

# Chapter 1

# REXX: INTRODUCTION

*Get on the  
Fast Track!*



TM

**SYS-ED/  
Computer  
Education  
Techniques, Inc.**

## Objectives

You will learn:

- REXX syntax rules.
- REXX language featureset and components.
- REXX syntax.
- Instruction - types.
- How to execute a REXX procedure.
- ALTLIB command.

---

## **1 REXX - What is it**

REXX stands for REstructured eXtended eXecutor.

REXX:

- Creates command procedures.
- Is an implementation of the SAA: Systems Application Architecture.
- Provides a common programming structure.
- Includes character manipulation and arithmetic functions.
- Can be intermixed with host commands from TSO, CMS, CP, etc.
- Can run in a TSO/E or z/OS address space.

---

### **1.1 REXX Featureset**

The REXX featureset includes:

- English like commands.
- Free format commands.
- Built-in functions which provide character manipulation, arithmetic, searching, host interface and formatting capabilities.
- Debugging capabilities including tracing and an interactive debug facility.
- An interpreted language.
- Parsing capabilities.
- Interactive conversational processing.
- File processing.

---

## 1.2 Components

The components of the REXX language include:

- Instructions.
- Keywords.
- Assignment.
- Labels.
- Nulls.
- Host commands.
- REXX built-in functions.
- External system functions.
- Data stack functions.

---

## 2 REXX Syntax

REXX programs are stored in a sequential dataset or a PDS: partitioned dataset.

- The record format of the file can be fixed or variable.
- The file can be allocated with JCL: Job Control Language, the TSO ALLOC command, or ISPF - option 3.2.
- The dataset name type should be EXEC unless it is being combined with a CLIST library.

### Example:

This ALLOC command allocates an EXEC PDS:

```
ALLOC DA(sample.exec) NEW DIR(20)      -  
      SPACE(20,5) TRACKS                -  
      RECFM(V B) LRECL(255) BLKSIZE(0)
```

---

## 2.1 Quotes and Strings

Literal strings can be placed in quotes.

- Strings not enclosed in quotes are changed to upper case and all multiple spaces are changed to single spaces.
- Strings can be enclosed in single or double quotes; however, the quotes at each end must match.
- A string delimited by single quotes can include a constant double quote and a string delimited by double quotes can contain a single quote.
  - Two quotes together also can be used to display a quote.
- A null string is two quotes together: " .
- Hexadecimal literals also can be used.

### Examples:

```
SAY 'Dad, Can I Have The Car Tonight'
```

```
SAY "Dad, Can I Have The Car Tonight"
```

```
SAY Dad, Can I Have The Car Tonight
```

```
SAY 'Dad, Can I Have Mom"s Car Tonight'
```

```
SAY "Dad, Can I Have Mom's Car Tonight"
```

---

## 2.2 Continuation

A REXX statement can begin in any column.

- It is preferable to have one statement per line.
- When a semicolon separates the commands; more than one statement can be put on a line.
- Statements that extend beyond one line can be continued by ending it with a comma and continuing on the next line.

### Examples:

```
SAY "What did the clock say at 12 o'clock?" ; SAY 'Hands Up'
```

```
SAY "Why are phone bills ",  
    "so high in a haunted house?"
```

```
SAY "There are a lot of ghost to ghost calls"
```

```
SAY  'This line continues on next line.',  
    'Notice opening and closing quotes on both.'
```

---

### 3 Instruction - Types

The REXX language uses the following instruction - types:

|         |            |       |      |         |
|---------|------------|-------|------|---------|
| Keyword | Assignment | Label | Null | Command |
|---------|------------|-------|------|---------|

| Instruction Type | Explanation   |
|------------------|---|
| Keyword          | <p>Most REXX statements begin with a keyword which is a REXX command.</p> <p><b>Example:</b><br/>SAY, PULL, IF, DO etc.</p>   |
| Assignment       | <p>The assignment statement will perform a calculation and initialize a field to the results. It does not begin with a keyword.</p> <p><b>Example:</b><br/>total = total + amt</p>  |
| Label            | <p>The label is used for subroutine processing and error checking. The use of the GOTO command is discouraged. A "read_loop:" is an example of a label.</p>   |
| Null             | <p>A comment or a blank line is used to improve readability. Comments are a string enclosed in a /* at the beginning and a */ at the end. A comment can span multiple lines.</p> <p>All REXX procedures should have a comment on the first line which includes the word REXX. This is used by the system to distinguish between a CLIST and a REXX command procedure.</p> |
| Command          | <p>A command is a statement that is not a REXX keyword, assignment, null or a label. Variables and expressions can be placed into a command.</p>  |

---

## 4 Executing a REXX Procedure

A REXX procedure can be executed in two ways:

|                    |                    |
|--------------------|--------------------|
| Explicit Execution | Implicit Execution |
|--------------------|--------------------|

In order to execute an EXEC, the EXEC command is used with the EXEC keyword parameter.

- If the high level qualifier is same as the user's prefix, quotes are not needed.
- If the low level qualifier already is EXEC, the low level qualifier can default, and be retrieved through ARG statement.

Quotes are not used with the implicit execution.

Parameters being passed into the routine should be enclosed in single quotes.

Error messages usually will be displayed on screen.

---

### 4.1 Explicit Execution

In TSO/E, an exec can be called explicitly using the EXEC command followed by the dataset name and the "EXEC" keyword operand of the EXEC command.

The "EXEC" keyword operand distinguishes the REXX exec from a TSO/E CLIST, which also can be called with the EXEC command.

#### Examples:

```
EXEC 'ptc001.sample.exec(mypgm)' EXEC
```

```
EXEC sample.exec(mypgm) EXEC
```

```
EXEC sample(mypgm) EXEC
```

```
EXEC sample(mypgm) '12 34 67' EXEC
```

---

## 4.2 Implicit Execution

Implicit execution allows a REXX procedure to be run by entering in the member name.

The format of standard TSO commands can be emulated and this can make it difficult to distinguish between a TSO command and a REXX execution.

Prior to running an implicit procedure, a SYSPROC or SYSEXEC must be allocated with the EXEC libraries. The order of concatenation will be an important consideration.

- SYPROC can contain both CLISTs and REXX procedures.
  - The REXX programs must have a comment in the first line with the word REXX on it.
- SYSEXEC can only contain REXX programs.

SYSEXEC is searched prior to SYSPROC; this command may be need to be executed prior to the SYSEXEC search:

```
EXECUTIL SEARCHDD (YES)
```

### Examples:

```
mypgm
```

```
mypgm 12 34 67
```

---

### 4.3 Example - Allocation Procedures

```
/* ALLOCATING SYSEXEC */

"EXECUTIL SEARCHDD (YES) "

"ALLOC FI (SYSEXEC) DATASET (MYSTUFF.EXEC) SHR REUSE"

IF RC = 0 THEN SAY "SUCCESS ... WE GOT LIFTOFF"
    ELSE SAY "ALLOCATION FAILED"

EXIT

/* ALLOCATING SYSPROC */

"ALLOC FI (SYSPROC) DATASET (MYSTUFF.exec",
    'SYS1.CLIST', ",
    'SYS1.CMDCLIST') SHR REUSE"

IF RC = 0 THEN SAY "GOOD GOING"
ELSE SAY "GET GOING"
```

---

## 4.4 ALTLIB Command

In a majority of situations, the ALTLIB command will be preferred to the EXECUTIL command.

ALTLIB affects how EXECUTIL operates in determining the search order.

The ALTLIB command is used to:

- Define alternative application-level libraries of REXX execs or CLISTs.
- Indicate that user-, application-, and system-level libraries of REXX execs and CLISTs are being searched.
- Exclude one or more library levels from being searched.
- Reset the search order to the system level.
- Obtain a display of the search order that is in effect.

The ALTLIB command operates on an application basis and is used for indicating that user-level, application-level, or system-level libraries are to be searched.

The ALTLIB command syntax is:

```
ALTLIB ACT APPL (EXEC) DS (dsname)
```

| Keyword             | Explanation  |
|---------------------|--|
| ACTIVATE or ACT     | Indicates that a specified library level is to be included when searching for a REXX exec or CLIST.                      |
| APPLICATION or APPL | Indicates that the application-level REXX execs and CLISTs are to be activated or deactivated.                           |
| DATASET(dsname)     | The DSNAM(dsname) or DS(dsname) specifies a dataset list to define an application-level library of REXX execs or CLISTs. |