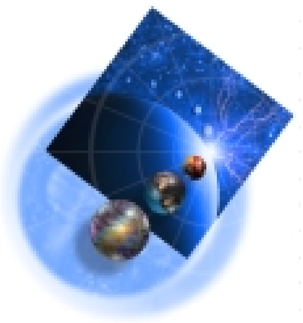
What's New in COBOL for z/OS



Tom Ross
SHARE New York
August, 2004

Enterprise COBOL

- A New name for COBOL compilers:
  - Enterprise COBOL for z/OS and OS/390 Version 3
  - PID 5655-G53
  - Release 1 GA: November, 2001
  - Release 2 GA: October, 2002
- Now ANOTHER new name!
  - Enterprise COBOL for z/OS Version 3 Release 3
  - Release 3 GA: February, 2004
- For migration/compatibility purposes
  - Think of Enterprise COBOL Version 3 Release 3 as VS COBOL II Release 11

Topics

- **First, Release 1 (2001) that added most new features:**
  - OO COBOL syntax for interoperability with Java
  - WebSphere support
  - High-speed XML PARSE support
  - Unicode support
  - Multithreading support
  - Integrated CICS translator
- **Then Release 2 (2002)**
  - Biggest new features are enhanced Debug support
  - Also some Java interoperability OO COBOL enhancements
- **Then Release 3 (2004)**
  - XML GENERATE
  - DB2 Version 8 SQL support in coprocessor
  - Debug Tool Version 4 support

Enterprise COBOL for z/OS R3

- **Run-time library pre-reqs:**
  - OS/390 V2R10 Language Environment
    - Plus PTFs for APAR PQ80358/PQ87307
  - z/OS Language Environment R1-R5:
    - Plus PTFs for APAR PQ80358/PQ87307
- **CICS: CICS TS 1.3**
  - Integrated CICS translator requires CICS TS V2
- **DB2: Version 6**
  - Integrated SQL coprocessor requires DB2 Version 7
  - New SQL features and DB2 support require DB2 Version 8
- **IMS: Version 6**
- **Pre-req for Java interoperability:**
  - IBM Developer Kit for OS/390, Java 2 Technology Edition, SDK 1.3.0 or later
- **Pre-req for Unicode and for Java interoperability:**
  - OS/390 Support for Unicode (HUNI2A0)
    - Included in z/OS R2 and later

COBOL or Java?   Both!

- **New object-oriented COBOL syntax for Java interoperability**
  - **Enable COBOL and Java to be mixed within a single application**
  - **OO COBOL syntax mapped to Java Virtual Machine "under the covers"**
  - **Based on facilities of Java Native Interface (JNI)**

e-business

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 Java Native Interface (JNI) services**
- **Code in COBOL classes can CALL existing procedural COBOL code**
  - Write *wrapper classes* for existing procedural COBOL code, enabling it to be invoked from Java programs
- **Java code can create instances of COBOL classes, invoke COBOL methods, extend COBOL classes**

OO syntax for Java interoperation:

- **There are three versions of Java for 390**
  - High Performance Java (HPJ)
  - Java JDK 1.1.8
  - Java SDK 1.3.0 or later   <- this is the COBOL pre-req
- **OO COBOL for Java interoperation works in any of the environments supported by Java SDK 1.3.0:**
  - UNIX System Services, WebSphere, Batch using BPXBATCH
- **Support for CICS and IMS will come later**
- **New capability complements existing COBOL:Java interoperation approaches:**
  - **Connector technologies**
  - **CICS TS V2 Java support**

OO syntax for Java interoperation:

- **New COBOL features that improve Java interoperability:**
  - **COBOL classes map to Java classes**
  - **Clearly defined list of compatible data types**

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

  - ▸ 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

OO syntax for Java interoperation:

- **OO syntax updated from COBOL V2R2**
  - **Closer to the proposed 2002 ANSI Standard**
  - **Extensions specifically for Java support**
- **Classes and methods:**
  - **CLASS-ID, METHOD-ID**
  - **REPOSITORY paragraph**
  - **FACTORY, OBJECT**
  - **INVOKE**
- **No real migration path from old OO to new OO**
  - **Much overlapping language syntax, but...**
  - **Goals are different**
  - **Some existing OO COBOL code from previous compilers could be migrated to Enterprise COBOL, with application rework.**

WebSphere Support

- **You can now use the Java interoperability extensions to access Enterprise Java Beans (EJB) that run on a J2EE-compliant EJB server**
  - WebSphere Application Server is J2EE-compliant
- **Client COBOL would access the following programming interfaces using INVOKE:**
  - Java Naming and Directory Interface (JNDI) to locate EJB services and components
  - Java ORB to invoke methods on enterprise beans
- **WebSphere requires several of the V3R1 features:**
  - Java-based OO *and therefore*
  - Unicode *plus*
  - Multithreading

Introduction to XML

- **What is XML?**
  - **A markup language, for describing the semantics of data (rather than the presentation)**
  - **Each piece of data is identified via the markup language**
  - **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 sender and receiver to evolve independently of each other (flexible interface)**
    - as opposed to Electronic Data Interchange (EDI) for example

Introduction to XML

- **Sample XML document:**

```
<?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 zSeries

- **IBM zSeries XML technology:**
  - **IBM announced:**
    - XML Toolkit for z/OS and OS/390 V1R2 on March 27, 2001
    - XML Toolkit for z/OS and OS/390 V1R3 on October 23, 2001
  - **These offerings include both:**
    - XML Parser for z/OS and OS/390, Java Edition
    - XML Parser for z/OS and OS/390, C++ Edition
  - **http://www.ibm.com/servers/eserver/zseries/software/xml/**
- **Now introducing the 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 MQSeries, CICS transient queue or COMMAREA, IMS message processing queue, WebSphere, etc.
- **XML Parser is part of the run-time library**
  - Can be used from Enterprise COBOL or Enterprise PL/I
- **Inbound XML documents only for V3R1 & V3R2**
  - Outbound can use MOVE CORRESPONDING, STRING, group declarations, etc. to create XML documents
  - More XML tools and support are on the way from IBM!

COBOL XML Parser support

- **Parses XML documents that are in memory, in a COBOL alphanumeric or national data item**
- **Used to parse XML documents into individual pieces**
  - Passes each piece to user-written processing procedure
- **During parsing you can populate COBOL data structures 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 new XML parser
- **New XML special registers**
  - XML-CODE: communicates status of parsing
  - XML-EVENT: describes each event in parse
  - XML-TEXT: contains XML document fragments
  - XML-NTEXT: contains NATIONAL XML doc fragments

```
XML PARSE XMLDOCUMENT
   PROCESSING PROCEDURE XMLEVENT-HANDLER
END-XML
...
XMLEVENT-HANDLER.
   EVALUATE TRUE
      WHEN XML-EVENT = 'START-OF-ELEMENT' AND
               XML-TEXT = 'TRADE'
         DISPLAY 'Processing new stock trade'
            ...
```

COBOL XML Generation support

▪ New XML GENERATE statement

- Generates XML message from COBOL group data items

```
1 Employee1.
  2 Name pic X(5) Value 'Tom'.
  2 Idn  pic 9(9) comp Value 123456789.
  2 Addr.
    3 Street pic X(20) Value '555 Bailey Ave'.
    3 City   pic X(20) Value 'San Jose'.
    3 State  pic X(20) Value 'California'.
  2 More.
    3 Age    pic +99.99  Value '45.9'.
    3 Firm   pic BBXXX9B Value 'IBM4'.
    3 Salary COMP-2      Value +.00012327E+06.
1 XMLDOCUMENT pic X(500).

Procedure division.

    XML GENERATE XMLDOCUMENT FROM EMPLOYEE1
```

COBOL XML Generation support

- **Output from sample XML GENERATE statement**
  - Using program 'PRETTY' from sample
    - Complete samples in Application Programming Guide

```
<Employee1>
  <Name>Tom</Name>
  <Idn>123456789</Idn>
  <Addr>
    <Street>555 Bailey Ave</Street>
    <City>San Jose</City>
    <State>California</State>
  </Addr>
  <More>
    <Age>45.9</Age>
    <Firm>IBM4</Firm>
    <Salary>1.23270000000000000E+02</Salary>
  </More>
</Employee1>
```

Support for Unicode

- **What is Unicode?**
  - Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language
    - Without Unicode you get many different code pages, reusing the same numbers for different characters
  - It enables you to handle text in any language efficiently
  - It 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

Support for Unicode

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

Support for Unicode

- **Unicode conversion services required for Unicode support and for Java interoperability**
  - Product "OS/390 Support for Unicode" (HUNI2A0), a base element of OS/390 and z/OS, provides conversion services
    - For OS/390, delivered via Web site
      http://www.ibm.com/downloads
    - Choose 'Software Downloads: Operating systems', then OS/390
    - Built in to base operating system, with z/OS R2 or later
  - Unicode conversion services must be configured
  - Enable conversions between pairwise combinations of:
    - Codepage(s) used for the COBOL CODEPAGE compiler option (default is CCSID 1140) or new intrinsic functions,
    - CCSID 1200, and
    - CCSID 1208

Support for 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.

Support for Unicode

- **Implicit conversions 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"

Support for Unicode

- Example: convert EBCDIC to ASCII

```
    77 EBCDIC-CCSID PIC 9(4) BINARY VALUE 1140.
    77 ASCII-CCSID  PIC 9(4) BINARY VALUE 819.
    77 Input-EBCDIC  PIC X(80).
    77 Temp-National PIC N(80) NATIONAL.
    77 ASCII-Output  PIC X(80).

  * Convert EBCDIC to NATIONAL
     Move Function
        National-of (Input-EBCDIC, EBCDIC-CCSID)
          to Temp-National

  * Convert NATIONAL to ASCII
     Move Function
        Display-of (Temp-National, ASCII-CCSID)
         to ASCII-output
```

Support for Unicode

- **Example: convert EBCDIC to ASCII (simplified?)**

```
   77 EBCDIC-CCSID PIC 9(4) BINARY VALUE 1140.
   77 ASCII-CCSID  PIC 9(4) BINARY VALUE 819.
   77 Input-EBCDIC  PIC X(80).
   77 ASCII-Output  PIC X(80).

 * Convert EBCDIC to ASCII
    Move Function
           Display-of
             ( Function National-of
                (Input-EBCDIC EBCDIC-CCSID),
               ASCII-CCSID
             )
       to ASCII-output
```

Multithread Support

- **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

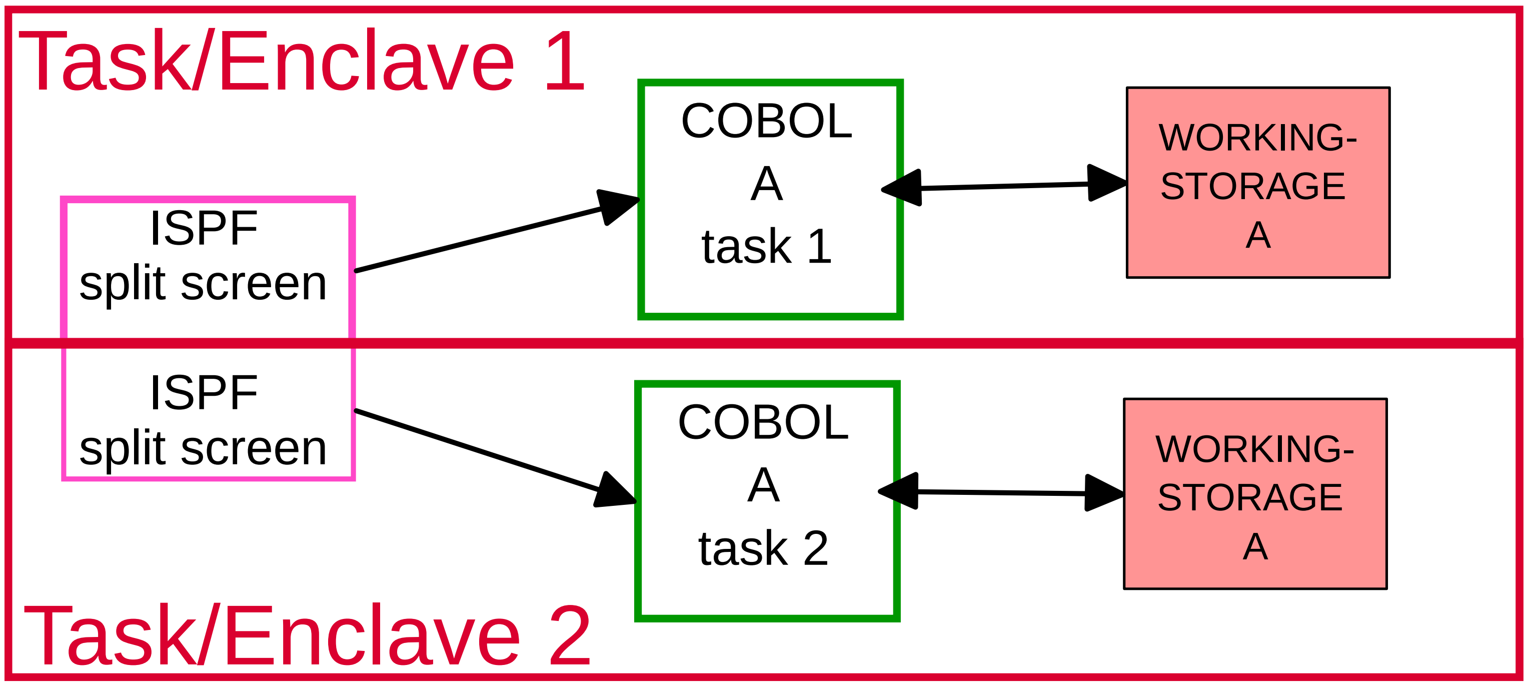
e-business

Multithread Support

- **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

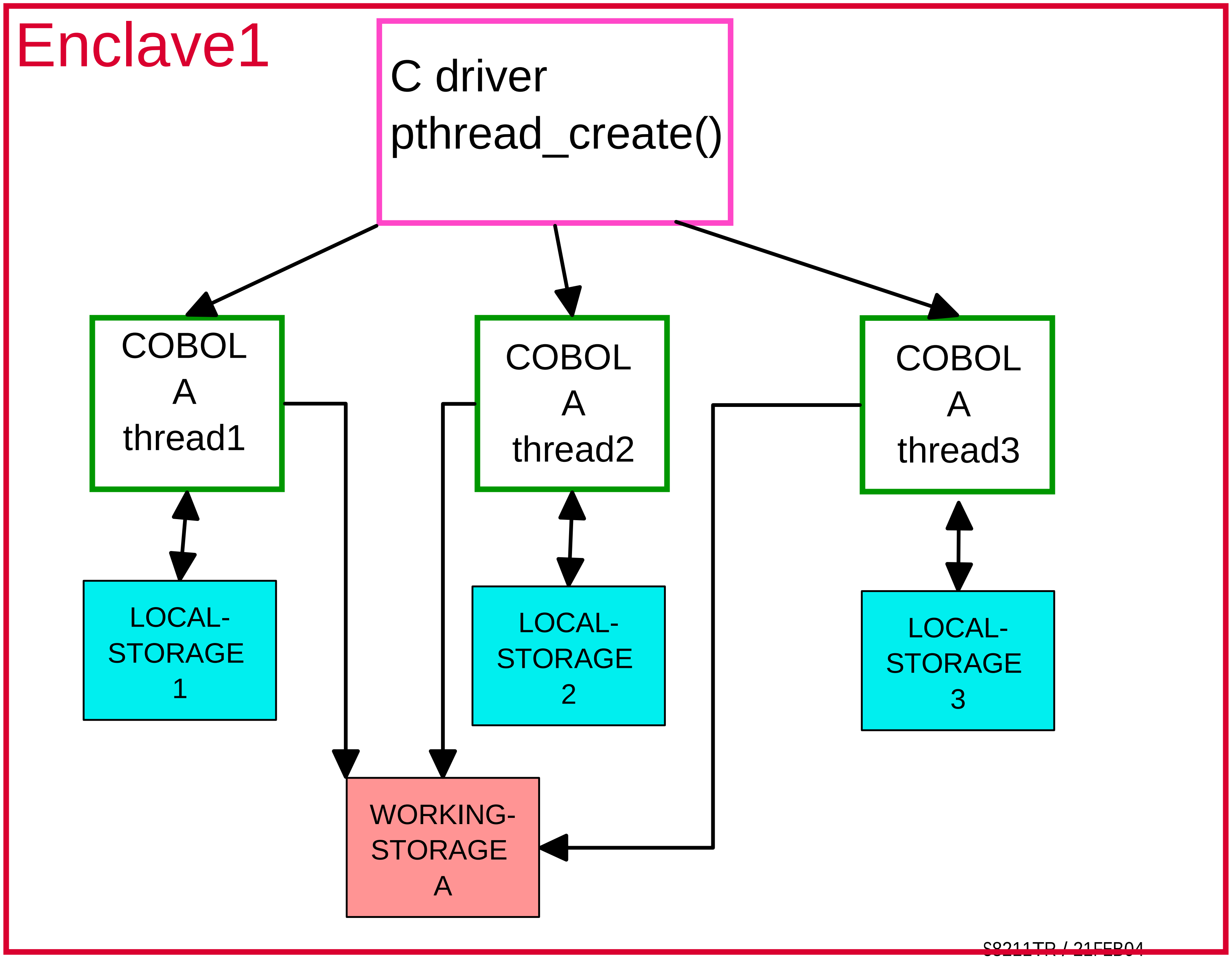
Multithread Support

- **Example of 'multitasking' COBOL:**



Multithread Support

- **Example of multithread COBOL:**



Integrated CICS Translator

- **Analogous to integrated SQL coprocessor shipped with IBM COBOL V2R2 and DB2 V7**
- **Enabled using CICS compiler option**
- **EXEC CICS statements in copybook members**
  - also EXEC DLI
- **Debug at EXEC CICS source level**
- **Requires CICS TS Version 2**
- **Can now have EXEC CICS and EXEC SQL together**
  - One compile step! No preprocessors!
  - EXEC CICS and EXEC SQL in copybook members

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 structures
- **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

Migration

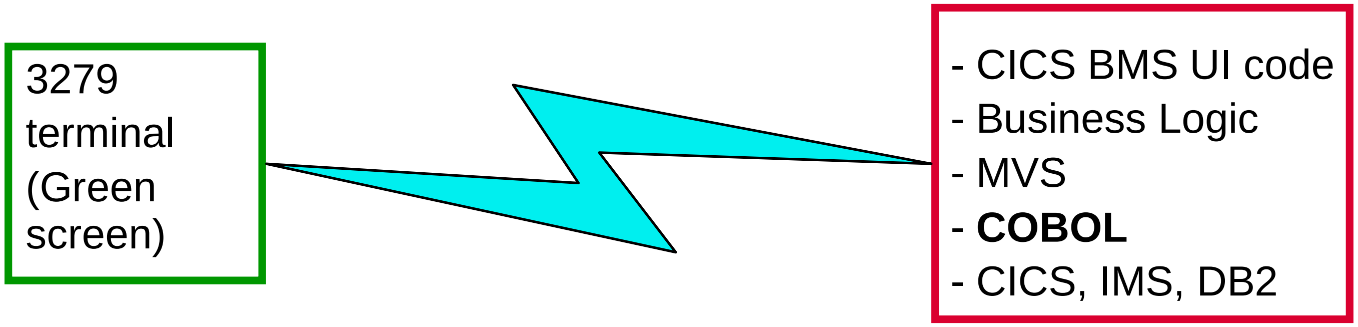
- **Enterprise COBOL is 1985 Standard only**
  - CMPR2 option has been removed
  - You can use CCCA to easily convert
- **New reserved words:**
  - JNIENVPTR, NATIONAL, XML, END-XML, XML-EVENT, XML-CODE, XML-TEXT, XML-NTEXT, FUNCTION-POINTER
- **Java-based OO only**
  - SOM-based OO COBOL has been removed
    - IDLGEN and TYPECHK options removed
  - SOM was removed from z/OS V1R2
- **OS/390 V2R10 or later only**
  - OS/390 2.9 is out of service anyway now
- **New default options:**
  - DBCS   FLAG(I,I)   RENT XREF
- **WORD(NOOO) no longer available**

Scenario for Putting it all Together

- **Scenario:**
  - **Insurance company with field offices connected to main office via mainframe network**
  - **Change to WEB interface, still have COBOL server code, but have Java business logic in middle tier**
  - **Want to consolidate servers, use COBOL strengths**
  - **Also want to expand business to other countries, need to address global marketplace**

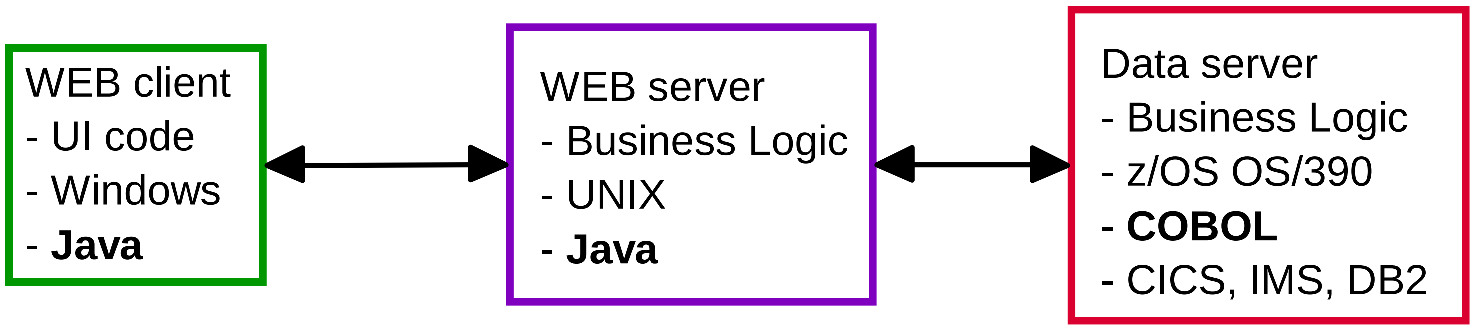
Scenario for Putting it all Together

- **Existing mainframe-based solution for insurance company field office:**



Scenario for Putting it all Together

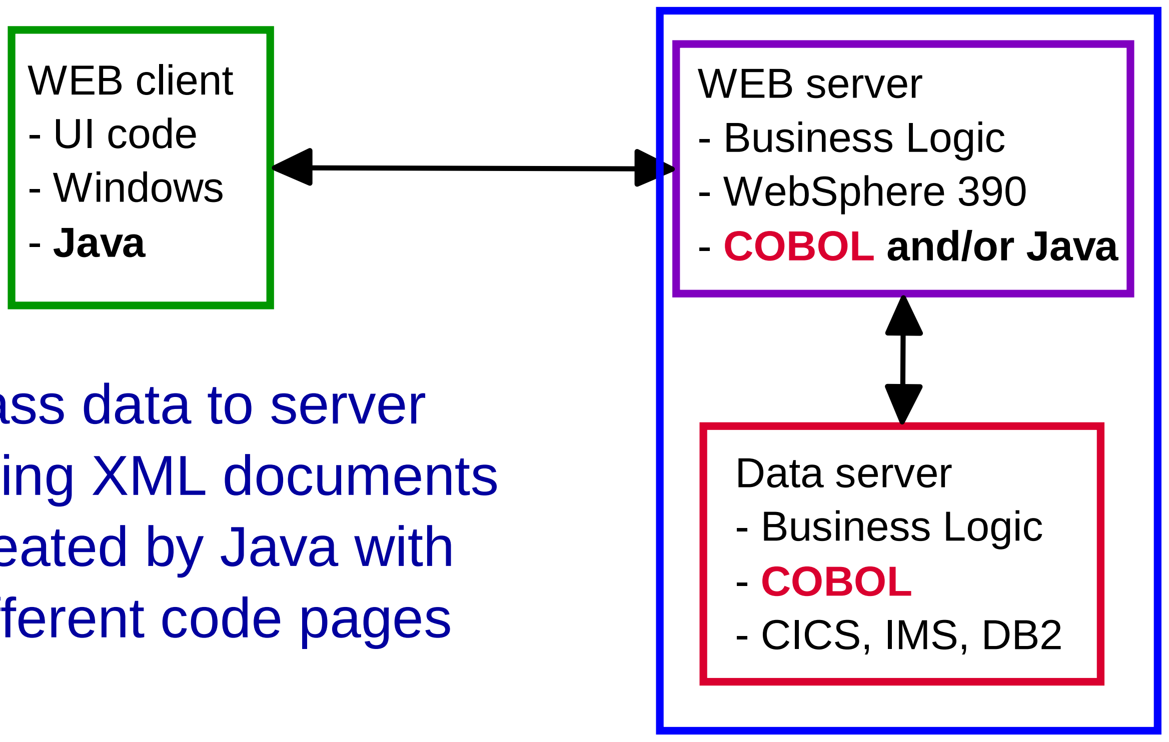
- **'Traditional' 3-tiered distributed application ... circa 1998:**



WEB client allowed flexibility, but middle tier added system complexity and difficult access to host COBOL assets

Scenario for Putting it all Together

- **Modern 3-tier distributed application ... circa 2002:**



Pass data to server using XML documents created by Java with different code pages

Convert XML to COBOL data structures, EBCDIC output

Process traditional data structures in EBCDIC

zSeries eServer (S/390 mainframe)

Enterprise COBOL

- COBOL publications in .pdf format:
  - www.ibm.com/software/awdtools/cobol/zos/library/
  - Includes updated Performance Tuning Paper
- COBOL publications in bookmanager format
  - www.ibm.com/servers/s390/os390/bkserv/

What's newer?   2002 Release

- **Enterprise COBOL for z/OS and OS/390 V3R2**
  - **Available September 27, 2002**
  - **Enhanced debugging information for new features of Debug Tool V3R1**
  - **Optimization of OO syntax for Java interoperability**
  - **Parameterized initialization of the JVM**
  - **Support for object arrays as method arguments**
  - **Support for COBOL class def. with a `main` method**
  - **Enhanced support for Unicode in DB2 COBOL apps**
  - **Execution of OO COBOL from batch JCL**
  - **Support for COBOL-Java interoperability in IMS**
    - Requires IMS Version 8 (or V7 w/APARS PQ53944 & PQ54039

What's newest?   2004 Release

- **Enterprise COBOL Version 3 Release 3**
  - **Available February 27, 2004**
  - **Enhanced XML capabilities, for generation of outbound XML from a COBOL data structure**
  - **Support for new functions in Debug Tool for z/OS, Version 4**
  - **Support for DB2 for z/OS Version 8**
    - Unicode features
  - **SQL coprocessor upgrade for new Version 8 SQL**
    - Multiple-Row Fetch
    - Multiple-row INSERT
    - Longer names for SQL identifiers, table names, and column names
    - And more! See DB2 V8 documentation for details

Enterprise COBOL for z/OS

- **Debug Tool Version 4 Release 1**
  - Included with Full Function Enterprise COBOL V3R3
  - Debug programs compile without hooks and OPTIMIZEd
    - TEST(NONE,SYM) OPTIMIZE
  - Playback mode (step backwards)
  - Automatic monitoring of variables referenced in statements
  - Can have debug information separate or in module
    - TEST(NONE,SYM,SEPARATE)
  - Support for debugging OS/VS COBOL source programs (CMPR2 source programs also)
  - Measure the code coverage of your test cases
  - ISPF interface to do program preparation, manage default settings and set up files
  - For COBOL:
    - Source-level debug for EXEC SQL
    - Source-level debug for EXEC CICS, EXEC DLI
    - XML PARSE and GENERATE statements
    - Multithreaded applications (basic level support)
    - All existing IBM COBOL language features
    - OO COBOL and Unicode support