

BMC AMI DevX Code Debug for TSO 17: Extended Facilities Using ISPF (WBT)



Learning Path >

Course Code: CPCD-EFIS-U1700

Modality	Duration	Applicable Version(s)	Target Audience
Web-based Training (WBT)	3 Hours	BMC AMI DevX Code Debug for TSO 17.02	Application Developers

Course Overview

BMC AMI DevX Code Debug for TSO is a source-level testing and debugging solution that helps developers test, diagnose, and resolve issues in mainframe applications. It supports interactive and batch debugging across multiple languages and environments, including COBOL, Assembler, PL/I, IMS, and Db2.

The course covers extended debugging capabilities, including automating workflows using Code Debug TSO scripts and performing subroutine debugging with the INTERCEPT facility. It explains how to apply debugging techniques in Db2 and IMS environments, including handling execution and binding scenarios. Additionally, the course introduces sourceless debugging methods and demonstrates how to manage file allocations effectively using the File Allocation Utility (FAU).

In addition to English, the textual content of this course is available in Chinese (Simplified), French, German, Italian, Japanese, Korean, Portuguese (Brazil), and Spanish (Mexico).

Prerequisites

- BMC AMI DevX Code Debug 17: Concepts (WBT)
- BMC AMI DevX Code Debug for TSO 17: Getting Started Using ISPF (WBT)

Recommended Knowledge/Skills

- NA

Learning Objectives

- Automate debugging workflows using Code Debug TSO scripts
- Debug subroutine interactions using the INTERCEPT facility
- Apply Code Debug in Db2 and IMS environments
- Identify scenarios that require Db2 plan or package binding
- Handle abends and detach scenarios in Code Debug IMS
- Analyze sourceless programs using pseudo-assembler support and memory inspection commands
- Manage file allocations required for debugging sessions using the File Allocation Utility (FAU)

Course Modules

Module 1: Code Debug TSO Scripts

- Describe the purpose and value of scripts in Code Debug TSO
- Identify script data sets and script libraries used by Code Debug TSO
- Differentiate initial, test, post, and abend script categories
- Demonstrate how to use scripts to initialize, control, and automate debugging sessions
- Execute test scripts and reuse them across debugging scenarios
- Apply post-script and abend-script processing to complete debugging workflows
- Explain script creation guidelines and restrictions
- Analyze practical use cases for scripts in debugging workflows
- Create, edit, and run scripts in Code Debug TSO

Module 2: Performing Subroutine Debugging

- Describe subroutine debugging in BMC AMI DevX Code Debug for TSO
- Demonstrate how to debug a calling program without intercepting the subroutine
- Trace program execution flow during subroutine calls
- Explain the purpose and capabilities of the INTERCEPT facility
- Execute the INTERCEPT command to debug subroutines
- Demonstrate subroutine execution and tracing using the GO and TRACE commands
- Monitor execution flow within a subroutine
- Return execution control from the subroutine to the calling program

Module 3: Using Code Debug TSO in Db2 and IMS Environments

- Describe how Code Debug TSO is used in Db2 and IMS environments
- Explain Db2 plan and package binding requirements for debugging
- Identify scenarios that require Db2 plan or package binding
- Demonstrate how to use the Code Debug TSO Bind Facility
- Demonstrate how to debug Db2 stored procedures with Code Debug TSO
- Identify IMS program types supported (MPP, BMP, IFP)
- Explain the IMS execution model and environment requirements
- Apply techniques to handle abends and forced termination in Code Debug IMS
- Demonstrate how to use the Detach Region Facility for IMS debugging
- Navigate ISPF panels to access and use Code Debug IMS options

Module 4: Debugging a Sourceless Program

- Explain sourceless debugging concepts and interpret system messages
- Demonstrate how to debug sourceless COBOL programs using BMC AMI DevX Code Debug for TSO
- Access and debug sourceless main programs and subprograms
- Apply traditional sourceless debugging techniques and perform dump-format analysis
- Execute key sourceless debugging commands: MEMORY, GPREGS, AT, SHOW AT, and DELETE AT
- Create and use pseudo-assembler source for COBOL programs
- Use the PSEUDO / PSEUDOSOURCE command for interactive debugging

Module 5: File Allocation Utility (FAU)

- Describe the File Allocation Utility (FAU) and its role in Code Debug TSO
- Create and edit file allocation lists using FAU
- Copy file lists from existing JCL, CLISTS, or other allocation lists
- Execute key FAU commands: COPY, SAVE, END, and ALLOCATE
- Navigate and use the EDIT FILE LIST panels
- Use the File PARMS MENU to view and manage detailed file information
- Allocate and manage different dataset types

Course Assessment

Discount Options

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

Contact us for additional information 