



### 6.1.1 Online execution

For the online calling of a program, you do not know whether this is a procedure or a compiled program. Moreover, you do not have to worry about the program requiring ISPF services, because you are already working in ISPF. Nevertheless, you have to know if there are parameters needed for the correct execution.

Programs can be executed online as follows.

- You can execute a REXX program by using the edit macro `##` contained in the SMART ISPF utilities when the program is currently being edited. This means simply enter in the command line of the editor `## test! This is a text`.
- Type in the command line `"iso test! This is a text"`.
- Enter in a DSLIST display panel TEST1 in front of a DSN. In this case, the DSN will be passed to the program as a parameter.
- Select the ISPF menu 6 Command and enter in the `==>` line `"test! This is a text"`.
- Insert a CMD command entry in the ISPF command table to execute the program. In this case, you can start the program by entering the command name in the command line of each ISPF panel. See the information to insert a command table entry in the chapter below. The SMART ISPF utility program SLE is a typical candidate for this type of call. See the following command table entry example:

```
_SLE      2  SELECT CMD(%SLE)
          REXX  SLE      CALL THE LASTEDIT FUNCTION
```

I will not demonstrate the online execution of the above program here. This is so simple that I really do not have to show it.

### 6.1.2 Batch execution

Executing a REXX program in a batch job is not as easy as in the online environment because an appropriate JCL program is required which is submitted to the batch system environment for execution. This is initially a difficult thing, because you need to know how the JCL must be designed to operate properly. In my work in the TSO / ISPF, this problem occurred earlier repeatedly. Therefore, long time ago I developed two EDIT macros that relieve me of creating each required JCL. These two edit macros are:

- **#TSOB** to submit and execute REXX programs which only need the TSO environment.
- **#ISPFB** to submit and execute REXX programs which additionally need the ISPF environment.

If a program requires input parameters, these can entered when calling. I executed the two programs TEST1 and TEST2 each with the correct edit macro. I have added the resulted JCL programs below.

JCL 6.1: JCL to execute TEST1 in a native TSO environment

```
//ESSMSTRU JOB (UNIVER),'#TSOB:TEST1',CLASS=A,
//  NOTIFY=ESSMSTR,MSGLEVEL=(0,0),MSGCLASS=H,
//  COND=(0,LT),TIME=100,USER=ESSMSTR
//*****
//* JOBPARM S=TESTMVS
//TSOBATCH EXEC PGM=JESJCL,REGION=80M,DYNAMNR=99,
//  ACCT=(UNIVER),TIME=100
//STEPLIB DD DISP=SHR,DSN=UNIVESS.LOAD
//SYSEXEC DD DISP=SHR,DSN=ESSMSTR.TEMP.SUBMIT
//          DD DISP=SHR,DSN=ESSMSTR.REXX
//          DD DISP=SHR,DSN=UNIVESS.REXX
//SYSPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD DUMP
//SYSOUT DD SYSOUT=*
//SYSTSIN DD *
//          PROFILE NOPREFIX NOMSGID
//          /* Executed REXX Program:ESSMSTR.TEMP.SUBMIT(TEST1)
//          %TEST1 param1 param2
```

When running this program, the following list is printed. It can be found in SDSF.

```
READY
PROFILE NOPREFIX NOMSGID
READY
/* Executed REXX Program: ESSMSTR.TEMP.SUBMIT(TEST1)
READY
%TEST1 param1 parama2
Entered parameter is: PARAM1 PARAM2
READY
END
```

JCL 6.2: JCL to execute TEST2 in a TSO/ISPF environment

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

```

//BSMSTRM JOB (UNIVER),'ISPPRT: TEST2',CLASS=A,
//          NOTIFY=BSMSTRM,MSGLEVEL=(1,1),MSGCLASS=J,
//          COND=(0,LT),TIME=100,USER=BSMSTRM
//          JOBPARN S=TESTV2U
//ISPP EXEC PGM=ISPPROF,REGION=80M,DYNAMNBR=99,
//          ACCT=(UNIVER),TIME=100
//STEP1LIB DD DISP=SHR,DDN=BSMSTRM.LOAD
//SYSEXEC DD DISP=SHR,DDN=BSMSTRM.TEMP.SUBMIT
//          DD DISP=SHR,DDN=BSMSTRM.D00K1.REXX
//          DD DISP=SHR,DDN=BSMSTRM.REXX
//          DD DISP=SHR,DDN=ISP.ISPEXEC
//          DD DISP=SHR,DDN=ISP.ISPEXEC
//ISP2LIB DD DISP=SHR,DDN=CHMTR.ISPSLIB
//          DD DISP=SHR,DDN=ISP.ISPSMENU
//          DD DISP=SHR,DDN=ISP.ISPSLIB
//          DD DISP=SHR,DDN=ISP.ISPEXEC
//ISP3LIB DD DISP=SHR,DDN=BSMSTRM.NMSG
//          DD DISP=SHR,DDN=ISP.ISPMENU
//          DD DISP=SHR,DDN=ISP.ISPSLIB
//ISP4LIB DD DISP=SHR,DDN=BSMSTRM.PANELS
//          DD DISP=SHR,DDN=ISP.ISPMENU
//          DD DISP=SHR,DDN=ISP.ISPSLIB
//ISP5LIB DD DISP=SHR,DDN=BSMSTRM.PANELS
//          DD DISP=SHR,DDN=ISP.ISPMENU
//          DD DISP=SHR,DDN=ISP.ISPSLIB
//          DD DISP=SHR,DDN=BSMSTRM.ISPPROF
//          DD DISP=SHR,DDN=BSMSTRM.VIO
//          DD DISP=SHR,DDN=BSMSTRM.D00K1.REXX
//          DD DISP=SHR,DDN=BSMSTRM.ISPPROF
//          DD DISP=SHR,DDN=BSMSTRM.ISPMENU
//          DD DISP=SHR,DDN=ISP.ISPSLIB
//ISPGLO DD SYSOUT=,DCB=(LRECL=121,RECFM=FB)
//SYSPRINT DD SYSOUT=
//SYSTSMT DD SYSOUT=
//SYSDUMP DD DUMP
//SYSTSIN DD *
//
// PROFILE NOPRFXIT NMSGID
//          EXECUTED REXX PGM=BSMSTRM.TEMP.SUBMIT(TEST2)
//          TESTPMT COM(TEST2,panel,panel2) NEWAPPAL (ISPP)

```

When running this program, the following list is printed. It can be found in SDFS.

```

/*-----*/
/* Display table */
/*-----*/
%BTOP tabname
%TDISPL tabname PANNEL(SLHP1) AUTOSEL(NO)*
/* Start error handling */
if rc > 8 then call ispf_error rc /*|||,
**%TDISPL tabname PANNEL(SLHP1) AUTOSEL(NO)*/*|||,
%TDISPL tabname PANNEL(SLHP1) AUTOSEL(NO)*
/* End error handling */

```

I think you now have the necessary knowledge to understand the following chapters.

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

PREV

5 The TSO/E REXX commands

NEXT ▶▶

7 Introduction to ISPF programming