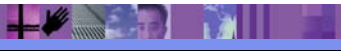
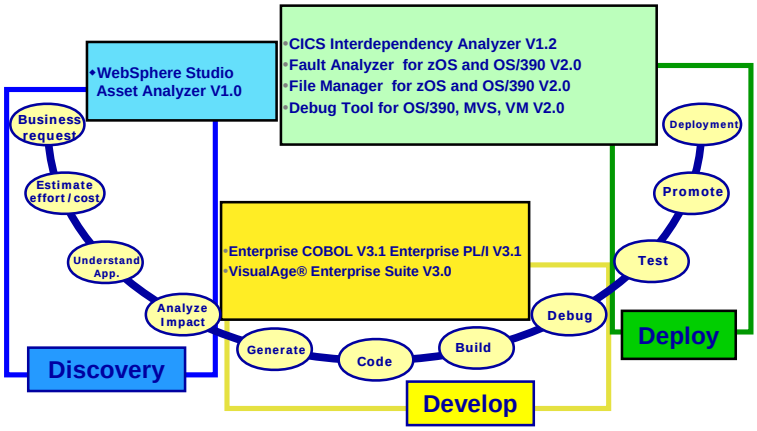

IBM Software Group
A trip through the new things you can do in CICS with COBOL, PL/I and C
Andy Krasun
WebSphere software
e business on demand software

Agenda

- Positioning
- Integrated CICS Translator, SQL Coprocessor
- Compiler currencies
- Enterprise compilers for COBOL and PL/I
  - Inbound XML parser support
  - Unicode support
- C/C++
- LE enhancements
- IBM Debug Tool
- IBM Fault Analyzer
- IBM File Manager
- CICS Interdependency Analyzer



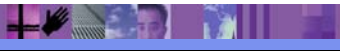
z-Series:Enterprise Application Development Offerings



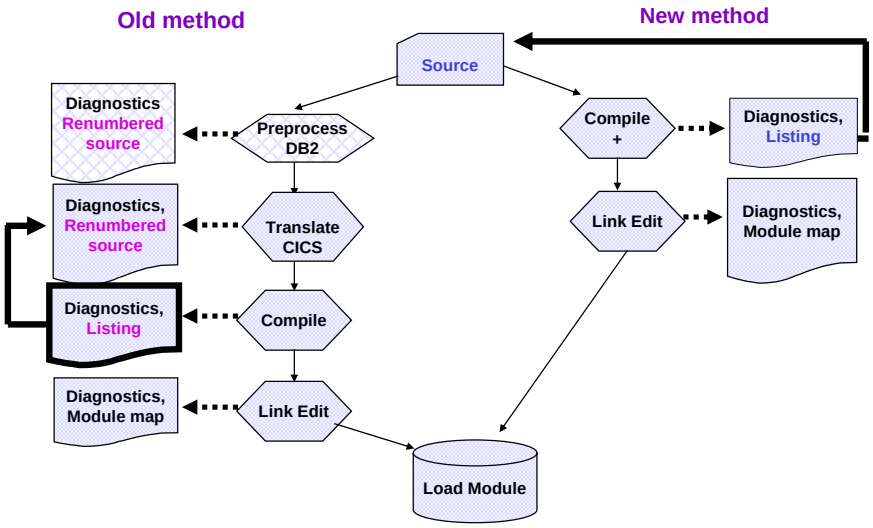
A solution to promote the discovery, development, and delivery of Enterprise applications !

Integrated CICS Translator and SQL Co-processor

- Commands translated during program compilation step
  - CICS Integrated Translator
    - EXEC CICS
    - EXEC DLI
    - EXEC CPSM
  - SQL Integrated Co-processor
    - EXEC SQL

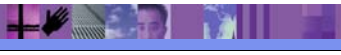


Traditional versus new process



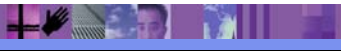
Integrated CICS Translator and SQL Co-processor

- Benefits
  - Ease of use
    - No translate step
    - No SQL preprocess step
    - Single output listing, corresponding to source – especially important for interactive Debug Tool
  - Function
    - Eliminates need to pre-translate or pre-process copybooks
    - Allows use of COBOL ‘REPLACE’ on EXEC statements
    - Simplifies nesting of COBOL programs containing commands
    - Allows nested programs to be COPYed
    - Makes CICS control blocks independent of COBOL TRUNC parameter (COBOL V3.1)



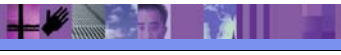
Integrated Translator - Migration

- Procedures
  - Remove translator step from CICS-supplied procedures, or
  - Use standard procedures with
    - CICSTS2x.CICS.SDFLOAD in STEPLIB for compile step
    - Include DFHELII in link step
- Compiler options
  - CICS for COBOL
  - PP (CICS) for PL/I
- Specifying translator options
  - COBOL
    - PARM = ‘ CICS (‘ ’opt1 opt2 . . .‘ ‘), SQL (’ ‘opta optb . . .’ ‘) ‘
    - CBL CICS (‘opt1,opt2, . . . ‘), SQL (‘opta,optb, . . .’)
    - COBOL and COBOL2 translator options not supported
  - PL/I – compiler preprocessor options
    - PP (CICS (‘opt1 opt2 . . . ‘), SQL (‘opta optb . . . ‘))
  - XOPT, XOPTS not supported



Integrated Translator - Migration

- COBOL
  - Nested procedures - omit DFHCOMMAREA, DFHEIBLK arguments on CALL and PROCEDURE statements
  - Comments - do not place before the CBL/PROCESS statement



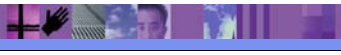
Integrated Translator, SQL Coprocessor Requirements

- CICS Integrated Translator
  - First shipped in CICS TS V2.1
  - IBM COBOL for OS/390 and VM, V2.2 with APAR PQ45462 or IBM Enterprise COBOL for z/OS and OS/390 V3.1
  - IBM Visual-Age PL/I for OS/390, V2.2.1 with APAR PQ45562 or IBM Enterprise PL/I for z/OS and OS/390 V3.1

- SQL Integrated Coprocessor
  - DB2 and UDB for OS/390 V7.1
  - COBOL for OS/390 and VM, V2.2 or IBM Enterprise COBOL for z/OS and OS/390 V3.1
  - IBM Enterprise PL/I for OS/390 and z/OS, V3.1

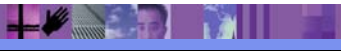
Compiler Currency in CICS TS V2.2

- COBOL
  - IBM COBOL for MVS and VM V1 (5688-197)
  - IBM COBOL for OS/390 and VM V2 (5648-A25)
  - IBM Enterprise COBOL for z/OS and OS/390 V3 (5655-G53)
- PL/I
  - OS PL/I V2.3 (5668-909, -910, -911)
  - IBM PL/I for MVS and VM V1 (5688-235)
  - IBM VisualAge PL/I for OS/390 V2 (5655-B22)
  - IBM Enterprise PL/I for z/OS and OS/390 V3 (5655-H31)
- C/C++
  - OS/390 C/C++ (5647-A01)
  - C/C++ for MVS/ESA (5655-121)
- Pre-LE compilers
  - No translator support in CICS TS V2.2 (COBOL and COBOL2 options removed)
  - Withdrawal of native run-time support in CICS TS V2.3
    - Will need run-time libraries distributed with LE to execute current load modules



Compiler Currency in CICS TS V2.3

- COBOL
  - IBM COBOL for OS/390 and VM V2 (5648-A25)
  - IBM Enterprise COBOL for z/OS and OS/390 V3 (5655-G53)
  - No translator support for pre-LE compilers
- PL/I
  - IBM Enterprise PL/I for z/OS and OS/390 V3 (5655-H31)
- C/C++
  - OS/390 C/C++ (5647-A01)
  - C/C++ for z/OS and OS/390 (5694-A01)
- Pre-LE compiled modules
  - Native run-time support withdrawn
    - Will need run-time libraries distributed with LE to execute current load modules

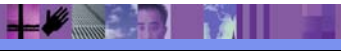


Some relevant dates

Operating System level	Announce	GA	End of Service
OS/390 V2.10	May 16, 2000	Sept 29, 2000	September 30, 2004
z/OS V1.1	October 3, 2000	March 30, 2001	March 31, 2004
z/OS V1.2	September 11, 2001	October 26, 2001	October 31, 2004
z/OS V1.3	February 19, 2002	March 29, 2002	March31, 2005
z/OS V1.4	August 13, 2002	September 27, 2002	March 31, 2007
z/OS V1.5	Preview: August 13, 2002	1Q2004	

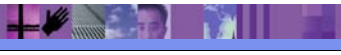
IBM Open Class library

- Present in z/OS 1.2, 1.3, and 1.4
- Built with standard linkages
  - Runs as all non Java™ on QR TCB
- Announced for phase out in z/OS 1.5
  - Source for IOC then available for a price
  - Recently advanced by one release
- Transition guide from IBM Open Class Library to C++ Standard Library available



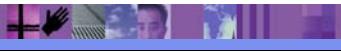
C++ Standard Library

- Implements ISO C++ 98 standard
  - 64 bit enabled
- Originally built with XPLINK only
  - Therefore unuseable in CICS TS
- NOXPLINK version for z/OS 1.2, 1.3 and 1.4 was made available in July 2003
  - Useable in CICS TS



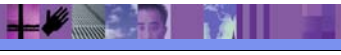
Parsing XML Documents

- Document Object Model(DOM) defines an object hierarchy, a tree model
  - Good points
    - Supports XML and HTTP
    - Very useful for repeated random access to a document
      - document editing
    - data retrieval
    - Platform and language neutral
  - Bad points
    - Entire document must be parsed and stored in memory
    - Unsuitable for document transformation applications
- Simple API for XML (SAX), an event based API for parsing XML
  - Good points
    - Does not need large amounts of memory
    - Document data is available as it is parsed
  - Bad point
    - No support for modifying or writing XML document data



COBOL and PL/I XML Parser

- Event-based interface (SAX-like)
- High performance, but non-standard interfaces
  - Gives capability to transform document contents to COBOL (PL/I) data structures
- Non-conforming
  - Well-formedness checks on basic grammar, but
  - No processing of internal DTD
    - Attribute values not normalized, though defaults supplied
    - No replacement text of internal entities unless predefined
  - Continuation allowed after some “fatal” errors
- Non-validating
- Multiple code pages supported
- Inbound XML only – no compiler facilities for XML creation at this time



XML Parsing – COBOL Implementation

- New XML PARSE verb

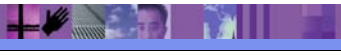
```
XML PARSE SANDWICH
   PROCESSING PROCEDURE TO-GO-OR-NOT-TO-GO
   ON EXCEPTION PERFORM SANDWICH-ERROR
   NOT ON EXCEPTION CONTINUE
END-XML.
```

New special registers to return results

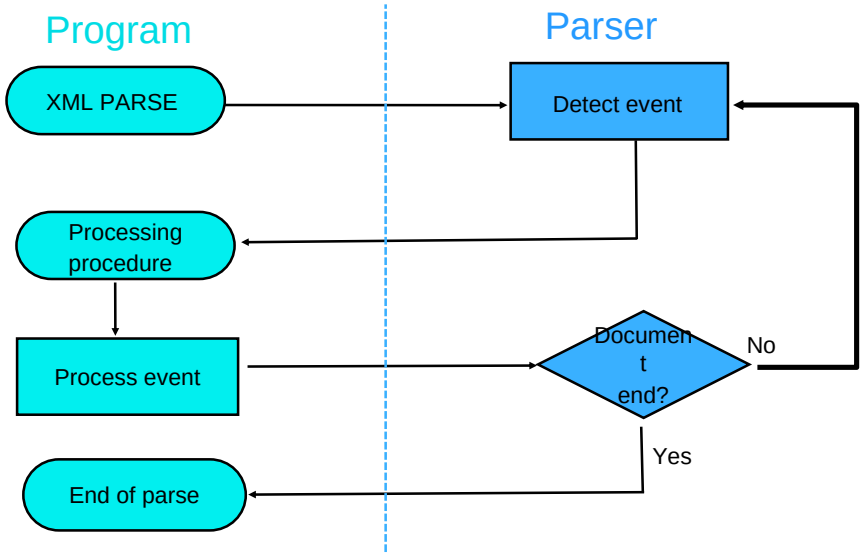
XML-CODE

XML-EVENT

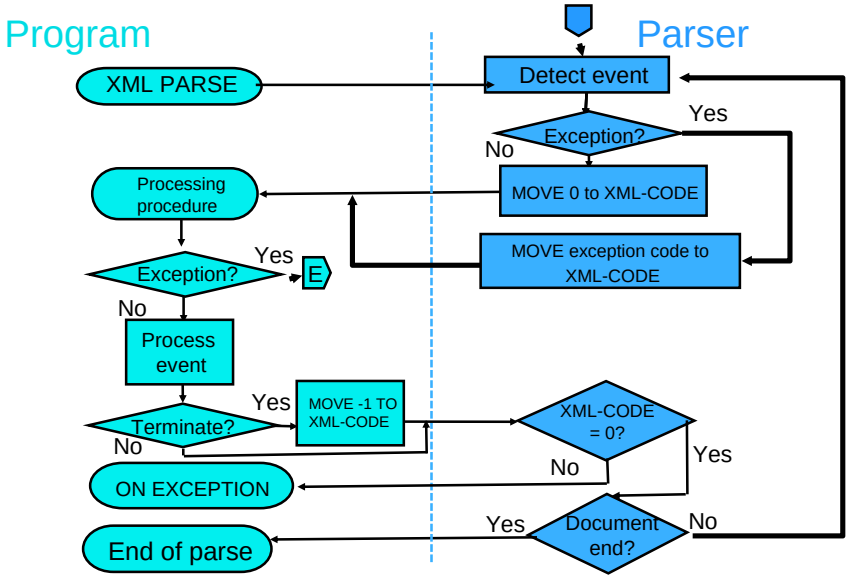
XML-TEXT, XML-NTEXT



XML parsing flow overview



XML parsing flow overview

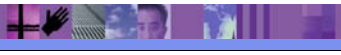


XML Parsing – PL/I Implementation

- New XML built-in function
  - PLISAXA with 5 arguments:
  - Event structure for 24 possible events
  - Pointer passed back to parsing events
  - Address of XML
  - Length of XML
  - Codepage of XML

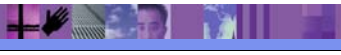
```
call plisaxa( eventHandler,
              addr(eventOrder),
              addrdata(xmlDoc(1).data),
               xmlDoc(1).len,
               xmlCodePage);
```

  - PLISAXB for parsing XML document in a file



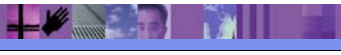
XML Document

```
<?xml version="1.0" encoding="ibm-1140" standalone="yes"?>
<!--This document is just an example-->
<sandwich>
  <bread type="baker&apos;s best"/>
  <?spread please use real mayonnaise  ?>
  <meat>Ham &amp; turkey</meat>
  <filling>Cheese, lettuce, tomato, etc.</filling>
  <![CDATA[We should add a <relish> element in future!]]>
  <listprice>$4.99 </listprice>
  <discount>0.10</discount>
</sandwich>
```



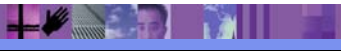
XML Document definition

```
 Process flag(i,i)
 Identification division.
   Program-id. xmlsampl.
 Data division.
  Working-storage section.
*****************************************************************
* XML document, encoded as initial values of data-items.        *
*****************************************************************
   1 xml-document.
   2 pic x(39) value '<?xml version="1.0" encoding="ibm-1140"'.
    2 pic x(19) value ' standalone="yes"?>'.
    2 pic x(39) value '<!--This document is just an example-->'.
    2 pic x(10) value '<sandwich>'.
    2 pic x(35) value '  <bread type="baker&apos;s best"/>'.
    2 pic x(41) value '  <?spread please use real mayonnaise  ?>'.
    2 pic x(31) value '  <meat>Ham &amp; turkey</meat>'.
    2 pic x(40) value '  <filling>Cheese, lettuce, tomato, etc.'.
    2 pic x(10) value '</filling>'.
    2 pic x(35) value '  <![CDATA[We should add a <relish>'.
    2 pic x(22) value ' element in future!]]>'.
    2 pic x(31) value '  <listprice>$4.99 </listprice>'.
    2 pic x(27) value '  <discount>0.10</discount>'.
    2 pic x(11) value '</sandwich>'.
   1 xml-document-length computational pic 999.
```



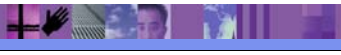
Definitions and XML Parse

```
*****************************************************************
* Sample data definitions for processing numeric XML content.   *
*****************************************************************
   1 current-element pic x(30).
   1 xfr-ed pic x(9) justified.
   1 xfr-ed-1 redefines xfr-ed pic 999999.99.
   1 list-price computational pic 9v99 value 0.
   1 discount computational pic 9v99 value 0.
   1 display-price pic $$9.99.

 Procedure division.
  mainline section.

     XML PARSE xml-document PROCESSING PROCEDURE xml-handler
       ON EXCEPTION
         display 'XML document error ' XML-CODE
       NOT ON EXCEPTION
         display 'XML document successfully parsed'
     END-XML
```



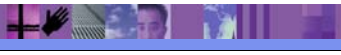
Output

```
*****************************************************************
*    Process the transformed content and calculate promo price.  *
*****************************************************************
     display ' '
     display '-----+++++***** Using information from XML '
         '*****+++++-----'
     display ' '
     move list-price to display-price
     display '  Sandwich list price: ' display-price
     compute display-price = list-price * (1 - discount)
     display '  Promotional price:   ' display-price
     display '  Get one today!'

     goback.
```



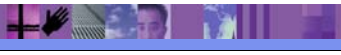
Outputting parsed data

```
*****************************************************************
*    Process the transformed content and calculate promo price.  *
*****************************************************************
     display ' '
     display '-----+++++***** Using information from XML '
         '*****+++++-----'
     display ' '
     move list-price to display-price
     display '  Sandwich list price: ' display-price
     compute display-price = list-price * (1 - discount)
     display '  Promotional price:   ' display-price
     display '  Get one today!'

     goback.
```



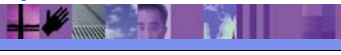
Event handler

```
  xml-handler section.
     evaluate XML-EVENT
* ==> Order XML events most frequent first
       when 'START-OF-ELEMENT'
         display 'Start element tag: <' XML-TEXT '>'
         move XML-TEXT to current-element
       when 'CONTENT-CHARACTERS'
         display 'Content characters: <' XML-TEXT '>'
* ==> Transform XML content to operational COBOL data item...
         evaluate current-element
           when 'listprice'
* ==> Using function NUMVAL-C...
             compute list-price = function numval-c(XML-TEXT)
           when 'discount'
* ==> Using de-editing of a numeric edited item...
             move XML-TEXT to xfr-ed
             move xfr-ed-1 to discount
         end-evaluate
       when 'END-OF-ELEMENT'
         display 'End element tag: <' XML-TEXT '>'
         move spaces to current-element
       when 'START-OF-DOCUMENT'
         compute xml-document-length = function length(XML-TEXT)
         display 'Start of document: length=' xml-document-length
             ' characters.'
       when 'END-OF-DOCUMENT'
         display 'End of document.'
       when 'VERSION-INFORMATION'
         display 'Version: <' XML-TEXT '>'
         display 'Unexpected XML event: ' XML-EVENT '.'
     end-evaluate.
```



Event Handler(continued)

```
        when 'ENCODING-DECLARATION'
          display 'Encoding: <' XML-TEXT '>'
        when 'STANDALONE-DECLARATION'
          display 'Standalone: <' XML-TEXT '>'
        when 'ATTRIBUTE-NAME'
          display 'Attribute name: <' XML-TEXT '>'
        when 'ATTRIBUTE-CHARACTERS'
          display 'Attribute value characters: <' XML-TEXT '>'
        when 'ATTRIBUTE-CHARACTER'
          display 'Attribute value character: <' XML-TEXT '>'
        when 'START-OF-CDATA-SECTION'
          display 'Start of CData: <' XML-TEXT '>'
        when 'END-OF-CDATA-SECTION'
          display 'End of CData: <' XML-TEXT '>'
        when 'CONTENT-CHARACTER'
          display 'Content character: <' XML-TEXT '>'
        when 'PROCESSING-INSTRUCTION-TARGET'
          display 'PI target: <' XML-TEXT '>'
        when 'PROCESSING-INSTRUCTION-DATA'
          display 'PI data: <' XML-TEXT '>'
        when 'COMMENT'
          display 'Comment: <' XML-TEXT '>'
        when 'EXCEPTION'
        compute xml-document-length = function length (XML-TEXT)
          display 'Exception ' XML-CODE ' at offset '
              xml-document-length '.'
        when other
          display 'Unexpected XML event: ' XML-EVENT '.'
      end-evaluate
      .
End program xmlsampl.
```

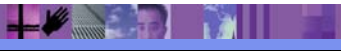


Output - XML added

```
Start of document: length=390 characters.
<?xml version="1.0" encoding="ibm-1140" standalone="yes"?>
Version: <1.0>
Encoding: <ibm-1140>
Standalone: <yes>
<!--This document is just an example-->
Comment: <This document is just an example>
<sandwich>
Start element tag: <sandwich>
Content characters: <  >
<bread type="baker&apos;s best"/>
Start element tag: <bread>
Attribute name: <type>
Attribute value characters: <baker>
Attribute value character: <'>
Attribute value characters: <s best>
End element tag: <bread>
Content characters: <  >
<?spread please use real mayonnaise  ?>
PI target: <spread>
PI data: <please use real mayonnaise  >
Content characters: <  >
<meat>Ham &amp; turkey</meat>
Start element tag: <meat>
Content characters: <Ham >
Content character: <&>
Content characters: < turkey>
End element tag: <meat>
Content characters: < >
```

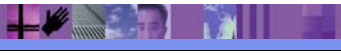
Output(continued)

```
<filling>Cheese, lettuce, tomato, etc.</filling>
Start element tag: <filling>
Content characters: <Cheese, lettuce, tomato, etc.>
End element tag: <filling>
Content characters: <  >
<![CDATA[We should add a <relish> element in future!]]>
Start of CData: <<![CDATA[>
Content characters: <We should add a <relish> element in future!>
End of CData: <]]>>Content characters: <  >
<listprice>$4.99 </listprice>
Start element tag: <listprice>
Content characters: <$4.99 >
End element tag: <listprice>
Content characters: <  >
<discount>0.10</discount>
Start element tag: <discount>
Content characters: <0.10>
End element tag: <discount>
</sandwich>
End element tag: <sandwich>
End of document.
XML document successfully parsed

-----+++++***** Using information from XML *****+++++-----

  Sandwich list price:  $4.99
  Promotional price:    $4.49
  Get one today!
```



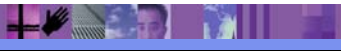
XML Parsing - Requirements

- CICS TS V1.3 or CICS TS V2.2 or later
  - IBM Enterprise COBOL for z/OS and OS/390 V3.1
  - IBM Enterprise PL/I for z/OS and OS/390 V3.1



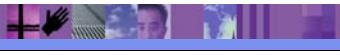
Unicode Support

- Unicode
  - All characters, all major languages
- In COBOL
  - New 'NATIONAL' data type
    - Unicode: UTF-16, CCSID 01200, big-endian
  - Character encoding *units* in COBOL
    - 1 byte for USAGE DISPLAY
    - 2 bytes for USAGE NATIONAL
  - Character encoding
    - EBCDIC – 1 encoding unit per character
    - UTF-16 – 1 or 2 encoding units per character (2 or 4 bytes)
- In PL/I
  - 'WIDECHAR' data type
  - 2 bytes per character
  - UTF-16, big-endian



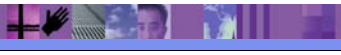
Unicode Support in COBOL

- Implementation
  - New ‘NATIONAL’ data type
  - NSYMBOL compiler option
    - Defines ‘N’ data type
    - DBCS (default) or NATIONAL
  - CODEPAGE compiler option (default 1140)
    - Alphameric and DBCS data items at run time
    - Alphameric, DBCS and NATIONAL literals
    - Default for XML documents encoded in EBCDIC
  - Code-page conversion functions
    - DISPLAY-OF converts from NATIONAL
    - NATIONAL-OF converts to NATIONAL



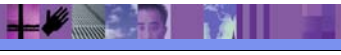
Unicode Support in PL/I

- Implementation -- 'WIDECHAR' data type
  - Most language functions
    - Conversion to and from other data types
    - Arithmetic and comparison, including mixed data types
  - Built-in functions
    - Most existing built-in functions support WIDECHAR variables
    - Built-in functions specific to WIDECHAR
  - Some limitations
    - No W string constants (WX only)
    - No WIDECHAR characters in source files
    - No WIDECHAR expressions in stream I/O
    - No automatic conversion in record I/O
    - No implicit endianness flags in record I/O

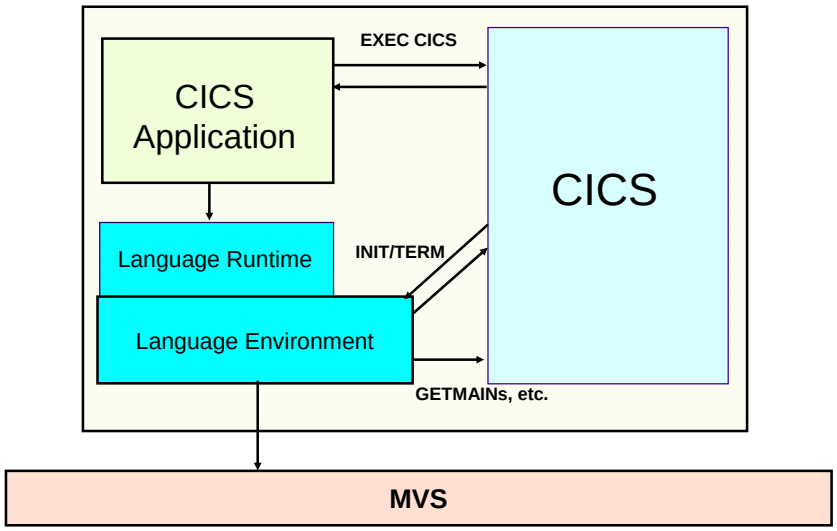


Unicode – software levels

- COBOL
  - IBM Enterprise COBOL for z/OS and OS/390 V3.1
- PL/I
  - Visual Age for PL/I V2.2 or
  - IBM Enterprise PL/I for z/OS and OS/390 V3.1
- DB2: V7 plus service
- IMS V6
- Integrated translator (CICS TS V2.2 or later)
- OS/390 V2.10 or Z/OS V1 at LE service level
  - PQ 52626 (all)
  - PQ 52338 (OS/390 V2.10)
  - PQ 53034 (z/OS V1.2)
- OS/390 Support for Unicode, 5647-A01

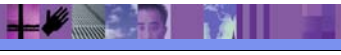


Where LE fits



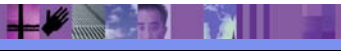
LE-CICS Enhancements

- CLER transaction
  - Display or print current global run-time options
  - Modify selected options
- Storage tuning exit
  - Collect storage tuning data without RPTSTG(ON)
  - Set STACK, LIBSTACK, HEAP, ANYHEAP, BELOWHEAP dynamically by enclave
  - Sample exit provided
- Dynamic storage tuning
  - SIT option AUTODST(YES)



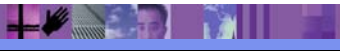
CLER

```
  CLER                                                          SUZY IYK2Z701

                Language Environment Region Level Runtime Options


Type in your Choices.

Runtime option      Choice      Possible choices.


TRAP           ==>  ON          ON, OFF
RPTOPTS        ==>  OFF         ON, OFF
RPTSTG         ==>  OFF         ON, OFF
ALL31          ==>  ON          ON, OFF
CBLPSHPOP      ==>  ON          ON, OFF
TERMTHDACT     ==>  TRACE       QUIET,MSG,TRACE,DUMP,UAONLY,UATRACE,UADUMP,UAIMM


When finished, press ENTER.


       PF1=Help     3=Quit     5=Current Settings    9=Error List
```



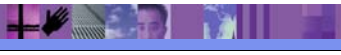
LE-CICS Enhancements...

- Dump and trace enhancements
  - Run-time options report in CEEDUMP
  - New reason codes for U4083
  - Control for storage around registers in CEEDUMP
  - Formatting of LE storage areas via CICS VERBX
  - RSA in to CICS trace
  - LE diagnostics in CICS transaction dump
  - LE control blocks formatted in transaction dump
  - Add LE trace component to CETR



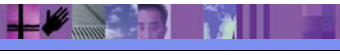
IBM Debug Tool

- Source-level view of point of failure
  - Diagnostics
  - Correction facilities
  - Re-execution without leaving debug session
  - Hookless debug for COBOL
- Supported for
  - COBOL
  - PL/I
  - C and C++
  - HPJava
- Access via
  - IBM VisualAge COBOL for Windows NT®
  - IBM VisualAge PL/I for OS/390
  - IBM C/C++ Productivity Tools for OS/390
  - VisualAge for Java, Enterprise Edition for OS/390, or
  - 3270



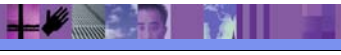
IBM Debug Tool...

- New in Version 2.1
  - Enhanced CICS save profile for breakpoints
  - Batch debug via TSO
  - Setup utility
  - Reliability improvements
- Later additions
  - More granularity
  - Disassembly view



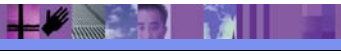
IBM Fault Analyzer for z/OS

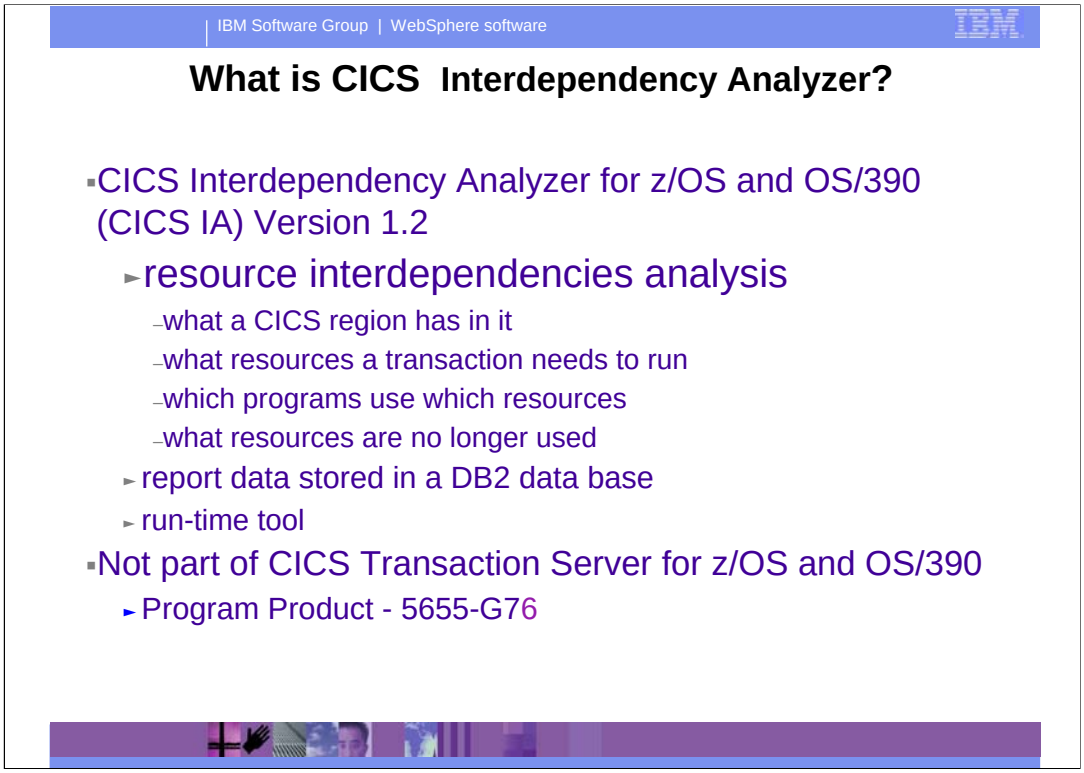
- Fault Analyzer V2.1 (GA July, 2001)
- Designed to do Error Capture and Reporting for operations, development and testing
  - Reduces skill necessary to diagnose an abend / dump
- MQ Series support
- CICS System Abend Support
  - Can re-analyze the dump after the initial capture/analysis (Permits post abend analysis when listing was not found at abend)
  - Complete user friendly full text error messages provided at analysis
  - Catalog of dumps maintained for future analysis and comparison
  - Handles COBOL, PL/I, C/C++ and Java
- WebSphere z/OS support



IBM File Manager for z/OS

- FM V2.1 (GA 7/2000)
- Designed to provide logical file viewing, editing, and copying for test and development
  - DB2, IMS, VSAM and QSAM file support
- ISPF Interface
- Online and Batch capabilities
- COBOL & PL/I Support
  - Ability to manipulate files utilizing COBOL & PL/I INCLUDE files in addition to COBOL COPYbooks

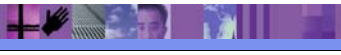


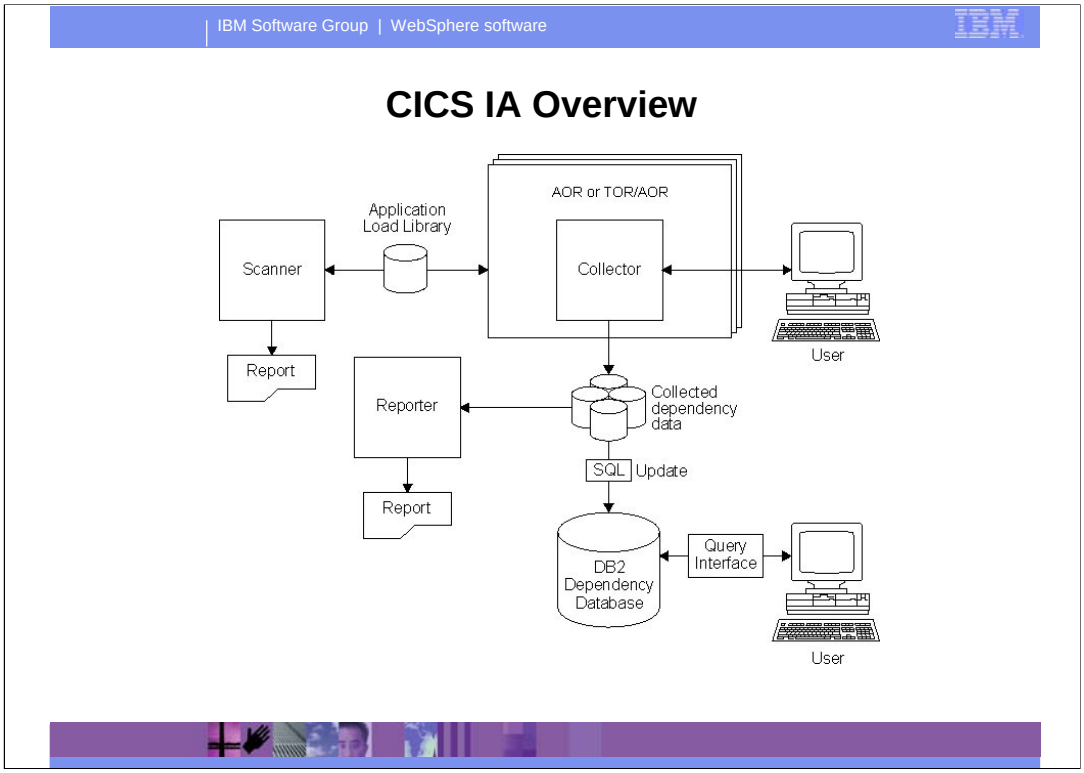


IBM CICS Interdependency Analyzer for z/OS and OS/390 (CICS IA) is a unique run-time tool that helps you identify resource interdependencies within your CICS system using report data stored in a DB2 data base. These reports, which can be interrogated on-line, will help you to improve your ability to maintain, enhance, and migrate your business applications.

CICS IA Benefits

- Helps to understand resource usage in your CICS systems to assist in:
  - making an informed decision on the best way to split workload and move applications to more CICS regions
  - fully exploiting the benefits of Workload Balancing across CICSPlex and Sysplex
  - improving your ability to maintain, enhance, and migrate your business applications
- Easy to use
  - Interface, familiar to any CICS customer
- Cross-System (Transactions, Programs, Applications,..)
- Optimized for run-time use





CICS IA automates the process of collecting the data on the interdependencies between CICS resources. CICS IA works in two ways - off-line and online.

Online Components

The Collector intercepts CICS commands which may be involved in an interdependency and records the details of the resources used. For efficiencies sake the data is stored in a data space, which under user control, may be offloaded to the Collected dependency data component. These data sets may be aggregated together and stored in a DB2 Database. The Query Interface provides the user with an online capability of interrogating the data using a comprehensive interface and gives a detailed level of understanding of relationships of:

what a CICS region has in it

what resources a transaction needs to run

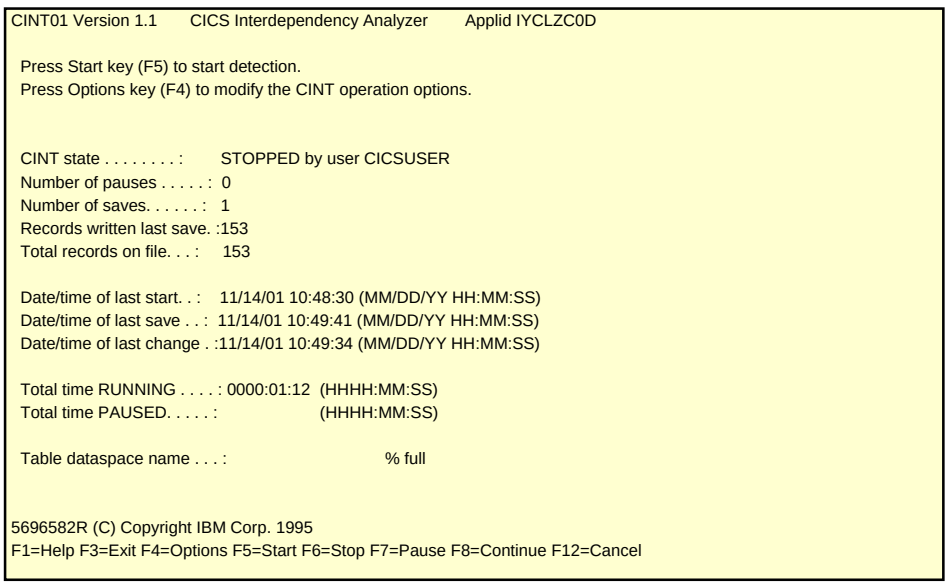
which programs use which resources

what resources are no longer used

Off-line Components

The Reporter presents the dependency information collected by the Collector in a structured format. The Scanner scans the load module data sets detecting EXEC CICS commands that may cause transaction resource dependency and produces a printed report.

CICS IA - CINT Transaction, Collector Component



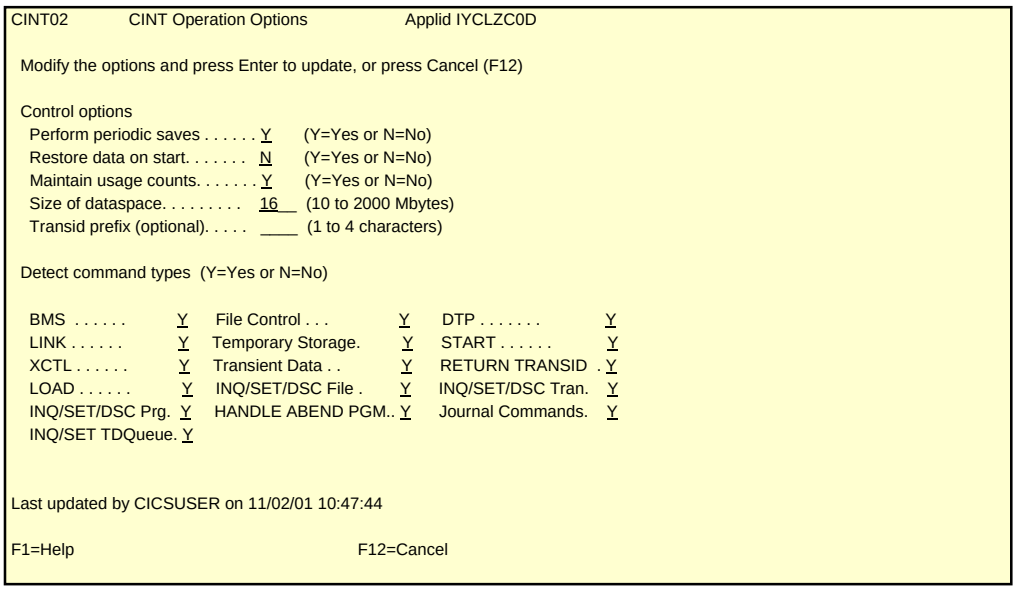
You can use the Collector in real-time to detect transaction resource dependencies in a running CICS region, and to save details of the dependencies in an MVS data space. This data is subsequently saved to DASD. The Collector consists of:
A control transaction, CINT

An autosave transaction, CINB

A global user exit program
The data is collected by the global user exit program at exit points XEIOUT and XEIIN. This exit program intercepts all the EXEC CICS commands that are needed to deduce dependencies. The exit program coexists with any other exit programs at the same exit points. (It can be placed before or after other exit programs, without any of the exit programs being affected.)
CINT gives the screen to start and stop the statistics gathering as in the screen-shot depicted in this foil.

PF4 gives the options

CICS IA - Controlling the Collector



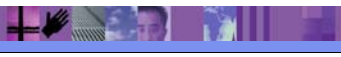
You can monitor and control the Collector through the CINT transaction, which enables you to start, pause, continue, and stop the collection of dependency data into the tables in the data space. Using the CINT transaction, you can also specify for which dependency commands, and for which transactions, data is to be collected.

The options that you specify to control the Collector for a CICS region are preserved in a recoverable VSAM control file.

This screen gives the options which can be selected for the next time the facility is started.

CICS IA - Wich transactions have run

```
CICS IA V1.1    CICS Interdependency Analyzer for z/OS and OS/390     CIUM100

For your query    WHICH TRANS ARE IN REGION CORD

   HOME   TRAN                  HOME   TRAN
   SYSID                        SYSID
   CORD   CINT
          EHLP
          EQRH
          EQRM
          EQSS
          VA10
          VA12
          VA20
          VA21
          VA30
          VA33
          V200
          V220
          V800
          V884

 No more details to display
PF3=End PF4=Exit PF7=Up PF8=Down              Page  1 of  1
```



Here we see that in CICS region CORD the following transactions were used.

CICS IA - Which Resources are in an Application

```
CICS IA V1.1     CICS Interdependency Analyzer for z/OS and OS/390      CIUM300

For your query  WHICH RESOURCES ARE IN Test application 1

 In      Tran    Program   Links/XCTL    Strts DTP to Remote File    Map      Inquire      Retn
Regn                        Loads        Tran  Tran  Sysid                    /Set         Tran
CORD  VA10 CAMA100C CAMA120C
      VA10 CAMA100C CAMA800C
      VA10 CAMA100C                                                CAM1001
      VA10 CAMA120C CAMI725C
      VA10 CAMA120C                                                         CAM1201
      VA10 CAMA120C                                                CAM1201
      VA10 CAMA120C                                                                  VA12
      VA10 CAMA800C                                                CAM8001
      VA10 CAMA800C                                                         EZPSCCIL
      VA10 CAMA800C                                                                  V800
      VA10 CAMI725C                                                         CAMIMROC
      VA10 CAMI725C                                                         EZPSCCIL
      VA12 CAMA100C CAMA115C
      VA12 CAMA100C CAMI725C
      VA12 CAMA100C                                                CAM1001

     PF3=End   PF4=Exit   PF7=Up   PF8=Down                                  Page  1 of  7
```

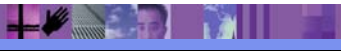
... the following output screen is displayed.

The screen shows the 1st of 7 pages of information gathered for the running of this application.

This is just to give an idea of the sort of things CICS IA can do.

Summary

- Positioning
- Integrated CICS Translator, SQL Coprocessor
- Compiler currencies
- Enterprise compilers for COBOL and PL/I
  - Inbound XML parser support
  - Unicode support
- LE enhancements
- IBM Debug Tool
- IBM Fault Analyzer
- IBM File Manager
- CICS Interdependency Analyzer




IBM Software Group
IBM Rich Media Presentations
Thank you for viewing
Further resources can be located
via the IBM CICS Website
www.ibm.com/cics
WebSphere software
business on demand software

Trademarks

- The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:
  - AIX
  - CICS
  - CICS Transaction Gateway
  - DB2
  - DFSMS
  - IBM
  - IMS
  - Language Environment
  - OS/390
  - RISC System/6000
  - RACF
  - RMF
  - S/390
  - VisualAge
  - WebSphere
  - z/OS
  - zSeries
- Java and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, othe countries, or both.
- SUN is a registered trademark of Sun Microsystems, Inc. in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a registered trademark owned by Linus Torvalds.
- Microsoft, Windows, Windows NT, Windows 2000 and the Windows logo are trademarks of Microsoft Corporation in the United States, othe coutries, or both.
- Other company, product or service names may be trademarks or service marks of others.