

# **Introduction**

## **Chapter 1:**

SYS-ED/Computer Education Techniques, Inc.

Ch 1: 1

## **Objectives**

You will learn:

- New features of Enterprise COBOL.
- Interface to COBOL and JAVA.
- Object-oriented features.
- Utilization of XML in COBOL.
- Extended operations and sizes.

## New Features

- **2001: Release 1**
  - OO COBOL syntax for interoperability with Java.
  - WebSphere support.
  - High-speed XML PARSE support.
  - Unicode support.
  - Multithreading support.
  - Integrated CICS translator.
- **2002: Release 2**
  - Enhanced debug support.
  - Java interoperability.
  - OO COBOL enhancements.
- **2004: Release 3**
  - XML GENERATE.
  - DB2 Version 8 SQL support in coprocessor.
  - Debug Tool Version 4 support.

## New Features

- **CICS:**
  - Integrated CICS translator requires CICS TS V2.
- **DB2:**
  - Integrated SQL coprocessor requires DB2 Version 7.
  - New SQL features and DB2 support requires DB2 Version 8.
- **Java interoperability:**
  - IBM Developer Kit for OS/390, Java 2 Technology Edition, SDK 1.3.0 or later.
- **Prerequisites for Unicode and for Java interoperability:**
  - Included in z/OS R2 and later.

## Java Interoperability

- **New object-oriented COBOL syntax for Java interoperability.**
  - Enables COBOL and Java to be mixed within a single application.
  - OO COBOL syntax is mapped to the Java Virtual Machine “under the covers”.
  - Is based on the facilities of JNI:  
Java Native Interface.

# OO Syntax for Java Interoperation

- Define classes, with methods and data implemented in COBOL.
- Create instances of Java and COBOL classes.
- Invoke methods on Java and COBOL objects.
- Classes can inherit from Java or COBOL classes.
- Define and invoke overloaded methods.
- Call JNI: Java Native Interface services.
- Code in COBOL classes can CALL existing procedural COBOL code.
- Wrapper classes can be written for existing procedural COBOL code enabling it to be invoked from Java programs.
- Java code can create instances of COBOL classes, invoke COBOL methods, and extend COBOL classes.

## Java Interoperability

- COBOL classes map to Java classes.
- There are a clearly defined list of compatible data types.
- There is automatic conversion of IEEE float to HEX float for INVOKE parameters.
- Unicode support in COBOL, plus CALLs to JNI services, enable interoperation with Java Strings Multithread support.
- Java runs in a multithread environment.

Java	byte	short, int, long	float	double	char	class types
COBOL	PIC X	BINARY	COMP-1	COMP-2	PIC N NATIONAL	OBJECT REFERENCE

## Migration Path

- **There is no real migration path from old OO to new OO.**
- **There is overlap in the language syntax, however, the goals are different.**
- **Some existing OO COBOL code from previous compilers can be migrated to Enterprise COBOL, with some reworking of the application.**

## WebSphere Support

- **The Java interoperability extensions can be used to access EJB: Enterprise Java Beans that run on a J2EE-compliant EJB server.**
  - WebSphere Application Server is J2EE-compliant.
- **Client COBOL would access the following programming interfaces using INVOKE:**
  - JNDI: Java Naming and Directory Interface to locate EJB services and components.
  - Java ORB to invoke methods on enterprise beans.
- **WebSphere requires several COBOL features:**
  - Java-based OO  
*and therefore Unicode plus*
  - Multithreading.

# XML: Introduction

- **What is XML?**
  - It is a markup language for describing the semantics of data; which is distinct from the presentation of data.
  - Each piece of data is identified via the markup language.
  - An unlimited number of tags can be defined.
- **Why XML?**
  - It is becoming the interconnection layer of e-business.
  - The industry direction for application integration and platform independent data interchange.
    - e.g. for Web Services.
- **Allows the sender and receiver to evolve independently of each other; this is a flexible interface.**

## XML Document - Sample

```
<?xml version="1.0" encoding="ibm-1140"?>  
<TRADE type="short sale">  
<SYMBOL>IBM</SYMBOL>  
<PRICE>$98.75</PRICE>  
<SHARES>200</SHARES>  
<COMMISSION>$29.95</COMMISSION>  
</TRADE>
```

## XML on z/OS

- **IBM zSeries XML technology:**
  - XML Toolkit for z/OS and OS/390
- **These offerings include both:**
  - XML Parser for z/OS and OS/390, Java Edition
  - XML Parser for z/OS and OS/390, C++ Edition
- **There is a COBOL High Speed XML parser:**
  - Faster and simpler than XML toolkit parsers.
  - COBOL parser does not validate XML documents.
  - COBOL parser does not process DTDs, even if internal DTD = Document Type Definition.
- **Tailored to integrate with COBOL programs.**

## COBOL XML Parser Support

- **Works with any transport mechanism for XML documents.**
  - Use WebSphereMQ, CICS transient queue or COMMAREA, IMS message processing queue, WebSphere, etc.
- **XML Parser is part of the run-time library/**
  - It can be used from Enterprise COBOL or Enterprise PL/I.
- **Inbound XML documents**
  - Outbound can use MOVE CORRESPONDING, STRING, group declarations, etc. to create XML documents.

# COBOL XML Parser Support

- **Parses XML documents that are in memory, in a COBOL alphanumeric, or national data item.**
- **It is used to parse XML documents into individual pieces:**
  - Passes each piece to user-written processing procedure.
- **During parsing COBOL data structures can be populated with the data from XML messages.**
  - Advantage:  
Non-COBOL programs can communicate data to/from COBOL without having to know the COBOL data structure formats.

# COBOL XML Parser Support

- **New XML PARSE statement.**
  - The COBOL interface to the new XML parser.
- **New XML special registers.**
  - XML-CODE: Communicates the status of parsing.
  - XML-EVENT: Describes each event in the parse.
  - XML-TEXT: Contains XML document fragments.
  - XML-NTEXT: Contains NATIONAL XML doc fragments.
- **New XML GENERATE statement.**
  - Generates XML message from COBOL group data items.

# Unicode

- **What is Unicode?**

- Unicode provides a unique number for every character; regardless as to the platform, program, and the language.
- Without Unicode there would be many different code pages; the same numbers would be reused for different characters.
- Enables text to be handled in any language efficiently.
- Allows a single application executable to work for a global audience.
- In Enterprise COBOL, Unicode is represented via multi-byte characters.
- Supports almost all characters for almost every country.

# Unicode

- **Why Unicode?**
  - Internationalization.
  - Java Interoperability.
- **Internationalization**
  - COBOL programs can generate reports for any country.
  - User interface/dialog/messages can be in any national language, with any characters.

# Unicode

- **National data type**
  - PIC N USAGE NATIONAL for data items.
  - N-literals: N'This is NATIONAL data'.
- **CODEPAGE(nnnnn) compiler option**
  - Specifies the code page CCSID used for:
    - Alphanumeric and DBCS data items at run time.
    - Alphanumeric, national, and DBCS literals in the source program.
    - Default code page for parsing XML documents.
- **National data in statements**
  - MOVE X TO national-item.
  - Relation conditions.
  - INITIALIZE, INSPECT, SEARCH, UNSTRING, etc.

## Unicode Conversion

- **Implicit conversions are performed as needed.**
  - MOVE *numeric-item* TO *national-item*
  - IF *alphanumeric-item* = *national-item* ...
- **New intrinsic functions for explicit conversion.**
  - DISPLAY-OF
    - Convert from USAGE NATIONAL to USAGE DISPLAY.
  - NATIONAL-OF
    - Convert from USAGE DISPLAY to USAGE NATIONAL.
- **Allow explicit CCSID specification.**
- **Can be nested, to support conversion of "any code page" to "any code page".**

# Multithreading

- **What is multithreading?**
- **How does it relate to 'COBOL multitasking'?**
  - **Multitasking:**
    - Multiple tasks running in the same address space sharing the same run-time library for programs compiled RES.
    - Sharing process resources.
    - One enclave per task/process.
    - One thread per enclave.
    - Supported for COBOL in 1991: COBOL/370 R1.
- **Multithreading:**
  - Multiple threads running in the same enclave.
  - Sharing enclave resources.
  - Supported for COBOL in 21st century: Enterprise COBOL.

# Multithreading

- **Multithreading is required for:**
  - COBOL programs called from multithreaded C programs.
  - COBOL programs called from PL/I tasks.
  - Java interoperation.
  - Multithreaded application servers.
- **THREAD compiler option:**
  - Required for multithreading with COBOL.
- **COBOL specific library is now thread safe.**
- **Multiple thread invocations of a program share:**
  - WORKING-STORAGE, record areas, buffer areas.
- **Multiple thread invocations of a program have separate copies of LOCAL-STORAGE.**
- **Thread-safe I/O statements:**
  - Use READ INTO Local-item and WRITE FROM Local-item.

## Miscellaneous Enhancements

- **Large Value clause literals for BINARY items:**
  - For TRUNC(BIN) or COMP-5.
  - 77 BIN1 PIC S(4) COMP-5 VALUE 32767.
  - Picture clause cannot have P (scaled).
- **FUNCTION-POINTER datatype:**
  - Same usage as PROCEDURE-POINTERS.
  - Same length as C/C++ function pointers.
  - Improved interoperability with C structure/s
- **ADDRESS OF WORKING-STORAGE for CALL arguments:**
  - CALL SUB USING BY VALUE ADDRESS OF WS-ITEM.
  - Recommended technique for calling C functions with pointer arguments.