

CICS Application Programming Workshop

Abstract

This workshop will give the student a comprehensive immersion into CICS program development. It utilises the 'pseudo-conversational' design model and teaches the implementation of separating the 'Business Logic' from the 'Presentation Logic' hence the students will be expected to write two programs.

Prerequisites

Prior attendance on the CICS Appreciation class is **NOT** required as appropriate introductory material is covered on the first day of this class. However, a working knowledge of TSO/ISPF and the ability to write in COBOL, PL/1 or Assembler is required.

Audience

Application designers/programmers who feel confident coding.

Prerequisites for on site courses.

As above plus:-

- A working CICS region which all student can access
- Procedure to Translate, Compile and Link BMS maps and the students' CICS programs
- A technical contact who will receive the Setup CD-Rom at least one week prior to the start of the course. The technical contact should be able to run all of the setup jobs (supplied) and make the appropriate resource definitions to the CICS region that will be used for the class.
- The technical contact should feedback to the lecturer at the start of the course any changes that had to be made due to installation standards.
- If the students will not be able to use their own TSO Ids for some reason, then the provision of enough 'training' TSO Ids and passwords will be required.
- Access to a printer would also be advantageous during the class.

Objectives.

At the end of the course the student should have a good understanding of the working of CICS and be able to:

- Demonstrate a considered approach to design and coding issues
- Write robust CICS code and be practised at diagnosing problems which occur during test.
- Suggest improvements to existing interfaces and file layouts
- Commence working on a real life project

Topics covered in this course include:

- Introduction to CICS concepts such as Transactions/Tasks; Multi-tasking/Multi-threading; CICS Dispatching; Resource Management and Intercommunication.
- Application Design: Comparison of Pseudo-Conversational, Conversational, Non-Conversational and Business Transaction Models.

- EXEC Interface: Command Format; Translation (where applicable); Command Execution; Condition handling and the EXEC Interface Block.
- Program Management: LINK vs XCTL vs Dynamic Calls; Commareas and passing data to the next Task or another Program in the same Task; ABEND, LOAD and RETURN commands and introduction to 'Affinities'
- Terminal Services: Native Terminal Control and Basic Mapping Support functionality; 3270 specific issues; 3270 Bridge; CWS API commands.
- Accessing Files: Majoring on VSAM both RLS and non-RLS; Updating and Timestamping; Locking scope and duration; avoiding Deadlocks; all API commands explored and many used in the workshop; Data Tables and CFDTs.
- Transient Data: What is it used for? Intra/Extra/Indirect/Remote functionality; recovery and Automatic Transaction Initiation (ATI).
- Temporary Storage: Likely usage; API calls; Main vs Auxiliary issues; REWRITE capability and the Read Cursor.
- Interval Control: Entire API explored; uses and abuses of some commands; ATI revisited with STARTs.
- Storage Management: GETMAIN and FREEMAIN; eight possible DSAs; SOS; Storage Violations; Subsystem Storage Protection and Transaction Isolation.
- Task Management: CPU hungry Tasks; serialization using (Global) Enqueue/Dequeue; Logical Units of Work and Syncpointing; Assigning State information about a task.
- Debugging: A walkthrough of typical Transaction dump solving techniques including solving an ASRA; HLLSAs and LE Blocks with TRAP=ON; use of CEDX/CEDF and DTCN Debug Tool.

Course Duration

5 days