



TSO/E REXX Programming; Parts I & II

Duration: 3-5+ Days (Depending upon selected content, Part I alone is 3 days, note the comments in Part II about module selection.)

Audience:

Delegates may be new to REXX, or have some exposure to REXX but need to underpin their experience by formal training.

Pre-requisites:

An understanding of computer concepts is assumed.

A working knowledge of TSO/ISPF is required. This can be gained from our z/OS TSO/ISPF Workshop.

Course Objectives

Part I

Each delegate will acquire a working knowledge of REXX in the z/OS TSO/E environment. Good programming practice is encouraged throughout. The course starts with the basics and furthers learning with ample hands-on exercises.

Part II

This is all about exploiting REXX with other components of z/OS. Whilst all modules could be covered, the thinking is that only the required modules would be selected providing focused flexibility with respect to content.

Course Content

Part I – Gaining the basics

Module 1: TSO/E REXX Environment

REXX Platforms

What is REXX?

Where is REXX code held?

How is REXX invoked?

REXX in batch; IKJEFT01 and IRXJCL



TSO/E REXX Programming; Parts I & II

Module 2: Structure and Syntax

What constitutes REXX code?

Changing default environments; ADDRESS instruction

The coding rules described

Definition of variables

STEM variables and the DROP instruction

Operator characters; Arithmetic, Logical, Comparison and Concatenation

Operator order of precedence

Module 3: Diagnostic Aids

Checking logic flow; SAY instruction

Immediate commands; HE, HI, HT, RT, TE and TS

Trapping errors; CALL and SIGNAL instructions

Analyzing instruction execution; TRACE instruction and EXECUTIL command

Module 4: File Processing

File processing overview

Making data sets available; TSO ALLOC command

Disposing of data sets; TSO FREE command

Read or Write a data set; EXECIO command

I/O data areas; STEM and Data Stack

File Allocation using BPXWDYN

Module 5: Control Instructions

Basic decision making; IF and ELSE instructions

Case statements; SELECT, WHEN and OTHERWISE instructions

Iterative and conditional processing; DO instruction

Iterative processing logic

Bypassing iterative instructions; ITERATE instruction

Terminating iteration; LEAVE instruction

Module 6: Parsing

What is Parsing?

PARSE instruction general syntax

Read from keyboard or Data Stack; PARSE PULL

Read from a Variable; PARSE VAR

Process the results of an expression; PARSE VALUE

Handling excess data generated by PARSE

INTERPRET instruction



TSO/E REXX Programming; Parts I & II

Module 7: Sub-routines and Functions

Naming internal sub-routines
Invoking internal sub-routines; CALL instruction
Hiding variables from a sub-routine; PROCEDURE instruction
Sharing hidden variables; EXPOSE instruction
Invoking external sub-routines; CALL instruction
Accessing passed data; PARSE ARG instruction and the ARG Built-in Function
Returning data to the caller; EXIT and RETURN instructions
The difference between a sub-routine and a Function

Module 8: REXX Built-in Functions

Conversion functions; C2D, C2X, D2C, D2X, X2B, X2C and X2D
Acquire date and time; DATE and TIME functions
Format a string; CENTRE, LEFT, RIGHT, SPACE, STRIP and TRANSLATE functions
Determine the size of a string; LENGTH, WORDLENGTH, and WORDS functions
Locate the position of something; FIND, POS, and WORD functions
Validating content; COMPARE, DATATYPE and VERIFY functions
Duplicate something; COPIES function
Shorten a string; DELSTR, DELWORD, SUBSTR and SUBWORD functions
Obtain session user identity; USERID function
Expand a string; INSERT function
Get highest/lowest value in a range; MAX and MIN functions
TSO/E functions; LISTDSI, MSG and PROMPT

Module 9: Data Stack Management

What is the Data Stack?
Add data to the stack; PUSH and QUEUE instructions
Read from the Data Stack; PARSE PULL
Stack management; DELSTACK, DROPBUF, NEWSTACK, QBUF, QELEM, and QSTACK Commands and the QUEUED Built-in Function
Some possible Data Stack uses



TSO/E REXX Programming; Parts I & II

Part II – Real world exploitation

It is recommended that if Part II is required, then only those modules that are required be selected otherwise the duration will exceed 5 days. Any selection of more than 15 modules total will require two separate sessions as delivery of more than 3 modules a day cannot be guaranteed.

Module 10: TSO/E Commands and Output Capture

List TSO/E Command plus IDCAMS affiliations
List of TSO/E built-in Functions
List the member of a PDS(E); TSO/E LISTDS command
Trapping command output; OUTTRAP Built-in Function
IDCAMS ALTER Command versus TSO/E RENAME
Delete a data set or member; IDCAMS versus TSO/E DELETE
Print a data set; IDCAMS PRINT versus TSO/E PRINTDS, plus the TSO/E OUTDES command
TSO/E SUBMIT Command
RACF Commands

Module 11: Invoking non-REXX programs

The TSO/E CALL Command
The TSO/E EXEC Command
The REXX LINK Environment
Pass and receive fixed length data; LINKPGM Environment
Pass and receive data of amended length; LINKMVS Environment
Search order for all LINK environments
Return Codes from LINKPGM and LINKMVS

Module 12: Using the SDSF Interface

Add/Delete SDSF Environment
Issuing SDSF primary commands; ISFEXEC instruction
Special panel commands and variable
Using the WHO and QUERY commands
Various Return Codes
Issuing SDSF line commands; ISFACT instruction
The SPOOL variables
Capturing JES2 SPOOL output to a data set
Issuing z/OS commands; ISFSLASH and its variables



TSO/E REXX Programming; Parts I & II

Module 13: ISPF Overview & Panels

ISPF structure

Libraries and data sets

Assigning alternative libraries; LIBDEF Command

Checking library status; ISPLIBD Command

ISPF Services; Browse, Control, Display, Edit, Select, VGET, VPUT and View

Establishing default libraries via TSO LOGON procedure

Diagnostic aids

ISPF in Batch

Starting ISPF; ISPSTART Command

ISPF Panel Components;)ATTR,)BODY,)INIT,)REINIT and)PROC

HELP Panels and Messages described

Panel logic; IF / ELSE, GOTO, EXIT

Panel Functions; PFK, TRUNC and VER

Embedding REXX on ISPF Panels

Testing Panels and Functions

Module 14: ISPF Variables & Tables

ISPF Variable Pools; Function pool, Shared pool and Profile pool

Defining variables on the panel

Using pre-defined variables

Using Input and output fields as variables

Using Variables in)INIT,)REINIT, and)PROC

Saving variables - VPUT

Acquiring variables - VGET

Handling short variable names – ZVARS

Control Variables

Variables in Tables

What are Tables

Positioning within a Table; TBTOP, TBBOTTOM, and TBSKIP

Table management; TBCREATE, TBOPEN, TBADD, TBDELETE, TBGET,

TBPUT, TBMOD, TBSORT and TBQUERY

Table display; TBDISPL

Table disposition; TBEND, TBCLOSE and TBSAVE



TSO/E REXX Programming; Parts I & II

Module 15: ISPF File Tailoring

Service overview
FTOPEN Service
FTCLOSE Service
FTINCL Service
File tailoring control statements
Conditional selection
Handling character conflicts
Nesting files
Variable assignments
Tables and File Tailoring

Module 16: ISPF Library Management

Overview of instruction relationships
Data set processing; LMINIT, LMOPEN, LMGET, LMPUT, LMCLOSE, LMFREE, LMERASE and LMRENAME
PDS/PDSE processing; LMMFIND, LMMADD, LMMDEL, and LMMREN and LMMREP
Simulate ISPF 3.3; LMCOPY and LMMOVE
Simulate ISPF 3.4; LMDINIT, LMDDISP, LMDLIST and LMDFREE
Library Management Return Codes
LMMFIND statistics

Module 17: Accessing z/OS UNIX System Services

Checking UNIX access; SYSCALLS Built-in Function
Awareness of variable isolation
Permission bit settings
Creating and reading a z/OS UNIX System Services file
Alternatives to readfile and writefile; EXECIO instruction
z/OS UNIX System Service file allocation using TSO ALLOC and BPXWDYN



TSO/E REXX Programming; Parts I & II

Module 18: REXX with DB2 and SQL

Ensuring DB2 is available; SUBCOM command and RXSUBCOM Built-in Function

Making DB2 the default environment; ADDRESS DSNREXX instruction

Connecting to DB2; DSNREXX "CONNECT"

Prohibited SQL statements

SQL statement rules

Using variable

Cursor and Statement naming

The difference between DB2 and REXX variables

Error handling

DB2 Command interface; -DIS and -TERM commands