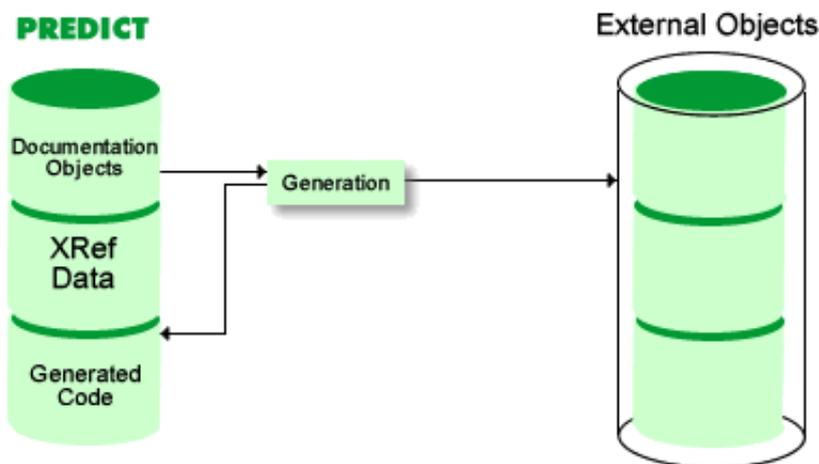


Generation

External objects can be generated from Predict objects.



There are basically two types of external objects:

- external objects owned by Predict (usually stored in the FDIC file)
- external objects **not** owned by Predict (usually stored in the external environment).

Note:

Special rules apply to DDMs and Natural processing rules. See the respective sections of this section for more information.

This section covers the following topics:

- Concepts of Generation
- Calling Generation Functions
- Generation Defaults
- Storage of External Objects Owned by Predict
- Common Parameters

Concepts of Generation

Data definitions generated from Predict objects are called **external objects**.

Two types of external objects can be generated:

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

External Objects Owned by Predict

- 3GL copy/include code (C, COBOL, Assembler, FORTRAN, PL/I)
- Adabas invert, compression and security definitions (ADAINV/ADAWAN/ADAFDU, ADACMP, ADASCR)
- SQL CREATE statements
- Adabas/VSAM transparency table

External objects owned by Predict are used at compile time and have to be copied (punched) to an operating system library before they can be used in the implementation of an application. Copying can be performed by Entire System Server or the Preprocessor.

A variety of parameters determine how these types of external objects are stored. See Parameters Specifying the Form of Output.

External Objects not Owned by Predict

- Databases (Adabas , DB2, IMS/DL1)
- Vista translation table
- DB2 tablespaces and storagegroups
- Files, tables and views (Adabas , DB2)
- IMS User-Defined Fields (UDFs)
- Natural DDMs (including Natural security definitions and/or Super Natural files).

External objects not owned by Predict are used at run time. Some of these objects are stored directly in the application environment, others are stored in the Predict system file. For objects that are stored in the Predict system file, Predict data must be accessible when an application using the external objects is running.

Connecting External Objects and Documentation Objects

An external object that has been generated from a documentation object is regarded as connected to the documentation object. The connection is established by adding an implementation pointer to the documentation object. External objects and documentation objects that are connected are - to a certain extent - protected from being modified independently from each other.

See also section Handling of External and documentation Objects in this documentation.

Overview of Generation Functions

The following table is sorted by the code used to call a generation function from the Predict main menu.

Object	Code	Command	Description of External Object	Predict owned
Adabas compression definition	AC	WAN, CMP	For use as input to the Adabas compression utility ADACMP, ADAWAN or ADAFDU. If generating for ADACMP, you can generate additional input for ADALOD.	Y
Adabas File	AF	FDT	Adabas files are loaded directly into an Adabas database. If the file is already loaded, the differences of the implemented file and the documented file are determined and update commands are generated in order to transform the implemented Adabas file according to the documented file.	
Adabas invert definitions	AI	ADAINV	For use as input to the Adabas invert utility ADAINV.	Y
Adabas security definitions	AS	SCR	For use as input to the Adabas security utility ADASCR.	Y
Vista table	AT	VISTATAB	This function implements Vista elements of Predict file and database objects of type A in the translation tables of Vista.	

Transparency table for Adabas VSAM bridge	AV	AVB, ADAVSAM	Supports conversion from VSAM files with KSDS organization to Adabas files without the need to change existing COBOL programs.	Y
Assembler Copy Code	BA	BAL, ASSEMBLER	Copy code for use in a 370 Assembler program.	Y
C Include Code	CC	LANG-C	Include code for use in a C program.	Y
COBOL Copy Code	CO	COBOL	Copy code for use in a COBOL program.	Y
SQL CREATE statement	CR	SQL-CREATE	CREATE TABLE, CREATE VIEW or CREATE CLUSTER statement. These SQL statements are stored as Natural members.	Y
Data Definition Module	DD	DDM	A collection of field definitions used by Natural for accessing a database.	
DB2 database	D2	DB2-DATABASE	A DB2 database is implemented directly as a physical DB2 database. Not applicable to SQL/DS.	
Adabas table/view	EQ	ESQ	Table descriptions or views in an Adabas SQL Server catalog.	
FORTRAN Copy Code	FO	FORTRAN	Copy code for use in a FORTRAN program.	Y
PL/I Include Code	PL	PLI	Include code for use in a PL/I program.	Y
Verification rule	RU	RULE	A rule must already have been generated using the Generate DDM function. Only the code of the rule is changed. The new rule will automatically be used by Natural maps that are cataloged. Not applicable to rules of status SQL or Natural Construct.	
DB2-Storagegroup	SG	STORAGEGROUP	Storagegroups are implemented directly from Predict storagespaces. If a storagegroup has already been implemented from the storagespace, the differences of the implemented DB2 storagegroup and the documented storagespace are determined and update commands are generated to adapt the implementation to the documentation.	
DB2 table/view	T2	TABLE	DB tables/views are implemented directly from Predict file objects of type D or E. If a table/view has already been implemented from the Predict table/view definition, the differences of the implemented DB2 table/view and the Predict table/view definition are determined and update commands are generated in order to change the implementation according to the documentation.	

DB2 tablespace, SQL/DS DBspace	TS	TABLESPACE	Tablespaces/dbspaces are implemented directly from Predict dataspace. If a tablespace/dbspace has already been implemented from the dataspace, the differences of the implemented DB2 tablespace/SQL/DS DBspace and the documented dataspace are determined and update commands are generated to transform the implementation according to the documentation.	
User-defined fields for IMS	UD	UDF	Definitions used by Natural to access an IMS database.	

Calling Generation Functions

Generation functions are called with G in the field Function and the appropriate code in the field Object Type in a Predict main menu or with a command. The following rules apply:

- All generation functions can be executed both online and in batch mode.
- If a system is not installed, the respective generation functions are not available. For example, if DB2 is not installed at your site, no generation functions for DB2 objects are available.

Calling Generation Functions with Commands

- Parameters are entered on the line following the command.
- Parameters can be entered in positional and/or keyword form. See the section Predict Commands in the **Predict Reference documentation**.
- To use the default parameters defined for the function, enter the command and the <object-ID> on the same line. This does not apply in batch mode.
- Some generation functions can be called with two or three commands. It makes no difference which command is used.

Generation Condition Codes

Condition codes are described in the section Predict in Batch Mode in the **Predict Administration documentation**.

Generation Defaults

Default generation values are set at installation. Most default values are displayed in the input screen of the respective generation function and can then be overwritten for temporary use. Changes to default values apply to subsequent generation runs until another Predict function is executed. This does not apply in batch mode.

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

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

```

13:35:47          ***** P R E D I C T 4.2.2 *****          2002-07-31
                    - Modify COBOL Defaults -
                                     Modified 2002-07-31 at 13:31
                                     by CHD

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

X Save as member .....
X Overwrite option ..... Y (Y,N)
X Punch / output .....* N
X List generated code ..... Y (Y,N)
X Generate format buffer ..* N
X Check field names .....* Y
X Start level ..... 1 (0-40)
X Level number increment.... 1 (1-40)
X Level shift increment .... 3 (0-9)
X Nr. of abstract lines .... 5 (0-16)
X Generate initial value ..* N
X Synchronized .....* Y
X Depending on ..... N (Y,N)
X Record buffer name .....
X Format buffer name .....

X Save in library .... COBLIB
X Op. system member ..
X List offsets .....* N
X ADABAS version ....* I7
X Field name prefix ..
X Field name suffix ..
X Validate ..... -
X Truncation .....* R
X With Cond. names ... Y (Y,N)
X Indexed by .....* N
X Literal delimiter .* S
X Decimal character .* P
X Redefinition name .* S

Compiler .....* 7
Preprocessor force ..... N (Y,N) Library system ..... 2

```

- Generation defaults can be protected by blanking out the X preceding the parameter in Modify ... Defaults screens. Protected default values cannot be changed when executing a generation function. The input fields of protected parameters are locked in the input screen of the generation function. These fields are skipped when positioning the cursor with the TAB key.
- Some default values are not displayed in the input screen of a generation function and can therefore only be changed using the Modify Generation Defaults function. These parameters are described under **Presetting** in the descriptions of individual generation functions later in this section.
- For Predict-owned external object types you can specify default values for storing generated code. See Specifying Entire System Server and VSE Librarian Options.

For a description of the parameters shown in the previous screen, see the relevant generation functions later in this documentation, the parameter Preprocessor force is described below.

Preprocessor Force

This parameter is used by the Predict Preprocessor and Adabas Native SQL. It can be specified for Assembler, COBOL, FORTRAN, PL/I, and ADA. It can only be defined in the respective Modify Generation Defaults screen.

Valid values:

Y

Both the Predict Preprocessor and Adabas Native SQL check that the program to be processed is documented. If no Predict object documenting the program is found, the task is not executed and a message is returned.

N

No check is performed. Default setting when Predict is installed.

Storage of External Objects Owned by Predict

There are different options to output/store external objects owned by Predict. The following types of external objects are owned by Predict:

- 3GL copy/include code (C, COBOL, Assembler, FORTRAN, PL/I)
- Adabas invert, compression and security definitions (ADAINV, ADAWAN/ADAFDU/ADACMP, ADASCR)
- Adabas/VSAM transparency table
- SQL CREATE statements.

Output / Storage Options for External Objects

Option 1 - Generating External Objects Temporarily

This option is used to have a first look at the results of a generation function. To generate objects temporarily the parameter Save as member is left blank. The parameter List generated code determines what is displayed.

Option 2 - Storing Generated External Objects as Members in a Natural Library of the Predict System File

If the parameter Member name is specified, the generated external objects are stored in a Natural library in the Predict system file (FDIC). The library must be specified.

For each file you can save up to 30 members per language.

For DB2 objects (these are not owned by Predict) a generation protocol can be created and stored in Predict by specifying the parameters Protocol saved in member and Protocol saved in library.

Members in a Natural library that have been created with generation functions can be read by the Preprocessor or written to workfile 1 using the command PUNCH or WRITE. See description of PUNCH/WRITE command in the section Predict Commands in the **Predict Reference documentation** and section The Preprocessor in this documentation.

Option 3 - Storing Generated External Objects Additionally in a Workfile or - with Entire System Server - as an Operating System Member

External objects can additionally be stored by Entire System Server as operating system members. The following rules apply:

- Storage by Entire System Server can only be executed in addition to storage in the Predict system file (Save as member parameter must be specified).
- Storage of generated external objects as operating system members is only possible in an OS/390 or VSE environment.
- If Library system is set to 3, the external object is stored additionally in a VSE Librarian library.
- One operating system member can be stored for each member in a Natural library that has been generated by a generation function.
- External objects are not processed by Entire System Server if errors occur during the generation.

Note:

See also Storing External Objects with Entire System Server.

Parameters Specifying the Form of Output

Parameters specifying the form of output of external objects owned by Predict are described below. Each parameter applies to different data storage options.

Parameters

<p>Save as member (Option 2)</p>	<p>If this parameter is specified, the generated code is stored as a member with the given name in a Natural library on the Predict system file (FDIC). The first character of the member name must be alphabetic, all others must be alphanumeric. If you enter an asterisk, the name of the new member is determined by Predict as follows:</p> <ul style="list-style-type: none"> ● If a Predict File object is used for the first time to generate a member, the member name is taken from the File ID (truncated if necessary). ● If at least one member has been generated before, a selection window containing all member names from previous generation runs is displayed. Select an existing member name with X or enter X in the line *new* to specify a new member name to be created. <p>In batch mode, a new member name is created in either case. If 30 members already exist, the oldest is overwritten.</p>
<p>Save in library (Option 2)</p>	<p>The generated definitions are saved in the specified Natural library on the Predict system file. The library name must not start with SYS; its first character must be alphabetic and all others must be alphanumeric. This parameter must be specified if the generated external object is to be stored in a Natural library.</p>
<p>Overwrite option (Option 2)</p>	<p>Y An existing member that has been generated from the same file object by Predict is overwritten. Existing members that were not generated by Predict are not overwritten. If the generated external object has additionally been stored as an operating system member, this member is overwritten, too. If an external member was created and the generation function is executed a second time without entering an external member, the external member is deleted if Overwrite option=Y.</p>
<p>Op. system member (Option 3)</p>	<p>If the generated code is to be stored in an operating system member, a member name must be specified with this parameter and the options Punch/Output, Save as member and Save in library must be specified. If an operating system member name is not specified explicitly, a name can be given with one of the following methods:</p> <ul style="list-style-type: none"> ● if an asterisk is entered, the first 8 characters of the file ID are used as the member ID. ● given that an external object has been stored both as a member of a Natural library and as an operating system member in the same generation operation, and that Entire System Server is used, the field Op. system member can be left blank. Predict will insert the name of the existing operating system member into the input field. This name can then be changed.

Punch/Output (Option 3)	<p>Determines further use of the generated code.</p> <p>N The generated external object is neither processed by Entire System Server nor written to a sequential file (workfile).</p> <p>Y The generated external object is punched to a sequential file (workfile 1): If Op. system member (see above) is specified, operating system statements preceding the data definitions are created in the external object depending on the type of operating system specified with the parameter Library system further down this table. Option Y is only valid in online environments where access to workfiles is possible, (for example, BS2000, CMS, TSO), and when running in batch mode.</p> <p>If Library system=3 the external object is additionally stored in a VSE Librarian library. See Specifying Entire System Server and VSE Librarian Options.</p> <p>P Entire System Server stores the generated external object as an operating system member. The parameter Op.system member and Entire System Server options specified in the generation defaults determine where the object is stored.</p> <p>S As above, but where the object is stored can be specified in a subsequent screen. See Specifying Entire System Server and VSE Librarian Options.</p>
List generated code (Option 1)	<p>Determines the amount of information to be displayed.</p> <p>Y The generated code is displayed.</p> <p>N Only the messages and the summary of the messages is displayed.</p>

<p>Library system (Option 3)</p>	<p>This parameter can only be specified in the generation defaults. Determines which type of operating system statements preceding the data definitions in the external object are created.</p> <p>blank Code is generated in IEBUPDTE format.</p> <p>A-Z CATALS and BKEND cards are generated for VSE and the member is added to the sublibrary identified by the specified letter. These cards will also be generated if the Predict command PUNCH is used in batch mode.</p> <p>1 Cards for The Librarian are generated.</p> <p>2 PANVALET cards are generated.</p> <p>3 VSE SP2/SP3 cards for use with LIBRARIAN system are generated. The ACCESS and CATALOG statements with the options Library name, Sublibrary name and Member type are added. If Punch/output is set to Y, the external object is stored additionally in a VSE Librarian library.</p> <p>The following table shows the statements generated for the various operating systems:</p> <table border="1" data-bbox="411 1055 1018 1350"> <thead> <tr> <th>Operating System</th> <th>Statement</th> </tr> </thead> <tbody> <tr> <td>OS</td> <td>IEBUPDTE ADD</td> </tr> <tr> <td>VSE</td> <td>CATALS, BKEND</td> </tr> <tr> <td>OS Librarian</td> <td>ADD</td> </tr> <tr> <td>Panvalet</td> <td>ADD</td> </tr> <tr> <td>DOS Librarian (VSE SP2/SP3)</td> <td>ACCESS,CATALOG</td> </tr> </tbody> </table> <p>Note: This parameter is evaluated only if the Punch/Output option is set to Y and an Op. system member is specified. This parameter is not applicable when generating ADAFDU definitions with function Generate ADACMP/ ADAWAN definitions with the following Adabas versions:</p> <ul style="list-style-type: none"> ● any version starting with U, ● V3 and above or ● P2 and above. <p>A complete table of Adabas versions can be found at the end of this section.</p>	Operating System	Statement	OS	IEBUPDTE ADD	VSE	CATALS, BKEND	OS Librarian	ADD	Panvalet	ADD	DOS Librarian (VSE SP2/SP3)	ACCESS,CATALOG
Operating System	Statement												
OS	IEBUPDTE ADD												
VSE	CATALS, BKEND												
OS Librarian	ADD												
Panvalet	ADD												
DOS Librarian (VSE SP2/SP3)	ACCESS,CATALOG												
<p>Workfile name</p>	<p>Only for Windows or UNIX platforms. Identifies the file for punch output. If punch is set to Y, default is taken from Natural Parameter Module -> Workfile name 1.</p>												

Storing External Objects with Entire System Server

External Objects owned by Predict can be stored directly as a member in an operating system library (partitioned data set) in an OS/390 and a VSE environment if Entire System Server is available.

This option is used by setting the parameter Punch/Output to P or S.

If Punch/Output is set to S additional parameters can be specified in a subsequent screen. See Specifying Entire System Server and VSE Librarian Options.

Additional storage of generated code with Entire System Server is notified in the generation log of the Predict file object from which the code was generated.

The following rules apply:

- **In a VSE environment**

Members are identified by member name and member type. Members of different types can therefore have the same name.

The code is stored as member in a VSE library.

The following prerequisites must be met:

- Librarian utility must be installed (VSE/SP2 or higher)
- library and sublibrary must be defined in VSE
- member type must be specified.

- **In an OS/390 environment**

Members are identified by name only.

The code is stored as a member in a partitioned data set.

The data set must be allocated. It is recommended that the data sets are cataloged. In this case, the VOLUME name need not be specified.

Note:

Parameters that have to be specified when storing generated code with Entire System Server are described in the section Specifying Entire System Server and VSE Librarian Options.

Using Entire System Server in Networks

Generated code can be stored in operating system libraries residing in remote environments. In this case Entire System Server calls are distributed by Software AG's Entire Net-Work facility.

Prerequisites for Using Entire System Server

- Depending on the setup parameters specified during installation of Entire System Server a LOGON is required.
 - Online, the LOGON is requested if necessary.
 - In batch mode, specify a logon command before the first Entire System Server command:

```
NPRLOGON <userid>,<password>,<Entire System Server database-id>
```

for example:

```
NPRLOGON AZ,XYZ,ENTIRE-SYSTEM-SERVER
```

To prevent the password being displayed in SYSOUT in batch mode, enter the terminal command %* before NPRLOGON.

- The database number 148 must be defined as Entire System Server in the Natural Parameter module by the NTDB macro (NTDB PROCESS,148).
The Entire System Server node can have a number other than 148, because the node number is always specified when an Entire System Server access is performed in Predict. In this case there must be an additional PROCESS node definition in the Natural parameter module.
- Predict supports multiple Entire System Server nodes. If the generated code is to be written by Entire System Server, the Entire System Server database ID must always be defined and the node number is filled with the logical database number.

Regenerating Code Handled by Entire System Server

When regenerating code that has been processed by Entire System Server, the members stored in a library or partitioned data set must be updated to ensure consistency.

Code can be regenerated with either the same or with different Entire System Server options:

- When regenerating code without changing the operating system member name, the Entire System Server options used for the previous generation are inserted in the generation screen (under the prerequisite that the options are valid according to the relevant defaults).
See *Specifying Entire System Server and VSE Librarian Options*.
- To regenerate code with different Entire System Server options, set the Punch/Output option to S. The options can then be changed in the second screen.

Example: A new Op. system member is specified. As the result the old operating system member is deleted and a new member is created.

If code is regenerated for which an operating system member was generated in a previous generation, and the Punch/Output option is set to N or Y, this will cause the operating system member to be deleted after a warning has been issued. The contents of the member will not be compared.

If code is regenerated and the operating system member documented in the generation log of the Predict file object is not found (this means that the member was deleted by an operating system utility) the code can be written to a new member or the generation log of the file is deleted depending on the Punch/Output option.

If Entire System Server is deinstalled, the generated code can be regenerated and the link in the generation log of the file is deleted.

If Entire System Server is temporarily not active (example: response code 5999), the generation function for this member is rejected. If it is necessary under these circumstances to change this generated code (saved with the current member and library name), it must be deleted with the Predict function Administration Implemented File.

The table below shows the effects of different generation settings when working with Entire System Server:

Member exists see ¹⁾	Overwrite Option	Code written with Entire System Server see ²⁾	Opsys Member exists	Name of Opsys Member changed	Entire System Server available	Action
-	-	Y	N	-	Y	Write to new opsys member
Y	Y	Y	Y	N	Y	Opsys member replaced
Y	Y	Y	Y	Y	Y	Old opsys member deleted; write to new opsys member
Y	Y	N	Y	-	Y	Old opsys member deleted
Y	Y	Y	N (but documented)	-	Y	Write to new opsys member
Y	Y	N	N (but documented)	-	Y	Delete references in generation log to opsys member
Y	Y	N	Y	-	N (not installed)	Delete references in generation log to opsys member
Y	Y	N	Y	-	N (not active)	Generation function rejected

¹⁾ Field Save as member filled with valid value

²⁾ Parameter Punch/output = P or S

Specifying Entire System Server and VSE Librarian Options

Additional parameters must be specified if generated external objects are written to an operating system member with Entire System Server (Punch/Output=S) or are written directly as members in an VSE librarian library (Punch/Output=Y and Library system=3).

```

13:59:30          ***** P R E D I C T 4.2.2 *****          2002-07-31
                    - Punch / Output Default Options -
                                           Modified 2002-07-31 at 13:28
                                           by CHD

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

X ENTIRE SYSTEM SERVER Database ID ..* PROCESS-148

MVS ENTIRE SYSTEM SERVER Defaults
X Data set .....
X Volume .....

VSE LIBRARIAN Defaults
X Library .....
X Sublibrary .....
X Member type .....
X VSAM catalog name ..          ( Required for ENTIRE SYSTEM SERVER)

Previous entered default options
Op. system member ..
Library system ..... 2
Punch / output ..... N
    
```

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

Parameters	
Entire System Server Database ID	ID of the Predict database object documenting the Entire System Server node. The logical database number specified in Predict is used as Entire System Server node number.
Data Set	Name of the PDS in an OS/390 environment used by Entire System Server.
Volume	Name of the Volume where the PDS used by Entire System Server is allocated. Volume must be specified if the data set is not cataloged in an OS/390 environment.
Library	Library name of the Librarian system (VSE SP2/SP3) or used by Entire System Server.
Sublibrary	Sublibrary name of the Librarian system (VSE SP2/SP3) or used by Entire System Server.
Member type	Member type of the Librarian system (VSE SP2/SP3) or used by Entire System Server.
VSAM catalog name	VSAM catalog name used only by Entire System Server.
Op. system member	The Op. system member name entered in the previous screen is displayed but cannot be modified in this screen.

Specify library system Librarian (VSE SP2/SP3) by setting option Library system to 3. Generated code written to workfile 1 is prepared for use as input of Librarian (VSE SP2/SP3) system: ACCESS and CATALOG statements are inserted at the beginning of the generated code.

Common Parameters

The parameters Adabas version and Preprocessor force can be specified with many generation functions. Preprocessor force can only be defined in the respective Modify Generation Defaults screen.

Adabas Version

Note:

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

Code	Version	Remarks
I1	V 5.1 for IBM/Siemens	Applicable to all external object types for which this parameter can be specified. When generating copy/include code, sub/superdescriptors are not included in the record buffer layout.
I3	V 5.3 for IBM/Siemens	As above.
I6	V 6.1 for IBM/Siemens	As above. This Adabas version supports larger database and file numbers.
I7	V 7.1 for IBM/Siemens	As above.
O4	V 4.1 for IBM/Siemens	
U1	V 1.1 for UNIX	
U2	V 1.2 for UNIX	
U3	V 2.1 for UNIX	
U4	V 2.2 for UNIX	
U5	V 3.1 for UNIX/NT	
V2	V 2.1 for VMS	
V3	V 3.1 for VMS	
V4	V 3.2 for VMS	
V5	V 4.1 for VMS	
P1	V 1.0 for OS/2.	
P2	V 1.2 for OS/2.	
R1	V 5.1 for IBM/Siemens	Only applicable to generation of copy/include code. Sub/superdescriptors are included physically in the record buffer layout. Code generated with this Adabas version cannot be used for update statements. Not applicable for files where parameter Adabas SQL usage =Y.
R3	V 5.3 for IBM/Siemens	As above.
R7	V 7.1 for IBM/Siemens	Similar to I7, sub/super and collation descriptors are included physically in the record buffer layout.