



ISPF Programmer's Guide

1 REVIEW

by [Franz Lanz](#)

Publisher: [De Gruyter Oldenbourg](#)

Release Date: *November 2015*

ISBN: 9783110407655

Topic: [Productivity Tools](#)

[CONTINUE READING](#)[View table of contents](#)

Book Description

This book is intended to support ISPF application programmers to become professional in the smart programming of ISPF applications using the REXX language. The contents are presented in a modular manner to suit reading with heterogeneous ISPF programming knowledge. The following topics are covered: Introduction to the programming language REXX as well as to ISPF programming, data processing in ISPF applications, use of messages, panels, skeletons, tables, ISPF variables and an introduction to creating and applying edit macros. Each theme is vividly illustrated by programming examples. The Smart ISPF Utilities contain some very useful programming aids that are often useful when programming ISPF applications. The book serves as a textbook as well as a manual for daily work. Many cross-references are included as well as an extensive index. Moreover, the author gives many helpful hints and tips on smart ISPF programming practices. The Smart ISPF Utilities contain many useful programming aids.



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

De Gruyter Oldenbourg, an imprint of De Gruyter since 2013, publishes content in the fields of history, economics, informatics, and engineering. Our reputation is strong for monographs and textbooks, and we have a close relationship ...



DE GRUYTER
OLDENBOURG

[More about De Gruyter Oldenbourg](#)

Table of Contents

 [Cover](#)

 [Title](#)

 [Copyright](#)

 [Preface](#)

 [1 Introduction](#)

[1.1 Users of this book](#)

[1.2 The significance of the REXX programming language in z/OS](#)

[1.3 Brochures for programming ISPF applications](#)

[1.3.1 The REXX programming literature](#)

[1.3.2 ISPF reference books](#)

[1.3.3 TSO reference books](#)

[1.3.4 The ISPF services](#)

 [2 Introduction to the REXX programming language](#)

[2.1 What is REXX?](#)

[2.2 Overview of REXX under TSO](#)

[2.2.1 Recognizing a REXX procedure by the TSO](#)

[2.2.2 Running REXX procedures in the TSO](#)

[2.3 Compile REXX procedures](#)

[2.4 Performance of REXX procedures](#)

[2.5 The syntax of the REXX language](#)

[2.6 Variables in REXX](#)

[2.7 Data types of REXX variables](#)



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

- 2.8.1 String operators
- 2.8.2 Arithmetic Operators
- 2.8.3 Compare operators
- 2.8.4 Logical operators

2.9 Stems in REXX

- 2.9.1 Initialize stems with null string
- 2.9.2 File processing in connection with stems
- 2.9.3 Multi-unit Stems



3 The REXX commands

3.1 ADDRESS – Connection to the subsystems

- 3.1.1 The host command environments
- 3.1.2 The active host command environment
- 3.1.3 The temporary addressing of external commands
- 3.1.4 Special case ISPEXEC and ISREDIT
- 3.1.5 The main host command environments

3.2 ARG – Retrieve the parameter string

3.3 CALL – Call other programs

3.4 DO groups and DO loops

- 3.4.1 The DO group
- 3.4.2 The DO loop

3.5 EXIT and RETURN – leaving the REXX procedure

3.6 IF and WHEN – check conditions

3.7 INTERPRET – generate REXX commands dynamically

3.8 NUMERIC – set computing options

3.9 PARSE – text fragmentation

- 3.9.1 PARSE ARG
- 3.9.2 PARSE VALUE WITH
- 3.9.3 PARSE VAR
- 3.9.4 PARSE SOURCE



3.10.1 Internal subroutines with PROCEDURE statement

3.10.2 Internal subroutines without PROCEDURE statement

3.11 QUEUE – Working with the TSO stack

3.11.1 The TSO/E data stack

3.11.2 Use options for the TSO/E Data Stack

3.12 SAY – Print texts

3.13 SELECT – Conditionally call alternative instructions

3.14 NOP – No operation

3.15 PULL – Enter data on the screen

3.16 TRACE – The strong debugging aid

3.17 SIGNAL – Jumping when errors



4 The REXX functions

4.1 General functions

4.1.1 ADDRESS – Get the active host command environment

4.1.2 ARG – Input parameter test or take

4.1.3 DATE – Date functions

4.1.4 TIME – Time functions

4.1.5 QUEUED – Number of records in the data stack

4.1.6 SOURCELINE – Return a program line

4.1.7 USERID – Return the TSO user ID

4.1.8 VALUE – Create variable names dynamically

4.2 Arithmetic functions

4.2.1 ABS – absolute value of a number

4.2.2 DIGITS, FORM, FUZZ – Query options for arithmetic operations

4.2.3 MIN, MAX – Minimum and maximum value

4.2.4 RANDOM – Generate random numbers

4.2.5 SIGN – Return of the sign

4.3 Comparison Functions

4.3.1 COMPARE – Compare texts



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

4.4 Conversion functions

- 4.4.1 C2D – Character to decimal
- 4.4.2 C2X – Character to Hexadecimal
- 4.4.3 D2C – Decimal to Character
- 4.4.4 D2X – Decimal to Hexadecimal
- 4.4.5 X2B – Hexadecimal to Binary
- 4.4.6 X2C – Hexadecimal to Character
- 4.4.7 X2D – Hexadecimal to Decimal

4.5 Formatting functions

- 4.5.1 CENTER – Centering a string
- 4.5.2 COPIES – Reproduce texts
- 4.5.3 FORMAT – Format numbers
- 4.5.4 JUSTIFY – Formatting a string
- 4.5.5 LEFT – Rearrange text left-justified
- 4.5.6 RIGHT – Arrange text right-justified

4.6 String functions

- 4.6.1 DELSTR – Delete substrings
- 4.6.2 INSERT – Insert text
- 4.6.3 SCHANG – Change of texts
- 4.6.4 LENGTH – Length of a text
- 4.6.5 OVERLAY – Superimpose text
- 4.6.6 POS – Search for text
- 4.6.7 STRIP – remove border characters
- 4.6.8 SUBSTR – Extract part of a text
- 4.6.9 TRANSLATE – translate characters
- 4.6.10 VERIFY – verify text

4.7 Word functions

- 4.7.1 WORD – return of a word
- 4.7.2 WORDINDEX – return the starting position of a word



4.7.4 WORDPOS – search for a word

4.7.5 WORDS – number of words in a string

4.8 System functions

4.8.1 LISTDSI – List data set information

4.8.2 MSG – control of the TSO messages

4.8.3 MVSVAR – Return z/OS system information

4.8.4 OUTTRAP – take TSO messages

4.8.5 SYSDSN – check data set status

4.8.6 SYSVAR – get system Information

4.8.7 STORAGE – read and write memory contents



5 The TSO/E REXX commands

5.1 EXECIO – Read and write data sets

5.2 DELSTACK – delete data stack contents

5.3 DROPBUF – Delete data stack buffers

5.4 TSO Commands



6 Execute REXX programs

6.1 Execution of programs in a TSO/ISPF environment

6.1.1 Online execution

6.1.2 Batch execution



7 Introduction to ISPF programming

7.1 Programming languages useable in ISPF

7.2 ISPF programming objects

7.3 Some typical examples

7.3.1 Example for use of ISPF panels

7.3.2 Example for use of skeletons

7.3.3 Example for use of tables

7.3.4 Example for use of ISPF variables

7.3.5 Example for data processing with TSO and ISPF



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

7.4 LIBDEF – Dynamic linking of ISPF libraries

7.5 ALTLIB – Dynamic linking of EXEC libraries

7.5.1 Search sequence in the procedures libraries

7.5.2 The ALTLIB command in ISPF

7.5.3 Stacking of the APPLICATION level ALTLIBs

7.5.4 The QUIET operand of the ALTLIB DISPLAY command



8 Data set processing using ISPF

8.1 The LM services

8.1.1 Grouping of LM services

8.1.2 LMINIT – Start of the data set processing

8.1.3 LMFREE – Free a data set

8.1.4 LMOPEN – Open a data set for processing

8.1.5 LMCLOSE – Close a data set

8.1.6 LMMFIND – Localize a member

8.1.7 LMMREP – Replace a member

8.1.8 LMMADD – Add a member

8.1.9 LMGET – Read a data records

8.1.10 LMPUT – Output data records

8.1.11 LMCOPY – Copy data

8.1.12 LMMOVE – Move data

8.1.13 LMMDEL – Delete members

8.1.14 LMMREN – Rename members

8.1.15 LMMSTATS – Display or change member statistics

8.1.16 LMCOMP – Compress a PDS

8.1.17 LMMLIST – Display a member list

8.1.18 LMMDISP – Display and edit a member list

8.1.19 LMDINIT – Initialize the LMDDISP service

8.1.20 LMDDISP – Data set list service

8.1.21 LMDLIST – List of data sets



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

8.1.23 MEMLIST – member list dialog service

8.2 Data set query services

8.2.1 LMDLIST – Data set list service

8.2.2 DSINFO – ISPF service which provides data set information

8.2.3 LISTDSI – REXX function to list data set information

8.2.4 QBASELIB – Query DSN information

8.2.5 QLIBDEF – Query LIBDEF Information

8.2.6 QUERYENQ – ENQs determination



9 Messages – Definition, setting, output

9.1 Error handling in ISPF

9.1.1 Returning error messages

9.1.2 Output of error messages

9.1.3 SETMSG – Set next message

9.1.4 Definition of ISPF messages

9.1.5 Naming convention of the ISPF message IDs

9.1.6 Definition of messages

9.1.7 The standard message member ISRZ00



10 Panels – create and use

10.1 The Dynamic Tag Language (DTL)

10.2 Panel types in ISPF

10.3 Definition of panels

10.3.1 The structure of a panel

10.3.2 Creation of panels and their call

10.3.3 The panel definition sections

10.3.4 Variables in panel definitions

10.3.5 Panel processing

10.3.6 Help panels

10.3.7 Panels to display ISPF tables



11.1 Creating skeletons

11.2 Steps to use the file-tailoring service



12 Tables – Create and edit

12.1 Locations for tables

12.2 Reading ISPF tables

12.3 Writing ISPF tables

12.4 Commands of the table services

12.5 Example of working with tables



13 Variables – Definition and using

13.1 Variables pools

13.1.1 Function pool

13.1.2 Shared pool

13.1.3 Profile pool

13.2 Saving the profile members

13.3 Creating profile members

13.4 Display of actually used profile member

13.5 The System Profile Pool

13.5.1 Frequently used ISPF variables

13.5.2 Process ISPF variables



14 Edit macros – Create and apply

14.1 What is an edit macro?

14.1.1 Naming conventions for edit macros

14.1.2 Example of two very useful edit macros

14.2 Table of edit macro commands

14.3 Operands and abbreviations used in the edit macro commands

14.4 Test aids in the creating macros

14.4.1 Prototyping

14.4.2 The REXX TRACE command as developing aid



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

- 14.5 System variables of the ISPF editor
- 14.6 Passing parameters to macros
- 14.7 Examples for editing and SUBMIT batch jobs
- 14.8 Mnemonics for macro programming



15 The SMART ISPF utilities

- 15.1 Naming conventions
- 15.2 The dynamic panel concept
- 15.3 List of executable programs
- 15.4 Program descriptions
 - 15.4.1 Edit macro ## – Execute a currently edited REXX procedure
 - 15.4.2 Edit macro #ALTXT – Realign line parts
 - 15.4.3 Edit macro #EDMEM – Edit of a member
 - 15.4.4 Edit macro #IMACROA – General ISPF edit macro
 - 15.4.5 Edit macro #IMACRO1 – Initial edit macro
 - 15.4.6 Edit macro #IMACRO2 – Edit session end macro
 - 15.4.7 Edit macro #ISPFB – Submit an ISPF batch job
 - 15.4.8 Edit macro #LCH – Perform long edit change commands
 - 15.4.9 Edit macro #SPLJ – SPLIT and JOIN lines
 - 15.4.10 Edit macro #SSS – Clear SCHFOR lists
 - 15.4.11 Edit macro #SSSCH – Mass update of members
 - 15.4.12 Edit macro #SU – Submit JCL without a JOB statement
 - 15.4.13 Edit macro #TSOB – Submit a TSO batch job
 - 15.4.14 Edit macro #VERASE – Erase ISPF profile variables
 - 15.4.15 Program SCURS0R – Calling a data set from an ISPF screen
 - 15.4.16 Program SDOC – Produce documentation members
 - 15.4.17 Program SLE – Display last edited data sets
 - 15.4.18 SLOGDSN – Data member containing DSNs for logon
 - 15.4.19 Program SLOGON – Personal logon procedure
 - 15.4.20 Program SPROFEDT – Store users ISPF profile variables



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

15.4.22 Program SSC – Super clone for data sets

15.4.23 Program SSS – Perform a Super-Search

15.5 Programming aids

15.5.1 Edit macro #C – Compile and execute a REXX program

15.5.2 Edit macro #RO – Online compile of a REXX program

15.5.3 Edit macro #RC – Compile a REXX procedure with a batch job

15.5.4 Edit macro #RCLOAD – Produce a load or a call module

15.5.5 Edit macro #IE – Insert a call to ISPF_ERROR

15.5.6 Subroutine ISPF_ERROR – Display ISPF error messages

15.5.7 Edit macro #PAN – Execute a panel source code

15.5.8 Subroutine DAYDIFF – Calculate number of days

15.5.9 Subroutine ENDDAY – Calculate a target date

15.5.10 Subroutine JULDATE – Translate a date to Julian and vice versa

15.5.11 Subroutine LEAPYEAR – Return the leap year information

15.5.12 REXX subroutine SCHANG – REXX change function

15.5.13 Program SDYNPAN – Convert a panel source code

15.6 Installation of SMART ISPF utilities

15.6.1 Download and unzip

15.6.2 Installation

15.6.3 ALTLIB command

15.6.4 Make the SMART ISPF utilities ready to run



List of programs



List of tables



List of screens



Bibliography



Index



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



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