COBOL/MVS Differences

Keane
enabling transformation

Course No.

2250

Description

This 1-day course provides training for personnel moving from the OS/VS COBOL language compiler to the COBOL/MVS environment. The student will gain an understanding of the COBOL/MVS product and the expanded capabilities available to the COBOL application developer.

Audience

This course is suggested for personnel moving OS/VS COBOL language compiler to the COBOL/MVS environment.

Prerequisites

Students should have appropriate college coursework or 6 months experience with COBOL.

Objectives

- Upon Completion of this course students will be able to:
- Describe and list the features of COBOL/MVS
- Explain the productivity features of COBOL/MVS
- Understand what Virtual Storage Constraint Relief is and which programs are executed above the 16M line and which are not
- Describe features no longer available and discuss alternatives
- Use IBM PROCs to compile, link and execute COBOL/MVS programs
- Explain the error types generated from the COBOL/MVS compiler
- Understand addressing mode (AMODE) and resident mode (RMODE)
- Use lowercase letters in a COBOL program
- Use INITIALIZE to prevent data exceptions
- Code special and symbolic characters within a program
- Manipulate character strings using reference modification
- Code nested programs
- Understand how the "ALL" subscript is used with intrinsic functions
- Delimit statements using scope terminators
- Use new features of the COBOL language

Major Topics

- Changes/Differences from OS/VS COBOL to COBOL/MVS
- Migration strategies and tools
- New language features
- Performance features
- Intrinsic functions
- Structured Programming enhancements
- Preparing COBOL/MVS programs
- Default settings
- Compile and link options
- Amode/Rmode/Data options
- Interactive debugging

Duration

1 day

Course Contents

1. **COBOL/MVS Overview**
   - COBOL Evolution (ANSI 1968, 1974, 1985)
   - Systems Application Architecture (SAA)
   - VS COBOL II, COBOL/370, COBOL/MVS
   - Programming Features
   - Productivity Features
   - Performance Features
   - Virtual Storage Constraint Relief
   - AMODE and RMODE
   - Performance Limitations
   - Migration Actions (Inventory, Prioritization, Education)
   - Summary
2. **Eliminated Features**
   - Causes of Change
   - Migration Impediments
   - Migration Strategy
   - Identification Division Changes
   - Environment Division Changes
   - Data Division Changes
   - Procedure Division Changes
   - COBOL/MVS Registers
3. **Preparing COBOL/MVS Programs**
   - Compiling COBOL/MVS Programs
   - COBOL/MVS Error Types
   - Compiler Options
   - DATA(24) and DATA(31)
   - FASTSRT
   - FDUMP
   - FLAG
   - FLAGSAA
   - SSRANGE
   - Reentrant Code (RENT)
   - Number Processing (NUMPROC)
   - Compiler Files, PROC and Run JCL
   - Compile and Link Files, PROC and Run JCL
   - Static and Dynamic Compile and Link
   - LOAD Module Run JCL
   - Compile, Link and Execute
   - Sample COBOL/MVS Listing
   - Sample Cross Reference Listing
   - Data Division Map Listing
   - Diagnostic Listing
4. **New Features**
   - Lower Case Letters
   - Nonnumeric Literal Increased Length
   - BINARY Usage Type
   - PACKED-DECIMAL Usage Type
   - POINTER Usage Types
   - Nested COPY
   - TITLE
   - *CBL
   - INITIALIZE
   - Complex OCCURS DEPENDING ON
   - LENGTH Register
   - DAY-OF-WEEK Register
   - Julian DAY Register
   - Hexadecimal Representation of Special Characters
   - Symbolic Characters
   - Reference Modification
   - De-Editing Numeric Fields
   - Nested Programs
   - Nesting Rules
   - ADDRESS OF Register
   - GLOBAL and EXTERNAL Data Elements
5. **Intrinsic Functions**
   - Definition
   - Examples
   - Summary of all Intrinsic Functions
   - Calendar Functions
   - Character and String Functions
   - Numerical Analysis Functions
6. **Structured Programming Enhancements**
   - The Importance of Structured COBOL
   - (Sequence, Selection and Iteration, Modularity, One entry, One exit, etc.)
   - CONTINUE
   - Scope Terminators
   - Statements With Scope Terminators
   - Scope Terminator Syntax
   - SET TO TRUE
   - New NOT Clauses
   - Inline PERFORM
   - PERFORM UNTIL WITH TEST BEFORE
   - PERFORM UNTIL WITH TEST AFTER

- EVALUATE
- CALL Extensions
- CALL Examples

7. Operational Differences

- Table Handling Enhancements
- Optional FILLER
- Alphabetic Changes
- Changes to COBOL/MVS Sort
- Two New Relational Operators
- ADD Processing
- ON SIZE ERROR Processing
- Determining Variable Record Length
- INSPECT CONVERTING
- REPLACE Statement
- EXIT
- Migrational Differences
- Number Processing Differences (NUMPROC)
- Binary Efficiency (TRUNC)

Appendices

A. IBM COBOL/MVS References

B. COBOL/MVS Publications

C. CICS® Considerations

D. The New, Improved "READY TRACE"

E. VSAM File Status and Feedback Codes