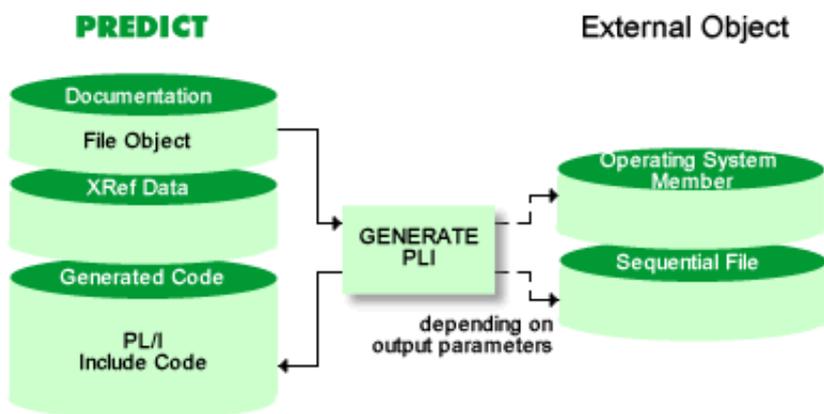


PL/I Include Code

The function Generate PL/I Include Code generates a record buffer for use in PL/I programs based on a Predict file object.

In addition, an Adabas format buffer can also be generated if required for files of the following types:

- Adabas C file (file type A) with parameter Adabas C SQL usage = N
- Adabas C userview (file type U)



Calling the Function

The Generate PL/I Include Code screen is displayed with function code G and object code PL in a Predict main menu, or with the command GENERATE PLI.

```

10:15:10          ***** P R E D I C T  4.2.1  *****          2001-07-12
Plan    0          - Generate PLI Copy Code -

File ID ..... CHD-A-FILE
Save as member ..... Save in library .... PLILIB
Overwrite option ..... Y (Y,N) Op. system member ..
Punch / output .....* N
List generated code ..... Y (Y,N) List offsets .....* N
Generate format buffer .....* N ADABAS version .....* I3
Check field name .....* A Field name prefix ..
Start level ..... 1 (0-40) Field name suffix ..
Level number increment ..... 1 (1-40) Validate ..... _
Level shift increment ..... 3 (0-9) Truncation .....* R
Nr. of abstract lines ..... (0-16) Align .....* Y
Generate initial value .....* N Static ..... N (Y,N)
Structure as char ..... N (Y,N) Numeric sign .....* T
With DCL ..... N (Y,N) Position of sign ..* R
Record buffer name .....
Format buffer name .....

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

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.

Presettings	
The parameters below can be changed in the Modify PL/I Defaults screen. See Generation Defaults.	
Library system	Library system for which the generated code is punched. Determines which additional cards need to be punched. An operating system member must be entered for the additional cards to be generated. See Parameters Specifying the Form of Output for more information.
Parameters	
The parameters Save as member, Save in library, Overwrite option, and Op. system member, Punch/output and List generated code are described in section Parameters Specifying the Form of Output.	
File ID	ID of the Predict file object from which the definitions are to be generated.
List offsets	<p>Y The offset of each item in the record buffer structure (relative to the beginning of the structure) in decimal and hexadecimal formats is to be included as a comment. The total length of each buffer is also included.</p> <p>P As above, but the absolute position (offset+1) is included as a comment.</p> <p>L The total lengths of the record buffer and the format buffer are to be included as a comment.</p> <p>V The file number and the calculated lengths of the record buffer and the format buffer are to be generated as constants in the include code. The name of the file number constant will be the record buffer name prefixed by N_. The name of each length constant will be the appropriate buffer name prefixed by L_. Each name is prefixed, suffixed, validated and truncated in the same way as any other field name.</p> <p>N No offset.</p>
Generate format buffer	<p>The contents of the format buffer will correspond exactly to the contents of the record buffer. Only valid for files of type A (with parameter Adabas C SQL usage set to N) or for files of type U. Valid values:</p> <p>Y Adabas format buffer is to be generated. Adabas groups, standard formats and lengths are used whenever possible. The resulting format buffers are then as short as possible.</p> <p>F Full format buffer is to be generated. Length and format of Adabas fields are included.</p> <p>N No format buffer is to be generated.</p> <p>Note: If you are generating for a WANG environment, you must set this parameter to F or N.</p>
Adabas version	The version of Adabas C for which the include code is to be generated. Enter an asterisk for valid values or see table in section Adabas Version for more information.

Check field names	<p>A The field names must be unique throughout the entire structure</p> <p>Y Structure levels are included in the validation check of the field names: if two fields have the same name, they must be separated by at least one field with a different name and a lower level number.</p> <p>N The field names in the generated PL/I code will not be checked for uniqueness.</p>												
Start level	<p>The starting level number to be used for the generated statements.</p> <p>Note: For fields within a redefinition, Start level is always 1.</p>												
Field name prefix	The prefix appended to each field name generated.												
Field name suffix	The suffix appended to each field name generated.												
Level number increment	<p>The increment to be used when assigning level numbers to the generated statements. This parameter is evaluated in conjunction with parameter Start level. See table below.</p> <p>Note: This parameter is ignored for fields within a redefinition.</p> <table border="1"> <thead> <tr> <th>Start Level</th> <th>Level number increment (Example)</th> <th>Level Numbers</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>5</td> <td>1, 6, 11...</td> </tr> <tr> <td>1</td> <td>5</td> <td>1, 5, 10 ...</td> </tr> <tr> <td>2 - 40</td> <td>5</td> <td>n, n+5, n+10...</td> </tr> </tbody> </table>	Start Level	Level number increment (Example)	Level Numbers	0	5	1, 6, 11...	1	5	1, 5, 10 ...	2 - 40	5	n, n+5, n+10...
Start Level	Level number increment (Example)	Level Numbers											
0	5	1, 6, 11...											
1	5	1, 5, 10 ...											
2 - 40	5	n, n+5, n+10...											
Level shift increment	<p>The number of positions to be shifted right when a level number which is higher than the current level number is encountered.</p> <p>Note: This parameter is ignored for fields within a redefinition.</p>												
Validate	<p>Determines how invalid characters are handled.</p> <p>blank Invalid characters in a field name will result in an error message but will not be modified.</p> <p>rep.char Invalid characters in a field name are replaced by this character. Valid replace characters: letters A-Z, digits 0-9, \$, %, # and _ (underscore).</p> <p>*</p> <p>Invalid characters in a field name are deleted.</p>												
Nr. of abstract lines	The number of Predict abstract lines per field to be included in the generated code.												

Truncation	<p>Specifies which characters are deleted if a generated field name is too long:</p> <p>L from the left</p> <p>R from the right</p> <p>M from the middle</p> <p>A warning is given if field names are truncated.</p>
Generate initial value	<p>Y Statements are generated to initialize the structure with the value for Init value defined for the field objects in Predict. Fields with no value defined for Init value are initialized with zeros or spaces.</p> <p>S Only fields which have a value defined for Init value in the corresponding Predict object are initialized.</p> <p>N No initialization.</p>
Align	<p>Determines which fields are to be aligned. In the record buffer and in the format buffer fields are aligned on the boundary shown in the table in section Field Format and PL/I Include Code.</p> <p>Note: This parameter only takes effect with fields that have the PL/I attribute FIXED BIN or FLOAT DEC.</p> <p>Y All fields are aligned.</p> <p>N No fields will be aligned. Predict will add the PL/I keyword UNAL to all appropriate fields.</p> <p>S Fields are aligned only if the 3GL specification Synchronized of the corresponding Predict object is set to S. Otherwise Predict adds the PL/I keyword UNAL.</p>
Static	The structure is declared with the attribute STATIC.
Structure as char	The entire generated structure is declared at the end of the record buffer as a single character string.
Numeric sign	Specifies which of the PL/I picture characters T, I or R is to be used for the representation of numeric values of format packed with sign or unpacked with sign.
With DCL	Y The generated code is preceded by a declare statement DCL and ends with a semicolon (not a comma). PL/I copy code can be inserted in any position of a PL/I program, it does not have to be inserted at the beginning.
Position of sign	<p>Defines the position of the sign in a numeric field</p> <p>R right</p> <p>L left.</p>

Record buffer name	Specifies the name of the record buffer in the generated structure. If omitted, the file ID is used.
Format buffer name	Specifies the name of the format buffer in the generated structure. If omitted, the file ID prefixed by FORBUF_ is used.

Generate PL/I Include Code in Batch Mode

Command: GENERATE PLI

Enter parameters on next line in positional or keyword form. File ID is obligatory, all other parameters are optional. If a parameter is not specified, the default value is taken.

Field	Keyword	Position
File ID	FILE-ID	1
Save as member	MEM	2
Save in library	LIB	3
Overwrite option	REPLACE	4
Op. system member	OS-MEMBER	5
Start level	START-LEVEL	6
Field name prefix	PREFIX	7
Level number increment	LEVEL-INCREMENT	8
Field name suffix	SUFFIX	9
Level shift increment	LEVEL-SHIFT	10
Nr. of abstract lines	NR	11
Validate	VALIDATION	12
Generate initial value	INIT	13
Truncation	TRUNCATION	14
Generate format buffer	FORMAT-BUFFER	15
List offsets	OFFSET	16
Structure as char	STRUCTURE	17
List generated code	LIST	18
Static	STATIC	19
Punch / output	PUNCH	20
Record buffer name	RECORD-BUFFER-NAME	21
Format buffer name	FORMAT-BUFFER-NAME	22
Align	SYNC	23
With DCL	DCL	24
Adabas version	ADA-VER	25
Check field name	CHECK-NAME	26

Numeric sign	NUM-SIGN	27
Position of sign	POS-SIGN	28
Workfile name (see note below)	WORKFILE-NAME	29
If Entire System Server is used		
- DB-ID	NP-DBID	30
- Dataset	NP-DSNAME	31
- Volume	NP-VOLSER	32
- Library	NP-LIB	33
- Sublibrary	NP-SUBLIB	34
- Member type	NP-MEMTYPE	35
- VSAM catalog	NP-VSAMCAT	36

Note:

Parameter Workfile name is obsolete in this version of Predict.

Parameters NP-LIB, NP-SUBLIB and NP-MEMTYPE must be specified if the generated code is written to workfile 1 (Punch/output=Y) and Library system=3.

Names in PL/I Include Code

The following rules apply to the generation of names for PL/I copy code.

- Field names are derived from Predict field objects: either the Field Name Synonym for PL/I or, if none exists, the Predict field ID.
- Alphabetic characters (letters) in Predict object IDs, such as file and field IDs, are converted to upper-case.
- All field names are prefixed, suffixed, validated and if necessary truncated to 31 characters according to the setting of these four options.

For example, with Field name prefix set to Adabas/, Field name suffix to *PERSONNELOFFICE, Validate to _ and Truncation to R the following names are generated for a field named "day" of type MC.

ADABAS_C_HOURS_DAY_PERSONNELOFF ADABAS_HOURS_DAY_PERSONNELOFFIC
--

- If a resulting field name is not unique, an error message may be issued if Check field name is either set to A or Y.
- The record buffer name and the format buffer name do not receive the prefix and suffix. The value(s) supplied must not contain invalid characters. If the file ID is taken as default record buffer name or, with the prefix FORBUF_, as default format buffer name, it is validated and if necessary truncated.

PL/I Names for Counter Fields - MC or PC

Before each field of type MC or PC, a counter field is generated, with a name created from the name of the MC or PC field prefixed by C_.

For example, a Predict field object HOURS-DAY with no field name synonym for PL/I and with type MC will result in two fields C_HOURS-DAY and HOURS-DAY.

PL/I Names for Additionally Generated Indicator Fields

An additional indicator field prefixed with S_ is generated in the following cases:

- a field is defined with Suppression option set to U (null allowed)
- a field of a file of type A, U or B is defined with Suppression option set to R (not null)

Note:

For Adabas C fields, the additional indicator field is only generated for the following Adabas versions:

- I3 or above
- U1 or above
- V4 or above.

Field Format and PL/I Include Code

Fields in the PL/I include code have a PL/I clause determined by the length and format of the corresponding Predict field object, as shown in the table below where s is the numeric sign whose content (T, I, or R) and position (left or right) are defined in the PL/I generation defaults; nn+m must not exceed 15; and if m is zero, V(m)9 is omitted.

Predict Format	File Type	Predict Length	PL/I Clause	Note
A		nnn	CHAR(nnn)	
B/I		1	BIT(8)	
B		3	BIT(24)	
B/I		2	FIXED BIN(15,0)	Aligned at half word boundary (if word length=4)
		4	FIXED BIN(31,0)	Aligned at word boundary (if word length=4)
B		<i>l</i> =>5	CHAR(<i>l</i>)	
I		8	CHAR(8)	
F		4	FLOAT DEC(6)	Aligned at word boundary (if word length=4)
		8	FLOAT DEC(16)	Aligned at double word boundary (if word length=4)
N/U		nn.m (nn+m<16)	PIC '(nn)9V(m)9'	If m = 0, V(m)9 is not applicable.
		nn.m (nn+m>15)	CHAR(nn+m)	
NS/US		nn.m (nn+m<16)	PIC '(nn)9V(m-1)9s' or PIC 's(nn-1)9V(m)9'.	s represents the numeric sign, the contents of which (T, I or R) and Position (left or right) depend on the parameters Numeric sign and Position of sign If m = 0, V(m) 9 is not applicable.
		nn.m (nn+m>15)	CHAR(nn+m)	
P/PS		nn.m (nn+m<16)	FIXED(nn+m,m)	
		nn.m (nn+m>15)	CHAR(<i>l</i>)	

D	D, E		CHAR(10)	
	other		FIXED(6)	
T	D, E		CHAR(8)	
	other		FIXED(12)	
L			BIT(8)	
DT	OT, OV		CHAR(9)	
	JT, JV		CHAR(25)	
G, GV, GL	D, E	n	CHAR(m)	m=2*n
JT, JV	BV	n	CHAR(n)	
	BL	n	CHAR(n)	Field is skipped if no length is specified
MO	JT, JV		FLOAT DEC(16)	
OK	JT, JV		CHAR(16)	
TK	JT, JV		CHAR(8)	

File Type in Predict	Predict Format	Predict Char. Set	Length	PL/I Clause	Note
OT, OV	A, AL	any	n	CHAR(n)	
	AV	there is only one char. set	n	Two-level group is generated: 02 Fieldname 03 Fieldname-LEN FIXED BIN (15,0) 03 Fieldname-TXT CHAR (n)	
JT, JV	A, AV	any	n	CHAR(n)	
	AL	there is only one char. set	n	CHAR(n)	Field is skipped if no length is specified
D, E	A, AL	any	n	CHAR(n)	
	AV	any	n	Two-level group is generated: 02 Fieldname 03 Fieldname-LEN FIXED BIN (15,0) 03 Fieldname-TXT CHAR (n)	
A, U	A	there is only one char. set	n	CHAR(n)	
	AV		n	Two-level group is generated: 02 Fieldname 03 Fieldname-LEN FIXED BIN (15,0) 03 Fieldname-TXT CHAR (n)	
A(SQL) AT, B	A, AV	there is only one char. set	n	CHAR(n)	

Additional Notes

An automatically generated counter field has the clause FIXED BIN (15,0).

A numeric or binary format field with a length not included in the table above is treated in PL/I include code as an alphanumeric format field and cannot have an initial value generated for it. A warning message is issued.

See also description of parameter Align.

If PL/I include code for DB2 tables/views is generated, the format NS or US is converted to the PL/I format FIXED. Any redefinitions that exist for this field are skipped.

A DB2 large object is generated as locator like an I4 field.

Sample Output

```

14:28:02          ***** P R E D I C T 4.2.1 *****          2001-08-20
                    - Generate PL/I Copy Code -                    Page: 1

File ID .. GENERATION-EXAMPLE

/*****
/* THIS RECORD-BUFFER LAYOUT WAS GENERATED BY PREDICT          */
/* FOR FILE: GENERATION-EXAMPLE                                */
/* ON: 2001-08-20 STARTING AT 14:28:02                        */
/* FILE-COMMENTS: Example file for the                         */
/* PREDICT generation subsystem.                               */
/* ..                                                         */
/*****

01 GENERATION_EXAMPLE,
  02 GROUP_1 ,
    03 ELE_N_9V5          PIC '(9)9V(5)9',
    03 GR_IN_GROUP ,
      04 ELE_B_4          FIXED BIN (31,0),
      04 ELE_PS_5V2       FIXED (7,02),
      04 MU_B_4 (5)       FIXED BIN (31,0),
    03 ELE_A_42          CHAR (42),
    03 ELE_F_8           FLOAT DEC (16),
    03 ELE_B_3           BIT(24) UNAL,
  02 C_PC_OCC_7         FIXED BIN (15,0),
  02 PC_OCC_7 (7) ,
    03 PC_ELE_DE_NS_7V3  PIC '(7)9V(2)9T',
    03 C_PC_MC_PS_6V1    FIXED BIN (15,0),
    03 PC_MC_PS_6V1 (11) FIXED (7,01),
    03 PC_GR ,
      04 PC_ELE_I_2       FIXED BIN (15,0),
      04 PC_PS_20V7      CHAR (14),
/* >>> DIC1818 WARNING: FORMAT(S) CHANGED ( ' PS ' --> ' A ' ) */
    03 PC_ELE_F_4       FLOAT DEC (6),
  02 ELE_D              FIXED (6,00),
  02 ELE_T              FIXED (12,00),
  02 ELE_L              BIT(8),
01 MU_B_4_STRUCT
  BASED(ADDR(GENERATION_EXAMPLE.MU_B_4)),
  02 MU_RED_B_7         CHAR (7),

```

```

14:28:02          ***** P R E D I C T 4.2.1 *****          2001-08-20
                   - Generate PL/I Copy Code -                   Page: 2

File ID .. GENERATION-EXAMPLE

/* >>> DIC1818 WARNING: FORMAT(S) CHANGED ( ' B ' --> ' A ' )      */
   02 MU_RED_US_13          PIC '(12)9T',

01 MU_B_4_STRUCT1
   BASED(ADDR(GENERATION_EXAMPLE.MU_B_4)),
   02 MU_RED_B_8          CHAR (8),
/* >>> DIC1818 WARNING: FORMAT(S) CHANGED ( ' B ' --> ' A ' )      */
/* ++ CONSTANTS FOR RECORD BUFFER LENGTH AND FILE NUMBER          */
01 L_GENERATION_EXAMPLE    FIXED BIN (15,0) UNAL INIT(674),
01 N_GENERATION_EXAMPLE    FIXED BIN (15,0) UNAL INIT(231),
/*****
/* THIS FORMAT-BUFFER WAS GENERATED BY PREDICT                    */
/* FOR FILE: GENERATION-EXAMPLE                                    */
/* ON: 2001-08-20 STARTING AT 14:28:03                            */
/*****
01 FORBUF_GENERATION_EXAMPLE_1 ,
   02 FILLE001 CHAR(34) INIT('AB,2X,AD,AE,AF1-5,AG,2X,AH,AI,1X,A'),
   02 FILLE002 CHAR(34) INIT('LC,2,AM1,AN1C,2,AN1(1-11),AO1,2X,A'),
   02 FILLE003 CHAR(34) INIT('R1,2X,AM2,AN2C,2,AN2(1-11),AO2,2X, '),
   02 FILLE004 CHAR(34) INIT('AR2,2X,AM3,AN3C,2,AN3(1-11),AO3,2X'),
   02 FILLE005 CHAR(34) INIT(' ,AR3,2X,AM4,AN4C,2,AN4(1-11),AO4,2'),
   02 FILLE006 CHAR(34) INIT('X,AR4,2X,AM5,AN5C,2,AN5(1-11),AO5, '),
   02 FILLE007 CHAR(34) INIT(' 2X,AR5,2X,AM6,AN6C,2,AN6(1-11),AO6'),
   02 FILLE008 CHAR(34) INIT(' ,2X,AR6,2X,AM7,AN7C,2,AN7(1-11),AO'),
   02 FILLE009 CHAR(21) INIT('7,2X,AR7,2X,AS,AT,AU. '),
01 FORBUF_GENERATION_EXAMPLE CHAR(293)
   BASED (ADDR( FORBUF_GENERATION_EXAMPLE_1 )),
/*++ CONSTANT FOR FORMAT BUFFER LENGTH                            */
01 L_FORBUF_GENERATION_EXAMPLE    FIXED BIN (15,0) UNAL INIT(293),
/*****

DIC1800 SUMMARY:      25 FIELD(S) PROCESSED
DIC1818 WARNING:      3 FORMAT(S) CHANGED
DIC1819 MESSAGE:      33 SLACK BYTE(S) GENERATED
DIC1847 MESSAGE:      3 FIELD(S) SKIPPED FOR RECORDBUFFER STRUCTURE

```