

# CICS/TS ADVANCED COMMAND LEVEL PROGRAMMING

## DESCRIPTION:

An advanced training class designed to take the experienced CICS application programmer to the final skill level in terms of the available Application Programming Interface (API).

### DURATION:

Four (4) Days.

### COURSE OBJECTIVES:

Through a combination of lecture, review exercises and workshops, the student will master many of the more advanced provisions of CICS command level Application Programming Interface. Upon completion of the course, the student will have:

- an expanded knowledge of BMS, File Control and Program Control
- an understanding of some of the more advanced aspects of the CICS command level API including facilities in Storage Control, Temporary Storage, Transient Data, Interval Control, SPOOL interface and Task Control
- an expanded appreciation of the more critical design and performance issues involved in application development

### AUDIENCE:

The experienced CICS Command Level Application Programmer.

### PREREQUISITES:

Familiarity with COBOL. Some knowledge of MVS Job Control Language. Substantial experience coding, compiling and link editing programs in general. A minimum of 6 months experience coding and testing CICS application programs. Experience using the CICS Execution Diagnostic Facility (an advantage, not a requirement)

## TOPICS OUTLINE:

### I. Overview

This module discusses various type of transactions, their coding particularities, and programming for performance. It also provides an overview of the product's functional areas namely transaction processing, inter-system communication, front-end programming interface, system programming interface and external CICS interface. By the end of this module, the participant will have acquired an awareness of the product's capabilities and will be in a better position to decide how a particular business problem is to be solved.

### II. Storage Control

This topic discusses various ways in which an application programmer can control and manage storage in a CICS application, how to obtain and access storage, COBOL Linkage Section issues. The lab on Storage Control will help understand differences between the "locate" and "move" modes in CICS; it will also provide the opportunity to exercise the GETMAIN command.

### III. Browsing Files

This module reviews the facilities provided by File Control Program. Browsing commands are discussed along with various advanced programming techniques. Updating files while browsing and also updating files with integrity are also covered. The lab is about developing a lock mechanism to protect a record being updated across a pseudo-conversational transaction.

### IV. Paging Transactions

Temporary Storage facilities and concepts are the main focus of discussion. Revision of techniques used in coding paging transactions that are efficient and easy to use. The lab provides an opportunity to experiment with browsing and terminal paging.

### V. Online Printing

This module describes the CICS/JES interface. It also discusses older methods for printing reports in CICS, namely Transient Data and Automatic task initiation. The exercise combines both mechanisms.

### VI. Field-level Help

This module addresses programming techniques used to make programs sensitive to the position of the cursor on the screen. It discusses BMS functions making this task possible and the exercise consist of a program providing field-level help.

## **CICS/TS ADVANCED COMMAND LEVEL PROGRAMMING**

---

### ***VII. Accessing System Information***

This module discusses the ASSIGN command and the values it can return. It also discusses some of the CICS commands used to manage resources and the use of CICS data values (CVDA). The exercise will give the programmer the opportunity to use both approaches.

### ***VIII. Recovery and Abend Handling***

Knowing the status of the data after a transaction or system ABEND is vital in the success of an application since it will influence the design of the program. Concepts of Logical Unit of Work, distributed UOW and remote UOW are explained. How does CICS recover the data? If the programmer has chosen to handle the ABEND, what should be done in order to maintain the integrity of the data.