



ISPF Programmer's Guide

PREV
11 Skeletons – Design and use

↩️ AA ⚙️ 🔍

NEXT
13 Variables – Definition and using

12 Tables – Create and edit

Here you will learn how to create, edit and view ISPF tables. The table service of ISPF provides the ability to create tables, to edit and to use them for your daily work. A row in the ISPF table consists of columns that are named by ISPF variables. Each line can be addressed using ISPF table service commands. After carrying out such a command, the names of the variables in the manufacturing procedure are known.

12.1 LOCATIONS FOR TABLES

Tables can be created either temporarily or permanently. A temporary table only exists in virtual memory. It cannot be saved in a permanent data set.

For processing, the entire table always resides in **virtual** memory. This means that an ISPF table is limited in its size by the available virtual memory. Normally tables are stored in memory above the 16-megabyte-line. This allows ISPF tables to become quite large.

12.2 READING ISPF TABLES

A table is read from the ISPTLIB data set chain with the TBOPEN statement into the memory or it is created there temporarily. When a table is read from ISPTLIB with the WRITE option, then an ENQ is set on the member in which this table resides. This prevents that this table can be read a second time. The ENQ is cancelled when the table is closed with TBCLUSE. When reading with the NOWRITE option, the ENQ is only active during the read operation. Under the DD name ISPTLIB several PDS/PDSE are normally allocated. When executing the TBOPEN, the member is searched in this data set chain. The first occurring member of the name will be loaded.

12.3 WRITING ISPF TABLES

When writing ISPF table members, you cannot use the DD name ISPTLIB because under this DD name usually a chain of PDS/PDSE is allocated. Moreover, we have learned above, that we cannot write to a DD, which addresses a concatenated data set. The DD name ISPTABL is used to write tables. Therefore, always only one data set is allocated to ISPTABL. If this applies, you must consider that the following question naturally arises: How can I maintain a table if I cannot write back to the file of which I have read?

A small **trick** solve this problem:

- You can read all the tables that are included in the PDS/PDSE concatenation of about the DD name ISPTLIB.
- Tables can only be written into a data set that is associated with the DD name ISPTABL.
- When the data set that is allocated with the DD name ISPTABL to write tables is also allocated as first data set in the ISPTLIB chain, then a recently written table will always be loaded with its latest contents.

This allocation looks in practice in my TSO user:

```
T1T105 SHR,KEEP > ISPTABL LAN2T.ISPF.TLIBTAB.LPRT
                                ISPTLIB >
T1T105 SHR,KEEP > LAN2T.ISPF.TLIBTAB.LPRT
PLPRT1 SHR,KEEP > SYST1.TSO.ISPTLIB
PLPRT1 SHR,KEEP > SYST1.JCLPLUS.ISPTLIB
OST0A9 SHR,KEEP > SYST1.DOTTLIB
T18001 SHR,KEEP > SYSP.T.CAI.CAIISPT
OST0A9 SHR,KEEP > ISP.SISPTENU
```

Find answers on the fly, or master something new. Subscribe today. [See pricing options.](#)



Rule:

When I write a table (whether new or read before), then it will be written via ISPTABL in the LANZT.ISPF.TLIBTAB.LPRT data set. When I read them afterwards, it is first found in the same physical data set under the DD name ISPTLIB and read from there. In order for the ongoing maintenance of a table will guaranteed.

12.4 COMMANDS OF THE TABLE SERVICES

In ISPF, a number of commands for working with tables is available. We distinguish the commands that relate to an entire table and those, which relate to the processing within the tables:

Table 12.1: Table services commands that refer to the entire table

Command	Description of the commands that refer to the entire table.
TBCLOSE	Closes a table and saves it in the output data set if it had been read before.
TBCREATE	Creates a new table and opens it for processing.
TBEND	Closes a table without saving them.
TBERASE	Deletes a table member in the ISPTABL data set.
TBOPEN	Opens an existing table for processing.
TBQUERY	Provides information about a table.
TBSAVE	Backs up a table in the member in ISPTABL without the table to close.
TBSORT	Sorts a table.
TBSTATS	Provides statistics information on the table.

Table 12.2: Table service commands that relate to individual rows

Command	Description of commands that relate to individual rows
TBADD	Inserts a new row in the table.
TBBOTTOM	Sets the CRP (Current Row Pointer) on the last line of the table.
TBDELETE	Deletes a row from the table.
TBEXIST	Tests if there is a line with a specific key.
TBGET	Stores the contents of the current row in the corresponding variables.
TBMOD	Changes the contents of an existing line or adds a new line.
TBPUT	Changes the contents of an existing line if it exists, and if the key matches.
TBSARG	Sets on a search argument for a subsequent TBSCAN or TBDISPL.
TBSCAN	Searches in the table for a row which is in accordance with the given arguments.

Find answers on the fly, or master something new. Subscribe today. [See pricing options.](#)

	stores the values of the addressed row into the corresponding variables.
TBTOP	Sets the CRP at the beginning of the table.
TBVCLEAR	Fills all the variables of the table with zero values.

Again, I will refrain from describing the above commands in detail. Instead, the following examples will show you how to create a table and how to use it.

12.5 EXAMPLE OF WORKING WITH TABLES

As I mentioned on several occasions, I wrote an ISPF application which displays the DSNs of the most recently edited data sets in a panel. The main program of this application is **SLE** which stands for **Smart Last Edit**. It is contained in the collection of the SMART ISPF utilities. All programs of this application are written in REXX. The following table shows the most important components that belong to this application:

Table 12.3: Components of the SLE application

Name	Type	Function
SLE	Main program	This program is called as ISPF function. It displays the panel SLEP1 and performs the work initiated there.
#IMACROA	Edit macro	If an edit session is started, this macro checks whether the IMACRO option is already set for the edited data set in the edit profile. If this is not the case, the option IMACRO is set to #IMACRO1 .
#IMACRO1	Edit macro	This macro defines an ALIAS so, that at the end of the edit session #IMACRO2 is called.
#IMACRO2	Edit macro	This macro will be called automatically when an edit session ends. It inserts the DSN of the currently edited data set into the ISPF table \$SLETAB . At the next call of SLE , the name of the recently edited data set is contained in the table \$SLETAB and it will be displayed in the panel SLEP1 .
SLEP1	ISPF panel	This is the main panel for execution of the SLE application. It shows the table \$SLETAB and offers the ability to perform some actions with the shown data sets.

Many help panels are defined to assist the user in the working with the SLE application. However, these panels are not important for the function of the application and are therefore not named here.

Here the part of the edit macro **#IMACRO2**, which shows the updating (or creation) of the ISPF table **\$SLETAB**:

Program 12.1: Excerpt from edit macro **#IMACRO2**

```
57 /*****
58 */ Insert the name of the edited data set into table $SLETAB
59 /*****
60 in tab
61 if edit <> ** then den = editn("editn") /* A number was edited */
62 else den = editn /* A sequential data set was edited */
63 address "ISPEXEC"
64 *VPUT (LSTOED) SHARED
65 tabname = "$SLETAB"
66 *TOPEN *tabname LIBRARY(ISPTAB) WRITE SHARE
67 openrc = rc
68 *FUNCTION = **, DATE = date("0"), TIME = time()
69 select
70 when openrc = 0 then do
71 *TIMED *tabname OPEN /* Update table $SLETAB */
72 *TSORT *tabname FIELD(DATE,C.D.TIME,C.D)
73 *TCLOSE *tabname LIBRARY(ISPTAB) REFLCOPT PAD(100)
74 end
75 when openrc = 8 then do /* Table $SLETAB does not exist. Create it */
76 *TCREATE *tabname REFL(DSH) NAMES(FUNCTION,DATE,TIME)
77 *TADD *tabname OPEN
78 *TCLOSE *tabname LIBRARY(ISPTAB) REFLCOPT PAD(100)
79 end
80 when openrc > 8 then do
81 *editn = editn /* Error
82 *RETURN MSG(LIB2001)
83 end
84 otherwise nop
```

Lines	Explanations
66	Table \$SLETAB is opened. The RC is saved in openrc.
68	Some table variables are set.
70–74	When the table \$SLETAB exists, it will be updated and written back to ISPTAB.
75–79	When the table \$SLETAB does not exist it will be created and written into ISPTAB as new member which then contains one row of data.

Here an excerpt of the panel description of panel SLEP1 where the)MODEL line defines the structure of the displayed rows of the table \$SLETAB.

Coding 12.1: Excerpt from panel definition of the panel SLEP1

```
21 *CHRecent DEN in edit Ahelp with PPI for all fields $Date $Time
22 *F-f
23 )MODEL
24 *S(den $date $time
25 )INIT
26 *vars = '(dynan insert maxrows allcha function)'
27 *help = SLEP1
28 *errorw = allch
29 *VPUT (ALLCHA) PROFILE
```

Lines	Explanation
23	Here the model definition segment begins. This line can be followed by up to 8 definitions of data structures.
24	Here only one line structure is defined. It contains the names and positions where the contents of the variables of the table \$SLETAB are displayed. Because in the previous panel definitions already four Z-variables are set, the Z-variable in line 24 for the name FUNCTION is assigned by the .ZVARS definition.

The complete definition of panel SLEP1 is found in [Program 10.1](#): Panel definition of panel SLEP1 on page 194.

The TBDISPL command to display the panel SLEP1 together with the table \$SLETAB is shown in the following excerpt from program SLE:

Coding 12.2: Excerpt from program SLE showing the TBDISPL command of panel SLEP1

```
/*-----*/
/* Display table
/*-----*/
*TOP *tabname
*TBDISPL *tabname PANEL(SLEP1) AUTOSEL(NO)
/* Start error handling */
if rc > 8 then call left_error rc /*|||,
**TBDISPL *tabname PANEL(SLEP1) AUTOSEL(NO) /*|||,
**TBDISPL *tabname PANEL(SLEP1) AUTOSEL(NO)
/* End error handling */
*VPUT (ALLCHA) PROFILE /* save insert indicator ALL or CHANGED */
if prc = 1 then leave k /* PPI was pressed, go to program end */
```

Here now a display of the panel SLEP1 containing the data of the table \$SLETAB:

Screen 12.1: Display of panel SLEP1 produced by the program SLE

Find answers on the fly, or master something new. Subscribe today. [See pricing options.](#)

HELP 8 ----- Table display of recently edited data sets Row 1 of 13		
COMMAND ----	SYMBOL ----	CUR
Pyramid- TB8-V10	Compile state of SLE	
Sort: Date of Release 2 Max Date: 76	RC	COMPILED
Insert control for edited data sets	----	All of Changes Add
C Recent DSN in edit Help with FPL for all fields	Date	Time

- USER001.BOOK1.REXX(SLE)	15/04/08	13:50:44
- USER001.BOOK1.REXX(SLE)	15/04/08	13:19:47
- USER001.BOOK2.REXX(TEST2)	15/04/08	09:43:01
- USER001.BOOK2.REXX(TEST1)	15/04/08	09:06:44
- USER001.BOOK1.EXAMPLE.REXX(TEST1)	15/04/07	14:22:16
- USER001.BOOK1.REXX(SFSPGFAR)	15/04/07	14:17:28
- USER001.BOOK1.EXAMPLE.REXX(CDSREXX)	15/04/04	14:40:18
- USER001.BOOK2.EXAMPLE.REXX(CDSREXX)	15/04/04	14:44:19
- USER001.TBM.REXX(CDSREXX)	15/04/04	14:12:28
- USER001.TBM.REXX(CDSREXX)	15/04/04	14:30:03
- USER001.TBM.REXX(CDSREXX)	15/04/04	14:29:44
- USER001.BOOK1.REXX(SLECOND)	15/04/03	15:45:47
- USER001.REXX(TEST1)	15/04/03	15:44:02

[Recommended](#) / [Playlists](#) / [History](#) / [Topics](#) / [Settings](#) / [Get the App](#) / [Sign Out](#)

 [PREV](#)
[11 Skeletons – Design and use](#)

[NEXT](#) 

[13 Variables – Definition and using](#)

Find answers on the fly, or master something new. Subscribe today. [See pricing options.](#)