

TSO REXX Programming in z/OS

TSO REXX Programming in z/OS - Course Objectives

On successful completion of this class, 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 alternate REXX library 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 29

REXX - Restructured Extended Executor

Basic concepts and terms

Basic REXX Instructions: SAY, PULL

TSO EXEC command

Computer Exercise: A First EXEC 69

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 112

Day Two

REXX PARSEing Capabilities

Computer Exercise: Parsing 136

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 176

TSO REXX Programming in z/OS - Topical Outline, 2

Day, Two, continued

Debugging and TRACE

TSO Commands for working with files:

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

Computer Exercise: Creating Data 235

Day Three

SMS - Storage Management Subsystem

More TSO Commands: ALTER, PRINTDS, PRINT, SMCOPY

Computer Exercise: Printing and Copying 259

LISTDSI TSO/E Function

Computer Exercise: The RENFILES EXEC 267

TSO/E EXECIO Command

Buffers and Stacks

Computer Exercise: The LISTEX EXEC 287

Compound Symbols and Stems

Computer Exercise: Compound Symbols 302

Day Four

Additional REXX Instructions and Functions

REXX Subroutines and User-Written Functions

Computer Exercise: Subroutines 337

Error Handling and Condition Traps

More Built-in Functions

Trapping Output from a command

Computer Exercise: Strings and Words 369

TSO REXX Programming in z/OS - Topical Outline, 3

Day Five

Arithmetic, Conversion, and Boolean Built-in Functions

Running programs in Foreground

TSO CALL Command

TSOLIB - TSO Command

Computer Exercise: Running a Program in the Foreground 403

Running jobs in the Background

TSO Commands: SUBMIT, STATUS, CANCEL, OUTPUT

Running EXECs in the batch

Computer Exercise: Running EXECs in the Batch 422

Host environments and the Dialog Manager

TSO TRANSMIT and RECEIVE Commands

LOG and NAMES datasets

[Alternate REXX Library Components]

[The REXX Compiler]

Section Preview

Introduction to TSO

The Environment

TSO Command Syntax

Comments and Continuation

LOGON

The Terminal Monitor Program (TMP)

SEND, TIME, LISTBC, PROFILE, HELP

LOGOFF

Working with TSO (Machine Exercise)

TSO / ISPF / PDF

TSO - Time Sharing Option

A powerful but awkward-to-use facility that allows users at any kind of terminal to:

- Create, modify, delete, rename files
- Maintain libraries of programs, JCL, data
- Assemble, compile, link, run programs in the batch (background) or under immediate terminal control (foreground)
- Monitor status of batch jobs, examine output
- Communicate to operator or other users

ISPF - Interactive System Productivity Facility

An extension to TSO, for users of full-screen terminals, that vastly simplifies using TSO by providing:

- Support for creating and using screens (panels) to gather, present, and modify data
- Support for creating and displaying embedded HELP and tutorial information
- Support for using programmable function (PF) keys
- Interfaces to programming languages such as CLIST, REXX, COBOL, Assembler, and FORTRAN

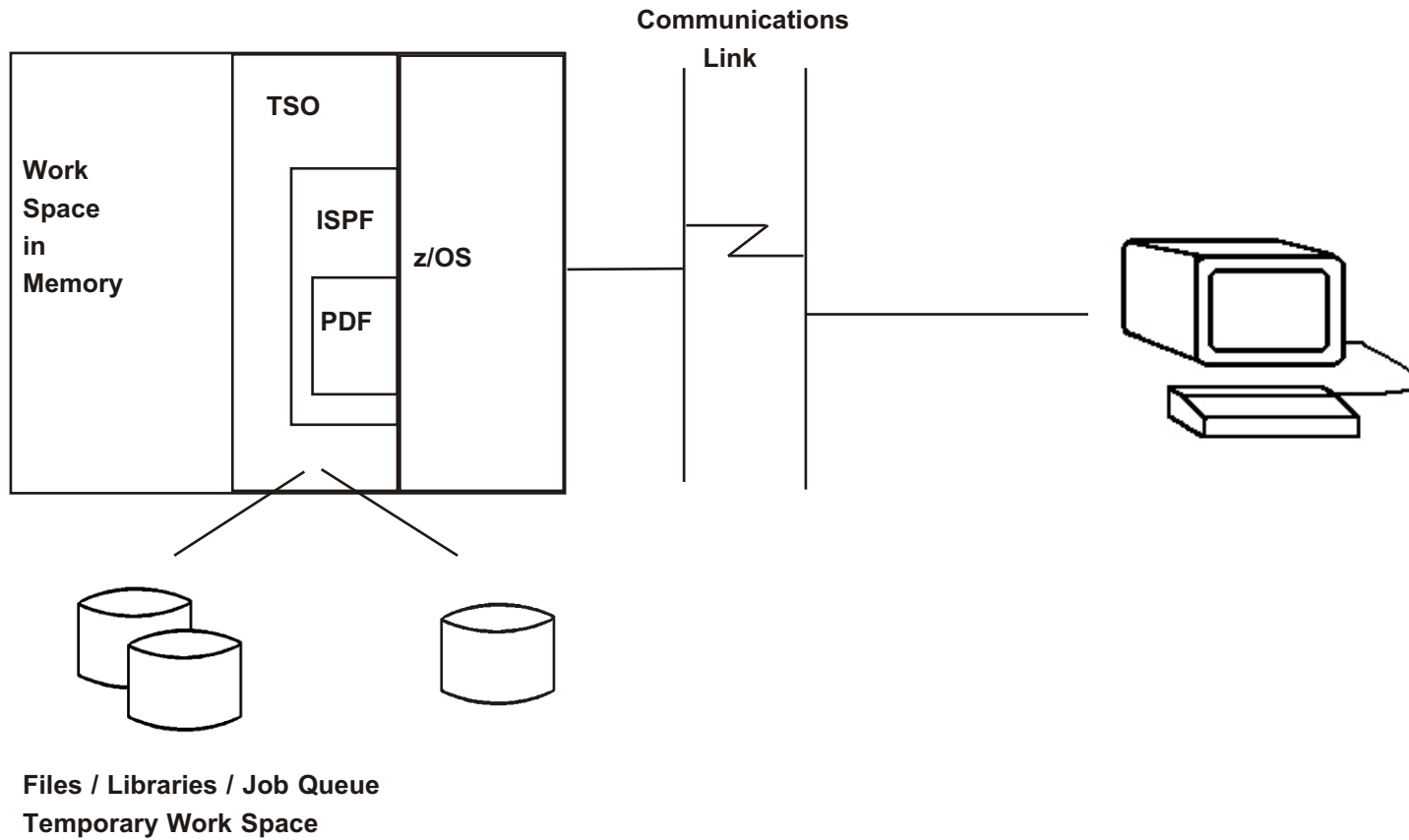
ISPF Is a Dialog Manager

PDF - Program Development Facility

A dialog that runs under ISPF that provides programmers assistance in using TSO through:

- Menu and fill-in-the-blanks approach
- Full screen editing and browsing of programs and data

The TSO / ISPF Environment



TSO Syntax

TSO Commands

COMMAND [OPERAND1 [OPERAND2 ... OPERANDn]]

Where Operands Are

Positional

Keyword, Two Types:

KEYWORD

KEYWORD(VALUE)

Both commands and operands may be abbreviated:

Each command may have one acceptable abbreviation

Keyword operands may be abbreviated to the fewest number of leading characters it takes to make the keyword unique among other parameters of the command, up to the entire word

Continuation

- ❑ If a command does not fit on a single line, use a continuation character to indicate the command is continued on the next line:

A '+' causes leading spaces to be removed from the continuation:

```
LISTDS (TRAIN.INPUTA.+  
BKUP) MEMBERS
```

is seen by TSO as:

```
LISTDS (TRAIN.INPUTA.BKUP) MEMBERS
```

A '-' causes leading spaces on the continuation statement to be preserved:

```
LISTDS (TRAIN.INPUTA.-  
BKUP) MEMBERS
```

is seen by TSO as:

```
LISTDS (TRAIN.INPUTA. BKUP) MEMBERS
```

(which is a syntax error, in this case)

- ❑ When TSO encounters a line ending in a continuation with no following line, TSO waits for you to enter a new line from the terminal

Comments

- Delimited by /* and */ :**

```
SEND 'Please mount tape "MICKEY"' /* Send message to operator. */
```

- To continue a comment, need a continuation character (+ or -) :**

```
FREE F(SYSIN SYSLIB SYSLIN) /* Free files -  
used in LINK step */
```

- Do not need to end a comment with */**

This is a convention, not a requirement

So, for example:

```
/*  
*****  
/* THIS OPEN BOX IS MADE UP OF COMMENTS  
/*  
*****  
*/
```

- If a comment is embedded in a command that continues, the continuation character must come after the comment:**

```
ALLOCATE DATASET(TRAIN.CLIST) /* Set up test library */ +  
FILE(SYSPROC) SHR REU
```

Data Set Naming

- A qualifier is a string of 1-8 alphanumeric or national (\$ # @) characters, the first of which is not numeric
- A data set name (DSNAME) consists of 1 or more qualifiers, separated by periods, up to a maximum of 44 characters

Examples:

MYFILE

SYS1.LINKLIB

\$TRNCM.TRAIN.LIBRARY

DEPT56.PAYROLL.EXTRACT.JANUARY.TEMP#1

- We use the term "level" to indicate where in a data set name a qualifier is, and you'll hear expressions like

High level qualifier (= leftmost qualifier)

Low level qualifier (= rightmost qualifier)

Fully-qualified data set name

TSO Data Set Naming Conventions

A Data set name consists of three qualifiers:

PREFIX.FILENAME.TYPE

PREFIX:

Default to current userid

FILENAME:

User choice

TYPE:

From list of standard choices, or user choice

TSO Standard Types List

<u>Type</u>	<u>Implied Data Set Contents</u>
ASM	Assembler source
CLIST	TSO commands / CLIST statements
CNTL	JCL and SYSIN for SUBMIT command
COBOL	COBOL source
DATA	Uppercase text
EXEC	REXX exec's
FORT	FORTRAN source
LINKLIST	Output listing from the Linkage Editor
LIST	Listings from assemblies / compiles
LOAD	Load modules
LOADLIST	Output from Loader program
OBJ	Object modules
OUTLIST	Listings from OUTPUT command
PASCAL	PASCAL source
PLI	PL/I source
TESTLIST	Listings from TEST command
TEXT	Uppercase and lowercase text
VSBASIC	BASIC source

Notices and Mail

Notices

Messages sent to all users

Sent by operator or users with operator authority

May only be deleted by same

Available at LOGON time, or through the LISTBC command

Mail

Messages sent to specific user(s)

May be sent by any user or operator

Deleted when user receives the message

Available at LOGON time, at time message sent, or through the LISTBC command

- Both stored in the BROADCAST data set (Note: mail may be stored in private user logs if installation desires)**

LOGON Parameters - I

USERID

Unique id for each TSO user

PASSWORD

To permit entry into the system

NEWPASSWORD

If current password has expired, need to supply a new one

- In z/OS 1.10 or later, you might have a password phrase instead of a password; RACF rules for password phrases:

14-100 characters from alphanumeric (case-sensitive), blank, national, punctuation, and special characters

At least 2 alphabetic characters and 2 non-alphabetic characters

Userid can not be included

No more than 2 consecutive characters can be identical

LOGON Parameters - II

ACCOUNT CODE

Optional; used for chargeback or accounting

One or more account codes may be associated with a password

PROCEDURE NAME

Optional; cataloged procedure to be used to start your TSO session

Defines what datasets to automatically allocate, any initial command to be executed on entry, and so on

One or more logon procedures may be associated with each account code

LOGON Parameters - III

REGION SIZE

Maximum allowed region for conditional GETMAIN requests

PERFORMANCE GROUP

Value from 1 - 999; identifies performance group this session to run under

GROUP NAME

A RACF group name; ignored if not using RACF

LOGON Command With Parameters

LOGON **userid[/password[/newpassword]]** **[ACCT(account)]**
[PROC(procedure)] **[SIZE(nnnnK)]** **[PERFORM(mm)]**
[[NO]RECONNECT] **[GROUP(name)]** **[OIDCARD]**
[NOTICES | NONOTICES] **[MAIL | NOMAIL]**

- RECONNECT** used when terminal connection was lost (power failure, say, or line problems) but the system still thinks the session is running

If you are RACF-defined, 'RECONNECT' attribute will be retained across sessions, unless you specify 'NORECONNECT'

'NORECONNECT' is used to override automatic reconnect for RACF-defined users who have previously specified 'RECONNECT' in some previous logon

- OIDCARD** used by operators with magnetically encoded card to place in special reader

LOGON Types

Line Mode:

logon userid [/password[/newpassword] ...

Full Screen Logon:

logon userid

Used if screen has 24 X 80 size or larger

Full Screen LOGON

----- TSO/E LOGON -----

Enter LOGON parameters below:

Userid ==> GRUMPY

Password ==> _

Procedure ==> TSOUSER

Acct Nmbr ==> (1350056,P09)

Size ==> 4096

Perform ==>

Command ==> CD F(TRAIN.EXEC,TRAIN.CLIST)

RACF LOGON parameters:

New Password ==>

Group Ident ==>

Enter an 'S' before each option desired below:

-Nomail

-Nonotice

-Reconnect

-OIDcard

PF1/PF13 ==> Help PF3/PF15 ==> Logoff PA1 ==> Attention PA2 ==> Reshow
You may request specific HELP information by entering a '?' in any entry field.

TMP

- A Terminal Monitor Program (TMP) is specified by the LOGON command (using the PROC parameter of the command) or a default supplied by the security package in use (RACF, ACF2, Top Secret, etc.)

- IBM supplies a standard TMP

But users may write their own

- The TMP is attached by the LOGON process

Prompts the user to enter a command
(IBM TMP: 'READY')

If the response is a command ...

- Attach the requested Command Processor (CP) which performs the requested function

TMP - Processing

- Handles error recovery
- Handles attention interrupts
- Accepts and interprets commands
- Schedules appropriate Command Processor(s)

SEND

Syntax

```
SEND 'msgtext'  {[USER(id [,id...])] [NOW| SAVE | LOGON]  
                | [OPERATOR(n)] [CN(mm)] }
```

- NOW** — Send message now; if user not logged on, forget it
- SAVE** — Save message until user logs on or issues 'LISTBC'
- LOGON** — Send message now if user logged on; otherwise, save until next time user logs on
- n** — Route code; default '2' (system console)
- mm** — Console id or console name (installation specified)

Up to 20 users may be specified (a userid of '*' will send the message to your own terminal)

Alternatively, send a message to operator or specific console

Examples

```
SEND 'Staff meeting now, rm 12' U(IS032,IS044,XY21)  
SE 'Please give me a call. -X334; Marc' U(RDVLY) logon  
se 'Please cancel job ISDRL05(J342). Thnx.' operator(2)  
se 'Don"t shut the system down yet! Thanks -SHC'
```


LISTBC

Syntax

LISTBC **[MAIL | NOMAIL]** **[NOTICES | NONOTICES]**

Results

Produces a list of the mail and / or notices data sets at your terminal

Examples

LISTBC

LISTBC NOMAIL

LISTBC NONOTICES

LISTBC NOTICES

- Here's an opportunity to demonstrate the use of parameter abbreviations:**

NONOTICES may be abbreviated as NONOTICE, NONOTIC, NONOTI, NONOT, NONO, or NON, but not as NO, since that could also mean NOMAIL

TIME

☐ **TIME** command returns:

Current time of day (hh:mm:ss)

Cumulative CPU time used since logon

Service units used

Cumulative session time (“wall time”) since logon

Current date

Syntax

TIME

PROFILE Command — TSO

```
PROFILE [ PROMPT | NOPROMPT ]  
        [ INTERCOM | NOINTERCOM ]  
        [ MSGID | NOMSGID ]  
        [ PREFIX (dsname-prefix) | NOPREFIX ]  
        [ VARSTORAGE({LOW|HIGH}) ]
```

```
PROFILE [ LIST ]
```

PROMPT — System should prompt for missing or invalid parameters

INTERCOM — Accept messages sent from other users

MSGID — Include message identifiers with any error messages

PREFIX — 1-7 A/N characters used for automatic high-level qualifier for data set names that are not fully-qualified (enclosed in single quotes); Default: *userid*

VARSTORAGE — HIGH allows CLIST variables or REXX OUTTRAP variables to reside above the 16MiB line (z/OS 1.8)

LIST — Display a list of profile attributes (this is the default if no parameters are coded on the PROFILE command)

Examples

```
PROFILE NOINTERCOM
```

```
PROFILE INTERCOM NOPREFIX
```

```
PROFILE PREFIX(PYRLL2)
```

HELP!

HELP

list of all commands that have entries in the HELP library

HELP *command-name*

all HELP information available about the named command

HELP *command-name*

[**FUNCTION** | **SYNTAX** | **OPERANDS** [(*list*)] | **POSITIONAL** (*nn*)]

part(s) of available HELP information you're interested in

HELP *command-name* **MSGID** (*list*)

explanation regarding specific error message(s) for specific command

only supported for following commands: MVSSERV, VSBASIC, TRANSMIT, RECEIVE

LOGOFF Command

Syntax

LOGOFF [DISCONNECT | HOLD]

HOLD is not supported for VTAM terminals

Computer Exercise: A First Encounter With TSO

Logon to TSO and perform the following tasks:

1. List your current profile settings; jot them down so you can examine them later.
2. Set your profile so no one can send you any messages.
3. Send the following message to yourself and at least one other student. Send the message so the student will get the message later even if they are not logged on now. The text of the message is:

I'm doin' OK so far

4. Issue a TIME command, and examine the results briefly.
5. Issue a HELP command. Explore some of the commands that have HELP information available. NOTE: If you get involved in a lengthy response, the ATTENTION key (often PA1) will get you out of it.
6. Reset your profile so you can receive messages.
7. Check if you have any mail pending.
8. Send another message to yourself and one other student:
The second time around from
9. Logoff TSO.

If you don't have a TSO id / password, the instructor will arrange for you to get one.