CICS Introduction and Overview

Ezriel Gross
Circle Software

November 4, 2015
Session AJ
Agenda

- What is CICS and Who Uses It
- Pseudo Conversational Programming
- CICS Application Services
- CICS Connectivity
- CICS Resource Definitions
- CICS Supplied Transactions
- CICS Web Services
CICS as a product

50,000 CICS Licenses

30 Million Users

950,000 Programmers

490 of Top 500 Companies

200 Billion Lines of COBOL

Installed on 85% of all IBM Mainframes

30 Billion Transactions/Day

Replacement costs estimated at $20 Trillion

Session feedback – Do it online at conferences.gse.org.uk/2015/feedback/AJ
What is CICS?

- CICS is an online transaction processing system.
- Middleware between the operating system and business applications.
- Manages the user interface.
- Retrieves and modifies data.
- Handles the communication.
CICS customers

• **Banks**
  – Mortgage
  – Account Reconciliations
  – Payroll

• **Brokerage Houses**
  – Stock Trading
  – Trade Clearing
  – Human Resources

• **Insurance Companies**
  – Policy Administration
  – Accounts Receivables
  – Claims Processing
Batch versus online programs

• The two ways to process input are **batch** and **online**.

• **Batch**
  – Batch requests are saved then processed *sequentially*.
  – After all requests are processed the results are transmitted.
  – Used for order entry processing such as warehouse applications.

• **Online**
  – Online requests are received randomly and processed *immediately*.
  – Results are transmitted as soon as they are available.
  – Response time tends to be sub-second.
  – Used for applications such as credit card authorization.
Transaction processing requirements

- Large volume of business transactions to be rapidly and accurately processed
- Multiple users, single/sysplex or distributed
- With potentially:
  - A huge number of users
  - Simultaneous access to data
  - A large volume of data residing in multiple database types
- Intense security and data integrity controls necessary
- The access to the data is such that:
  - Each user has the perception of being the sole user of the system
  - A set of changes is guaranteed to be logically consistent
  - If a failure occurs, any intermediate results are undone before the system becomes available again
  - A completed set of changes is immediately visible to other users
A business transaction

- A *transaction* is a sequence of related operations that performs a function

- It might perform a single action
  - Example: Retrieve an account balance

- It can also perform a set of operations:
  - Read credit limits
  - Check if amount of purchase is greater than limit
  - Subtract funds or deny purchases

- A transaction has a 4-character id
CICS tasks and programs

- A **task** is an instance of a transaction entered by a user.
- When a user types in data and presses the Enter or a function key, CICS begins a task and loads the necessary programs.
- Tasks run concurrently. Therefore, a user can run the same transaction simultaneously.
- CICS uses multitasking to provide fast response times.
- Programs can be loaded once and then shared by transactions.
- CICS runs each task individually, briefly giving CPU to each one.
- If a user updates a file or database, the change is immediately available.
Conversational vs pseudo-conversational

• *Conversational* programs…
  – Run and *stay in memory* for the duration of the transaction.
  – All resources are *held/locked* for this duration.
  – If a user went to lunch in the middle of a conversational transaction, other users *may have to wait!*

• *Pseudo-conversational* programs…
  – Overcome this by *terminating* when the first response is produced.
  – Usually when the 3270 screen is displayed.
  – This frees up the resource should the user go to lunch.
  – A transaction is re-started when the user presses the Enter or a function key.
  – This involves more difficult program design (but is well worth it).
  – *Most applications are coded in a pseudo-conversational manner.*
Application development

- CICS application programs are generally divided into 3 categories:
  - **Presentation services**: communication between users and the transaction processing system.
  - **Business logic**: data manipulation and computation required by the transaction (most of processing).
  - **Data services**: retrieve/update data.

- Benefits of separation:
  - Components can be invoked/reused by other applications.
  - Allows for plug and play component changes when necessary.
  - Business transactions can mix and match program languages and data types.
Application services

- The **CICS API** allows programmers to request services using EXEC CICS commands.

- Many programming languages are supported in the CICS environment.

- CICS provides built-in transactions to assist the programmer with development.
  - **CEDF / CEDX**: Execution diagnostic facility transactions. Provides interactive debugging.
  - **CADP / DTCN**: Provides access to the CICS Debug Tool, a Source Level Debugger supplied with LE370.
  - **CECI**: Command interpreter transaction. Allows EXEC CICS statements without coding a program.
  - **CEBR**: Allows a programmer to browse through CICS Temporary Storage or Transient Data Queues.
  - **CMAC**: CICS Message and Codes online transaction.
EXEC interface

• CICS command format
  – The general format of CICS commands for the COBOL language is:

  ```cobol
  EXEC CICS
     FUNCTION
     OPTION (Argument)
  END-EXEC
  ```

  where:
  FUNCTION: Describes the CICS operation
  OPTION: Describes the options available with each function
  ARGUMENT: A data value used to qualify the option

• CICS programs look like batch with the insertion of EXEC (execute) CICS commands.

• The CICS commands are used to request services.

• CICS commands must be translated into COBOL prior/during program compilation.
Integrated translator

- Translation integrated into the compiler
  - In the past, this was a step prior to compilation
- The CICS command is commented out and replaced with valid COBOL statements
- The stub is link-edited with the load module and it is used to find the DFHEIP program

**Before translation**

```
EXEC CICS READ
  FILE (‘STOCK’)
  RIDFLD(KEY)
  INTO(INPUT-AREA)
  UPDATE
  RESP(WS-RESP)
END-EXEC

MOVE INPUT-AREA TO...
```

**After translation**

```
*EXEC CICS READ
  * FILE (‘STOCK’)
  * RIDFLD(KEY)
  * INTO(INPUT-AREA)
  * UPDATE
  * RESP(WS-RESP)
*END-EXEC

CALL DFEHI1 USING...

MOVE INPUT-AREA TO...
```
Execution flow

• The program **DFHEIP** gives control to the management module that will satisfy the request.

• The EXEC Interface Block (EIB) copybook contains fields to pass data and receive responses from CICS.

• The EIB is read-only. The contents should not be modified.

Supplied by IBM:  
* Must be linked in with Application code.  
** The EXEC CICS Command is commented out by the CICS Translator, and replaced by Language compatible Statements.
# EIB fields

<table>
<thead>
<tr>
<th>NAME</th>
<th>COBOL</th>
<th>PL/I</th>
<th>C</th>
<th>ASM</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIBTIME</td>
<td>PIC S9(7)comp-3</td>
<td>FIX DEC(7,0)</td>
<td>char[4]</td>
<td>PL4</td>
<td>TIME IN OH:MM:SS FORMAT</td>
</tr>
<tr>
<td>EIBDATE</td>
<td>PIC S9(7)comp-3</td>
<td>FIX DEC(7,0)</td>
<td>char[4]</td>
<td>PL4</td>
<td>DATE IN OCY/DDD FORMAT</td>
</tr>
<tr>
<td>EIBTRNID</td>
<td>PIC X(4)</td>
<td>CHAR(4)</td>
<td>char[4]</td>
<td>CL4</td>
<td>TRANSACTION IDENTIFIER</td>
</tr>
<tr>
<td>EIBTASKN</td>
<td>PIC S9(7)comp-3</td>
<td>FIX DEC(7,0)</td>
<td>char[4]</td>
<td>PL4</td>
<td>TASK NUMBER</td>
</tr>
<tr>
<td>EIBTRMID</td>
<td>PIC X(4)</td>
<td>CHAR(4)</td>
<td>char[4]</td>
<td>CL4</td>
<td>TERMINAL IDENTIFIER</td>
</tr>
<tr>
<td>EIBRSVD1</td>
<td>PIC XX</td>
<td>CHAR(2)</td>
<td>char[2]</td>
<td>CL2</td>
<td>RESERVED</td>
</tr>
<tr>
<td>EIBCPSN</td>
<td>PIC S9(4)comp</td>
<td>FIX BIN(15)</td>
<td>signed short</td>
<td>H</td>
<td>CURSOR POSITION</td>
</tr>
<tr>
<td>EIBCALEN</td>
<td>PIC S9(4)comp</td>
<td>FIX BIN(15)</td>
<td>signed short</td>
<td>H</td>
<td>COMMAREA LENGTH</td>
</tr>
<tr>
<td>EIBAUD</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>ATTENTION IDENTIFIER</td>
</tr>
<tr>
<td>EIBFN</td>
<td>PIC XX</td>
<td>CHAR(2)</td>
<td>char[2]</td>
<td>CL2</td>
<td>FUNCTION CODE</td>
</tr>
<tr>
<td>EIBCODE</td>
<td>PIC X(6)</td>
<td>CHAR(6)</td>
<td>char[6]</td>
<td>CL6</td>
<td>RESPONSE CODE</td>
</tr>
<tr>
<td>EIBDS</td>
<td>PIC X(8)</td>
<td>CHAR(8)</td>
<td>char[8]</td>
<td>CL8</td>
<td>DATASET NAME</td>
</tr>
<tr>
<td>EIBREQID</td>
<td>PIC X(8)</td>
<td>CHAR(8)</td>
<td>char[8]</td>
<td>CL8</td>
<td>REQUEST IDENTIFIER</td>
</tr>
<tr>
<td>EIBRSRCE</td>
<td>PIC X(8)</td>
<td>CHAR(8)</td>
<td>char[8]</td>
<td>CL8</td>
<td>RESOURCE NAME</td>
</tr>
<tr>
<td>EIBSYNC</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF SYNCPNT REQUESTED</td>
</tr>
<tr>
<td>EIBFREE</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF FREE REQUESTED</td>
</tr>
<tr>
<td>EIBRECV</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF RECEIVE REQUIRED</td>
</tr>
<tr>
<td>EIBSEND</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>RESERVED</td>
</tr>
<tr>
<td>EIBATT</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF ATTACH RECEIVED</td>
</tr>
<tr>
<td>EIBEOC</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF EOC RECEIVED</td>
</tr>
<tr>
<td>EIBFMH</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF FMHS RECEIVED</td>
</tr>
<tr>
<td>EIBCOMPL</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF DATA COMPLETE</td>
</tr>
<tr>
<td>EIBSIG</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF SIGNAL RECEIVED</td>
</tr>
<tr>
<td>EIBCONF</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF CONFIRM REQUESTED</td>
</tr>
<tr>
<td>EIBERR</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF ERROR RECEIVED</td>
</tr>
<tr>
<td>EIBERRCD</td>
<td>PIC X(4)</td>
<td>CHAR(4)</td>
<td>char[4]</td>
<td>CL4</td>
<td>ERROR CODE RECEIVED</td>
</tr>
<tr>
<td>EIBSYNRB</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF SYNCE ROLLBACK REQ'D</td>
</tr>
<tr>
<td>EIBNODAT</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>XTF NO APPL DATA RECEIVED</td>
</tr>
<tr>
<td>EIBRESP</td>
<td>PIC S9(8) comp</td>
<td>FIX BIN(31)</td>
<td>signed long</td>
<td>F</td>
<td>CONDITION NUMBER</td>
</tr>
<tr>
<td>EIBRESP2</td>
<td>PIC S9(8) comp</td>
<td>FIX BIN(31)</td>
<td>signed long</td>
<td>F</td>
<td>Additional details for some Responses</td>
</tr>
<tr>
<td>EIBRLDBK</td>
<td>PIC X</td>
<td>CHAR(1)</td>
<td>char</td>
<td>CL1</td>
<td>ROLLED BACK</td>
</tr>
</tbody>
</table>
Passing data between programs

- Communicate via **channels and containers** or via **COMMAREA**.
  - **Channels and containers**
    - Unlimited number of containers in a channel, unlimited amount of data per container
  - **COMMAREA**:
    - Older style of communication
    - Maximum size is 32K: if more was needed temporary storage was often used
CICS connectivity

- Multiregion operation (MRO)
- Intersystem communication (ISC)
- External CICS interface (EXCI)
- External Call Interface (ECI)
- External Presentation Interface (EPI)
- Web Support / Services (TCP/IP)
Communications

- **Transaction Routing**
  - Allows users from terminals connected in one CICS System to run transactions in another CICS system.

- **Function Shipping**
  - Allows a CICS Transaction in one system to access the resources owned by a connected CICS system.

- **Asynchronous Processing**
  - Allows distributed processing of an application asynchronously, and can be used cross system.

- **Distributed Program Link (DPL)**
  - Allows a program to link to another program in a remotely connected system.

- **External CICS Interface (EXCI)**
  - Enables an MVS Batch Program to call a program in a CICS region.
  - Same as External Call Interface (ECI), but with ECI the call is made from another platform.

- **External Presentation Interface (EPI)**
  - Allows a program running on another platform to emulate a 3270 terminal into CICS.

- **Web Support / Services**
  - Allows applications running on other platforms to communicate using a SOAP / XML message in an HTTP format over TCP/IP with CICS programs.
CICS resources

CICS is a table driven product that requires the definition of resources prior to use.

These are some of the resources defined to CICS...

![Diagram showing CICS resources]

- Terminals
- Files
- Programs & Mapsets
- Transient Data Queues
- Intra-partition temporary storage queues
- Terminals
- Files
- Programs & Mapsets
- Transient Data Queues
- Intra-partition temporary storage queues
- Logs
- TCP/IP Services
- EJBs
- CORBA Servers
- RDO
- Transactions
- Web Documents
- DB2 Threads
- ISC & MRO
- Monitoring Services
- Dynamic Storage Areas
- Printers
- VSAM/BDAM
- DFHLOG
- DFHRPL
- DFHINTRA
- Log files
- Program Library
- Extra-partition temporary storage queues
- Temporary storage queues
- AUX TS
- DFHCSD
- Monitor Data
- AUX Trace
- TXN Dumps
- DFHxUXT
- DFHDMPx

☆ Session feedback – Do it online at conferences.gse.org.uk/2015/feedback/AJ
CICS system definitions

• Resource Definitions are descriptions of resource types.
  – Example: The name of a transaction and the first program to execute.

• Resource Definitions provide CICS with the information to recognize and manipulate data appropriately.

• The information in the resource definitions may also contain the properties and interactions between resources.

• If a resource is not defined or defined incorrectly to CICS, it may not be recognized or cause errors and Transaction failures.

• Resource definitions are mostly stored on the CICS System Definition (CSD) File.
Methods for resource definition

• Resource Definition Online (RDO)
  – Uses CICS supplied transactions (CEDA, CEDB and CEDC) while a CICS region is running, to make definitions that are stored in the CICS System Definition (CSD) file.

• DFHCS DUP Offline Utility
  – Operates like RDO, but offline through a batch job.

• Automatic Installation (Autoinstall)
  – Works only with user modifications through a definition model. The utility then dynamically creates new definitions based on the model which can prevent the manual creation of large numbers of definitions.

• System programming
  – Using the EXEC CICS CREATE command, creates resources that are independent of the CSD.

• Macro Definition
  – Using assembler macros, creates definitions and stores them in assembled tables in a program library. The definitions are installed during CICS initialization.
**Resource Definition Online (RDO)**

```
CEDA DEF
ENTER ONE OF THE FOLLOWING

CONnection   PROCasesstype
CORbaserver  PROFILE
DB2Conn       PROGRAM
DB2Entry      Requestmodel
DB2Tran       Sessions
DJar          TCpipservice
DOctemplate   TDqueue
Engmodel      TERMINAL
File          TRANclass
Ipconn        TRANSaction
Journalmodel  TSMODEL
Library       TPETERM
LSrpool       URMAP
Mapset        Webservice
PARTItionset  
PARTner       
PIpipeine     

SYSID=C320 APPLID=CICSTS32

PF 1 HELP  3 END  6 CRSR  9 MSG  12 CNCL
```

☆ Session feedback – Do it online at conferences.gse.org.uk/2015/feedback/AJ
The CICS Master Terminal (CEMT) transaction can be used to get information about resources and their definitions.

CEMT has four commands and can be used to alter resource definitions that have already been installed in CICS.

Only some attributes of a resource may be changed using CEMT, others require complete re-installation.
Access to CICS

- CICS provides access to applications from a variety of sources.
- Client applications can be developed on any platform and in any language.
- CICS can also be used as a client to other applications running on different platforms.

© Copyright Circle Software Incorporated 2015
CICS Web Services

- Described by **Web Services Definition Language (WSDL)**

- CICS can be a service **requestor** or **provider** in a SOA environment

- CICS provides utilities to assist in:
  - Converting applications into web services
  - Accessing web services from external providers

- **CICS TS Feature Pack for Mobile Extensions:**
  - Adds support for web service requests using **JavaScript Object Notation (JSON)** and the conversion between JSON and high-level language data structures

★ Session feedback – Do it online at conferences.gse.org.uk/2015/feedback/AJ
CICS Service Oriented Architecture (SOA)

- Integrated into most current release
- Defined by Web Service Description Language (WSDL)
- CICS role in SOA can be service requestor, service provider, or both
- CICS Web Services utility programs
  - Assist in converting existing application into a Web Service
  - Use a Web Service provided by an external provider
- Support for web services standards and technologies
  - WSDL 2.0
  - WS-I Basic Profile 1.1
  - WS-Security
  - WS-Trust
  - WS-Addressing
  - Message Transmission Optimization Mechanism /XML – Binary Optimized Packaging (MTOM/XOP)
CICS event processing

• An event is anything of significance to an enterprise

• CICS allows users to capture, format and emit business events from CICS

• Events can be sent via HTTP, MQ queue, TS Queue or Start Transaction for further processing

• Events are bound to a CICS system using an event binding editor built into CICS Explorer and Rational Developer for System z with Java (RDz)

• The bindings are enabled using a BUNDLE resource
  – CICS Explorer or Web User Interface (WUI)
  – RDO or CEMT
CICS tools

- If CICS does not provide the needed functionality there are many tools available from IBM and other vendors to assist in creating and managing an online transaction processing system.
- For example, IBM provides the following:
  - CICS Batch Application Control
  - CICS Configuration Manager for z/OS
  - CICS Interdependency Analyzer
  - CICS Online Transmission Time Optimizer for z/OS
  - CICS VSAM Recovery for z/OS
  - CICS Performance Analyzer for z/OS
  - CICS Business Events Publisher
  - CICS VSAM Transparency
  - CICS Deployment Assistant for z/OS
  - CICS Service Flow Runtime
  - IBM Tivoli OMEGAMON XE for CICS on z/OS
  - REXX for CICS Transaction Server for VSE/ESA
  - Extensions to the CICS Information Center
- Many tools also provide a CICS Explorer plug-in.
New in CICS TS 5.3

New and enhanced capabilities are delivered in three main focus areas, each with four core capabilities:

• **Service agility**: Enhanced support for Java and the WebSphere® Liberty profile and includes:
  – Additional Liberty profile features
  – Enhanced interoperability
  – Simplified management
  – Enhanced Java SE support

• **Operational efficiency**: Includes performance optimizations, enhanced metrics, and additional security:
  – Web service optimizations
  – Performance improvements
  – Enhanced metrics
  – Additional security options

• **Cloud with DevOps**: Includes new cloud and DevOps support to automate CICS deployments:
  – Automated builds
  – Scripted deployments
  – UrbanCode Deploy support
  – Enhanced cloud enablement
Summary

• CICS is ideal for existing transactional environments and your new ones too.

• Attributes of CICS:
  – Availability, Maintainability, and Scalability.
  – Continues exploitation of new hardware and software technology.
  – Plenty of education is available.
## Further reading

<table>
<thead>
<tr>
<th>Reading</th>
<th>URL</th>
</tr>
</thead>
</table>
Further reading: IBM Redbooks

• Introduction to CICS Dynamic Scripting

• Threadsafe Considerations for CICS

• Extend the CICS Explorer: A Better Way to Manage Your CICS

• IBM CICS and the JVM server: Developing and Deploying Java Applications

• Implementing CICS Web Services

• Exploring Systems Monitoring for CICS Transaction Gateway

• CICS Web Services Workload Management and Availability

• CICS Systems Manager in the WUI as the Principle Management Interface

Session feedback – Do it online at conferences.gse.org.uk/2015/feedback/AJ
Contact us

- Ezriel Gross' Email: ezriel@circle-us.com
- General Email: mail@circle-us.com
- Website: www.circle-us.com/education

CA: CICS Concepts and Facilities
Session feedback

• Please submit your feedback at

  http://conferences.gse.org.uk/2015/feedback/AJ

• Session is AJ