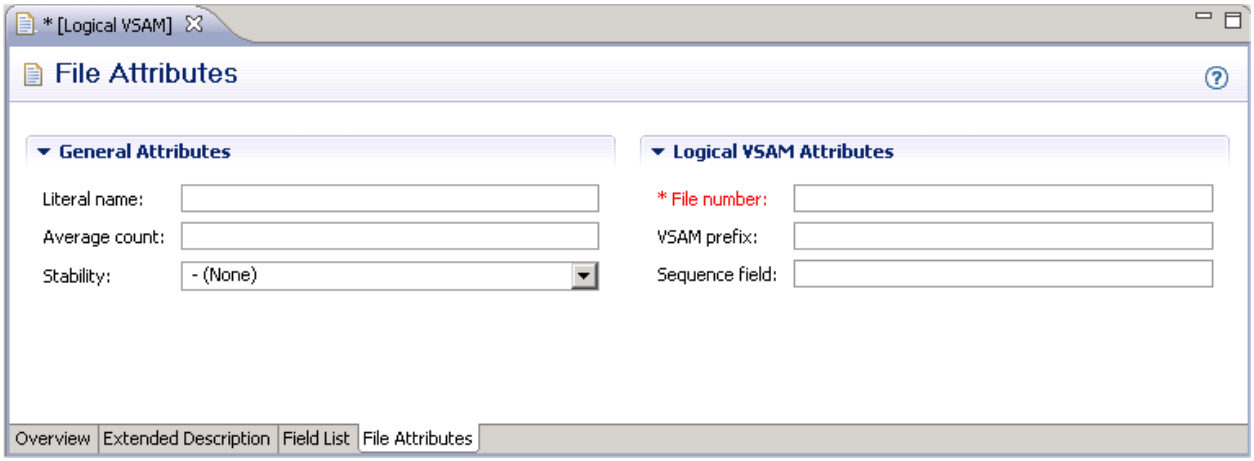


Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters	
Sequence field	The descriptor to be used by Natural for logical sequential reading. Determines the sequence in which records are delivered by the `READ LOGICAL` statement. The function Generate DDM uses this attribute as the default `READ LOGICAL` field in the Natural data definition module.
VSAM attributes	
VSAM DD name	This parameter refers to a DD card in batch mode, or to a CICS FCT object. See the *Natural Operations* documentation.

Parameters	
VSAM file org	Valid values: K — KSDS (key-sequenced data set) E — ESDS (entry-sequenced data set) R — RRDS (relative-record data set) V — VRRDS (Variable-length relative record data set)
	The above mentioned file organizations do not use extended VSAM files. This means a four byte *ISN is used to hold the relative byte address (RBA) for ESDS files and the relative record number (RRN) for RRDS and VRRDS files.
	F — ESDS ext. S — RRDS ext. L — KSDS ext. W — VRRDS ext.
	The above mentioned file use extended VSAM files. This means for ESDS, a larger *ISN (up to eight bytes) may be used for VSAMs relative byte address (RBA). The relative record number (RRN) for extended RRDS and VRRDS files remain four bytes long, but will be displayed using this extended ISN field.
Compressed file	Only applicable to files with organization K (KSDS). Y — The record will be truncated if the trailing byte positions are unused.
Numeric zones	Valid entries are C and F. This field affects the representation of positive numbers in packed decimal format. The sign position holds hexadecimal C or F respectively.
Location	
Volume 1 - 5	The volume(s) on which the file is located. Up to five volumes can be specified.
Data set attributes	
CI size - Data	The data control interval size.
CI size - Index	The control interval size for the primary index.
Recsize - Min	The minimum record size.
Recsize - Max	The maximum record size.
Free space	The free space to be allocated (in percent).

VSAM Logical Files, VSAM Userviews - File Types L, W and R

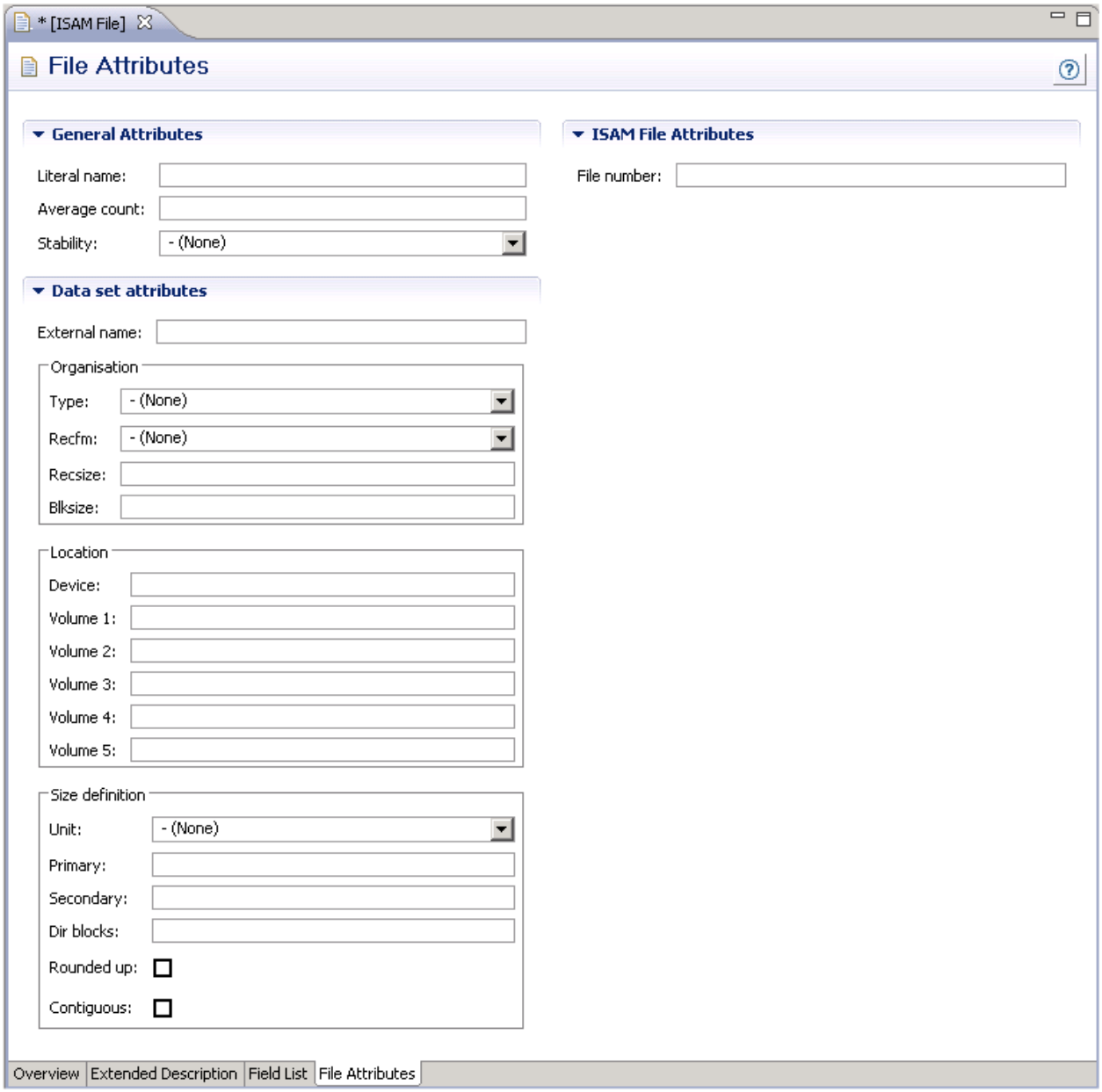


Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters	
VSAM prefix	Only applicable to files of types L and R. If this field is left blank, the last 3 digits of the file number are taken as the prefix. Otherwise, a string of up to 20 characters can be specified. The records in the corresponding physical VSAM file (type V) whose primary keys begin with the specified prefix string will be considered as belonging to the logical VSAM file. The length of the primary key specified for the logical VSAM file must be equal to the length of the primary key specified for the physical VSAM file minus the length of the prefix. A dummy field (corresponding to the prefix) preceding the primary key in the logical VSAM file must be defined for the field offsets to be calculated correctly.
Sequence field	The descriptor to be used by Natural for logical sequential reading. Determines the sequence in which records are delivered by the `READ LOGICAL` statement. The GENERATE DDM function will use this field as the default `READ LOGICAL` field in the Natural data definition module.

28 ISAM

ISAM Files and Sequential Files - File Types M and S

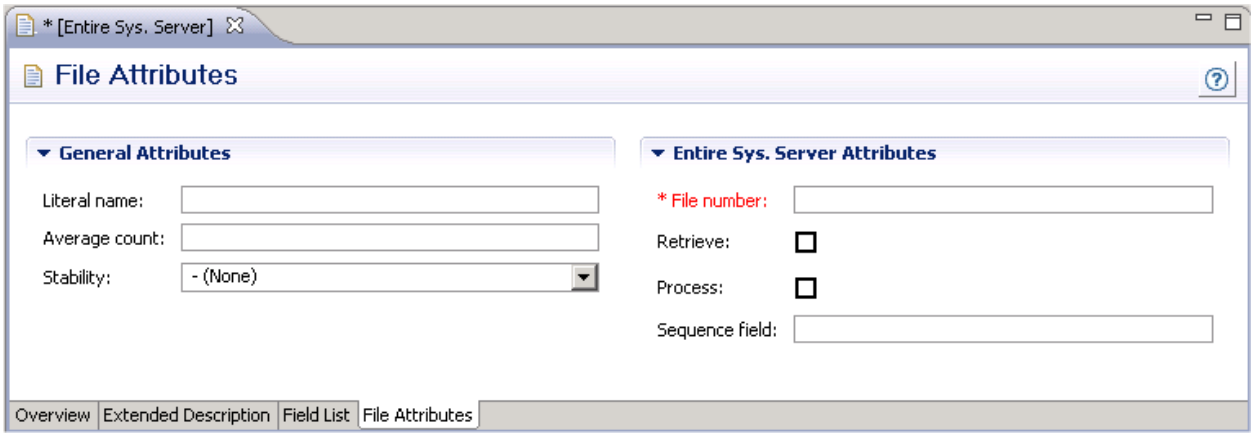


Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters	
Data Set Attributes	
External name	Name of the physical file in operating system. Up to 250 characters can be specified.
Organization	
Type	The organization of the data set: DA – Direct access PO – Partitioned PS – Sequential blank – None of the above applies
Recfm	The record format of the file: F – Fixed FB – Fixed block FS – Fixed block standard V – Variable VB – Variable blocked VS – Variable blocked standard U – Undefined *blank* – None of the above applies
Recsize	The record size of the file.
Blksize	The block size of the file.
Size Definition	
Unit	The units in which storage space has been allocated to the file: BL – Blocks CY – Cylinders TR – Tracks
Primary	The number of units of storage space allocated to the primary extent of the file.
Secondary	The number of units of storage space allocated to the secondary extent of the file.
Dir blocks	The number of blocks reserved for the directory of the file.
Rounded up	Y – Each space allocation is rounded up to full cylinders.
Contiguous	Y – The space allocated to the secondary extent of the file is contiguous with the space allocated to the primary extent.
Location	
Device	The type of storage device on which the file is located.
Volume 1 - 5	The volume(s) on which the file is located. Up to five volumes can be specified.

29 Entire System Server

Entire System Server Files and Userviews - File Types P and Q



Note: Parameters not listed below are described in other sections of this documentation: Parameters common to all object types, for example Keys, are described under *Global Attributes*. Parameters common to all file types, for example Literal name, are described under *Common File Attributes*. See also *Common Parameters for SQL File Types*.

Parameters		
Sequence field	The descriptor to be used by Natural for logical sequential reading. Determines the sequence in which records are delivered by the `READ LOGICAL` statement. The GENERATE DDM function will use this field as the default `READ LOGICAL` field in the Natural data definition module.	
Retrieve	If checked	Operation system information can be read with this file.
Process	If checked	Operation system activities can be performed via this file.

Note: You cannot add files of type P. Files of this type are added automatically when Entire System Server is installed.

30 File-Specific Maintenance

- Purge File 240

Standard maintenance functions applying to files as well as to most other types of Predict Objects are described in the section *Maintenance* in the *Predict Reference* documentation.

File-specific maintenance functions and aspects of standard maintenance functions specific to files are described below.

Purge File

The following files *cannot* be purged with the **Delete** command.

- all SAG-owned file objects
- Files of type I (IMS segment). Files of type I can be purged by scratching the IMS database (type I) containing the file.

The following objects are purged if you confirm this function:

- the file and all its userviews
- all fields of the file and its userviews
- generated code of the file and userview
- all links to databases
- all links from the file to children/from parents
- all links from/to objects that are also purged with this function.

In addition, all file relations using this file are set to D (documented).

When an Adabas file is purged, all Adabas attributes and Vista elements of the file are also deleted.

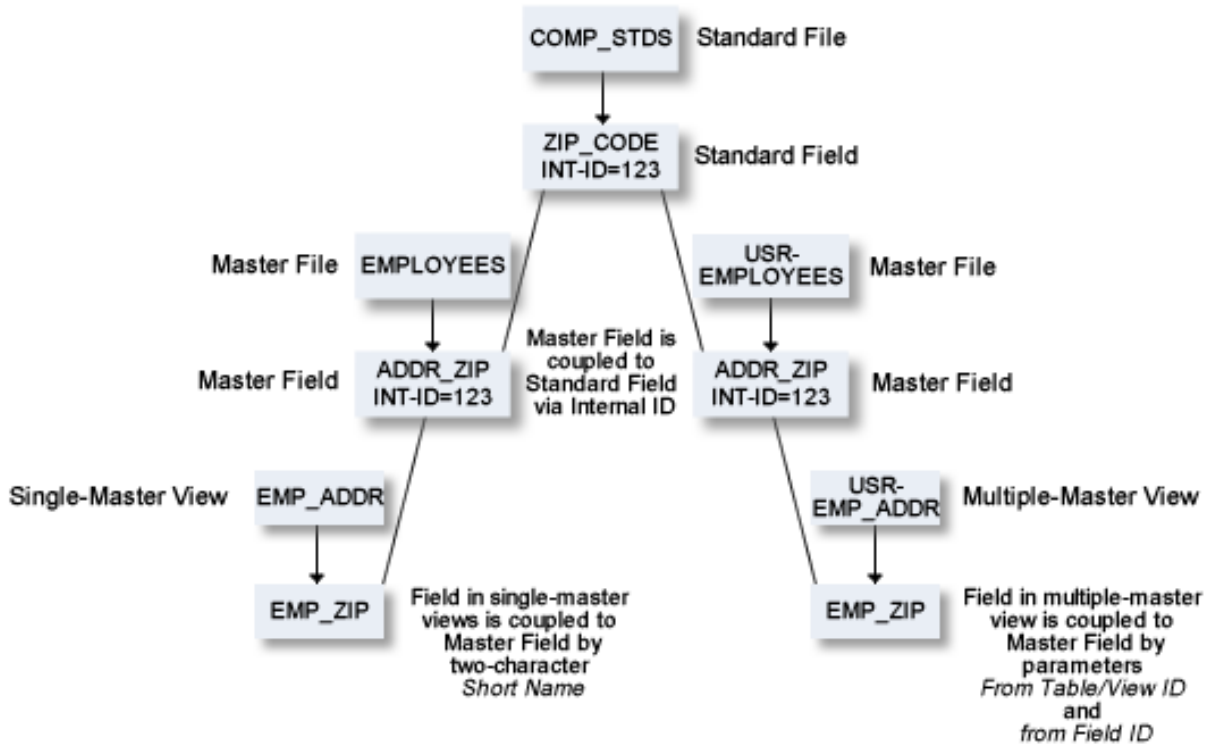
Note: A file cannot be deleted if a DDM for the file exists or the file is implemented.

31 Rippling - Ensuring Consistent Data Definitions

- Rippling - Overview 242
- Rippling from Standard Files 243
- Rippling from Master Files to Views/Userviews 247

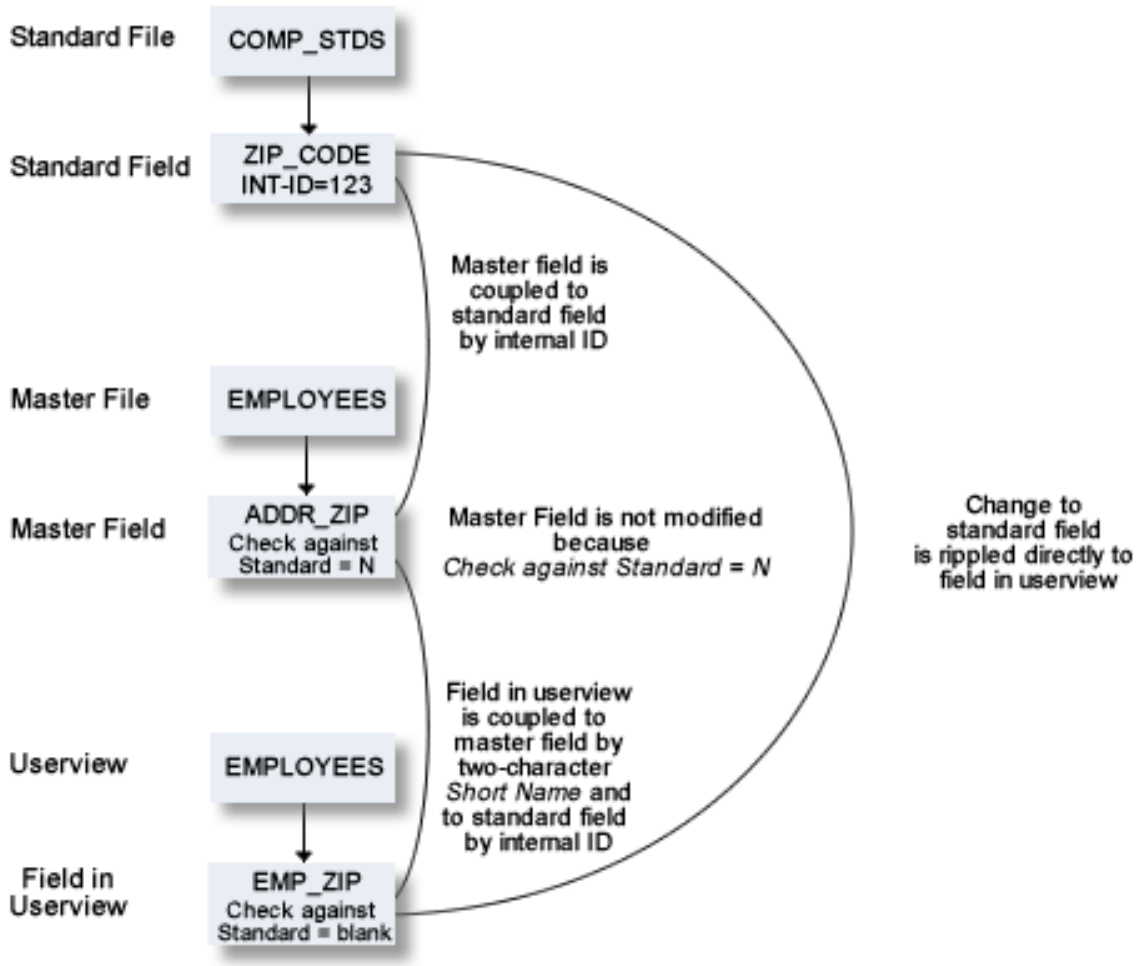
Rippling - Overview

Predict rippling options can be used to define a standard, hierarchical data structure and to ensure consistent use of this structure throughout an organization: Whenever field definitions on higher levels are changed, all data definitions on lower levels (including views/userviews) are automatically updated.



Check against standard

This option determines whether attribute changes in standard fields are rippled to connected fields. See also **Check against standard** in the section *Field*.

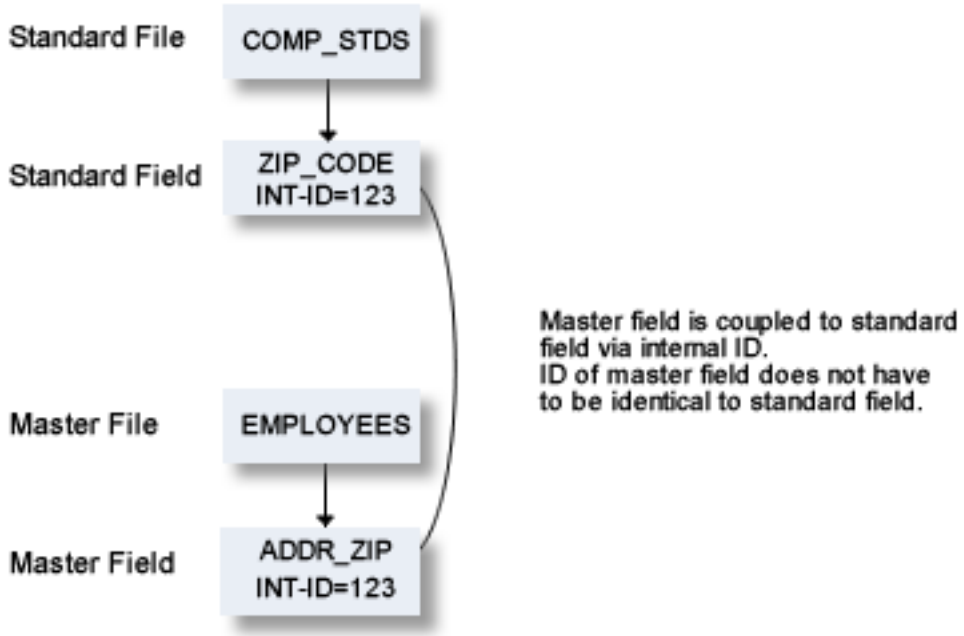


Rippling from Standard Files

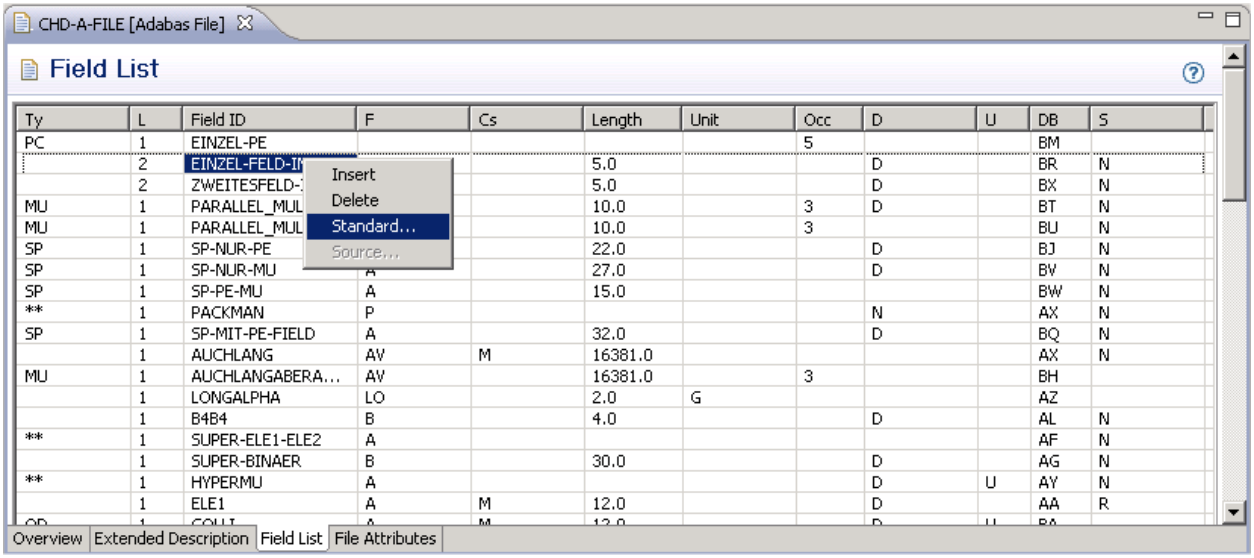
Coupling of Standard Fields

Standard fields and connected fields are coupled internally by means of Internal ID.

The coupling remains intact even if the connected field is subsequently renamed.

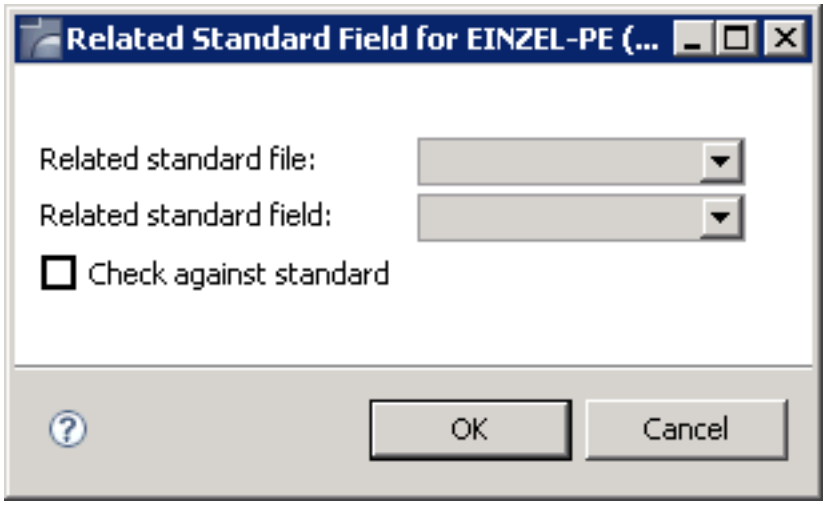


To couple fields select the **Field List** tab and select a field.



Ty	L	Field ID	F	Cs	Length	Unit	Occ	D	U	DB	S
PC	1	EINZEL-PE					5			BM	
	2	EINZEL-FELD-IN			5.0			D		BR	N
	2	ZWEITESFELD-			5.0			D		BX	N
MU	1	PARALLEL_MUL			10.0		3	D		BT	N
MU	1	PARALLEL_MUL			10.0		3			BU	N
SP	1	SP-NUR-PE			22.0			D		BJ	N
SP	1	SP-NUR-MU	A		27.0			D		BV	N
SP	1	SP-PE-MU	A		15.0					BW	N
**	1	PACKMAN	P					N		AX	N
SP	1	SP-MIT-PE-FIELD	A		32.0			D		BQ	N
	1	AUCHLANG	AV	M	16381.0					AX	N
MU	1	AUCHLANGABERA...	AV		16381.0		3			BH	
	1	LONGALPHA	LO		2.0	G				AZ	
	1	B4B4	B		4.0			D		AL	N
**	1	SUPER-ELE1-ELE2	A							AF	N
	1	SUPER-BINAER	B		30.0			D		AG	N
**	1	HYPERMU	A					D	U	AY	N
	1	ELE1	A	M	12.0			D		AA	R

Choose the **Standard** button and select a related standard file and field in the resulting window.



Functional Scope of Standard Files

The following attributes of a standard field can be rippled to coupled fields at lower levels.

- Field length
- Field format
- Field type
- Suppression option
- Uniqueness option
- Descriptor type (see below)
- Character set
- Timestamp, timezone and precision
- Inline length

If an attribute is not defined in a standard field (which means the attribute is blank if it is alphabetic, or zero if it is numeric), no rippling takes place for this attribute and the lower-level object can be modified without restriction. It is therefore possible to have some field attributes defined centrally and others modifiable without restriction at lower levels. See also *Changing Coupled fields*.

Note: If one of the attributes above is changed and this change is not compatible with the coupled field, the attribute Check against standard of the field is set to N. For example: If you change a field type to HY (hyperdescriptor, this change is not rippled to coupled fields in DB2 files and the attribute Check against standard of the coupled fields is set to N.

Rippling the Attribute Descriptor Type

The attribute Descriptor type of a standard field can have the following values:

D	Disallowed. The descriptor type of coupled fields must be blank. All non-blank descriptor types in coupled fields are set to blank.
F	Force. The descriptor type of coupled fields may not be blank. If a coupled field has a non-blank descriptor type, no rippling is performed. If a coupled field has descriptor type blank, the descriptor type is set to N and a message is given.
blank	Undefined. The descriptor type of coupled fields can be any value, including blank. No checks are performed, no rippling takes place.

Rippling Verifications

When the verification list of a standard field is edited, corresponding changes are automatically made in the verification list of every field derived from the standard field. The following rules apply:

- Every verification contained in the verification list of a standard field must also be contained in the verification list of a field coupled to that standard field. However, the sequence of verifications in the lists can differ.
- If a verification is removed from the verification list of a standard field, the verification is automatically removed from the verification lists of all coupled fields.
- If a verification is added to the verification list of a standard field (at any position), the verification is automatically added to the end of the verification list of all coupled fields.
- If the parameter Check against standard is set to N in connected fields, the checks listed above are not performed.

Changing Coupled Fields

The following rules apply when changing fields at lower levels:

- Attributes not defined in a standard field can be modified in coupled fields.
- Attributes that have been defined in standard fields cannot be modified in coupled fields.
- If an attribute of a coupled field that is defined in the standard field has to be changed, the fields must be uncoupled. See below.

Uncoupling Fields from Standard Fields

Fields can be temporarily or permanently uncoupled from the standard field with the parameter Check against standard.

- **Temporarily**
  From the **Fieldlist** tab choose the **Standard** button. In the upcoming window empty the box **Check against Standard**. The field is uncoupled temporarily from the standard field from which it was derived. The coupling can be reactivated by filling the box **Check against standard**.
- **Permanently**
  From the **Fieldlist** tab choose the **Standard** button. In the upcoming window remove the names of the standard file and field. The field is uncoupled permanently from the standard field from which it was derived. The coupling cannot be reactivated with the parameter Check against standard. To recouple a field, you have to enter the names of the standard file and field.

Rippling from Master Files to Views/Userviews

The following rules apply:

- Changes to master fields are rippled to fields in userviews that were derived from master files. If the master field is coupled to a standard field, changes to the standard field are rippled to the coupled master field and to the derived field in the userview.
- Changes to fields in userviews are rejected if they are not compatible with the master field.

For example: if a field in a userview is derived from a master field of type T (time), the field in the userview can only be changed to format P with length 13.

All other changes are rejected.

Coupling of Master Fields and Fields in Views/Userviews

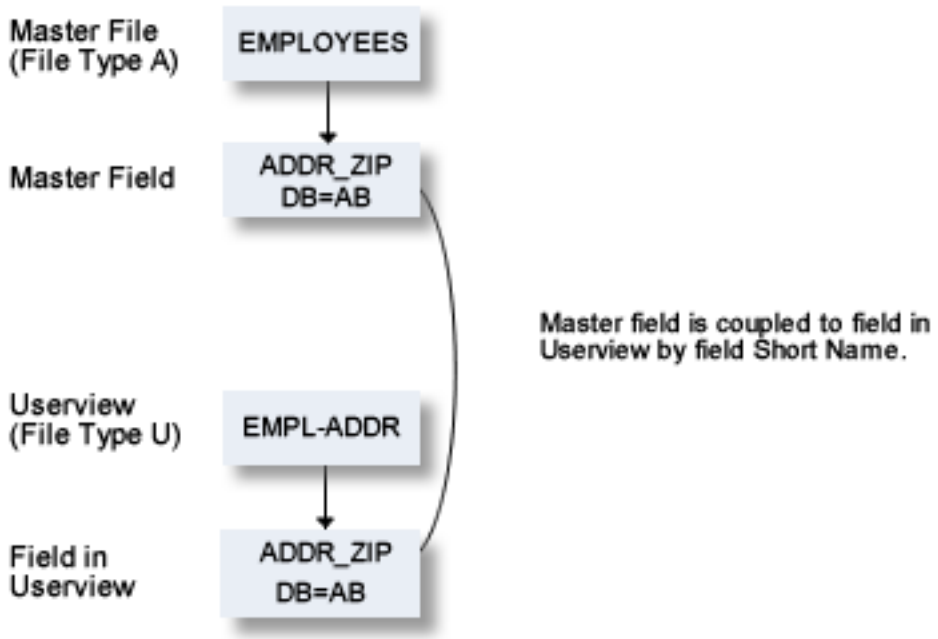
The coupling between master files and views/userviews depends on whether the view is derived from a single master file or from one or several master files.

Single-Master Views

Userviews are derived from one of the following master files:

- Adabas file
- Physical and logical VSAM files
- IMS Segments
- Entire System Server files

Master fields and fields of Userviews are coupled by field short name (column DB in field maintenance screens).



The following table indicates the valid combinations of view types and master file types:

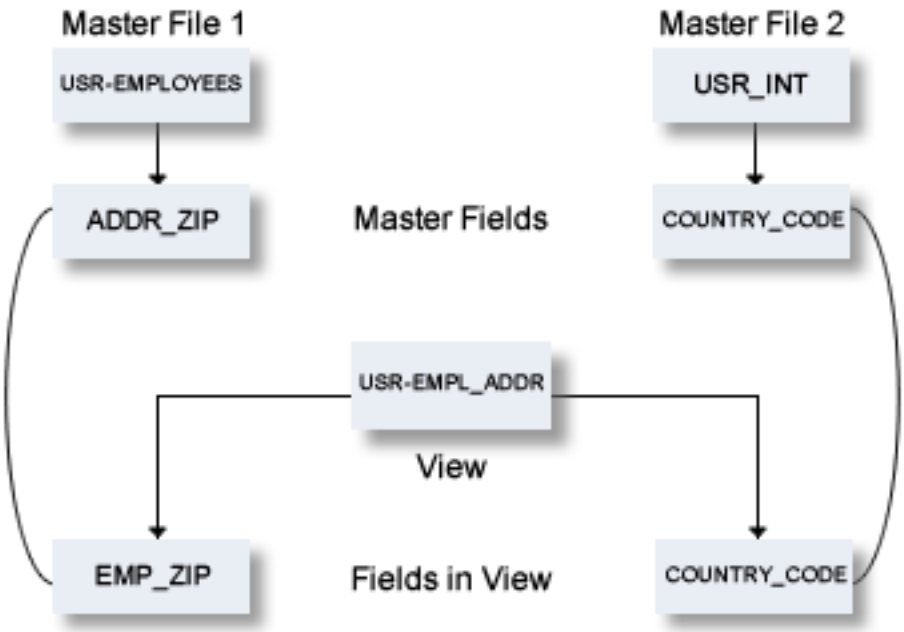
Type of View	Type of Master File
AT	A
B	A(SQL) AT, B
BV	BT, BV
E, IV	D, E, IV
J	I

Type of View	Type of Master File
JV	JT, JV
K	I
L	V
OV	OT, OV
Q	P
R	L
U	A
W	V
XV	XT, XV
YV	YT, YV

Multiple-Master Views

For views which can be derived from several master files, the coupling is established by parameters from Table/View ID and from Field ID in the field List of the file documenting the view. This applies to the following master file types:

- Adabas Files (with SQL usage set to Y)
- Adabas Cluster Tables
- Adabas D Table
- DB2 Table
- Informix Table
- Ingres Table
- Oracle Table
- Sybase Table



Functional Scope for Master Files

If fields in a master file are modified, views and userviews coupled to these fields are changed accordingly. The following rules apply for this rippling:

Attributes which are always Rippled

The following attributes are always rippled:

- short name (if applicable)
- Field type
- suppression / null value option
- uniqueness option
- character set
- null default option

Attributes which are Rippled if Identical

The following attributes are rippled if the attribute values in the userview and the master field were identical before the master field was modified:

- Field ID
- length, format (both must be identical)
- max. occ.
- gr. structure
- justify
- header / edit mask
- Field/View name name synonym

Abstract

The abstract of a field is rippled according to the setting of the following parameter in the Profile SYSTEM

Ripple abstract	N	Abstract is not rippled.
	T	Abstract is rippled.
	L	Abstract is rippled only if the abstract was identical in the view/userview and the master file before the abstract was changed in the master file.

Rippling Verifications from Master Field to View/Userview

When a verification list of a master field is edited, corresponding changes are automatically made in the verification list of fields in the view/userview derived from the master file. The following rules apply:

- The verification list of a field in a userview does not have to contain all the verifications that are contained in the list of the master file field from which the userview field has been derived.
- If a verification is removed from the verification list of a master field, the verification is automatically removed from the verification list of coupled fields.
- If a verification is added to the verification list of a master field, it is automatically added to the verification list of coupled fields.

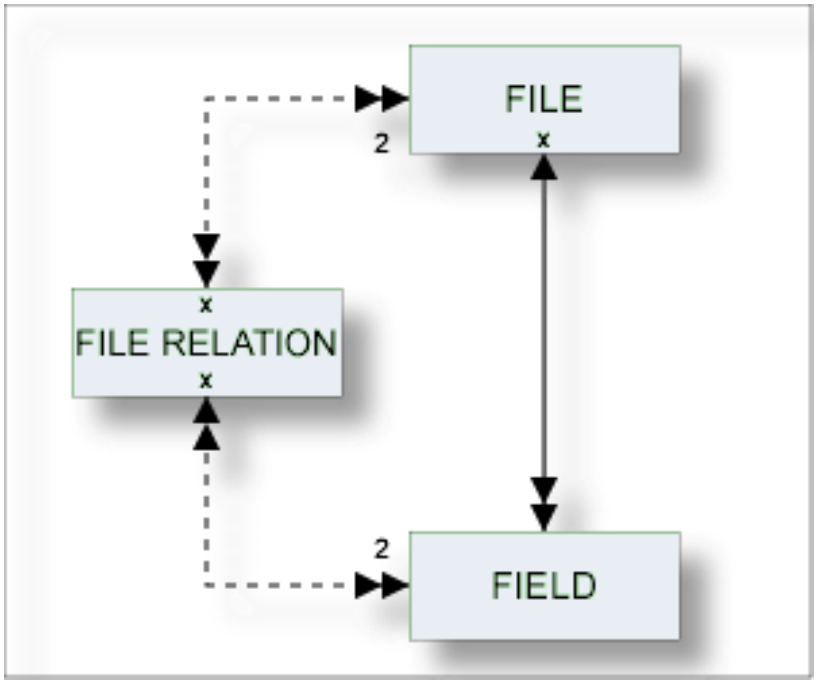
VII

- 32 File Relation 255
- 33 Interface 263

32 File Relation

- File Relation Maintenance 256

The object type File Relation documents relationships between files. The relationship is established by means of references to fields.



File Relation Maintenance

The following topics are covered below:

- File Relation Types
- Add a File Relation
- Validity Checks for File Relations

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

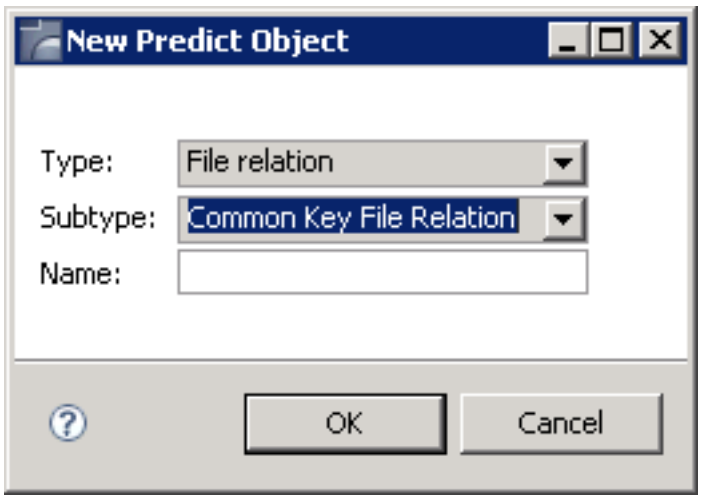
File Relation Types

The table below contains a list of all valid file relation types.

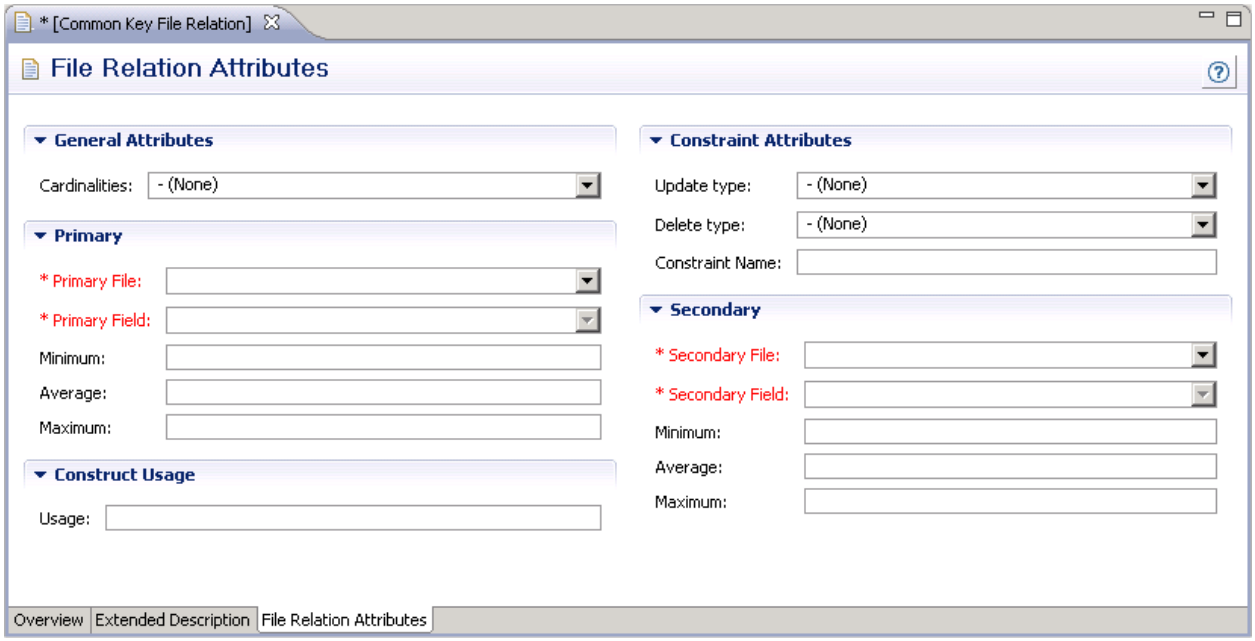
Code	File Relation Type
C	Two files of type A are physically coupled.
D	The file relation is only documented.
K	Common keys. This file relation type is only valid for file types YT and YV (Sybase tables and views). The field linked to the file relation must have a non-blank descriptor type. Predict checks whether the number, formats and character sets of the fields - or source fields in the case of superdescriptors - in file 1 and file 2 agree. For Sybase, you can generate a common key from a file relation of this type. For other database management systems, file relations of this type are used for documentation purposes only.
N	This file relation type documents the models used by Natural Construct. See *Defining File Relations for Objects in Predict* in the *Natural Construct User's* documentation.
R	Ref. Constraint. Files of type AT, BT, D, JT, OT, X, XT, XV, Y, and YV are connected by referential integrity.
S	Files of type A are soft coupled.

Add a File Relation

When you add a file relation, you first have to specify the file relation type in the **New Predict Object** dialog box.



When you choose the **OK** button, a file relation type-specific window appears. The file relation type is indicated in the title bar.



The parameters listed below can be found on the following tabs: **File Relation Attributes**, **Construct Usage** and **Constraint Attributes**.

Note: Parameters not listed here are described under *Global Attributes*.

Parameters		
Cardinalities	The number of records of each file that is permitted in any occurrence of the file relation. Valid values:	
	1	one (must be one)
	C	none or one (can be one)
	CM,CN	one or one or more (can be many)
	M, N	one or more (must be at least one)
Primary File/Field	One of the related files. If the type of **file relation is R**, the field which is used to link this table must be a unique descriptor or ISN (Adabas), a primary index (for DB2) or a unique key (for other SQL systems).	
Secondary File/Field	The other related file. If the type of **file relation is R**, the field which is used to link this table must be one of the following: ■ descriptor (**descriptor D**) ■ foreign key (**descriptor E**) ■ foreign index (**descriptor F**) ■ primary index (**descriptor P**)	
Minimum	The minimum number of occurrences of a field from the primary or secondary file in the file relation.	

Parameters	
Average	The average number of occurrences of a field from the primary or secondary file in the file relation.
Maximum	The maximum number of occurrences of a field from the primary or secondary file in the file relation.
Constraint Attributes	
Update type	The type of constraint to be applied. C – Cascade. R – Restricted. L – Suffix as line number (file relation type D or N). N – Renumber suffix (file relation type D or N). S – Set NULL. D – Set Default. A – No Action.
Delete type	The type of constraint to be applied. C – Cascade. R – Restricted. L – Suffix as line number (file relation type D or N). N – Renumber suffix (file relation type D or N). S – Set NULL. D – Set default. A – No Action.
Constraint name	The constraint name for a file relation of type D and R. For **files of type A**, the constraint name must follow the Adabas short name conventions. For details refer to *Field Names* in *Record and Field Definitions* of the section *Adabas Design* in the *Adabas Concepts and Facilities* documentation.
Enforce	Only applicable to file relations of type D or R. Specifies whether or not the referential constraint is enforced by DB2 during normal operations such as insert, update or delete. Check the box to enforce the referential constraint..
Construct Usage	
Usage	Only applicable to file relations of type N or D. Describes how the file relation is evaluated in Natural Construct: A – Construct aggregate. I – Construct inheritance.

Validity Checks for File Relations

The validity checks performed by Predict depend on the file relation type:

Code C

Type	Applicable for	Validity Checks
Physically Coupled	Adabas	May not be any of the following: ■ redefined field ■ group ■ periodic group ■ member of a periodic group ■ hyperdescriptor ■ phonetic descriptor The two fields in the file relation must be descriptors with the same length and format.

Code D

Type	Applicable for	Validity Checks
Documented	all types	None

Code K

Type	Applicable for	Validity Checks
Common Keys	Sybase tables and views	The field linked to the file relation must have a non-blank descriptor type

Code N

Type	Applicable for	Validity Checks
Natural Construct	all types	Both the field and file containing the file relation must be defined in Predict.

Code R

Type	Applicable for	Validity Checks
Referential Constraint	Adabas File Adabas Cluster Table DB2 Table Oracle Table Adabas D Table Informix Table or View	Must be marked in the table of *file 1*: For file type DB2 table or Informix table/view: ■ as primary index (descriptor type P), ■ foreign index (descr. type F) ■ or index (descr. type D), ■ and as unique (unique option U) For file type Adabas file: as unique descriptor or ISN. For file type Adabas cluster table: as primary index (descriptor type P). For other file types: as unique (unique option U). Must be marked in the table of *file 2*: for file type Adabas file: as descriptor. For file type Adabas cluster table: ■ as foreign index (descr. type F) ■ or foreign key (descr. type E); For other file types: ■ as primary index (descr. type P), ■ foreign index (desc. type F) ■ or foreign key (descr. type E).

Code S

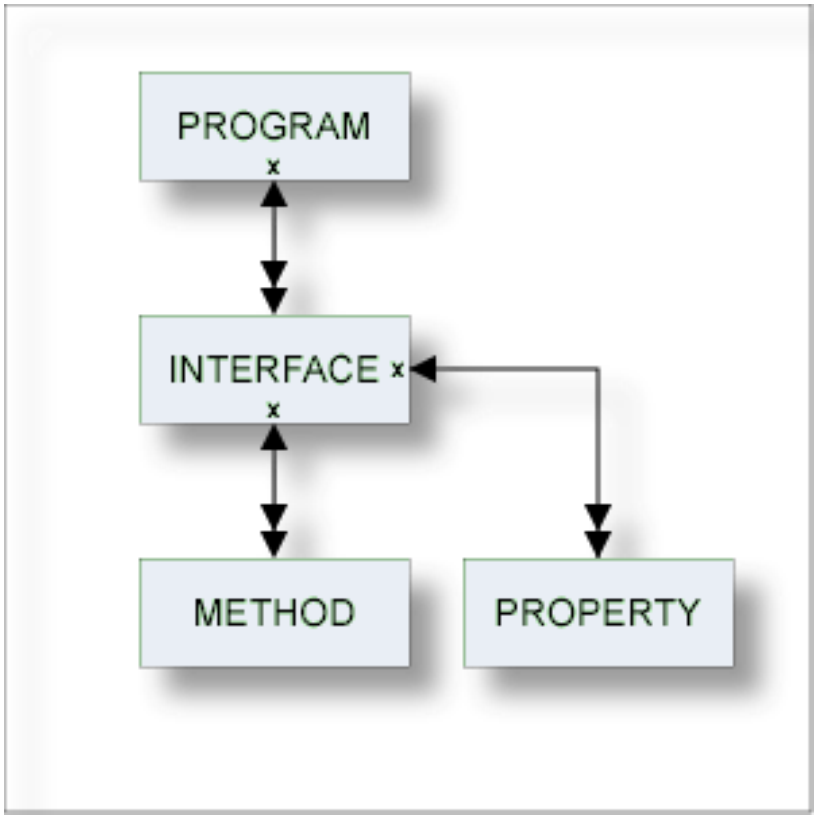
Type	Applicable for	Validity Checks
Soft-coupled	Adabas	May not be any of the following: ■ redefined field ■ group ■ periodic group ■ member of a periodic group ■ hyperdescriptor ■ phonetic descriptor

Type	Applicable for	Validity Checks
		The first field in the file relation must be a descriptor; the second field must have the same format.

33 Interface

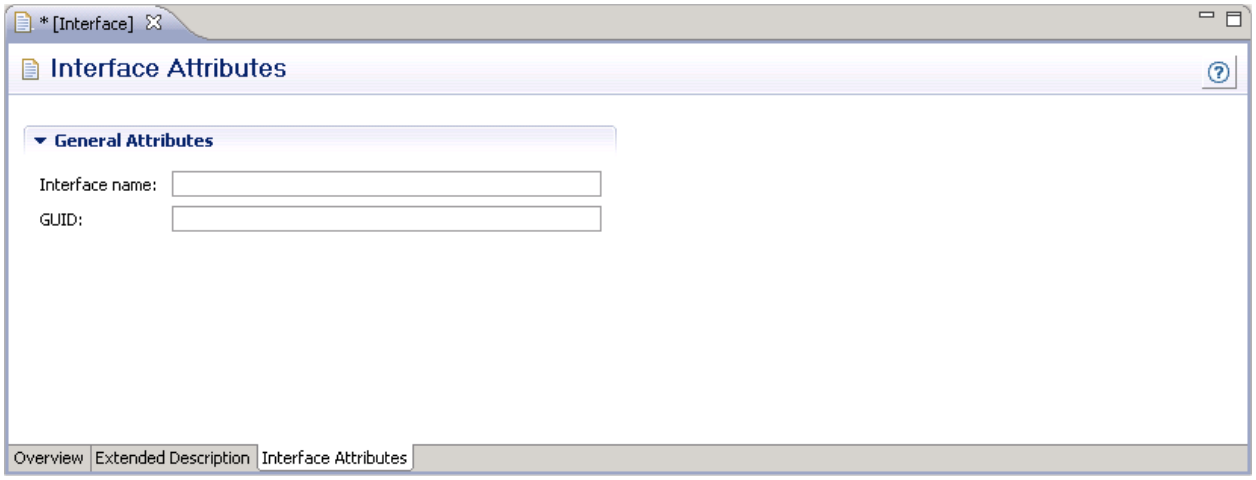
- Defining Basic Attributes of an Interface 265

This object type, together with object types Method, Property and Program, is used to document the Natural programming object class.



For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Defining Basic Attributes of an Interface

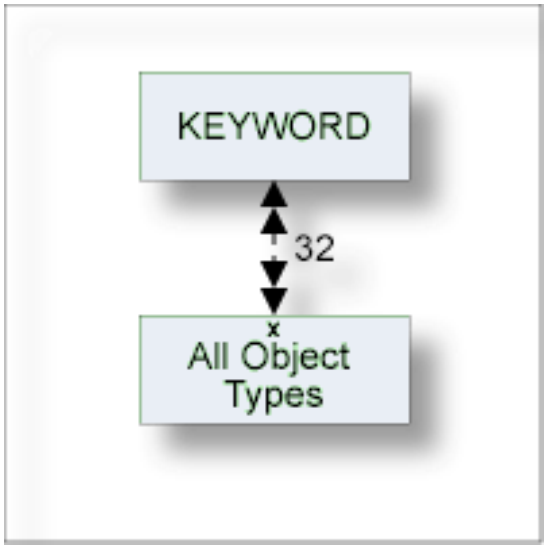


Note: Parameters not listed here are described under *Global Attributes*.

Parameters	
Interface name	Name of the interface.
GUID	The globally unique ID of the interface.

VIII Keyword

Predict objects of type Keyword are used to relate objects logically, for example, all objects belonging to an application or all objects used in a particular business context.



In the predefined Predict metastructure, a keyword can be related as a child object to objects of all types including other keywords.

The description of object type Keyword is organized under the following headings:

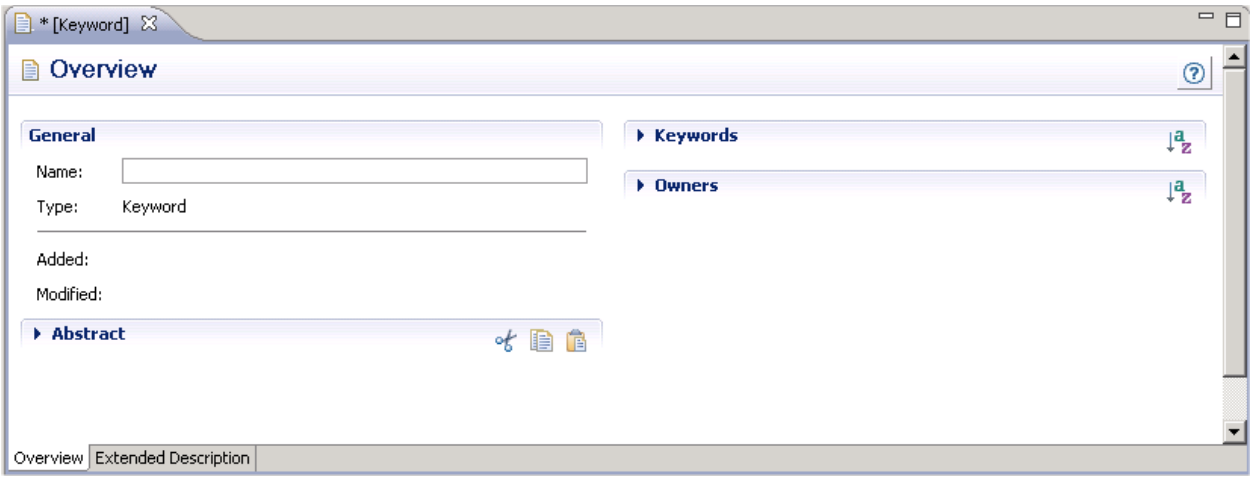
Maintaining Objects of Type Keyword

34 Maintaining Objects of Type Keyword

- Defining Basic Attributes of Keyword 270
- Keyword Maintenance Functions 270

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Defining Basic Attributes of Keyword



The parameters are described under ***Global Attributes***.

Keyword Maintenance Functions

Standard maintenance functions are described in the section *Maintenance* in the *Predict Reference* documentation. The following functions are described below.

- Purge Keyword

Purge Keyword

Predict objects of type Keyword are purged with the **Delete** command.

The following are deleted:

- the keyword object
- all links to child objects
- all links from parent objects

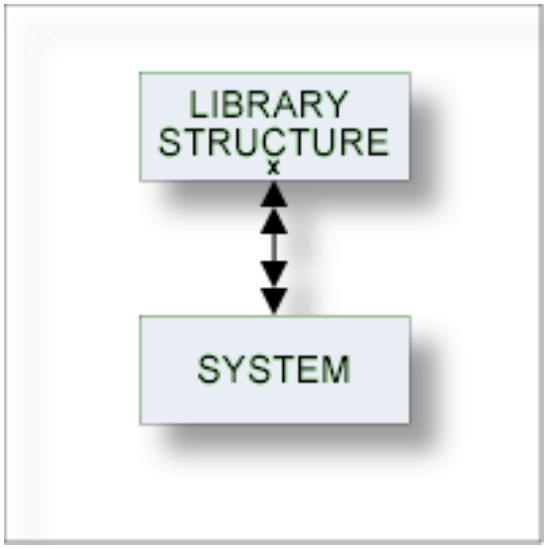
IX

- 35 Library Structure 273
- 36 Method 277
- 37 Network 281
- 38 Node 285
- 39 Packagelist 289

35 Library Structure

- Add/Modify Library Structure 275

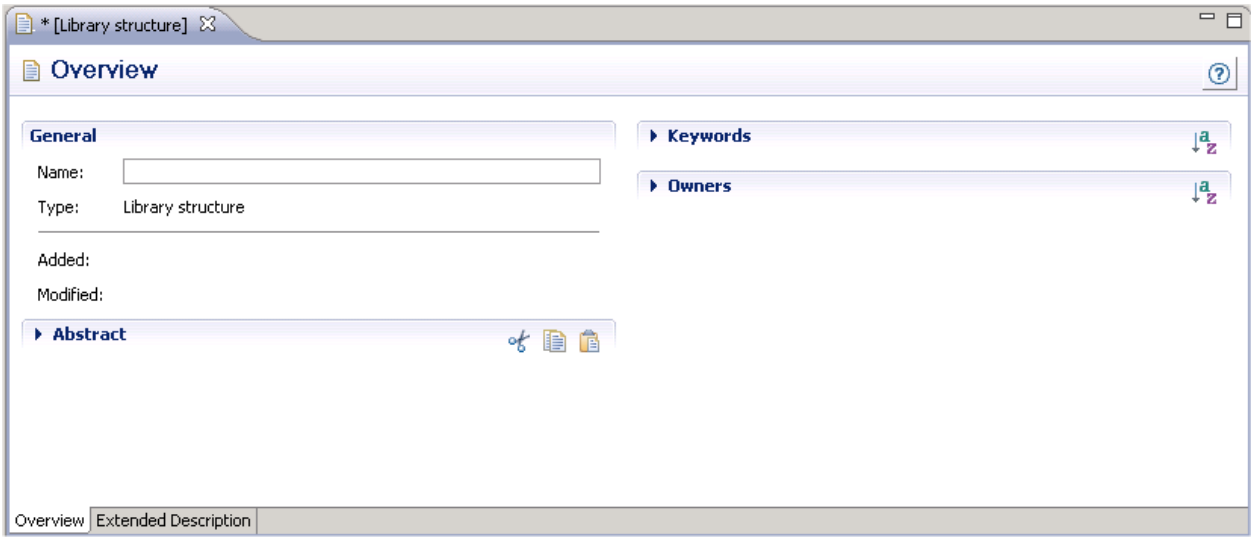
Programs that are called by another program are not necessarily in the same library as the calling program: it is possible that they are loaded from a steplib at runtime. An object of type Library Structure documents a structure which describes a runtime or development environment (for example libraries for copy code). The corresponding systems are linked as child objects of type System to the library structure.



See also section *Steplib Support* in the *Predict Reference* documentation for more information.

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Add/Modify Library Structure



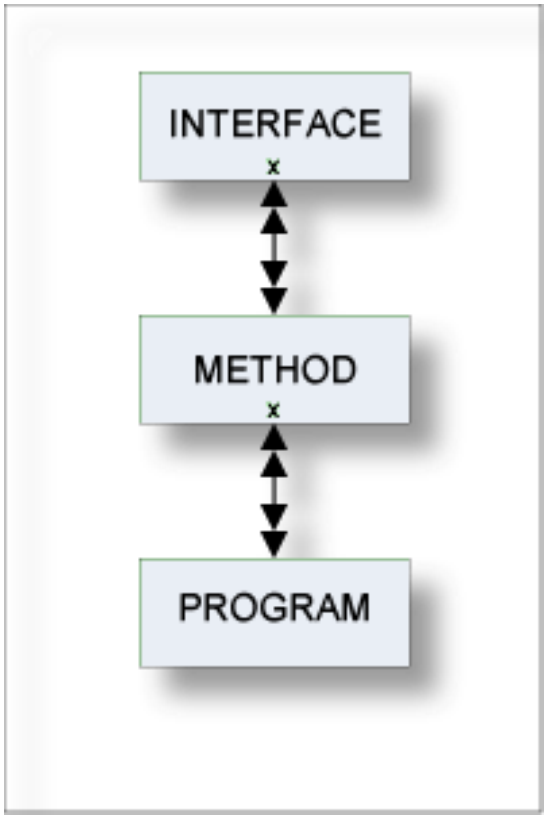
Parameters

The parameters are described under *Global Attributes*.

36 Method

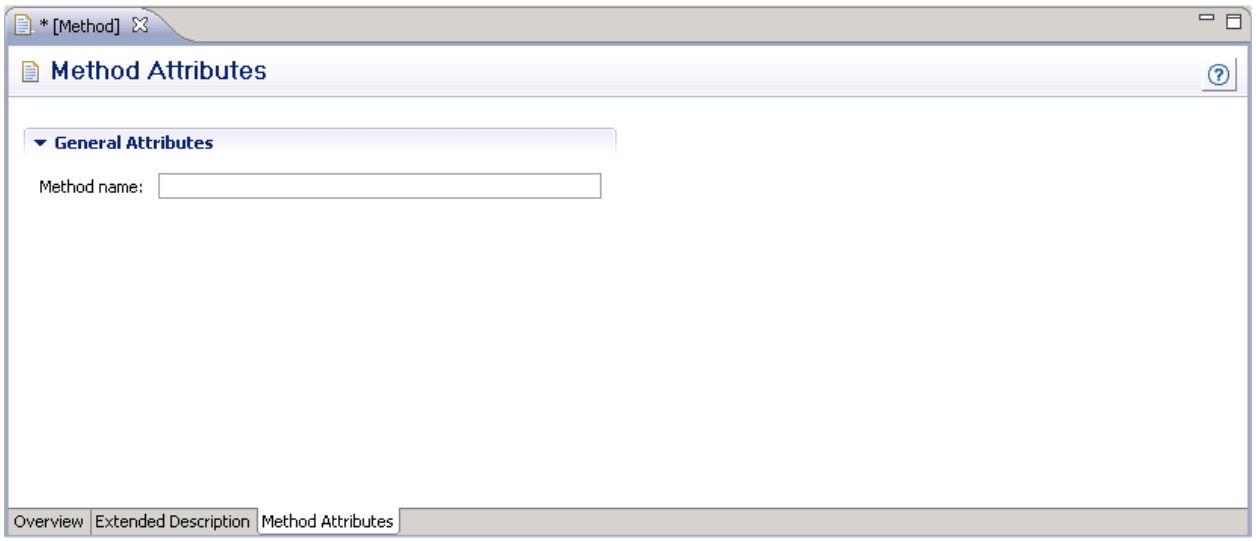
- Add/Modify a Method 279

This object type is used to document the methods of an interface.



For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Add/Modify a Method

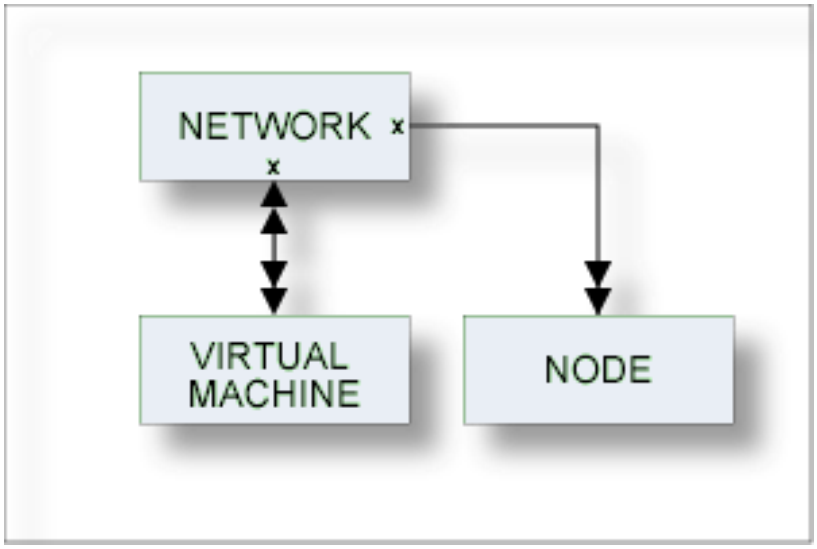


Note: Parameters not listed here are described under *Global Attributes*.

Parameters	
Method name	Name of the method.

37 Network

- Add a Network 283
- Network-Specific Maintenance 283



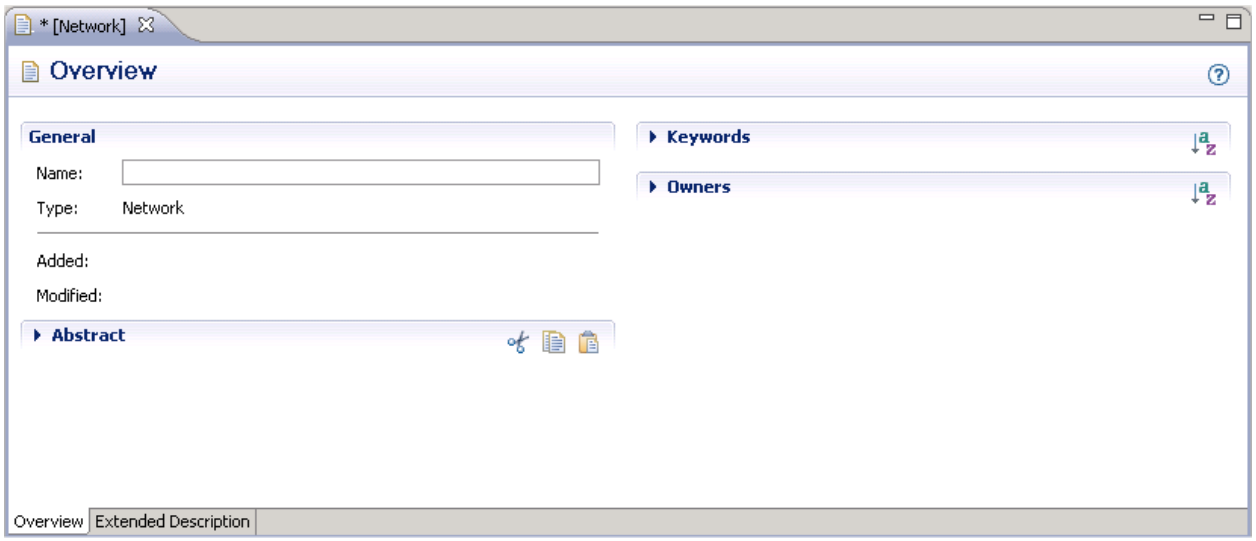
The location of a database must be specified by linking each database to an object of type Virtual Machine and each virtual machine to an object of type Network.

The current network will be taken as default for virtual machine objects if no network is specified.

See *Defining the Distribution of Data in Predict* in the section *Adabas Vista* in the *Predict and Other Systems* documentation for a description of how to define the distribution of data.

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Add a Network



The parameters are described under *Global Attributes*.

Network-Specific Maintenance

Standard maintenance functions are used for maintaining networks. These are described in the section *Maintenance* in the *Predict Reference* documentation.

The special rules applying to function Purge Network are described below.

Purge Network

Predict objects of type Network are purged with the **Delete** command.

The following rules apply:

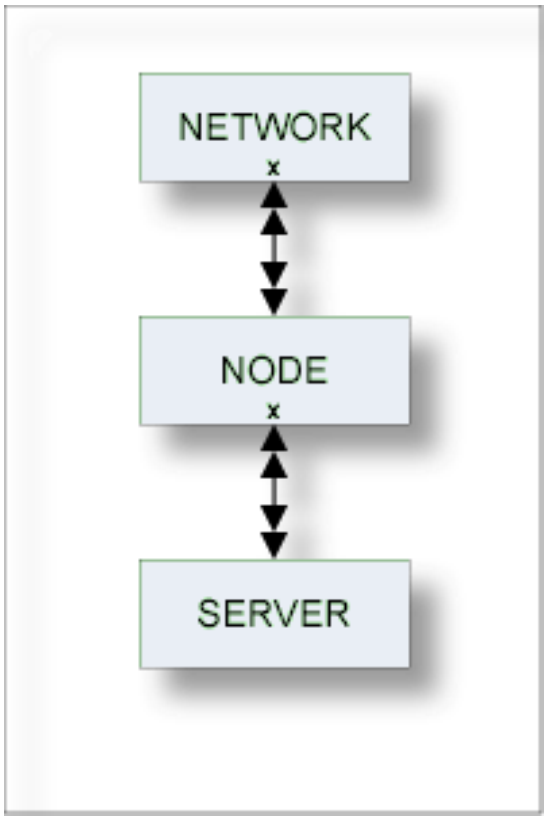
- A network that is linked to a virtual machine via "Uses VM" cannot be deleted.
- The network defined as current network in the General defaults cannot be deleted.

38 Node

- Add a Node 287

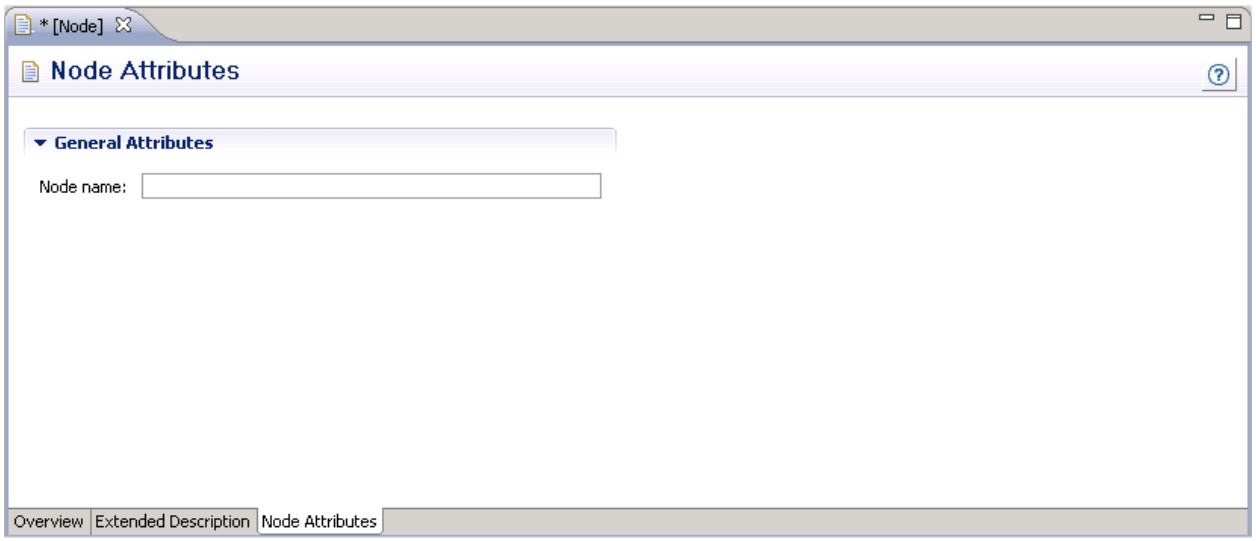
This object type, together with object type Server, is used to document remote procedure calls.

An object of type Node documents the physical machine containing the server.



For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Add a Node



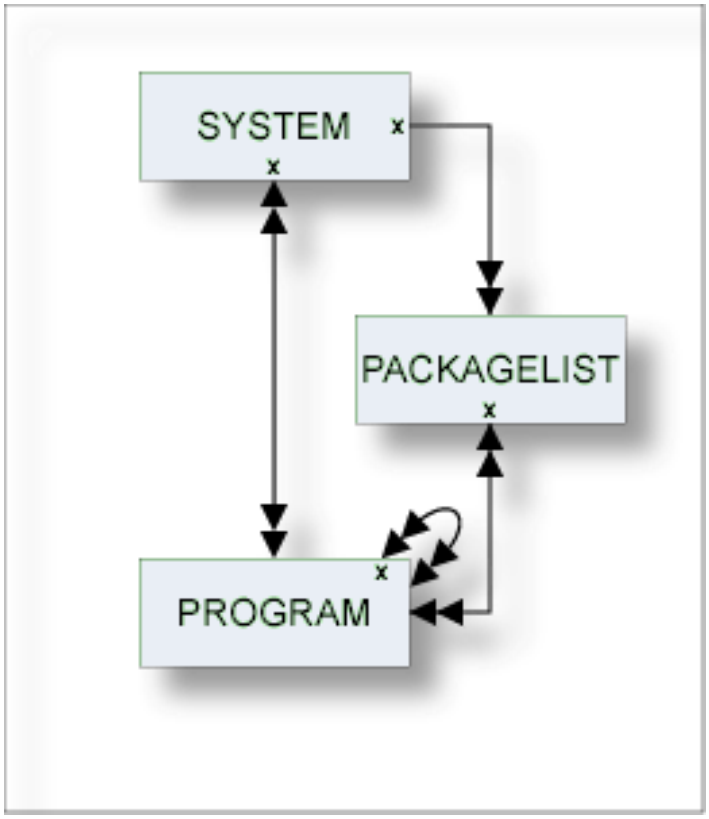
Note: Parameters not listed here are described under *Global Attributes*.

Parameters	
Node name	Name of the node. Up to 8 characters.

39 Packagelist

- Packagelist Types 290
- Add a Packagelist 291
- Packagelist-Specific Maintenance 292

The Predict object type Packagelist is used to document DB2 packages.



Note: Packagelists of type T and packagelists of type S are related using the parameters Collection name and Location name.

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

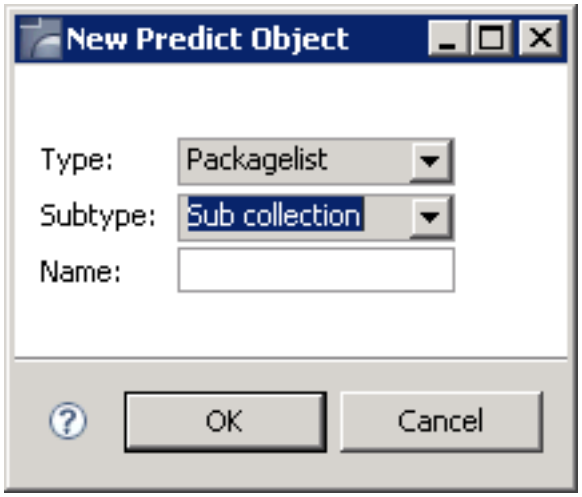
Packagelist Types

The table below contains a list of all valid packagelist types.

Code	Packagelist Type
Q	Database request module (DBRM). Packagelists of type Q contain one DBRM which is directly bound to the plan.
S	Subcollection. Packagelists of type T and packagelists of type S are connected using the parameters Collection name and Location name. Each package in a packagelist of type S is also contained in a packagelist of type T.
T	Total collection. Packagelists of type T provide an overview of all packages used in a collection. The parameters Collection name and Location name are mandatory for packagelists of type T.

Add a Packagelist

When you add a packagelist, you first have to specify the packagelist type in the **New Predict Object** dialog box.

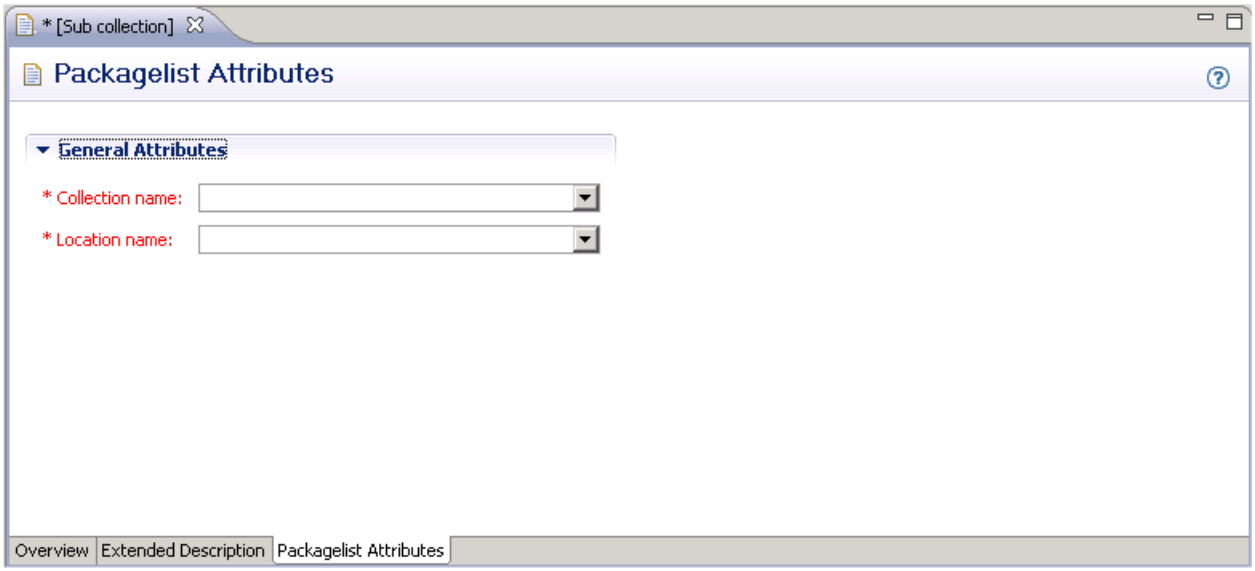


When you choose the **OK** button, a packagelist type-specific window appears. The packagelist type is indicated in the title bar.

Predict ensures the consistency of related packagelists (types T and S):

- If a package is purged from a packagelist of type T, it is purged automatically from corresponding packagelists of type S.
- If a package is added to a packagelist of type S, it is added automatically to the corresponding packagelist of type T.

The **Packagelist Attributes** tab is shown for packagelists of type T and S. For Packagelists of type Q only the tabs for the global attributes are provided. Parameters not listed here are described under *Global Attributes*.



Parameters	
Collection name	From version 2.3 of DB2 and above, packages are always referenced via collections. A collection is a virtual summary of packages, used to simplify references to packages. In Predict, collections are documented as attributes of packagelists. Packagelists are grouped by including several packages to the same collection. A collection is documented in Predict with the attributes collection name and location name. A collection name can be up to 18 characters long.
Location name	Together with collection names, location names identify collections uniquely. A location name can be up to 16 characters long.

Packagelist-Specific Maintenance

Purge Packagelist

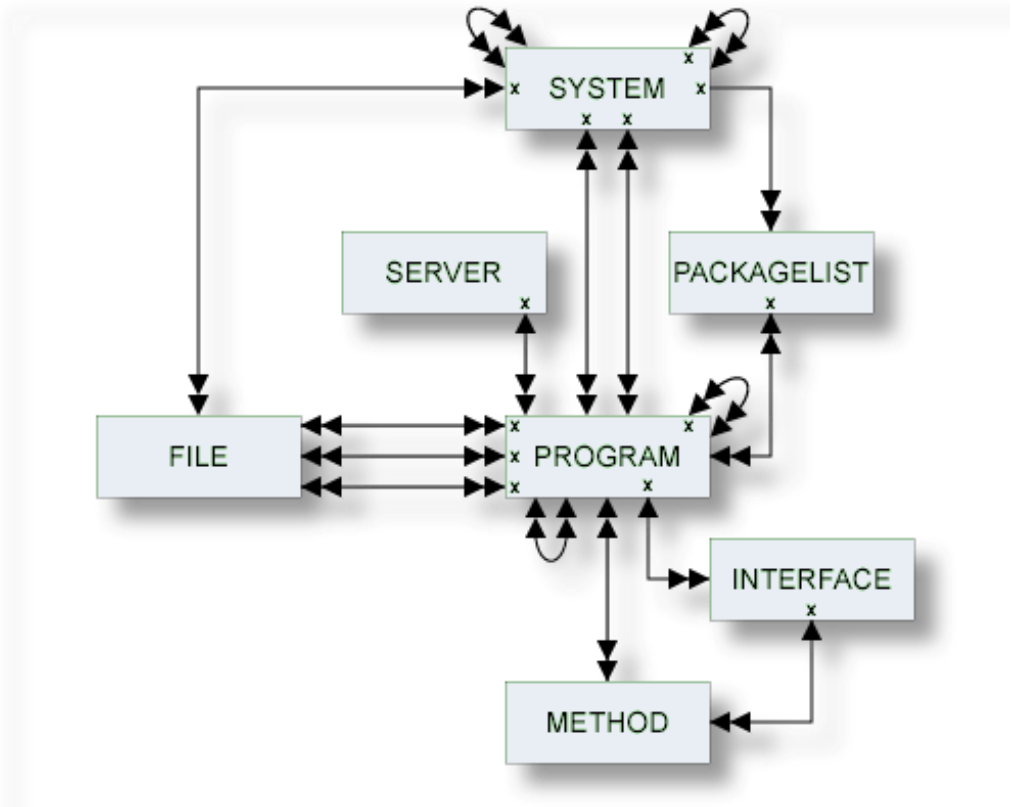
Predict objects of type Packagelist are purged with the **Delete** command.

The following rules apply:

- The following objects are deleted:
  - the packagelist object
  - all links to child objects
  - all links from parent objects
- With packagelists of type T, all packagelists of type S connected to the packagelist via the attributes Collection/Location name are deleted as well.

X Program

Predict knows more than a dozen different types of programs, ranging from parameter data area to Natural Expert model. About a dozen different programming languages are supported.



The description of object type Program is organized under the following headings:

Maintaining Objects of Type Program
Defining Additional Attributes of Programs
Program Specific Maintenance

40 Maintaining Objects of Type Program

- Program Types 296
- Languages 297
- Program-Specific Libraries 297
- Add a Program 298

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Program Types

The table below contains a list of all valid program types.

Code	Program Type
A	Parameter data area
C	Copy Code
D	Documented program
E	External program
F	Function
G	Global data area
H	Help routine
I	Dynamic (see ***Programs of Type dynamic***)
J	Job
K	ISPF Macro
L	Local data area
M	Map/Help map
N	Subprogram
O	Natural command processor
P	Main program
R	SQL procedure
S	Natural subroutine
T	Dialog
U	Database function
X	Text
Y	Expert Model
1	Error Message
4	Class
5	Resource
8	Adapter
blank	Undefined

Languages

The table below contains a list of all valid languages.

Code	Language
B	BAL (Assembler)
C	COBOL
E	Natural EL
F	FORTRAN
G	ADA
H	C
J	Job Control Language
N	Natural
O	Other
P	PL/I
Q	Static SQL
R	REXX
S	SQL Procedure Language
V	Java
Z	System Program, see *System Programs*
0	Language 0
1	Language 1
2	Language 2
3	Language 3
blank	Unknown

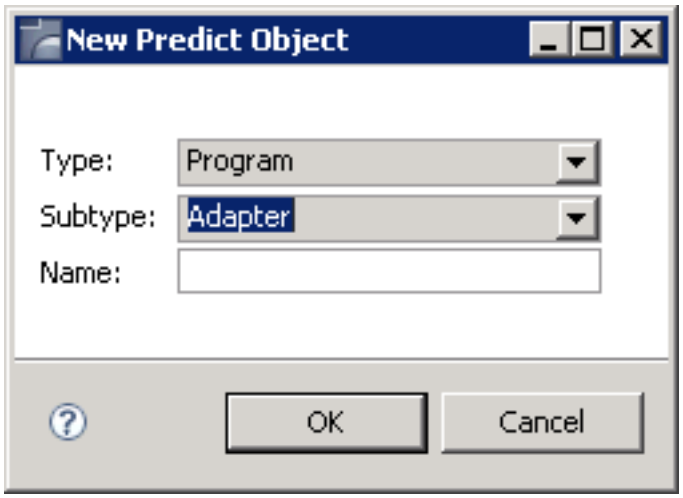
Program-Specific Libraries

The table below contains a list of all valid program-specific libraries.

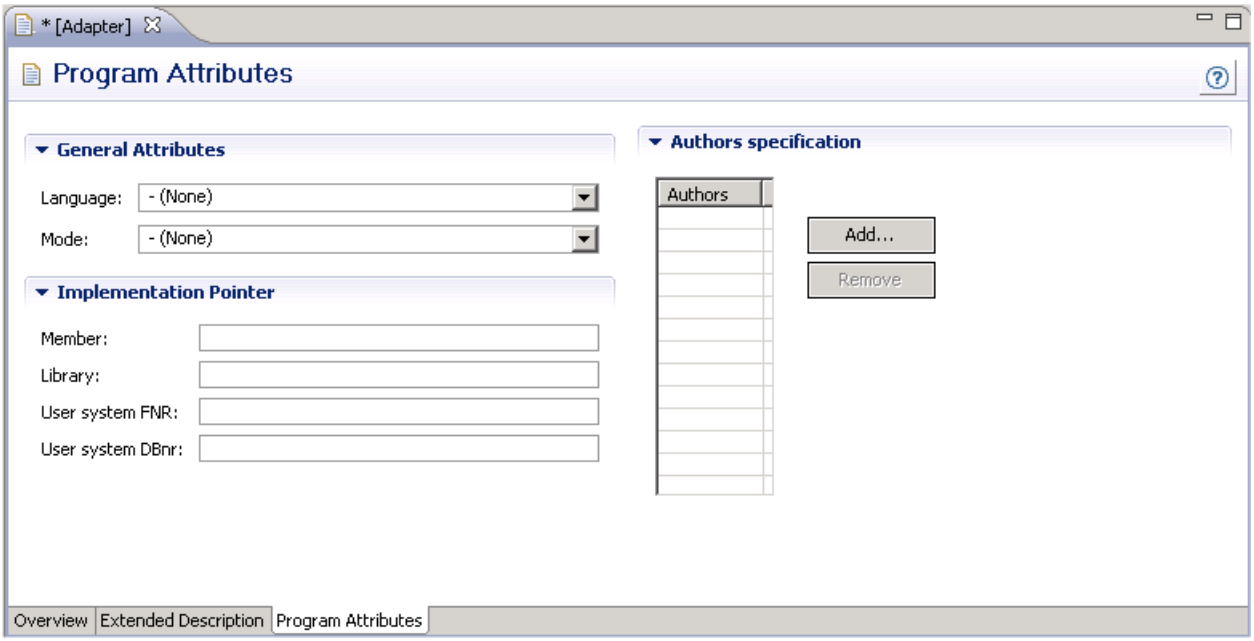
Library Name	Description
SYSADA	for ADA
SYSBAL	for ASSEMBLER
SYSCCC	for C
SYSCOB	for COBOL
SYSFOR	for FORTRAN
SYSPLI	for PL/I
SYSSTA	for Static SQL
SYSSYS	for system programs
user-defined	library of a 3GL application environment; must be documented in an object of type System

Add a Program

When you add a program, you first have to specify the program type in the **New Predict Object** dialog box.



When you choose the **OK** button, a program type-specific window appears. The program type is indicated in the title bar.



Note: Parameters not listed here are described under *Global Attributes*.

Parameters	
Program Attributes	
Language	The language in which the program is written. See *Overview of Language-Specific Program Types* for a table of valid program type/language combinations.
Mode	Mode of operation in which the program is used. A — All (both online and batch modes) B — Batch mode O — Online blank — Undefined
Load-Lib	The load library. Not available for all program types.
Implementation Pointer	
Member	Member documented by the Predict program (not applicable to programs of type 5).

Library	The name of the library in which the member is stored (not applicable to programs of type D).
	■ For Natural programs: see the table in *Overview of Language-Specific Program Types*.
	■ For 3GL programs:
	  ■ one of the standard 3GL libraries (see *Program-Specific Libraries*),
	  ■ any library, provided that it is documented in a Predict system object of type G.

User system FNR	The number of the user system file. For 3GL programs, the number is always 255.

Parameters	
User system DBnr	The number of the database in which the user system file is located. For 3GL programs, the number is always 255.
NAT-Func	Applicable only to Natural subroutines (type S). The name of the function of the subroutine (DEFINE SUBROUTINE). If an asterisk is entered, Predict derives the function name from XRef data if XRef data exists for the specified member.

41 Defining Additional Attributes of Programs

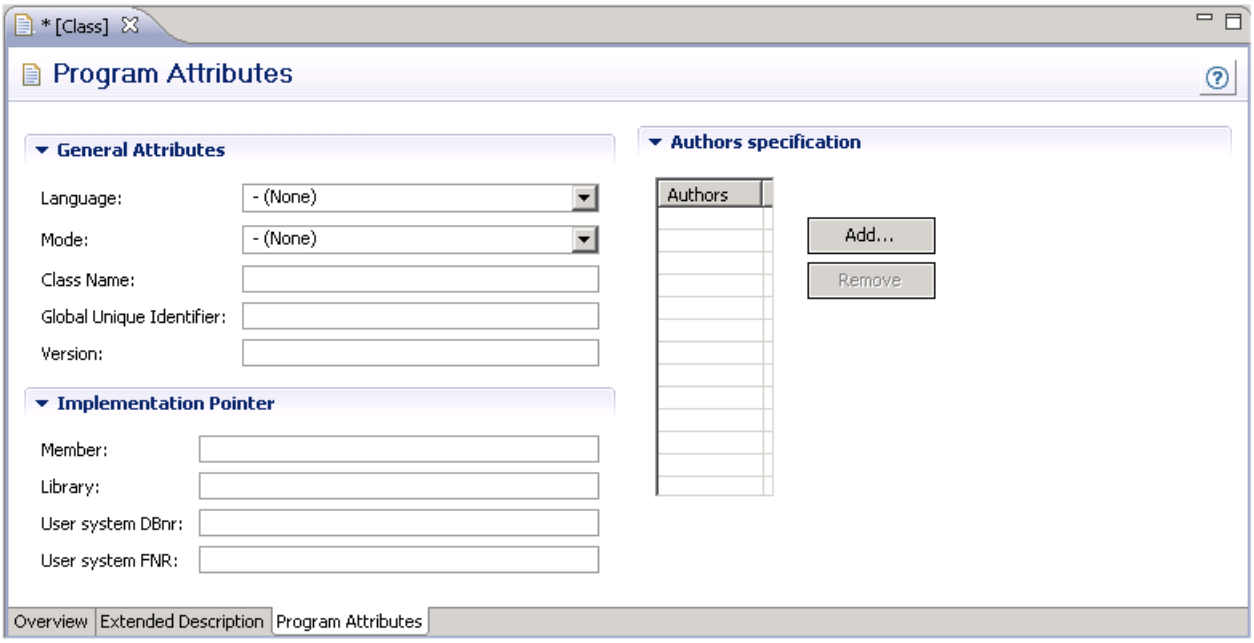
- Programs of Type Class 303
- Programs of Type Resource 304
- Programs of Type SQL Procedure 305
- Programs of Type Database function 313
- System Programs 316
- Programs of Type dynamic 316

- **Class definition**
  See *Programs of Type Class*.
- **Resource definition**
  See *Programs of Type Resource*.
- **Database function options**
  See *Programs of Type Database function*.
- **Procedure options**
  See *Programs of Type SQL Procedure*.
- **Entry points**
  Entry points are to be modified. This is valid only for programs written in certain languages.
  See *Overview of Language-Specific Program Types*.
- **SQL procedure code**

The following rules apply:

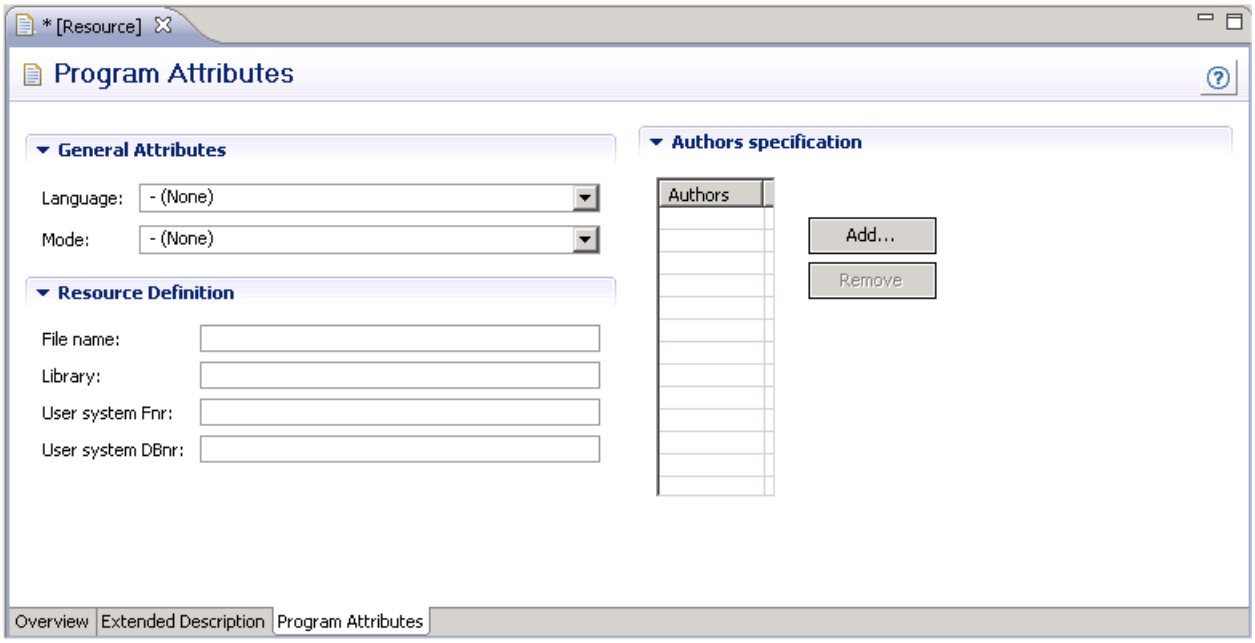
- Only those types of additional attributes appear in the window that apply to the type of program. For example: the option Class definition is not contained in the list when a program of type Resource is processed.
- More than one choice can be made at a time. The respective input maps are then displayed one after the other.

Programs of Type Class



Parameters	
Class Name	The name of the class.
Global Unique Identifier	The globally unique ID of the class.
Version	The version number of the class.

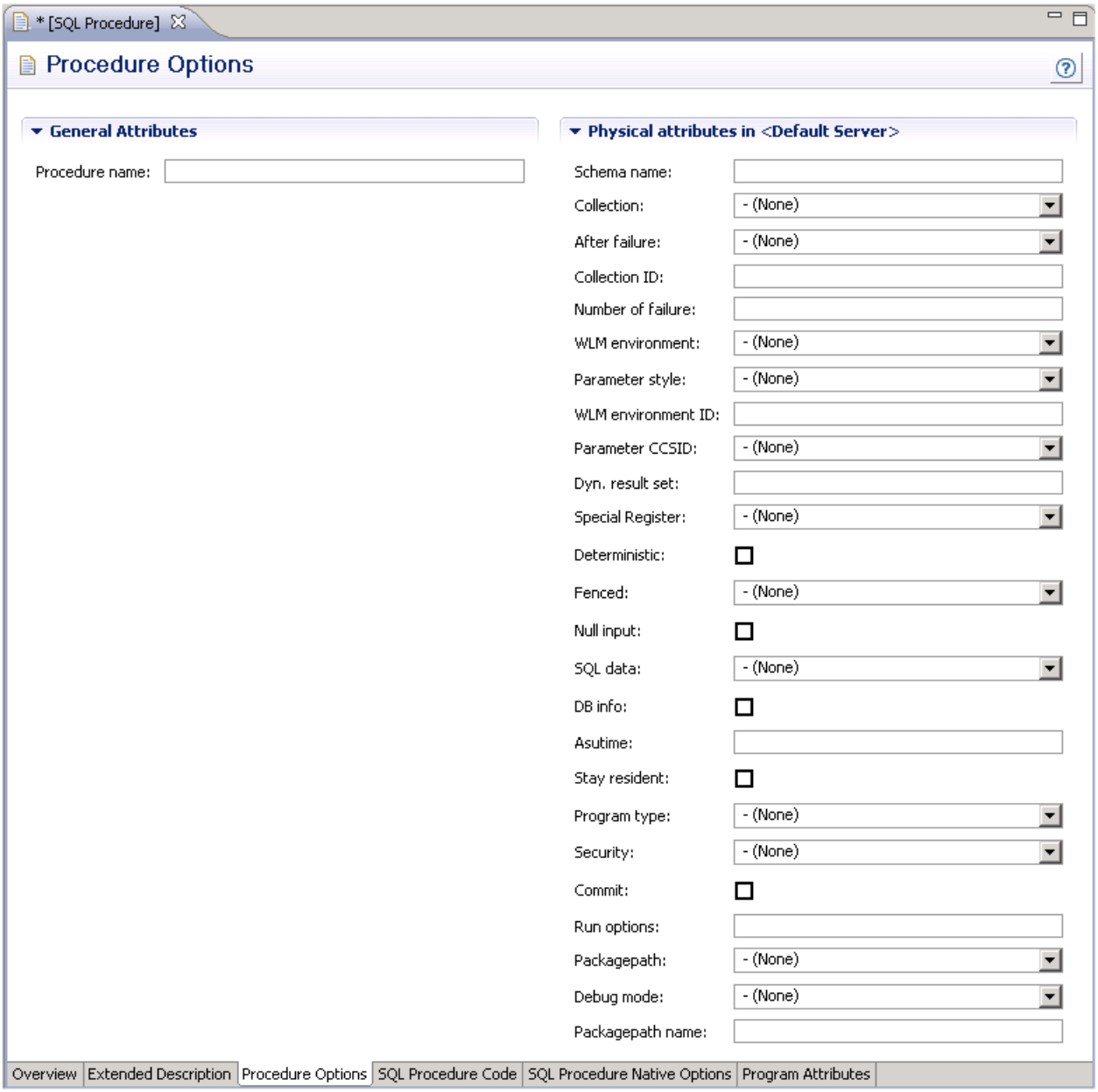
Programs of Type Resource



Parameters	
File name	File name documented by the Predict program.
Library	The name of the library in which the file name is stored.
User System Fnr	The number of the user system file.
User System DBnr	The number of the database in which the user system file is located.

The type of Resource can be documented in the language field of a Predict program object. There is a user exit program U-PGMLAN that allows dynamic extension of possible languages in each installation.

Programs of Type SQL Procedure



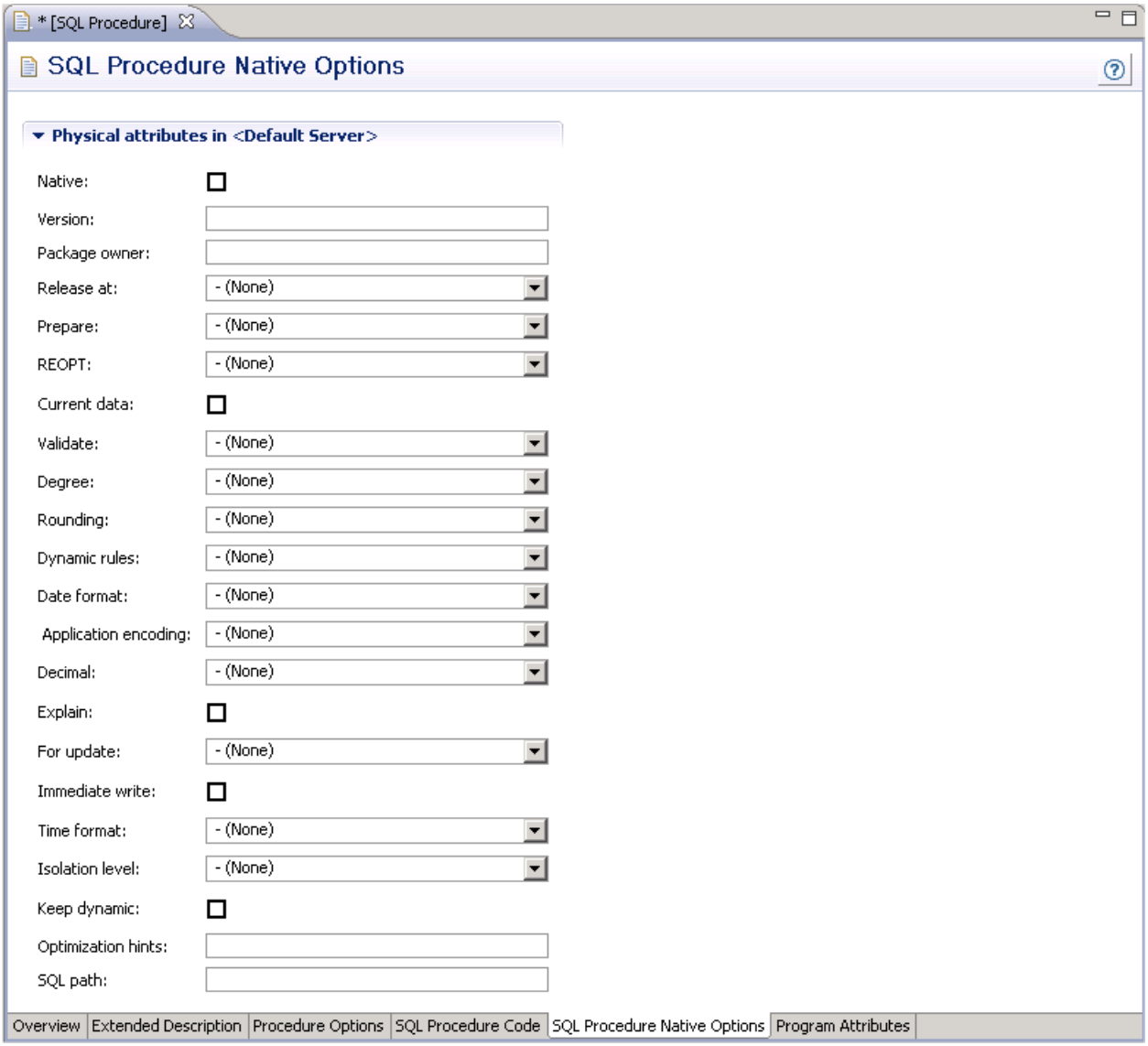
Parameters	
Procedure name	This name must comply with SQL naming conventions. See the section *Naming Conventions for SQL Objects* in the section *Adabas D and Other SQL Systems* in the *Predict and Other Systems* documentation.
Schema name	Used as a qualifier for an unqualified procedure name.
Specific name	Specifies a unique name for the procedure.
Collection	Identifies the package collection. N: NO COLLID Y: Use collection-ID. A collection-ID must then be specified. blank: none
WLM environment	Identifies the MVS workload manager application environment.
Dyn. result set	Specifies the maximum number of query result sets that the stored procedure can run.
Deterministic	Specifies whether the procedure returns the same results for identical arguments. Y: Yes N: No blank: undefined
Null input	Specifies whether the procedure is called if any of the input arguments is null at execution time. Y: Yes N: No
DB info	Specifies whether specific information that is included in DB2 is passed to the procedure when it is invoked. Y: Yes N: No
Stay resident	Specifies whether the load module for the procedure remains resident in memory when the procedure ends.
Security	Specifies how the procedure interacts with an external security product. D: DB2 F: Definer U: User blank: none
After failure	Specifies the action to be taken after a failure has occurred. Valid values: D: Stop (system default). Stops after number of failures defined in the system defaults. N: Stop (number). Stops after number of failures defined by the user. If N is specified, enter a numeric value in the field **Number of failure**. C: Continue.

Parameters		
	blank	none
Run options	Specifies the language environment run-time options to be used for the procedure.	
Packagepath	Specifies the package path to use when the procedure is run	
	N	No packagepath.
	Y	Use packagepath. A list of package collections must be specified.
	blank	none
Parameter style	Identifies the linkage convention use to pass parameters to the procedure.	
	D	DB2SQL
	G	General
	N	General with nulls
	J	Java
	blank	none
Parameter CCSID	Specifies the encoding scheme. Valid values:	
	blank	none
	A	ASCII
	E	EBCDIC
	U	Unicode
Special Register	Valid values:	
	I	Inherit. The values of special registers are inherited.
	D	Default. Special registers are initialized to the default values.
	blank	none
Fenced	Determines that the external procedure runs in an external address space.	
	Y	Yes
	N	No
SQL data	Indicates whether the procedure can execute any SQL statements.	
	M	Modifies SQL data
	N	No SQL
	R	Read SQL data
	S	Contains SQL
	blank	none
Asutime	Specifies the total amount of processor time.	
Program type	Specifies whether the procedure runs as a main or a subroutine.	
	S	Sub

Parameters		
	M	Main
	blank	none
Commit	Indicates whether DB2 commits the transaction immediately on return from the procedure.	
	Y	Yes
	N	No
Debug Mode	Specifies whether the procedure can be run in debugging mode. The default is D (Disallow) when no Dynamic rules run behavior is in effect. Valid values:	
	D	Disallow.
	A	Allow.
	I	Disable.
	blank	None.

- Native SQL Procedure

Native SQL Procedure



Valid values are:

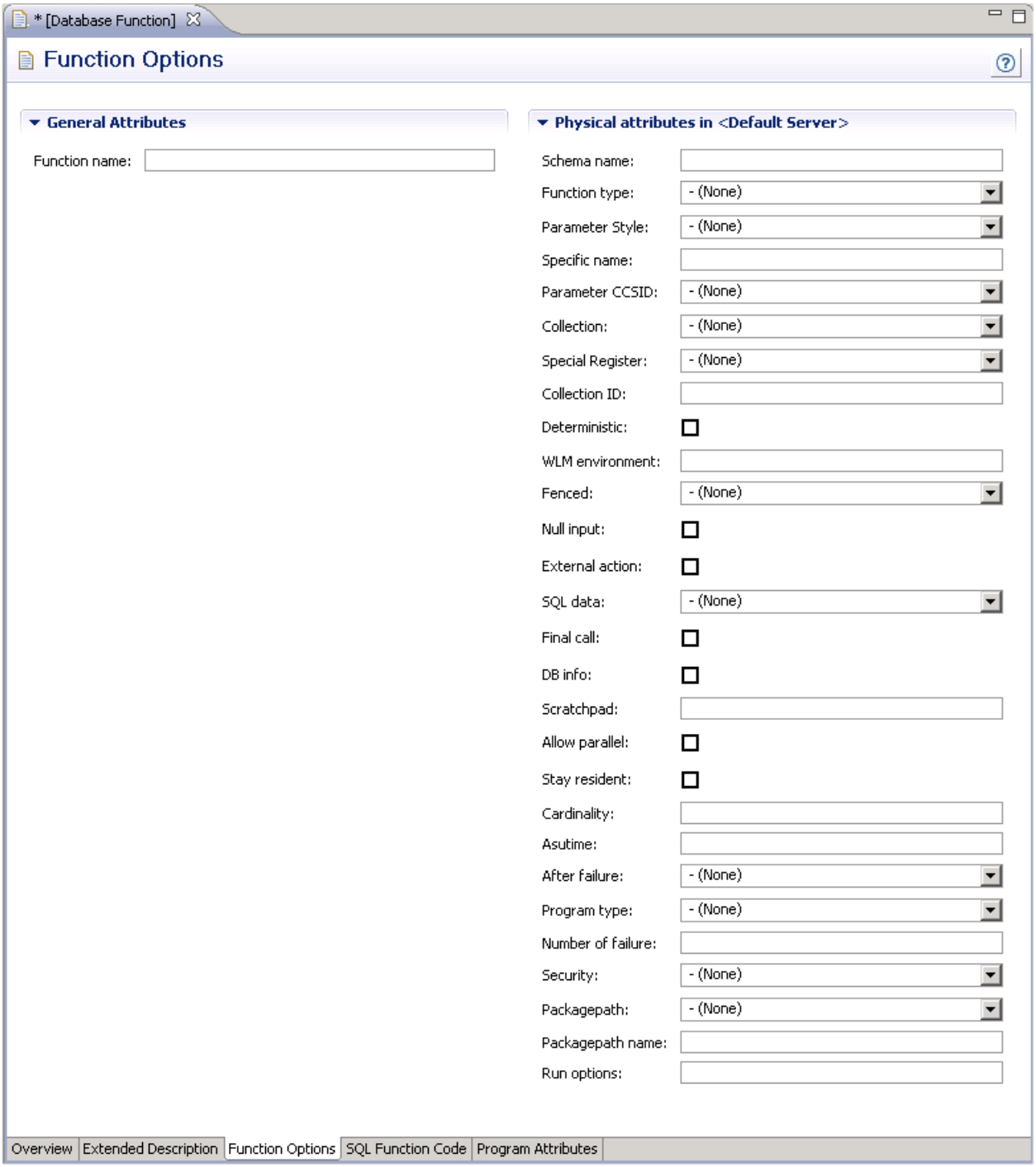
Parameters		
Native	To indicate if the definition is for a native SQL procedure.	
	Y	Yes.
	N	No. This is the default.
	blank	none
Version	Specifies the procedure version identifier. The default is V1.	

Parameters		
Package owner	Specifies the owner of the package.	
Prepare	Specifies whether to defer preparation of dynamic SQL statements that refer to remote objects, or to prepare them immediately.	
	D	Defer.
	N	Nodefer.
	blank	Not specified. This is the default.
Current data	Specifies whether to require data currency for read-only and ambiguous cursors when the isolation level of cursor stability is in effect.	
	Y	Yes.
	N	No. This is the default.
	blank	none
Degree	Specifies whether to attempt to run a query using parallel processing to maximize performance.	
	1	One. This is the default.
	A	Any.
	blank	Not specified.
Dynamic rules	Specifies the values that apply, at run time, for the following dynamic SQL attributes:	
	R	Run. This is the default.
	B	Bind.
	D	Definebind.
	E	Definerun.
	I	Invokebind.
	N	Invokerun.
	blank	Not specified.
Appl. encoding	Specifies the default encoding scheme for SQL variables in static SQL statements in the routine body.	
	A	ASCII.
	E	EBCDIC.
	U	Unicode.
	blank	Not specified.
Explain	Specifies whether information will be provided about how SQL statements in the routine will execute.	
	Y	Yes.
	N	No. This is the default.
	blank	Not specified.

Parameters		
Immediate write	Specifies whether immediate writes are to be done for updates that are made to group buffer pool dependent page sets or partitions.	
	Y	Yes.
	N	No. This is the default.
	blank	Not specified.
Isolation level	Specifies how far to isolate the routine from the effects of other running applications.	
	C	Cursor stability.
	S	Read stability.
	R	Repeatable read.
	U	Uncommitted read.
	blank	Not specified.
Keep dynamic	Specifies whether DB2 keeps dynamic SQL statements after commit points.	
	Y	Yes.
	N	No. This is the default
	blank	Not specified.
Optimization hints	Specifies query optimization hints.	
SQL path	Specifies the SQL path.	
Release at	Specifies when to release resources that the procedure uses: either at each commit point or when the procedure terminates.	
	C	Commit. This is the default.
	D	Deallocate.
	blank	Not specified.
REOPT	Specifies if DB2 will determine the access path at run time by using the values of SQL variables or SQL parameters, parameter markers, and special registers.	
	N	None. This is the default.
	A	Always.
	O	Once.
	blank	Not specified.
Validate	Specifies whether to recheck, at run time, errors of the type `OBJECT NOT FOUND` and `NOT AUTHORIZED` that are found during bind or rebind.	
	R	Run. This is the default.
	B	Bind.
	blank	Not specified.
Rounding	Specifies the desired rounding mode for manipulation of DECFLOAT data.	
	C	Ceiling.

Parameters		
	D	Down.
	F	Floor.
	1	Half down.
	2	Half even.
	3	Half up.
	U	Up.
	blank	Not specified. This is the default.
Date format	Specifies the date format for result values that are string representations of date or time values.	
	I	ISO.
	E	EUR.
	U	USA.
	J	JIS.
	L	Local.
	blank	Not specified. This is the default.
Decimal	Specifies the maximum precision that is to be used for decimal arithmetic operations.	
	15	
	15,*n* (where *n* must be a number between 1 and 9)	
	31	
	31,*n* (where *n* must be a number between 1 and 9)	
	blank	Not specified. This is the default.
For update	Specifies whether the `FOR UPDATE` clause is required for a `DECLARE CURSOR` statement if the cursor is to be used to perform positioned updates.	
	R	Required. This is the default.
	O	Optional.
	blank	Not specified.
Time format	Specifies the time format for result values that are string representations of date or time values.	
	I	ISO.
	E	EUR.
	U	USA.
	J	JIS.
	L	Local.
	blank	Not specified. This is the default.

Programs of Type Database function



Parameters	
Function name	This name must comply with SQL naming conventions. See the section *Naming Conventions for SQL Objects* in the section *Adabas D and Other SQL Systems* in the *Predict and Other Systems* documentation.
Function type	The type of the function. S: Scalar T: Table
Schema name	Used as qualifier for an unqualified function name.
Specific name	Specifies an unique name for the function.
Collection	Identifies the package collection. N: NO COLLID Y: Use collection-ID. A collection-ID must then be specified. blank: none
WLM environment	Identifies the MVS workload manager application environment.
Special Register	Valid values: I: Inherit. The values of special registers are inherited. D: Default. Special registers are initialized to the default values. blank: none
Deterministic	Specifies whether the function returns the same results for identical arguments. Y: Yes N: No
Null input	Specifies whether the function is called if any of the input arguments is null at execution time. Y: Yes N: No
External action	Specifies whether the function takes an action that changes the state of an object that DB2 does not manage. Y: Yes N: No
Final call	Specifies whether final call is made to the function. Y: Yes N: No
DB info	Specifies whether specific information that DB2 knows is passed to the function when it is invoked. Y: Yes N: No

Parameters	
Asutime	Specifies the total amount of processor time.
Program type	Specifies whether the function runs as a main or a subroutine. S: Sub M: Main blank: none
Run options	Specifies the language environment run-time options to be used for the function.
Packagepath	Specifies the package path to use when the function is run N: No packagepath. Y: Use packagepath. A list of package collections must be specified. blank: none
After failure	Specifies the action to be taken after a failure has occured. Valid values: D: Stop (system default). Stops after number of failures defined in the system defaults. N: Stop (number). Stops after number of failures defined by the user. If N is specified, enter a numeric value in the field **Number of failure**. C: Continue. blank: none
Parameter Style	Specifies the conventions used for passing parameters to and returning the value from functions. Valid values: D: DB2SQL J: Java blank: not specified
Parameter CCSID	Specifies the encoding scheme. Valid values: blank: none A: ASCII E: EBCDIC U: Unicode
Fenced	Determines that the external function runs in an external address space. Y: Yes N: No
SQL data	Indicates whether the function can execute any SQL statements. M: Modifies SQL data N: No SQL R: Read SQL data

Parameters		
	S	Contains SQL
	blank	none
Scratchpad	Specifies whether DB2 provides a scratchpad for the function.	
Allow parallel	Specifies whether parallelism can be used.	
	Y	Yes
	N	No
Cardinality	Specifies an estimate of the expected number of rows that the function returns.	
Stay resident	Specifies whether the load module for the function remains resident in memory when the function ends.	
Security	Specifies how the function interacts with an external security product.	
	D	DB2
	F	Definer
	U	User
	blank	none

System Programs

Programs that are only available as object code and hence have no language are documented with programs of type E (external object) and language Z (system program). Predict creates XRef data for these so called system programs because neither the preprocessor nor Natural can create XRef data for object code.

The implementation pointer for a system program has to be specified explicitly. One entry point (with the ID of the program object) is created by Predict, additional entry points have to be specified manually.

Programs of Type dynamic

Programs of type dynamic are used to document calls of programs of the same name from different steplibs depending on the library structure. The following rules apply:

- Because programs of type dynamic document any number of implemented members, no check is performed as to whether the members documented by the program are actually implemented.
- With the active retrieval function Programs using programs, programs of type dynamic are ignored as current objects.
- Programs of this type can only have children for association "Uses PR concept".

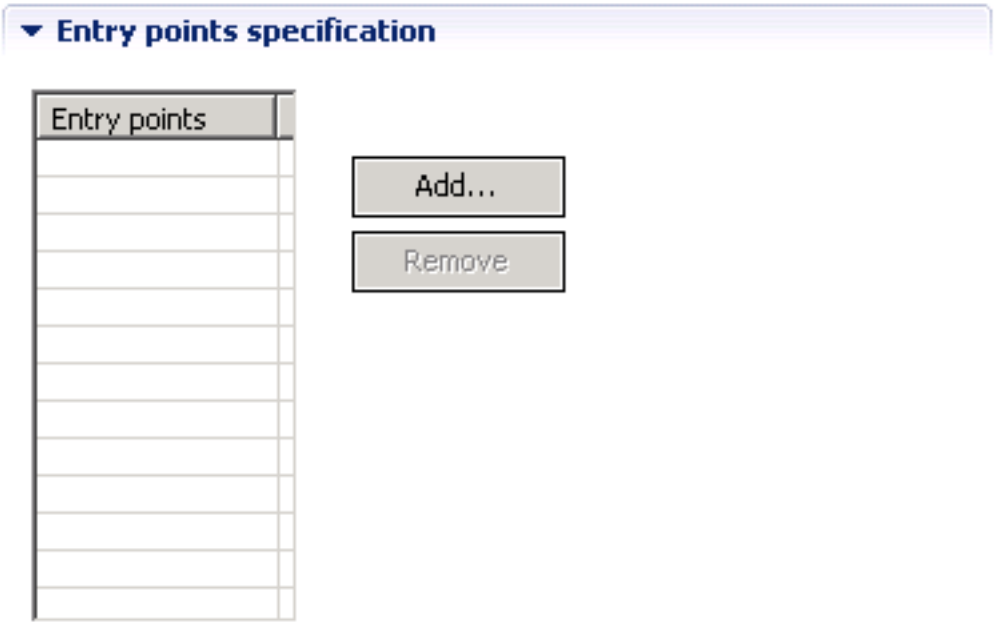
42 Program-Specific Maintenance

- Editing Entry Points 318
- Overview of Language-Specific Program Types 318
- SQL Procedure Code Tab 320

Standard maintenance functions are described in the section *Maintenance in Predict* in the *Predict Reference* documentation.

Editing Entry Points

To edit the lists of entry points use the **Entry points specification** section.



Overview of Language-Specific Program Types

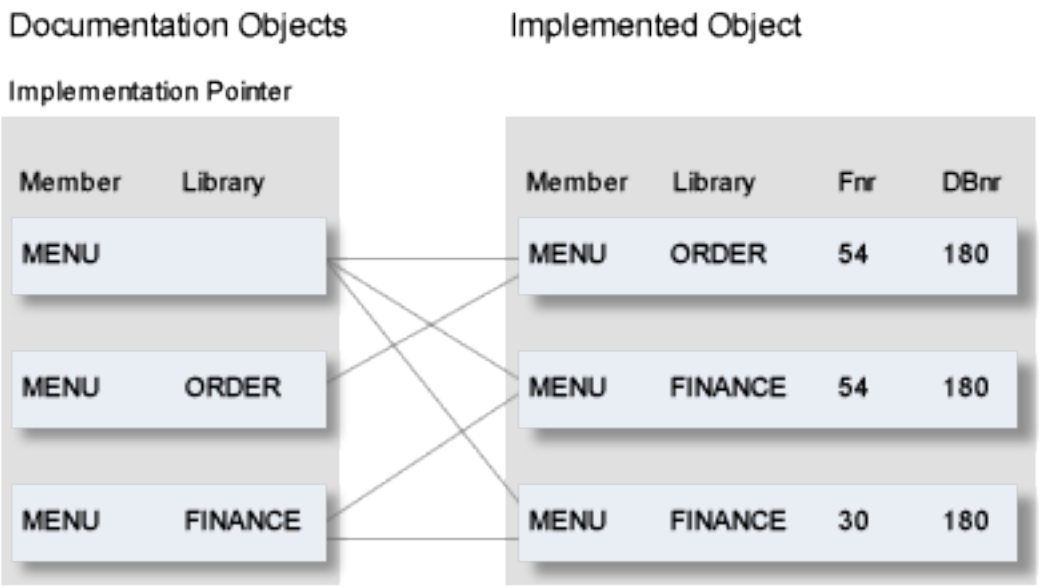
The table below lists the program types permitted for a program written in a particular language and indicates whether the program can have a list of entry points. In third generation languages, marked * in the table below, functions and subprograms can be documented as programs of type F and N respectively, but any active references for these programs will have type P (main program). The active references of these programs will be correctly connected in the active retrieval functions to programs of types P, N and F.

Language		Permitted Program Types	Entry Points allowed?
B	BAL (Assembler)*	C D F I N P U	yes
C	COBOL*	C D F I N P U	yes
E	Natural EL	D Y	no
F	FORTRAN*	C D F I N P	yes
G	ADA*	C D F N P	yes
H	C*	C D F I N P U	yes
J	Job Control Language	D J	no
N	Natural	A C D G H I K L M N O P S T X 1 4 5	no
O	Other	C D F H M N P	yes
R	Rexx	R	yes
P	PL/I*	C D F I N P U	yes
Q	Static SQL	D I P	yes
S	SQL procedure language	R U	no
V	Java	R	yes
Z	System program	D E	yes
0 - 9	user-defined	C D F N P	no

New languages (code 0 - 9) are defined with the program U-PGMLAN. See the section *U-PGMLAN - Define New Program Language* in the section *User Exits* in the *Predict Administration* documentation.

Combinations of Parameters for Natural Programs

If the same member is used in several libraries, multiple documentation of this member can be avoided by omitting parts of the implementation pointer. Predict then finds out for itself all the libraries in which this member exists. In the example below, the library name is omitted.



The valid combinations of implementation pointer parameters permitted for Natural programs are shown below.

Member	Y	Y	Y	Y
Library		Y	Y	Y
Fnr			Y	Y
DBnr				Y

SQL Procedure Code Tab

This function can only be executed for programs of type SQL procedure or Database function with language SQL procedure.

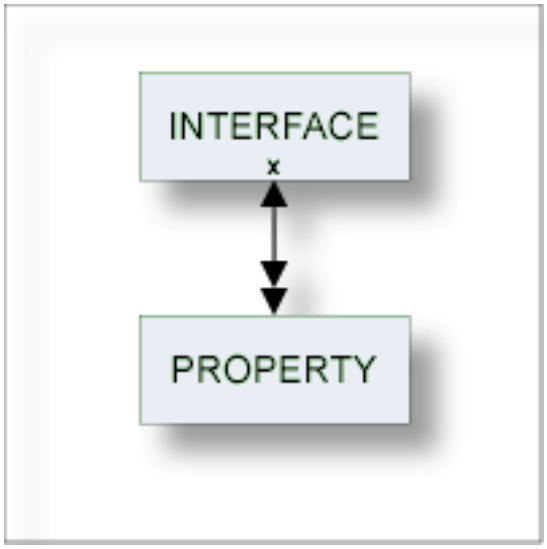
XI

- 43 Property 323
- 44 Report Listing 327
- 45 Server 331

43 Property

- Add a Property 325

This object type is used to document the properties of an interface.



For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Add a Property

Note: Parameters not listed here are described under *Global Attributes*.

Parameters		
Property name	Name of the property.	
Readonly	Y	Variables cannot be modified.

44 Report Listing

- Report Listing ID 328
- Modify Report Listing 330

Objects of type Report Listing log

- a transfer operation of the Predict Coordinator, or
- a conversion operation.

Report listings are added automatically with an ID assigned by the system. For this reason, the functions Add and Copy are not available for this object type.

When transferring data with the Predict coordinator, the extract containing the objects to be transferred is automatically linked as a child to the report listing.

See the *Predict Coordinator* documentation for more information.

Report Listing ID

The report listing ID is assigned automatically when an object is added and is composed as follows:

- **USR - User ID**
  The ID of the user who performed the coordinator function.

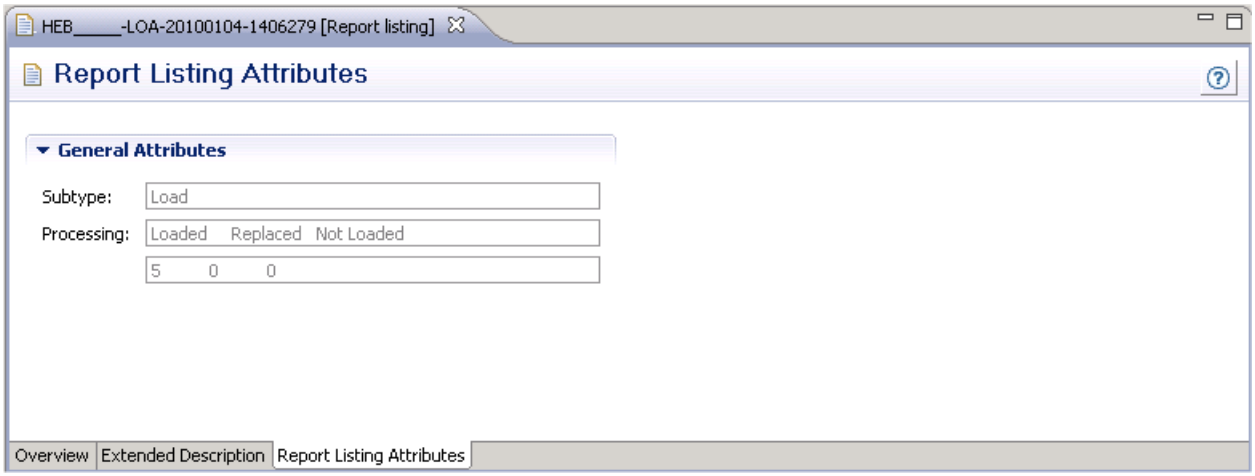
  In batch mode: the job name. This section is appended by underscore characters if less than 8 characters.
- **TYP - Subtype**
  One of the following:
  - EXP Export
  - IMP Import
  - CON Conversion

- TRC Trace
- UNL Unload
- LOA Load
- ALF* - ALF to Migrate conversion
- MIG* - Migrate to ALF conversion

Note: * Report listings of type ALF and MIG are no longer created with this version but objects of this type may exist from earlier versions.

- **200940803 - Date**
  Date on which the report listing was added. Format YYYYMMDD
- **1522453 - Time**
  The time at which the report listing was added. Format HHMMSST

Modify Report Listing



Note: Parameters not listed here are described under *Global Attributes*.

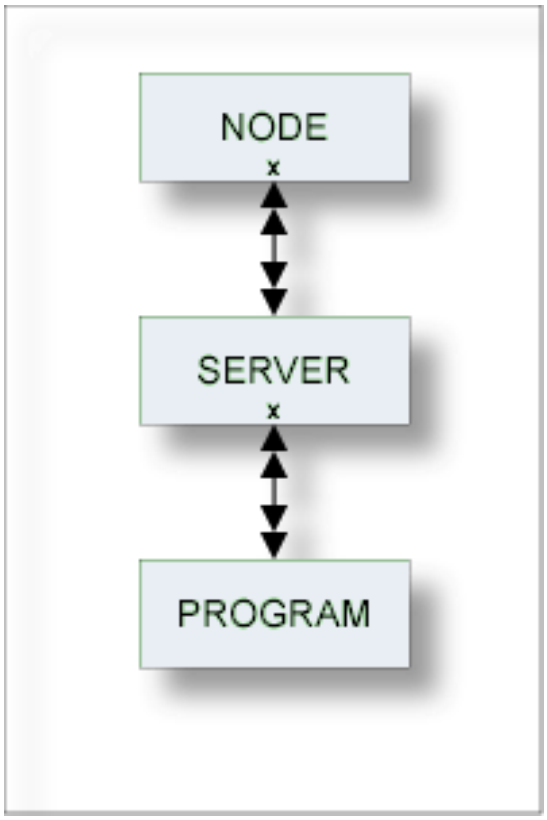
Explanation	
Subtype	Subtype of report listing. ■ Conversion (ALF to Migrate or Migrate to ALF conversion) ■ Export ■ Import ■ Trace ■ Load ■ Unload
Processing	
Exported / Not Exported	For the function Export: The number of objects successfully exported / objects not exported due to errors. **Note:** See the extended description of the report listing for a complete list of these objects.
Loaded / Replaced / Not Loaded	For the function Import: The number of new objects successfully loaded / existing objects overwritten / objects not loaded due to errors. **Note:** See the extended description of the report listing for a complete list of these objects.

45 Server

- Add a Server 333

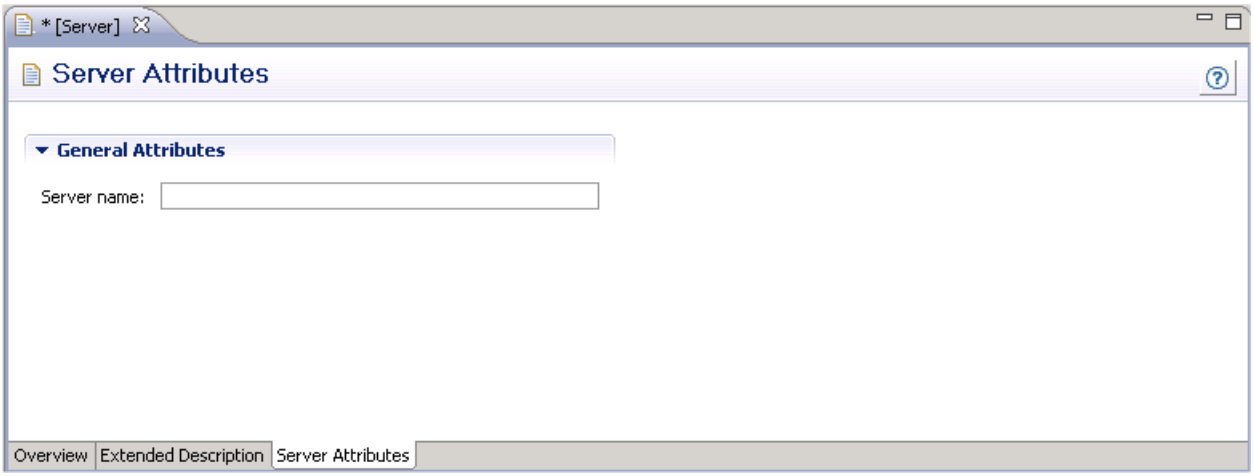
This object type, together with object type Node, is used to document remote procedure calls.

An object of type Server documents all programs available on a logical server.



For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Add a Server

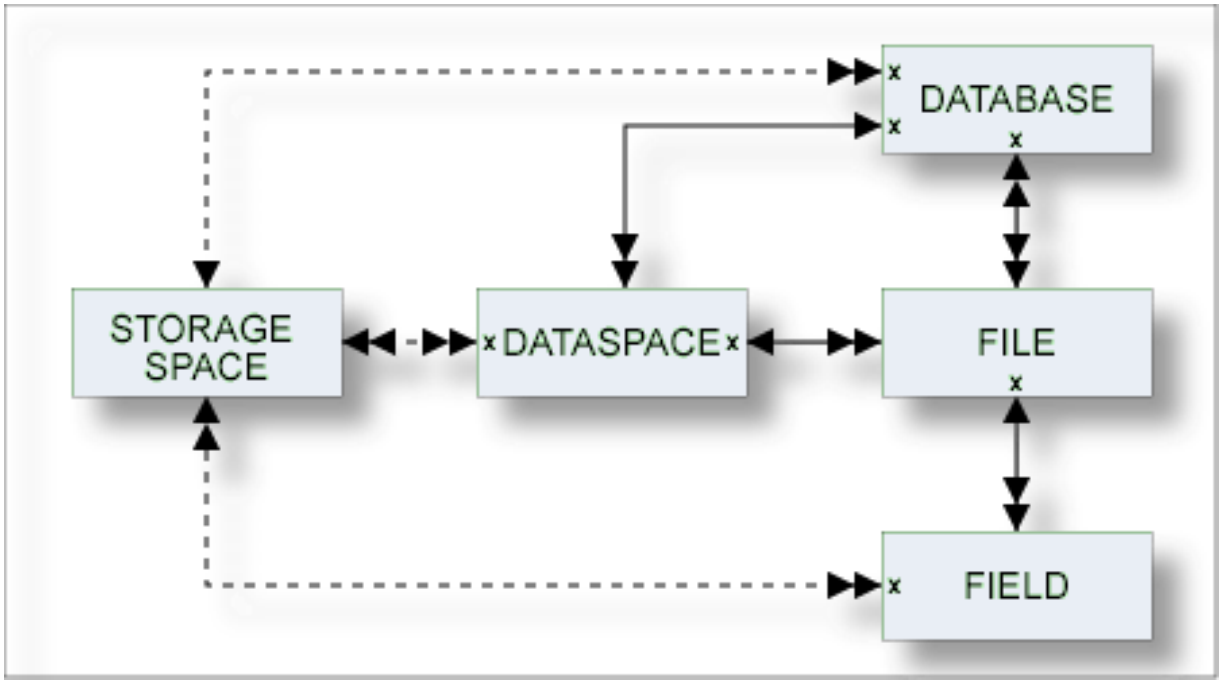


Note: Parameters not listed here are described under *Global Attributes*.

Parameters	
Server name	Name of the server must be specified. Up to 8 characters.

XII Storagespace

DB2 storagegroups are documented in Predict with the object type Storagespace. See the section *DB2 and SQL/DS* in the *Predict and Other Systems* documentation.



In the predefined Predict metastructure, a storagespace has no predefined association. References to storagespaces are realized with the attribute (Default) Storagespace of objects of type Database, Dataspace and Field.

The description of object type Storagespace is organized under the following headings:

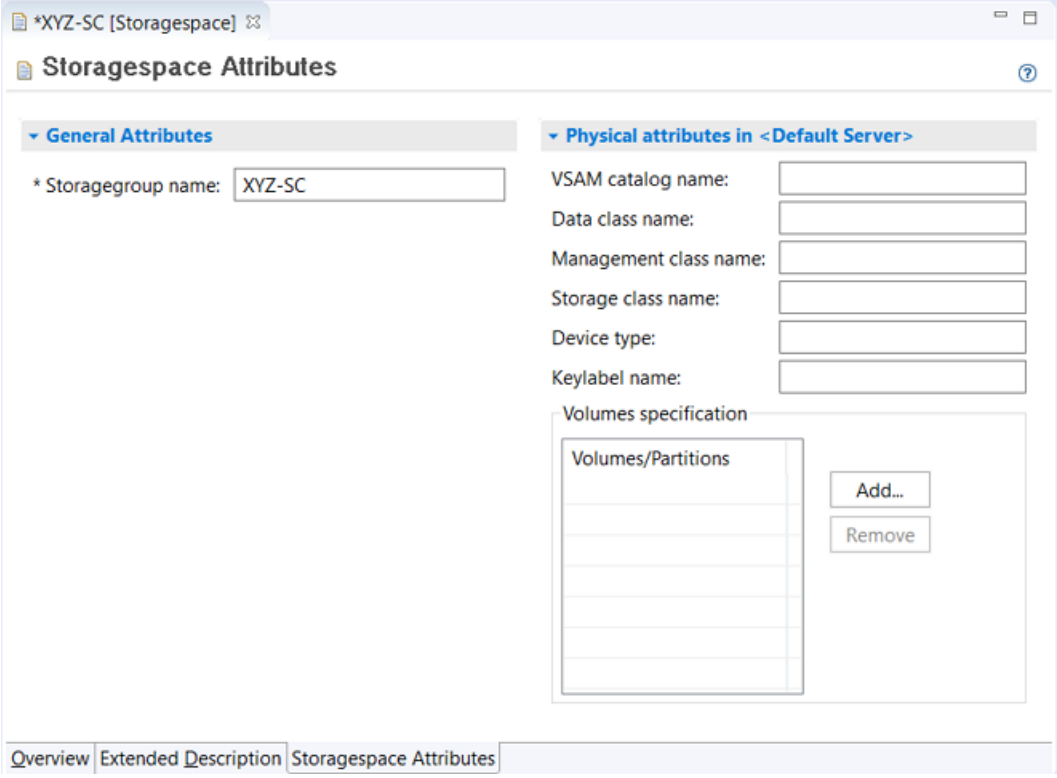
Maintaining Objects of Type Storagespace

46 Maintaining Objects of Type Storagespace

- Add a Storagespace 338
- Storagespace-Specific Maintenance 339

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Add a Storagespace



Note: Parameters not listed here are described under *Global Attributes*.

Parameters	
Storagegroup name	Name of the storagegroup in DB2.
VSAM catalog name	Name or alias of an ICF catalog. Aliases are used for names of ICF catalogs that are longer then eight characters.
Data class name	Identifies the name of the SMS data class to associate with the DB2 storage group. The SMS data class name must be from 1-8 characters in length. The SMS storage administrator defines the data class that can be used. A data class must not be specified more than one time.
Management class name	Identifies the name of the SMS management class to associate with the DB2 storage group. The SMS management class name must be from 1-8 characters in length. The SMS storage administrator defines the management class that can be used. A management class must not be specified more than one time.

Parameters	
Keylabel name	Used for encryption.
Storage class name	Identifies the name of the SMS storage class to associate with the DB2 storage group. The SMS storage class name must be from 1-8 characters in length. The SMS storage administrator defines the storage class that can be used. A storage class must not be specified more than one time.
Device type	For documentation purposes.
Volumes/Partitions	Physical volume(s)/partition(s) where the storagespace resides.

Storagespace-Specific Maintenance

Purge Storagespace

Predict objects of type Storagespace are purged with the **Delete** command.

The following restriction applies to this function:

A storagespace cannot be deleted if it is still referenced by a database, a dataspace or a file.

Otherwise this function behaves as described in the section *Maintenance* in the *Predict Reference* documentation.

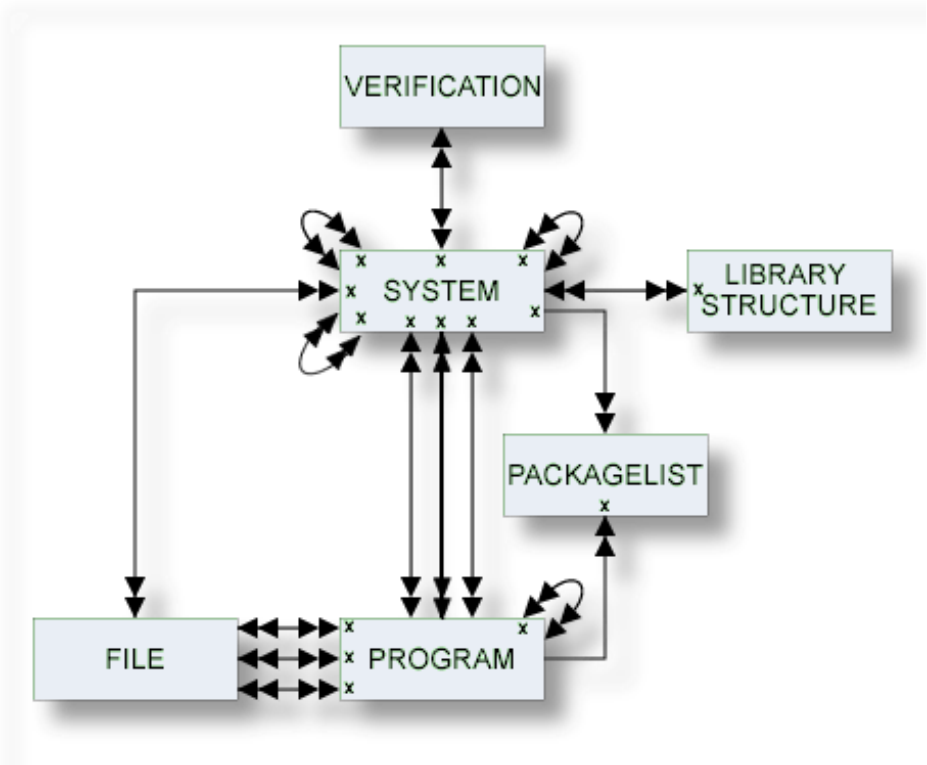
XIII

- 47 System 343
- 48 Trigger 349

47 System

- System Types 344
- Add/Modify System 345
- System-Specific Maintenance 347

An application can be documented with a Predict object of type System. See *System Types* for a list of possible system types.



For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

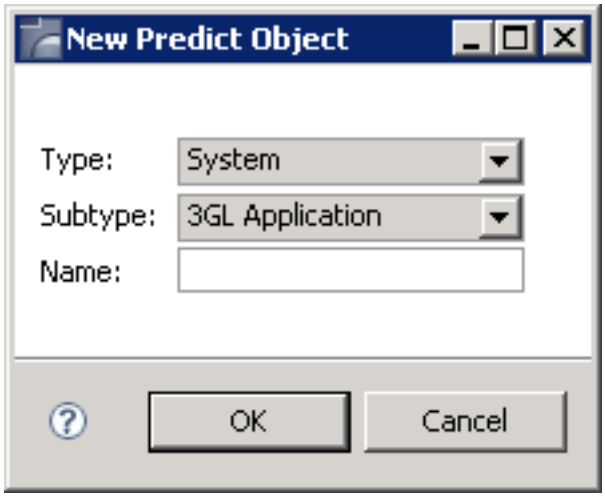
System Types

The table below contains a list of all valid system types.

Code	System Type
A	Application Library
B	Base Application
C	Conceptual. Used to outline the preliminary description of an application in the design phase.
G	3GL Application
O	Compound Application
P	DB2 plan. Used to document a DB2 application.

Add/Modify System

When you add a system, you first have to specify the system type in the **New Predict Object** dialog box.

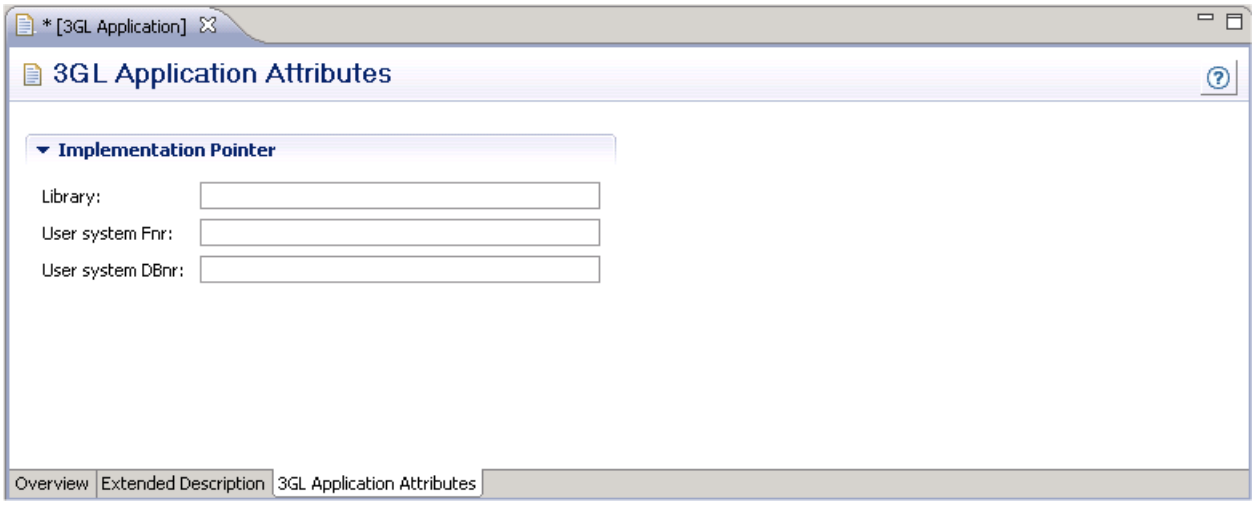


When you choose the **OK** button, a system type-specific window appears. The system type is indicated in the title bar. Different or sections with different names are used, depending on the system type:

- **Implementation Pointer** section (3GL Application and Application Library)
- **DB2 Plan Attributes** tab (DB2 Plan)
- **Base Application Attributes** tab (Base Application)

For system types Compound Application and Conceptual only the tabs for the global attributes are provided.

The following is an example of the **Implementation Pointer** section.



Note: Parameters not listed here are described under *Global Attributes*.

Parameters		
Implementation pointer	This information is shown in the **Implementation Pointer**section (types 3GL Application and Application Library).	
	Library	The name of the library. For type G: The library cannot be changed if XRef data exists (the library is used by a 3GL program).
	User system Fnr	The file number of the user system file (FUSER).
	User system DBnr	The database number of the user system file.
DB2 plan name	This information is shown on the **DB2 Plan Attributes** tab (type DB2 Plan). Unique DB2 plan name.	
Profile	This information is shown on the **Base Application Attributes** tab (type Base Application).	
	Name	The name of the profile.
	Fnr	The number of the user system file.
	DBnr	The number of the database in which the user system file is located.
Port	This information is shown on the **Base Application Attributes** tab (type Base Application). The port number.	
Server name	This information is shown on the **Base Application Attributes** tab (type Base Application). The name of the server.	
Development platform	This information is shown on the **Base Application Attributes** tab (type Base Application). The development platform. Enter one of the following values: ■ MAINFRAME ■ Linux	

Parameters	
	■ PC This parameter specifies for which type of server the application is developed.

System-Specific Maintenance

Identifying Systems

Systems documented with Predict objects of type System can be identified with three parameters: library, file number and database number. The three possible combinations of these parameters are shown below.

Library	Y	Y	Y
File number		Y	Y
Database number			Y

Purge System

Predict objects of type System are purged with the **Delete** command.

The following rules apply to this function:

- A system of type A (Application Library) cannot be deleted if it is linked to one or more systems via association "Has library SY".
- A system of type G (3GL application) cannot be deleted if XRef data exist.
- The following objects are deleted:
  - the system object
  - all links to child objects
  - all links from parent objects

Rename System

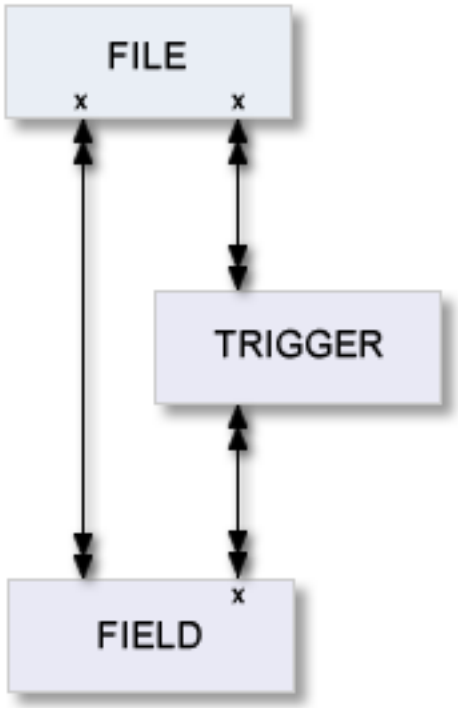
Use this function to change the ID and/or type of a system object. The following restriction applies:

- You cannot change the type of a system of type 3GL application for which XRef data exists.
- You cannot change the type of a system of type A (Application Library) if it is linked to one or more systems via association "Has library SY".

48 Trigger

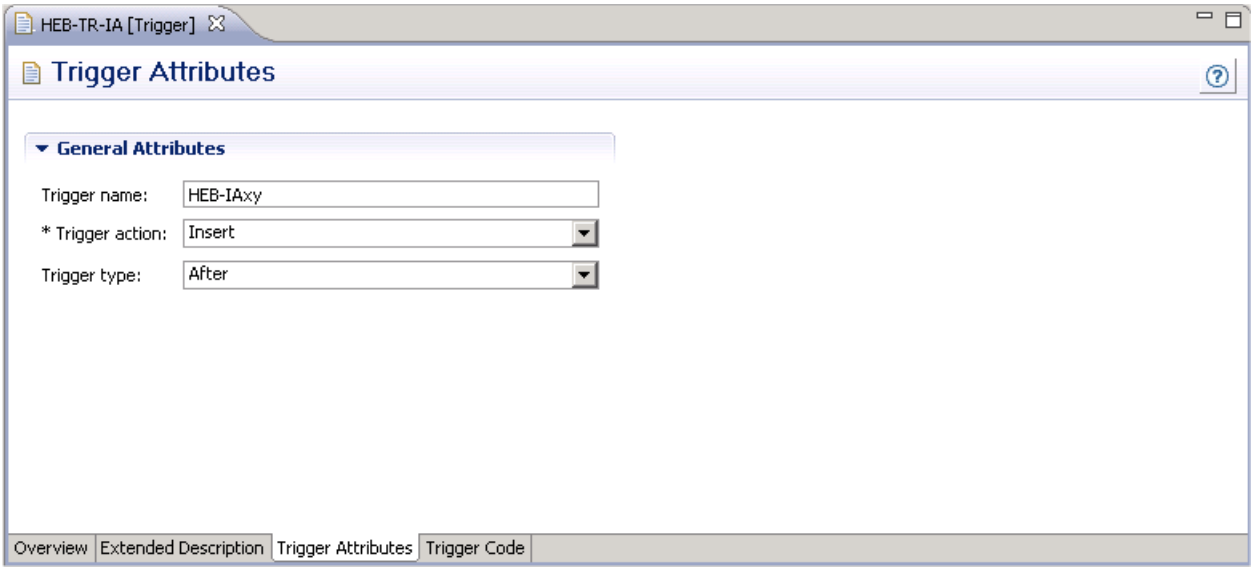
- Add a Trigger 351
- Editing the Trigger Code of a Trigger 352

This object type is used to define triggers for SQL tables and SQL table fields.



For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Add a Trigger

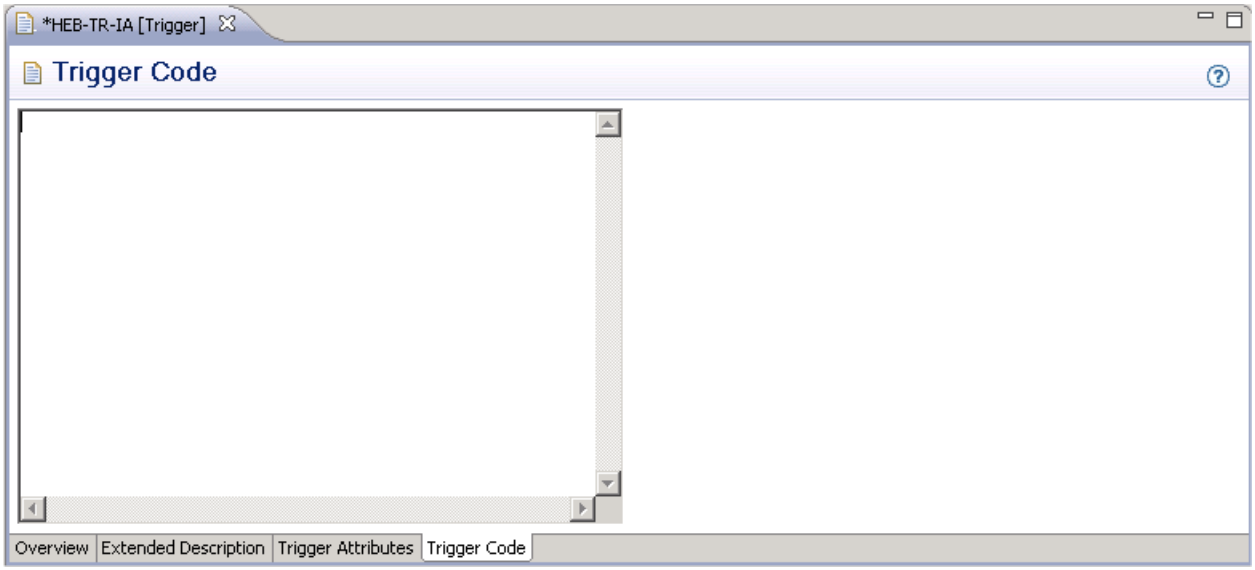


Note: Parameters not listed here are described under *Global Attributes*.

Parameters	
Trigger name	Name of the trigger.
Trigger action	Activating a trigger with the statement: ■ Insert ■ Update ■ Delete
Trigger type	Activation time of a trigger: ■ After ■ Before ■ None

Editing the Trigger Code of a Trigger

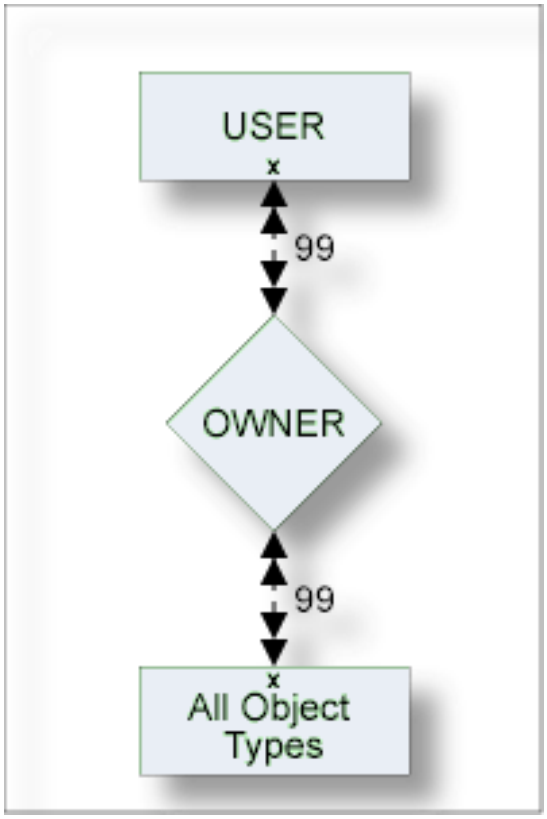
The trigger code can be edited on the **Trigger Code** tab.



XIV User

The object type User contains information on users and organizational units, such as name, ID or position within the company.

One attribute of this object type is Owner. Groups of users reflecting organizational units, such as project teams, can be formed by assigning individual users to an owner. Each user can belong to several owners. Owners can be associated to other types of Predict objects. See also *User/Owner* and *Keyword* in the section *Overview of Predict* in the *Introduction to Predict* documentation.



The description of object type User is organized under the following headings:

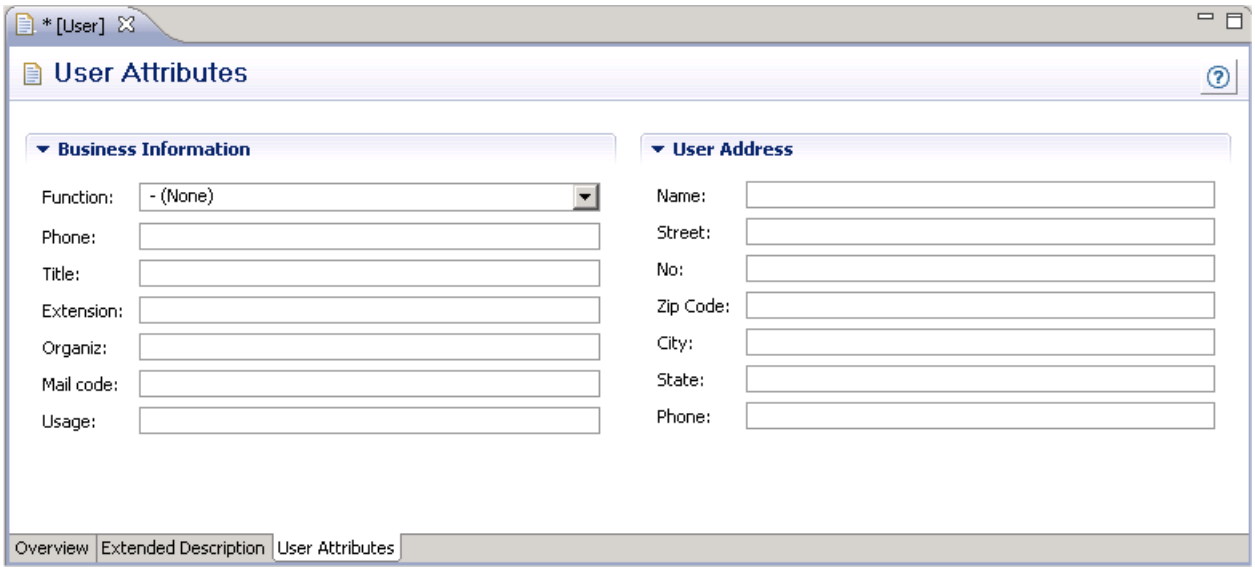
Maintaining Objects of Type User

49 Maintaining Objects of Type User

- Add/Modify a User 356
- User Maintenance 357

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Add/Modify a User



Note: Parameters not listed here are described under *Global Attributes*.

Parameters	
Name	The name of the user is specified in the field `Name` on the **User Address** tab.
Business Information	Various attributes describing the user's position within the organization, telephone number and access privileges (parameter Usage with values ACCess or UPDate). The attributes are used for documentation purposes only.
User Address	Various address data for the user.

User Maintenance

Purge User

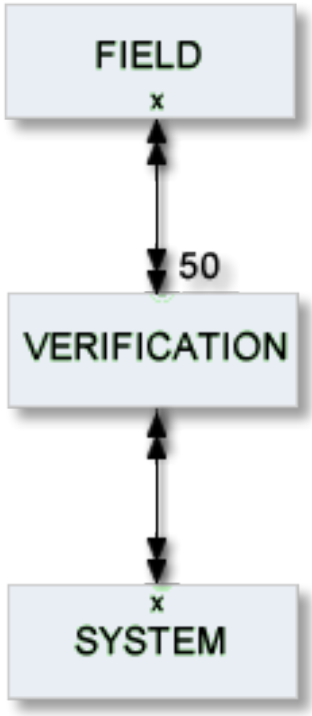
Predict objects of type User are purged with the **Delete** command.

The following rules apply:

- The following objects are deleted:
  - the user
  - all links to child objects
  - all links from parent objects
  - all sets created by this user
  - the workplan of the user
  - the Predict and LIST XREF profiles of the user
  - the filter definitions of the user
- A user will not be deleted if
  - he is the only user in the user list of an owner and
  - this owner is assigned to an object where the option OWNER=FORCE has been defined in the metadata administration for this object type.

XV Verification

Objects of type Verification can contain code for processing rules. Verifications can have as status: documented, conceptual, free, automatic, Natural Construct or SQL.



The description of object type Verification is organized under the following headings:

Maintaining Objects of Type Verification
Verification-Specific Maintenance

Additional Information on Verifications/Processing Rules

- See the section *Verifications and Processing Rules* in the *Predict and Other Systems* documentation.
- See also *Rippling Verifications* in the *Predict and Other Systems* documentation.

50 Maintaining Objects of Type Verification

- Verification Status 362
- Verification Formats 362
- Add a Verification 363

For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Verification Status

The table below contains a list of all valid verification status.

Code	Verification Status
A	Automatic
C	Conceptual
D	Documented (no rule)
F	Free
N	Natural Construct
S	SQL

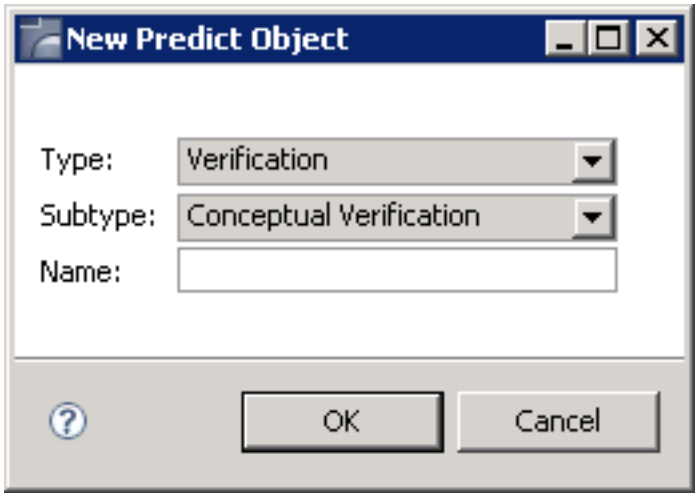
Verification Formats

The table below contains a list of all valid verification formats.

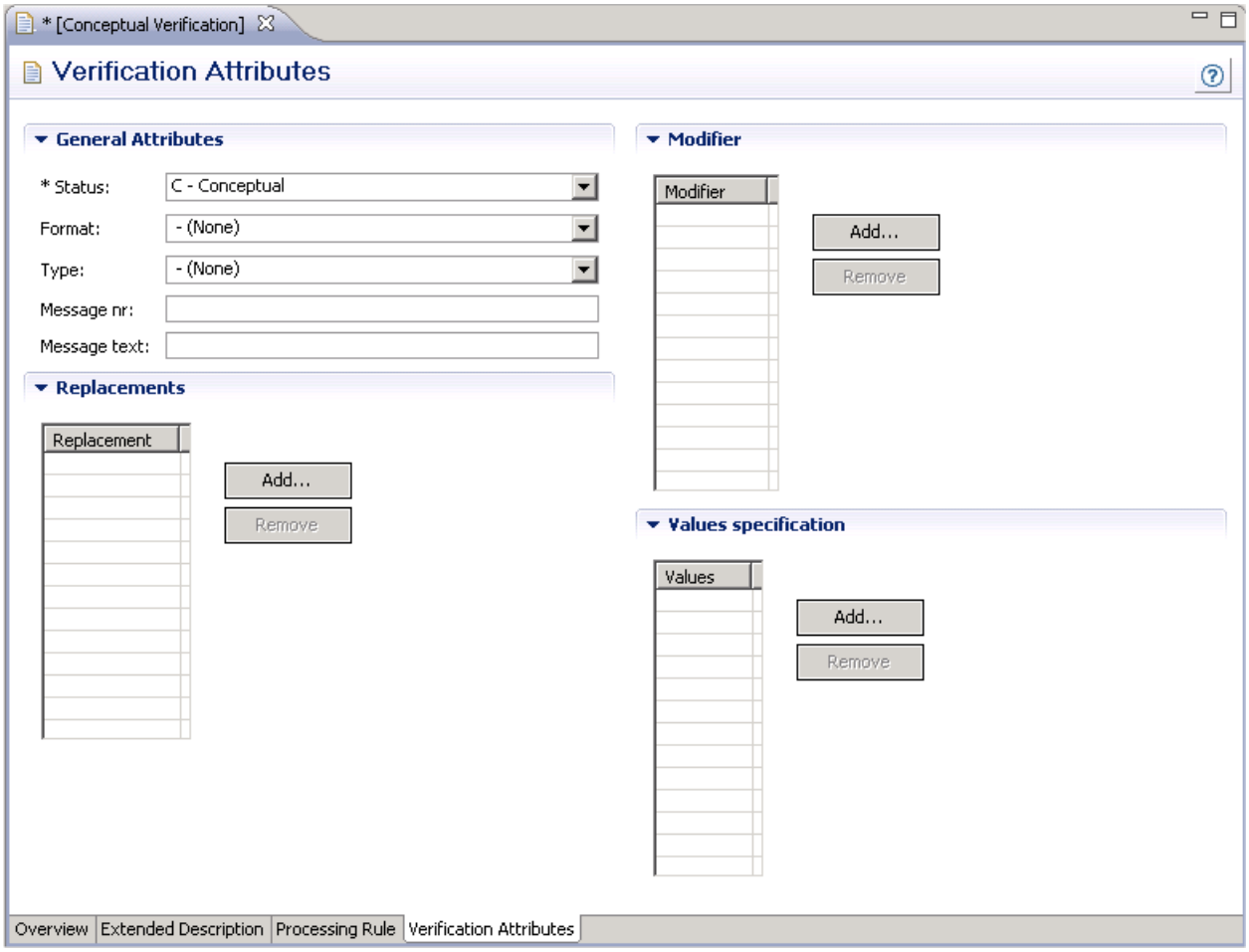
Code	Format
A	Alphanumeric
B	Binary
D	Date/time
K	Function key
L	Logical
N	Numeric
blank	Unknown (no rule defined)

Add a Verification

When you add a verification, you first have to specify the verification status in the **Type** dialog box.



When you choose the **OK** button, a verification status-specific window appears. The verification status is indicated in the title bar.



Note: Parameters not listed here are described under *Global Attributes*.

Parameters	
Status	The status assigned by Predict to the verification rule. See *Verification Status* for list of valid values.
Format	The format of the verification rule. See *Verification Formats* for list of valid values.
Modifier	User and or user groups defined in Natural Security who can be authorized to modify free rules of the verification. This information is specified in the **Modifier** section. The parameter is evaluated by Predict according to the setting of the default parameters Rule in Map Editor / Rule in SYSDIC. If any of these parameters is set to force, Predict checks the following: ■ that at least one modifier is specified, ■ that each modifier of the object is a Natural Security administrator, person or group, ■ that the user is listed as a modifier of the object.

Parameters	
	See also description of Rule in Map Editor / Rule in SYSDIC in the section *Defaults* in the *Predict Administration* documentation and *Protecting Processing Rules* in the section *Protecting External Objects in Predict with Natural Security* in the *Predict Security* documentation.
Type	The type of rule. The table also shows the number of values to be specified with each type of rule:

Code	Type of Rule	No. of Values
E	Equal to	1
		0 or 1 for format logical
G	Greater than	1
L	Less than	1
N	Not equal to	n
		0 or 1 for format logical
R	Range of values	2
T	Table of values	n
U	User routine	
B	Range, but not	3 or 4
I	Not in range	2
M	Mask	n
O	Not Equal Mask	n
S	Scan	n
V	Not Equal Scan	n
blank	(none) - no rule defined	

For a list of the generated code, see *Rule Editor* in the *Predict Reference* documentation. .

Message nr	Number of Natural error message. The message will be displayed if a validation fails. Up to three replacement strings can be inserted into an error message if the respective targets (:1:, :2:, :3:) are provided.
Replacement	Strings to be inserted into a Natural message. See description of Message nr above.
Message text	Message to be displayed if a validation fails. A standard message will be created if neither Message text nor Message nr have been specified.

Values	The values used to perform the verification. The following rules apply:
	■ The number of values to be specified depends on the verification type. See table above.
	■ Values are delimited
	  ■ with blanks
	  ■ with the Natural INPUT delimiter character (ID) defined in the Natural environment
	  ■ by entering them in separate lines.
	■ Hexadecimal values can be specified in two ways:
	  ■ if Format=B, hexadecimal values can be specified directly. Example: F0

Parameters	
	■ if Format=A, hexadecimal values must be preceded by uppercase X or H and be enclosed in single quotes. Example: X'F0' or H'F0' ■ Blanks can be specified in one of the following ways: ' ', BLANK or SPACE. Strings that include blanks must be enclosed in single quotes, apostrophes in strings have to be doubled (for example: 'six o''clock'). ■ Line comments can be specified by preceding them with /* (a slash and an asterisk). Line comments can be used by SYSHELP as descriptive text in input windows. Strings that include the comment delimiter /* must be enclosed in single quotes.

51 Verification-Specific Maintenance

- Purge Verification 368

Purge Verification

Predict objects of type Verification are purged with the **Delete** command.

A verification of type automatic cannot be purged. To purge a verification of this type, perform the following steps:

- Remove all links from fields to the verification
- Regenerate DDMs that were generated from the files linked to these fields.

When the verification is no longer connected to any fields, the status is changed to conceptual and the rule can be purged.

XVI Virtual Machine

52 Virtual Machine

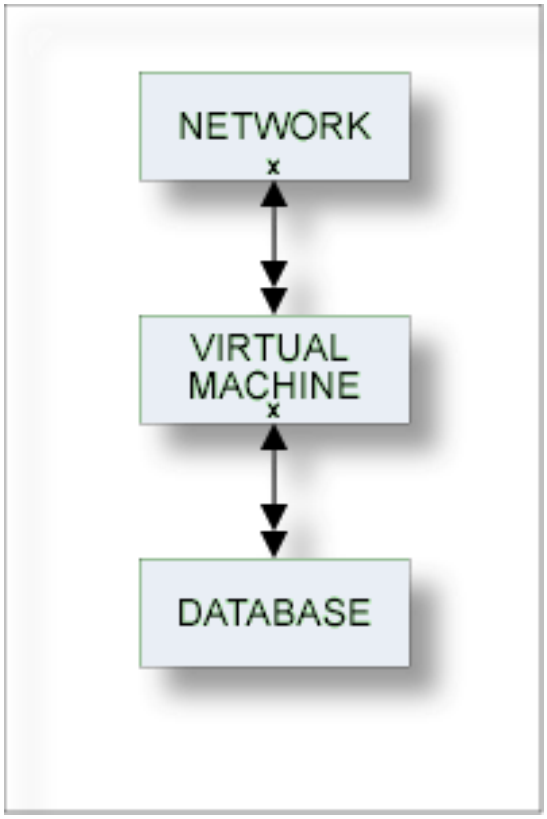
- Add a Virtual Machine 373

Since data can be distributed across several databases, the exact location of data storage has to be specified: databases are linked to objects of type Virtual Machine and virtual machines are linked to objects of type Network.

The Predict object virtual machine identifies the hardware and operating system environment of a database.

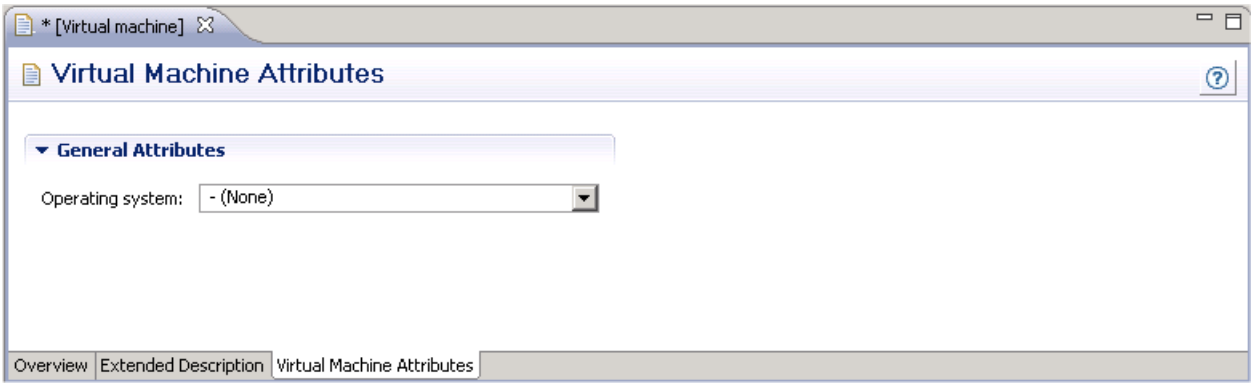
See the section *Adabas Vista* in the *Predict and Other Systems* documentation for a complete description of how to define distributed data structures with Predict.

Note: Links between networks, virtual machines and databases are established with the parameters "Belongs to NW" and "Belongs to VM", and not with active/passive associations.



For general information on how to manage objects (for example, how to add or copy an object), see the *Predict Description and Generation* documentation.

Add a Virtual Machine



Note: Parameters not listed here are described under *Global Attributes*.

Parameters	
Belongs to NW	This attribute can be found in the **Mandatory Association** section of the **Overview** tab. The ID of the network containing the virtual machine.
Operating system	Select an operating system from the drop-down list box.