

## TSO REXX Programming in z/OS (5 Days)

**Benefits:**

Students who complete this course will be able to accomplish work using native TSO commands and the powerful REXX capability of TSO. REXX execs are developed that can be immediately useful in day-to-day work.

**Audience:**

Applications and systems programmers, who need to know how to use TSO commands and how to create or maintain REXX execs. Anyone who will be creating or maintaining applications written to run under the Dialog Manager.

**Prerequisites:**

At the very least, the student entering this course should have experience in using ISPF/PDF, especially the editor. Additionally, experience submitting jobs to run in the batch and some programming background are helpful.

**TSO REXX Programming in z/OS - Course Objectives:**

On successful completion of this course, the student, with the aid of the appropriate reference materials, should be able to:

1. Describe the TSO environment, and describe the distinctions between TSO commands and REXX instructions
2. Write REXX EXECs to accomplish useful functions
3. Use TSO commands to work with datasets, either in native mode or in EXECs
4. Use REXX instructions to work with records in files
5. Use subroutines as a coding technique for EXECs
6. Use TSO and EXECs to run programs in the Foreground or the Background (batch)
7. Run EXEC's in the batch, in TSO/E-integrated address spaces or non-TSO/E-integrated address spaces
8. Use TSO commands to send and receive datasets between users
9. Use the REXX compiler, if it is available
10. Use the level 2 REXX constructs, if they are available.

## **TSO REXX Programming in z/OS - Topical Outline:**

### Day One

#### Introduction

Review of TSO LOGON command and parameters

TMP - The Terminal Monitor Program

Basic TSO commands: SEND, LISTBC, TIME, PROFILE, HELP, LOGOFF

Computer Exercise: A First Encounter With TSO

#### REXX - Restructured Extended Executor

Basic concepts and terms

Basic REXX Instructions: SAY, PULL

TSO EXEC command

Computer Exercise: A First EXEC

#### RC - Return Code special variable

Assignment Instruction

Built-in Functions

Numerics Issues

TSO/E Functions

The Data Stack

Computer Exercise: Assignments, Functions, and The Stack

### Day Two

#### REXX PARSEing Capabilities

Computer Exercise: Parsing

#### Clearing the Screen

TSO Commands: LISTCAT, LISTDS

More built-in functions

REXX Instructions: IF-THEN-ELSE, LEAVE, ITERATE, DO-Sequences, SELECT

Logic structures in REXX

Computer Exercise: The TSOUTIL EXEC

#### Debugging and TRACE

TSO Commands for working with files:

ALLOCATE, FREE, LISTALC, ALTLIB, RENAME, REPRO, DELETE, DEFINE CLUSTER

Computer Exercise: Creating Data

### Day Three

#### SMS - Storage Management Subsystem

More TSO Commands: ALTER, PRINTDS, PRINT, SMCOPY

Computer Exercise: Printing and Copying

#### LISTDSI TSO/E Function

Computer Exercise: The RENFILES EXEC

TSO/E EXECIO Command

Buffers and Stacks

Computer Exercise: The LISTEX EXEC

Compound Symbols and Stems

Computer Exercise: Compound Symbols

#### Day Four

Additional REXX Instructions and Functions

REXX Subroutines and User-Written Functions

Computer Exercise Subroutines

Error Handling and Condition Traps

More Built-in Functions

Trapping Output from a command

Computer Exercise: Strings and Words

#### Day Five

Arithmetic, Conversion, and Boolean Built-in Functions

Running programs in Foreground

TSO CALL Command

TSOLIB Command

Computer Exercise: Running a Program in the Foreground

Running jobs in the Background

TSO Commands: SUBMIT, STATUS, CANCEL, OUTPUT

Running EXECs in the batch

Computer Exercise: Running EXECs in the Batch

Host environments and the Dialog Manager

TSO TRANSMIT and RECEIVE Commands

LOG and NAMES datasets

[Level 2 REXX Components]

[The REXX compiler]