

# BMC AMI Ops Automation 8.x: Fundamentals

## Managing REXX EXECs (WBT)

[Learning Path >](#)

Course Code: AOAU-RXEM-F803

| Modality                 | Duration | Applicable Versions  | Target Audience  |
|--------------------------|----------|--|--|
| Web-based Training (WBT) | 4 Hours  | BMC AMI Ops Automation 6.1, 6.2, 6.3, 8.1, 8.2, 8.3, and 8.3.1 | <ul style="list-style-type: none"><li>Programmers (System Programmers/Application Programmers)</li></ul> |

### Course Overview

BMC AMI Ops Automation (BMC AMI OpsA) product is part of the BMC AMI Ops Automation for Data Centers solution. This product provides tools and applications to improve an operator's ability to provide high system availability and helps the operator through error minimization, improved productivity, and problem determination and prevention.

BMC AMI Ops Automation provides two mechanisms to automate workflow, Rules and EXECs.

This course will help System Programmers to understand the basics of REXX EXECs like datatypes, statements, instructions, and functions. This course will also introduce them to EXEC-related views in BMC AMI Ops Automation.

**Note:** BMC AMI Ops Automation was formerly known as MainView AutoOPERATOR™.

### Prerequisites

- NA

### Recommended Trainings

- BMC AMI Ops Automation 8.x: Fundamentals Managing

### Learning Objectives

- Understand the concept of EXECs
- Identify when to use an EXEC and when to use a Rule
- List actions an EXEC can perform
- Describe EXEC life cycle events, such as creation, invocation, execution, and data exchange
- Understand REXX programming basics
- Use REXX built-in functions
- List EXEC-related views available in the product in all interfaces. (Full-screen mode/ Windows mode/ MVE)

# Course Modules

## Module 1: Automation and EXECs

- Overview of BMC AMI Ops Automation
- Rule Organization
- Advantages of Rule-based Automation
- Need for EXEC-based Automation
- Key Features of EXEC-based Automation
- EXEC Actions
- Notifications: Messages
- Notifications: Alarms
- Notifications: Alerts
- Commands Overview
- Transactions: IMS
- Transactions: CICS
- Transactions: MQ
- Defining REXX EXEC
- Creating Hello World EXEC
- Running REXX EXEC
- Running REXX EXEC: EX Command
- REXX Programs: Example 1
- REXX Programs: Example 2

## Module 2: EXEC Life Stages

- EXEC Life Stages
- Scheduling an EXEC
- Scheduling an EXEC: Rule
- Scheduling an EXEC: Alerts
- Scheduling an EXEC: User
- Scheduling an EXEC: Time
- New CALLX Request: Block of Requests
- New CALLX Request: Individual Request
- Scheduling an EXEC: From Another EXEC
- Scheduling an EXEC: Externally
- Scheduling an EXEC: From TOM
- Executing an EXEC
- Using ISPF Panels
- Using BMC AMI OpsA Views
- Executing an EXEC as a Batch Job
- Sharing Data with EXEC
- Command Line Parameters
- ARG Statement
- Variables
- Variables: Example

## Module 3: REXX Fundamentals

- REXX Introduction
- REXX Program
- Documentation
- Terminal Input Output
- Data Types
- Variables
- REXX Operators and Expressions
- REXX Operators and Expressions: Arithmetic
- REXX Operators and Expressions: Comparison

- REXX Operators and Expressions: Logical or Boolean
- REXX Operators and Expressions: Concatenation
- Instructions
- Instructions: Conditional If-Then-Else
- Instructions: Conditional Select
- Instructions: Looping
- REXX Loop Constructs
- Instructions: Interrupt
- The EXIT Keyword
- The SIGNAL Keyword
- The CALL / RETURN Keywords

## Module 4: REXX Functions

- Built-in function Types
- REXX String Functions
- REXX String Function: LENGTH
- REXX String Function: SUBSTR
- REXX String Function: COMPARE
- REXX String Function: DELWORD
- REXX String Function: INDEX
- REXX String Function: POS
- REXX String Function: STRIP
- REXX String Function: WORD
- Conversion Functions
- REXX Conversion Function: C2D
- REXX Conversion Function: D2C
- REXX Conversion Function: D2X
- REXX Conversion Function: X2D
- Arithmetic Functions
- REXX Conversion Function: ABS
- REXX Conversion Function: RANDOM
- REXX Conversion Function: TRUNC
- Comparison Functions
- REXX Function: COMPARE
- REXX Function: DATATYPE
- Inferring Datatype
- Formatting Functions
- REXX Function: FORMAT
- REXX Function: LEFT
- Miscellaneous Functions
- REXX Function: ARG
- REXX Function: DATE
- REXX Function: TIME
- REXX Function: USERID

## Module 5: BMC AMI OpsA REXX EXECs

- Overview of BMC-supplied EXECs
- QAOSMTP1: SMTP Server Setting
- Sending an Email
- Calling Email EXEC From a Rule
- REST API
- QAOCRA1
- Distributed Utility EXECs
- REXX EXEC Conventions
- Expressions and Operators in REXX EXECs
- TSO/E REXX MVS Environment Commands
- Unsupported TSO/E Commands
- Other Restrictions in REXX EXECs
- Tokenized REXX EXECs Considerations
- Minimizing EXEC Processing Time
- Best Practices
- Best Practices: Naming Conventions
- Best Practices: Documentation
- Best Practices: Diagnostic Information
- Introduction to BMC AMI Ops EXEC APIs
- IMFEXEC
- AOEXEC
- MV/MVX

## Module 4: BMC AMI Ops Automation EXEC Views

- Navigating EXEC views
- EXEC Management Application
- EXEC Management View
- Line Commands
- Line Commands: Browse
- Line Commands: Disable
- Line Commands: Enable
- Line Commands: Execute EXEC
- Line Commands: Test EXEC
- EXEC Management View: Commands
- Expand Command
- Time initiated EXECs
- Windows Mode Views: Access
- EXEC Manager View
- Using the Expand Command
- List of All EXECs View
- Viewing the Execution Log
- AutoRptr EXEC History
- EXEC Message Log
- Navigating MVE Views
- EXECMGR
- AREXECs
- EXEC Command Views
- ARXCMDG

### Discount Options

**Have multiple students?** Contact us to discuss hosting a private class for your organization.

Contact us for additional information 