



BMC Mainframe: REXX Programming under TSO/ISPF

COURSE ABSTRACT

COURSE CODE

» MGRS-RPTI-2021

APPLICABLE VERSIONS

» Not Applicable

DELIVERY METHOD

» Instructor-led Training (ILT)

COURSE DURATION

» 3 Days

PREREQUISITES

- » An understanding of programming at a conceptual level and a good knowledge of TSO and its commands.
- » A superficial knowledge of ISPF/PDF is insufficient for those who wish to gain full value from this course.

RECOMMENDED TRAININGS

» NA

Course Overview

The course is developed and delivered by © RSM Technology.

REXX is a powerful multi-purpose and user-friendly programming tool, used to exploit many IBM and third-party operational products. It is the tool of choice in most (if not all) of today's larger data centers for those who need to implement new operational procedures both quickly and effectively.

This three-day course is designed for TSO/ISPF users who wish to learn how to create and use programs in the REXX language. The course is taught combining formal classroom teaching with many practical exercises and problem scenarios, thus ensuring a full understanding of this powerful procedural language.

Target Audience

Systems programmers, operations analysts, senior operators, operations support staff and application programmers who will be writing and/or maintaining TSO REXX programs.

Learner Objectives

- » Describe the concepts and structure of the REXX language environment
- » Write and debug REXX programs executing in the TSO environment
- » Use the tracing and debugging aids effectively
- » Use the loop control and decision-making instructions
- » Identify and put into practice the most commonly used REXX built in functions
- » Parse data strings
- » Use TSO commands and functions within REXX EXECs
- » Manage I/O using data stacks and stems
- » Use sub-routines and functions



BMC Mainframe: REXX Programming under TSO/ISPF

COURSE ABSTRACT

COURSE ACTIVITIES

- » Classroom Presentations
- » Demonstration


BMC MAINFRAME INFRASTRUCTURE AND PLATFORMS LEARNING PATH

- » <https://www.bmc.com/education/courses/find-courses.html#filter/%7B%22type%22%3A%22edu-specific-types-159150236%22%7D>

CERTIFICATION PATHS

- » This course is not part of a BMC Certification Path.

DISCOUNT OPTIONS

- » Have multiple students? Contact us to discuss hosting a private class for your organization
- » [Contact us for additional information](#) 

Course Modules

Introduction to REXX

- » Executing REXX EXECs
- » elements of the REXX language: Clauses, Tokens, Labels, Assignments, Instructions and Commands
- » constant and variable symbols
- » Simple, Compound & Stem variable symbols
- » operators for concatenation, arithmetic, comparison & logical operations. *This segment introduces the terminology associated with REXX and shows how data fields are defined and initialized*

Tracing and Debugging

- » Using TRACE instruction for debugging
- » using immediate commands HI, HT, TS and TE
- » use of EXECUTIL
- » interpreting TRACE output. *This segment describes the tracing facilities available in REXX to enable more rapid problem determination*

Parsing Strings

- » Parsing variables and strings
- » basic parsing techniques

- » special templates
- » place holders, literal patterns, and numeric patterns.
- » Explains how input from any source can be split into useable entities using various techniques

Control Instructions

- » Decisions: IF, THEN, ELSE, SELECT, WHEN, OTHERWISE
- » creating and controlling loops: DO WHILE, UNTIL, FOREVER, LEAVE, ITERATE
- » introduction to sub-routines.
- » This segment describes the IF, THEN, ELSE logic used in REXX

Sub-routines and Functions

- » Internal and external sub-routines
- » how to create internal and external functions
- » passing arguments and results to and from sub-routines and functions
- » CALL
- » RETURN
- » EXIT
- » PROCEDURE
- » RESULT.

- » This segment shows how to create subroutines and functions, and identifies the difference between them

REXX Built-in Functions

- » Introduction to REXX supplied built in functions
- » how to use the most useful ones: ARG, DATE, TIME, DATATYPE, LENGTH, POS, WORDS, LEFT, RIGHT, STRIP, SPACE, COPIES and WORD.
- » This segment explains the commonly used REXX built-in functions, and indicates their use in REXX programs

Data Stack Management

- » LIFO and FIFO stacking
- » avoiding the stack
- » writing to the stack: PUSH and QUEUE
- » reading from the stack: PARSE PULL
- » creating and managing extensions to the stack: MAKEBUF, QBUF and DROPBUF
- » interrogating the stack: QUEUE, QELEM
- » creating and managing private stacks: NEWSTACK, DELSTACK and QSTACK.
- » This segment describes the stack mechanisms used by TSO and REXX

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners. ©2021 BMC Software, Inc. All rights reserved.



BMC Mainframe: REXX Programming under TSO/ISPF

COURSE ABSTRACT

Using TSO Commands & Functions

- » Using TSO commands and functions: SYSDSN, LISTDSI, MSG, OUTTRAP, PROMPT and SYSVAR
- » reading and writing datasets with EXECIO.
- » Shows how to invoke and use TSO commands from a REXX program