



IBM Software Group

## CICS TS V2.3 Presentation Guide



@business on demand software

This presentation explains how the CICS Transaction Server for z/OS V2.3 further enhances the functions and value of the world's most popular transaction monitor

CICS has led the way in the implementation of new technology to support the transaction processing of your mission critical business applications. This is just one reason why CICS is the leader in large scale transaction processing today and is probably one of the reasons why you originally chose CICS as your transaction manager. CICS has evolved even further into an e-Business application server, providing improved application enabling capabilities as well as powerful web and Java support. This presentation will cover the value benefits and functions of the CICS Transaction Server for z/OS.

## Acknowledgements

- The following are trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM, CICS, CICS/ESA, CICS TS, CICS Transaction Server, DB2, MQSeries, OS/390, S/390, WebSphere, z/OS, zSeries, Parallel Sysplex.
- Java, JavaBeans, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Other company, product, and service names and logos may be trademarks or service marks of others.



During this presentation we may use some trademarks or abbreviations. They are shown here.

## Agenda

- CICS Strategy
- CICS Transaction Server for z/OS V2.3
  - ▶ e-Business Access to CICS Applications
  - ▶ Application Modernization
  - ▶ Enterprise Management
- Planning Information for CICS TS 2.3
- Summary of CICS TS 2.3
- Appendix: Useful Information

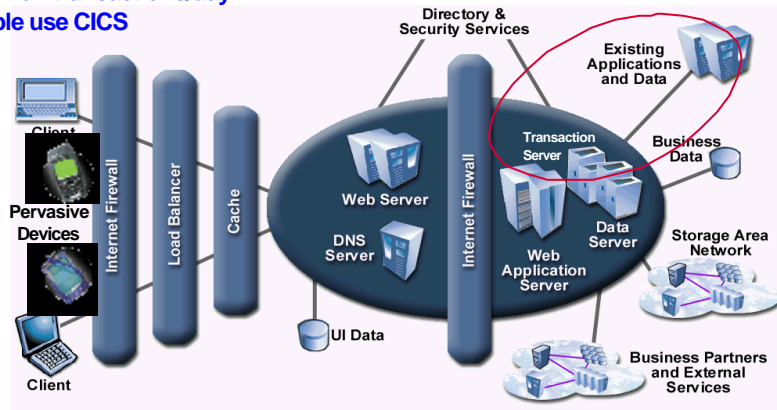
During this presentation we will first review the CICS strategy. This will cover how this strategy will create value to our customers and focus areas are for new function. We will review how much has already been delivered in pursuit of this strategy and what is the roadmap for future function.

We will then review the new function in CICS TS 2.3 in the areas of Java and EJB support and improvements in application modernisation and enterprise management.

We will then deal with planning for the new product and summarise what we have covered

- ✓ Over 30 years and \$1 Trillion invested in Applications ... IDC
- ✓ Over \$1 trillion processed/day
- ✓ Over 30 billion transactions/day
- ✓ Most people use CICS

## CICS Today ...



*Among the highest revenue earning software products in the world*

For over 30 years, CICS has led the way in the implementation of new technology to support the transaction processing of your mission critical business applications.

Many of today's largest enterprises, having assessed the opportunities and advantages presented by e-business technologies are now aggressively and decisively implementing e-business solutions to gain advantage in the marketplace.

An enterprise moving to the e-business approach will be looking to build new applications as well as to extend their existing, proven, core business applications. This new version of CICS TS provides a platform from which an enterprise can evolve towards the adoption of e-business exploiting all the advantages of the evolutionary approach while capturing new opportunities derived from the latest industry-standard e-business technologies.

## Customer Comments on CICS V2

- "The direction CICS is taking with EJB is right for us as it provides us with a highly scalable and reliable application server. We already know of the great reliability of CICS, having been a user for over 30 years. Now, it is also demonstrating its great scalability and, with the help of Version 2 EJB, we are integrating our existing and future applications."
- "CICS TS 2.2 is rock solid. Everything we have tested has performed as expected with no problems."
- "Benchmark results were great. Today, we have first 200 Java-CICS transactions in production environment! Everything works fine. Customer is very satisfied with the performance, stability, functionality etc. We are continuing writing Java-CICS transactions and there is no doubt that Java in CICS is one of the best software architecture choices available today."



This foil shows some custom comments we have received for CICS TS V2

## Strategic Themes for CICS TS v2

### **e-Business Access to CICS Applications**

- ▶ enable re-use of CICS applications within broader e-Business scenarios, via standard APIs and protocols

### **Application Modernization**

- ▶ enable enhancement of existing applications, and construction of new applications, using contemporary programming languages, constructs and tools

### **Enterprise Management**

- ▶ enable effective management of large runtime configurations via modern user interfaces, so that demanding service level objectives can be met



We have three strategic themes for CICS TS V2. They are e-Business Access to CICS Applications to enable re-use of CICS applications within broader e-Business scenarios, via standard APIs and protocols, Application Modernization

enable enhancement of existing applications, and construction of new applications, using contemporary programming languages, constructs and tools

and Enterprise Management to enable effective management of large runtime configurations via modern user interfaces, so that demanding service level objectives can be met

Lets consider these in the context of CICS TS V2

## CICS TS 2.3 Continues the Strategy ...

	e-Business Access to CICS Applications	Application Modernization	Enterprise Management
<b>CICS TS V2.2</b>	Java SDK 1.3.1 ECI over TCP/IP LINK3270 Bridge XML capability for COBOL and PL/I	EJB Security (Roles) support JNDI (LDAP service provider) WAS Appl Assembly Tool Simplified EJB deployment Remote debug for Java Record level sharing for FC JDBC 2.0	Signon retention CF mngd rebuild /duplexing DB2 group attach CICS-DB2 Performance RMI purge CEMT INQ DB2 FC GLUE CPSM NT Agent
<b>CICS TS V2.3</b>	Java evolution Java SDK 1.4.1 EJB hardening SOAP for CICS	Common Client Interface JCICS APIs for WEB, DOCUMENT and EXTRACT  End-to-end debug for Java and COBOL DB2 restart-light support  Samples supplied compiled	CPSM definitional WUI WLM for LINK3270 Bridge CPSM new function support CPSM architectural limits  MRO Timeout Display TN3270 IP address Java and EJB trace/ srvcblty Statistics enhancements NLS support for GB18030 Info Center enhancements

We have delivered various functions in V2.2 in support of these themes.

Leading functions are support for JAVA SDK 1.3, and support for sessions beans conforming to the J2EE Enterprise JavaBeans 1.1 programming model.

Functions to aid efficient application development include an enhanced 3270 bridge, an integrated translator for use with COBOL and PL/I, CICS COBOL and PL/I XML application capability, and enhancements to function shipping of remote file requests. Improvement to connectivity include support for external call interface over TCP/IP, improved CICS exploitation of TCP/IP services, connection optimization, and support for VTAM LU alias facility. Enhancements in the area of availability include sign-on retention for persistent sessions, automatic restart of CICS data servers and system managed rebuild of coupling facility structures. DB2 users benefit from CICS-DB2 performance improvements, DB2 group attach and the RMI purge option.

## CICS TS V2.3 new features

### e-business Access to CICS

- f* Java evolution including the SDK 1.4.1
- f* EJB hardening and performance
- f* Enhancements to IIOF authentication and encryption

### ▪ Application Modernization

- f* New JCICS classes for the CICS APIs WEB, DOCUMENT, and EXTRACT
- f* New CCI Connector for CICS TS
- f* Interactive end-to-end debugging enhancements
- f* DB2 restart-light and JDBC support
- f* SOAP for CICS feature

### ▪ Enterprise Management

- f* CICSplex SM enhancements
  - Definitional web user interface
  - Workload management for the LINK3270 Bridge

Lets begin with the theme of e-business access to CICS



## Java evolution - exploiting the latest Java SDK 1.4.1

### IBM Developer Kit for z/OS, Java 2 Technology Edition V1.4.1

*f* full function, passed the Java compatible test suites, carries Java compatible logo

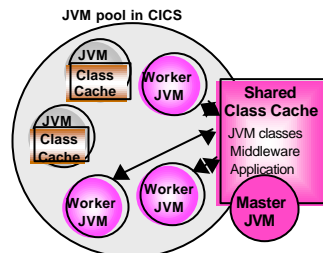
#### Shared class cache

*f* Multiple JVMs can share a single class cache

- JVM base classes (Primordial)
- Trusted middleware classes
- Sharable application classes - user applications

*f* JVMs in CICS

- Master JVM using a JM TCB
  - One per CICS region
  - Can be started at CICS startup
  - Initializes and owns the shared class cache and JIT compiled code
- Worker JVM using J8/J9 TCBs
  - Run Java applications and continuously reused
  - Uses the classes loaded in the shared class cache
  - Working storage for objects, static variables and non-shareable classes
  - Reduced storage footprint, startup time, and time for JVM reset



CICS Transaction Server for z/OS, Version 2 Release 3 supports the JVM provided by the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.1, which features the persistent reusable JVM technology. This JVM includes several optimizations designed for the execution of CICS transactions.

The first one is the ability for JVMs to share a cache of commonly-used class files, that are already loaded. This enables faster JVM startup and reduced cost of class loading. When a new JVM that shares the class cache is initialized, it can use these pre-loaded classes instead of reading them from the file system. Also, if the JVM performs just-in-time (JIT) compilation for any of the classes, it can write the results back to the shared class cache, and other JVMs can then use the compiled classes. All the heap data (objects and static variables) are owned by the individual JVMs; this maintains the isolation between the applications being processed in the JVMs.

The shared class cache is initialized by a JVM referred to as the master JVM. The master JVM cannot be used to run Java programs; it exists only to initialize and own the shared class cache. The master JVM obtains shared memory in which its system heap is allocated. The system heap contains class files that can be shared by all the worker JVMs, and the rest of the shared memory contains other information that is common to the master and worker JVMs. The JVMs that share the class cache are referred to as worker JVMs, and they can be used to run Java applications. The worker JVMs use the classes loaded in the shared class cache, instead of having to load these classes from the file system. Although the worker JVMs share the class cache, each worker JVM owns all the working data (objects and static variables) for the applications that run in it. This helps to maintain the isolation between the Java applications being processed in the system.

CICS supports one active shared class cache in each region. The shared class cache can support the majority of the JVMs in each region. Some of the JVMs in the region might not be suited to sharing the class cache, because they are debug JVMs used for problem diagnosis, or because they have an inappropriate level of reusability. These JVMs can still run as standalone JVMs, and have their own cache of classes in their storage heaps.

Use of the shared class cache improves the performance of JVMs in two respects:

- Average startup time for a new JVM is reduced, because shareable classes will, in most cases, have been preloaded in the shared class cache.
- The memory used by shareable classes is reduced, because there is a single copy of each class in the shared class cache, rather than one copy for each JVM.

## Java evolution - increasing performance

### ▪ Reduced JVM startup time

- f CICS TS V2.2 builds middleware infrastructure for EJB + CORBA requests for JVMs
  - Not required for basic Java programs in CICS
    - Overhead in CPU, storage and garbage collection
- f CICS 2.3 JVMs build the minimum set of objects to process a Java transaction
  - EJB and CORBA infrastructure build when first required
  - Faster JVM startup times
  - Reduced objects in heap, leading to reduced time for reset
  - Reset times are lower because there are fewer objects in the heap
- f Also delivered on CICS TS V2.2 in APAR PQ68286

### ▪ Enhanced mechanism for allocating free JVMs

- f Avoids where possible having to stop restart JVMs to match profile + execution key
  - More efficient dispatching of free JVMs
  - Wait some Java programs based recent usage and urgency
- f Results in greater overall throughput of Java programs

The performance of enterprise beans has been improved. In particular, when a JVM is reset some objects are cached for reuse, which greatly improves performance. In previous releases of CICS, an EJB deployed JAR file was considered to contain only application code. In CICS TS for z/OS, Version 2.3, the CICS-generated code within the deployed JAR file (the implementation of the beans' home and component interfaces) is treated as middleware. This means that some objects that were previously held in the application heap and discarded when the JVM was reset have become long-lived middleware heap objects. Now, when a JVM is reused to process requests against the same enterprise beans, the cached objects are reused, which greatly improves performance.

CICS Transaction Server for z/OS, Version 2 Release 3 implements a shared class cache for the JVM. The shared class cache is created using the JVMs' -Xjvmset option. Previously, classes were cached in the appropriate storage heap in each JVM. Now, multiple JVMs can share a single cache of class files that have already been loaded, including some that have been optimized by compilation. The shared class cache replaces the system heap and the application-class system heap for those JVMs, and it can contain middleware and application classes. JVMs that use the shared class cache start up more quickly, and have lower storage requirements, than JVMs that don't.

A new selection mechanism for CICS Transaction Server for z/OS, Version 2 Release 3 enables CICS to make more efficient use of free JVMs when the number of JVMs is at its maximum limit. The mechanism is also used when MVS storage is severely constrained, because in this situation CICS behaves as if the number of JVMs is at its maximum limit. The new mechanism aims to avoid mismatches (the allocation of a JVM with the wrong profile), and steals (the allocation of a JVM in the wrong execution key), *except* in situations where mismatching or stealing a JVM is actually the most advantageous course of action. With the new mechanism, when the number of JVMs is at the MAXJVMTCS limit, or MVS storage is severely constrained:

If there are free JVMs but none have the correct profile and execution key for an incoming request, the selection mechanism decides whether the request should wait for a JVM with the correct profile and execution key, rather than take, destroy and re-initialize one of the free JVMs. CICS can then make the request wait until a suitable JVM is available, and the free JVMs are left to be used by other requests for which they are more appropriate.

If there are no free JVMs, requests are queued. When a JVM becomes free, it does not have to be assigned to the first request in the queue. CICS examines the queue, and if it finds a request that needs a JVM of that profile and execution key, it assigns the free JVM to that request. If none of the requests need a JVM of that profile and execution key, the selection mechanism looks for a request where it will be an advantage to destroy and re-initialize the free JVM, and give the replacement JVM to the request. If it does not find such a request, the JVM is kept free to await a more appropriate request.

## Java evolution - increasing flexibility and faster resets

### Three modes of reuse for the JVM:

1. **Single use** - destroyed at task termination
2. **Resettable** - undergoes a JVM reset between transactions
3. **Continuous** - no reset between transactions – recommended for production

The continuous JVM has a greater transaction throughput and lower CPU usage than the resettable JVM

CICS TS for z/OS, Version 2.3 introduces a new level of reusability for Java virtual machines (JVMs). The previous release supported:

#### **The single-use JVM**

The JVM, and any application state, are destroyed after use and cannot be reused.

#### **The resettable JVM**

The JVM can be reused, and it is reset after each use. During the JVM reset, any application state is destroyed, and checks are made for any actions during the last use of the JVM that modified the internal state of the JVM in a way that cannot be properly reset. If any such actions are identified, the JVM is destroyed rather than being made available for reuse.

This release introduces support for:

#### **The continuous JVM**

The JVM can be reused, but it is not reset after each use. This means that the JVM's internal state is not checked, and application state can remain in the JVM across reuses.

The continuous JVM is an enhancement to CICS support for the JVM provided by the IBM Software Developer Kit for z/OS, Java 2 Technology Edition, Version 1.4.1, which features the persistent reusable JVM technology.

The continuous JVM has a greater transaction throughput and lower CPU usage than the resettable JVM, because it is not performing a reset between program invocations. If your applications have been designed and tested to ensure that they can run in a continuous JVM without causing any undesirable consequences for subsequent program invocations in the same JVM, then they can benefit from this performance enhancement.

## Java evolution - improving robustness

### ▪ Protection against short-on-storage conditions

- f CICS tracks MVS storage requests (Batch LE)
- f Normal CICS short-on-storage protection when fixed storage cushion breached
  - eg. Notify domains, wait transactions, throttle creation of new JVMs by acting as if MAXOPENTCBs was reached

### ▪ Storage protection for Java programs

- f Extends CICS storage protection
- f USER key mode for Java applications can be specified on Java PROGRAM RDO
  - CICS selects or creates a JVM that runs in user key
  - J9 TCB (new type) used for the JVM
  - MVS storage for LE enclave obtained in user key
  - By default enterprise beans run in USER key
  - Class cache is also in user key (not protected) to enable reuse by J8 + J9 JVMs
  - note: user key was the default RDO even at CICS TS V2.2
- f CICS key programs continue to run on J8 TCB
  - may be required if TASKDATA storage is in CICS key

JVMs use storage that is managed by MVS(TM), and so they cannot make use of the storage management facilities available from the existing CICS storage manager. For CICS Transaction Server for z/OS, Version 2 Release 3, a new storage monitor is provided for MVS storage.

As JVMs make requests for MVS storage, the storage monitor checks whether the availability of MVS storage has dropped below pre-set thresholds and notifies CICS when this is the case. Once CICS has been notified that MVS storage is constrained, the actions it takes, depending on the seriousness of the situation, are as follows:

- When MVS storage is constrained, CICS deletes JVMs in the JVM pool that are not currently in use, together with their TCBs. However, new JVMs can still be created for incoming requests.
- When MVS storage is severely constrained, CICS temporarily prevents the creation of new JVMs for incoming requests, and behaves as though the MAXJVMTCBS limit has been reached and the JVM pool is full. CICS then terminates all JVMs as soon as they finish running their current Java programs. If limited MVS storage is available, and the storage monitor is still receiving requests from CICS to create JVMs, it queues any such requests that cannot obtain sufficient MVS storage.
- When CICS manages to reduce its use of MVS storage by these methods, and the availability of MVS storage has risen above the pre-set thresholds, the storage monitor informs CICS that it can return to normal operation. As CICS returns to normal operation, operator messages inform you when MVS storage is no longer constrained.

You can now run Java programs in a JVM that executes in user key. A new type of open TCB, the J9 TCB, is used for these JVMs. Running applications in user key extends CICS storage protection, and the J9 TCB uses less storage below the line than other TCBs.

When you set the EXECKEY parameter on the PROGRAM resource definition for a Java program to USER, a J9 TCB is used for the JVM, and MVS storage is obtained in user key. When the EXECKEY parameter is set to CICS, a J8 TCB is used, and MVS storage is obtained in CICS key.

The default for the EXECKEY parameter is USER. As running applications in user key extends CICS storage protection, it could be beneficial to let most of your Java programs run in a JVM in user key. However, you might need to execute a JVM in CICS key if the program that uses the JVM is part of a transaction that specifies TASKDATAKEY(CICS). If this is the case, you need to invoke the JVM using a PROGRAM resource definition that specifies EXECKEY(CICS).

For enterprise beans, CIRP (the default transaction for REQUESTMODEL definitions) specifies TASKDATAKEY(USER), and the PROGRAM resource definition for DFJIIRP (the default request processor program) specifies EXECKEY(USER), so by default enterprise beans run in user key.

The master JVM that initializes the shared class cache is invoked in user key, so that worker JVMs that were invoked in user key can read and write to the shared class cache. Even if all the worker JVMs that share the class cache are invoked in CICS key, the master JVM and the shared class cache are still in user key.

## Java evolution - enhancing the management of Java

### ▪ Invoke multiple Java programs in the same CICS task

- f Removal of one JVM per stack restriction
- f Enables development of Java programs without worrying about the calling stack
- f A strategy for piece-by-piece replacement of COBOL / PLI / C

### ▪ Flexible tracing from the Java environment

- f `USEROUTPUTCLASS=classname` allows STDERR and STDOUT to be redirect
  - e.g. to CICS transient data queue
  - consolidate / merge log files, prefix / suppress records
- f STDOUT primarily used for development / debug
- f STDERR useful in production for early problem detection and automation
- f Activate JVM trace dynamically using CETR

### ▪ JVM profile management simplified

- f JVM profile **and** properties files stored on the HFS
  - No longer split over PDS and HFS files
- f Referenced by Java PROGRAM RDO
- f Bring in a new JVM profile dynamically using CEMT

You can now use the CICS program-link mechanism to invoke more than one Java program within the same CICS task. The one-JVM-per-stack restriction has been removed.

Previously, a stack of programs formed by a succession of EXEC CICS LINK commands, or JCICS program invocations, within the same CICS task, could not contain more than one JVM. As a CICS task could only use one JVM, applications designed in this way could only contain one Java component.

Now, multiple JVMs can be allocated to a CICS task, so you can create an application that links together multiple Java programs in the same CICS region. You can also convert your existing complex applications piece by piece, by replacing programs written in other languages with programs written in Java.

Enterprise beans can now link to another Java program within the same CICS task. However, a single CICS task still cannot contain more than one enterprise bean, because CICS treats an execution of an enterprise bean as the start of a new task. You can create an application that includes more than one enterprise bean, but the application will not operate as a single task.

You can now control tracing for JVMs using the CICS-supplied transaction CETR. Previously, tracing for JVMs could be controlled using the `ibm.dg.trc.external` system property in the JVM properties file, and tracing was active for the lifetime of the JVM. Now, using CETR, you can set tracing options for each transaction that uses the JVM. You can set the JVM trace options for each trace level, turn tracing on or off at each level for transactions, or reset to the supplied settings.

CICS Transaction Server for z/OS, Version 2 Release 3 includes features to help you manage JVM profiles more efficiently.

JVM profiles are now stored as text files on HFS, rather than as a PDS member. This means that they can be edited using any standard text editor.

The new `JVMPROFILE` option on the EXEC CICS and CEMT [SET PROGRAM](#) command enables you to change the JVM profile that a program uses during a CICS run, without having to re-install the program definition. Any instances of the program that are currently running in a JVM with the old JVM profile are unaffected, and are allowed to finish running their current Java program. New instances of the program will use a JVM with the new JVM profile that you have specified.

## J2EE EJB session beans

### ▪ Performance enhancements

- f* WebSphere AAT generated code in deployed EJB-JAR loaded as middleware
  - Implementation of the beans' home and component interfaces
  - Reused when a JVM reused for same enterprise beans
- f* JNDI cache
  - Improves the performance of JNDI lookups of Enterprise Javabeans
- f* Improved DJAR loading performance
  - EJB 2.0 deployment descriptor toleration allows use of EJB 2.0 tools, such as WebSphere Studio V5 for the deployment of EJB 1.1 session beans

### ▪ Enable and disable CORBASERVERs

- f* New SPI to dynamically quiesce CORBASERVERs to install new enterprise beans
  - No need to shut down CICS
- f* Recommended procedure to maintain continuous availability

### ▪ Tracking IOP tasks

- f* New 'stack' counts and request information to trace enterprise bean work in CICS
- f* Track chain of requests multiple method requests
- f* New SPIs for INQUIRE / SET WORKREQUEST

Important improvements have been made to enterprise bean performance. The performance of CICS TS V2.3 as an EJB server is better than that of CICS TS V2.2. In CICS TS V2.3, the generated code from within the deployed ejb-jar is treated as middleware code. As a result, various items previously held in the application heap are now long-lived middleware heap items and are not discarded at JVM tidyup; they persist in middleware caches and are reused when the JVM itself is reused. This caching also means that a CICS JNDI cache can be used to greatly improve the performance of JNDI lookups under CICS TS V2.3, provided the potential consequences concerning data integrity are understood.

Changes have been introduced in CICS TS V2.3 that offer better performance on installing a DJAR (also delivered on CICS TS V2.2 by the service channel, in APAR PQ62280), and improved diagnostics in the event of a failure on DJAR installation. These changes also mean that EJB 2.0 deployment descriptors can now be read so that EJB 2.0 JAR files are now tolerated. This allows the use of EJB 2.0 tools, such as WebSphere Studio V5.0, for the deployment of session beans; however, CICS TS V2.3 ignores information not relevant to EJB 1.1.

A facility has been introduced to allow a CorbaServer to be enabled and disabled. This makes systems administration easier, since in order to apply maintenance (to alter the set of installed DJARs, for example), there is no need to shut down the whole CICS region. Instead, only the EJB workloads for the CorbaServer itself need to be dynamically quiesced, the maintenance applied, and then the workload brought back up.

## IIOOP authentication and encryption enhancements

### ▪ Authentication

- f Supports the following authentication for IIOOP clients
  - SSL client certificate authentication
    - Supported in CICS TS V2.2
  - Asserted identity authentication
    - Used when an IIOOP client communicates with the target server through an intermediate server

### ▪ Encryption

- f CICS uses cipher suites supported by z/OS or OS/390
- f Enables secure inter-operation between CICS and IBM and non-IBM systems
- f Supports a variety of authentication protocols

CICS Transaction Server for z/OS, Version 2 Release 3 adds new function to the existing support for the Secure Sockets Layer (SSL). For the Internet Inter-Orb Protocol (IIOOP), you can use asserted identity authentication between client and server, to identify and authenticate the client to the target application. For the Hypertext Transfer Protocol (HTTP), and for IIOOP, you can select the level of encryption used between client and server.

Previous releases of CICS supported SSL client certificate authentication for IIOOP clients; CICS Transaction Server for z/OS, Version 2 Release 3 supports asserted identity authentication protocol for IIOOP clients.

Asserted identity authentication can be used when an IIOOP client communicates with the target server through an intermediate server. The target server verifies that the intermediate server can be trusted to authenticate its clients. When the intermediate server receives a request, it authenticates the client using whatever authentication protocol is appropriate. If the client is successfully authenticated, the intermediate server passes the request to the target server. Because the target server trusts the intermediate server to authenticate the client, it makes no further checks of the client's authenticity before processing the client's request.

There are many different algorithms which can be used for encrypting data, and for computing the message authentication code. Some provide the highest levels of security, but require a large amount of computation for encryption and decryption; others are less secure, but provide rapid encryption and decryption. The length of the key used for encryption affects the level of security-- the longer the key, the more secure the data.

To allow users to select the level of security that suits their needs, and to enable communication with others who may have different needs, SSL defines **cipher suites**, or sets of ciphers. When an SSL connection is established, the client and server exchange information about which cipher suites they have in common. They then communicate using the common cipher suite that offers the highest level of security. If they do not have a cipher suite in common, then secure communication is not possible.

CICS Transaction Server for z/OS, Version 2 Release 3 extends the possibilities for secure interoperation between CICS and other IBM and non-IBM systems. CICS supports a variety of authentication protocols suitable for use in all environments, ranging from highly controlled secure intranets, where -- in many cases -- simple authentication protocols provide sufficient protection, to the open environment of the internet, where only the most rigorous authentication protocols can be relied upon to protect critical data.

Support for secure interoperation with non-IBM systems is provided by the SSL client certificate authentication protocol. In addition, SSL client certificate authentication, and asserted identity authentication support secure interoperation with WebSphere(R) Application Server.

Support for asserted identity authentication enables secure interoperability between CICS and WebSphere Application Server z/OS and OS/390(R).

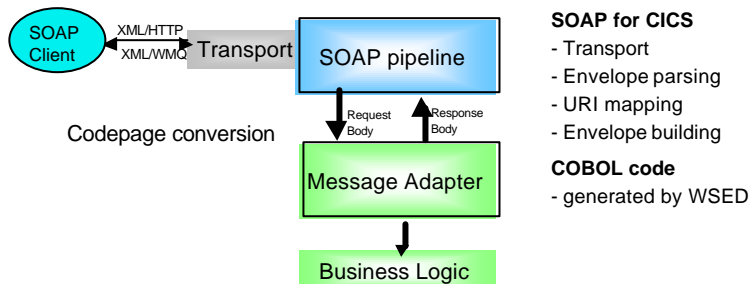
Support for PRIVACY enables you to specify the level of encryption that is best for your installation:

You can specify PRIVACY(REQUIRED) if you want all communications between client and server to be encrypted. You can specify PRIVACY(NOTSUPPORTED) if encryption of communications between client and server is not required. Using this option avoids the overhead of encryption.

## SOAP for CICS feature

### Why process SOAP in CICS?

- f Easily and efficiently reuse CICS assets from SOAP clients
  - drive existing CICS programs with user supplied XML<->COMMAREA mapping
  - Supports IBM WSED generated code for mapping
- f Performance focused
- f Language neutral approach, works for COBOL, C and Java
- f Synergy with WebSphere and industry direction for Web Services



This feature provides an infrastructure to allow CICS applications to service and make SOAP message requests. It includes transports over HTTP and WebSphere MQ. The HTTP based support builds on top of functions provided by CWS (CICS Web Support) and uses that support to manage security and transaction attributes. This includes the ability to use an SSL connection via the HTTPS protocol. No SOAP specific security mechanisms have been included.

SOAP for CICS allows you to create custom handlers to process his requires an application layer to be written to provide the mapping from a XML based SOAP payload to commarea and back again. This can be accomplished in a number of ways. WebSphere Studio Enterprise Developer (WSED) provides tools to generate converter routines from COBOL copybooks, that can perform these mappings. Custom mappings can be developed using SAX like parse support built into Enterprise Edition Cobol and PLI.

It can also be used to implement new applications that are XML aware that can be driven via a SOAP message. This allows scope for custom parsing and handling of such messages. Finally it allows the development of new CICS applications to make SOAP requests, targeting a remote server, over HTTP or WebSphere MQ. This allows CICS to obtain services, that use a SOAP and XML transport, provided by a remote server or business partner.



## SOAP for CICS feature

- **Enables CICS to be a SOAP 1.1 server and client**

- f* SOAP over HTTP uses the proven and easy to use CICS Web support, incl. SSL support
  - f* CICS TS V2 required to initiate outbound HTTP
- f* SOAP over WebSphere MQ provides guaranteed delivery
- f* Simple SOAP processing node - limited SOAP header processing

- **SOAP for CICS optional feature**

- f* Free orderable feature with CICS TS V2.2 and V2.3
- f* Inbound and outbound HTTP support
- f* Fully supported for production use by normal IBM CICS service

- **SOAP for CICS SupportPac CA1M**

- f* Free download from [ibm.com/software/http/cics/soap/index.html](http://ibm.com/software/http/cics/soap/index.html)
- f* Requires CICS TS V1.3 or above
- f* Supplied AS-IS and unsupported although there is a public newsgroup

- f* **Requirements**

- f* OS/390 V2R10 LE with APAR PQ65085 or above
- f* Some means to parse XML eg. Enterprise COBOL or Enterprise PL/I in user program

Through the availability with CICS TS V2 of the SOAP for CICS optional feature, CICS-based applications can be accessed as Web Services within a service-oriented architecture. These may be existing or new applications written in any supported programming language. The applications can be invoked by SOAP requests over either HTTP or WebSphere MQ, without the need for an intermediary application server. The SOAP for CICS feature also enables CICS applications to invoke Web services which are hosted on other systems. This enables an additional form of connectivity appropriate for Business to Business applications.

**SOAP FOR CICS FEATURE:** The SOAP for CICS optional feature was made available on September 26, 2003. It provides a mechanism which allows CICS applications, written in any supported programming language, to communicate via the Simple Object Access Protocol (SOAP). Transports are provided over Hypertext Transfer Protocol (HTTP) and WebSphere MQ. Both inbound and outbound function is provided. The implementation supports SOAP 1.1.

The SOAP for CICS feature is available for both CICS TS V2.2 and CICS TS V2.3. For more information, refer to Software Announcement 203-199, (RFA38441) dated August 5, 2003.

There is also a SOAP for CICS technology preview download available. Using this support you can write and deploy CICS applications that process SOAP messages. This technology preview is suitable for prototyping applications and for investigating SOAP technologies. It is available on an ASIS basis and is unsupported.



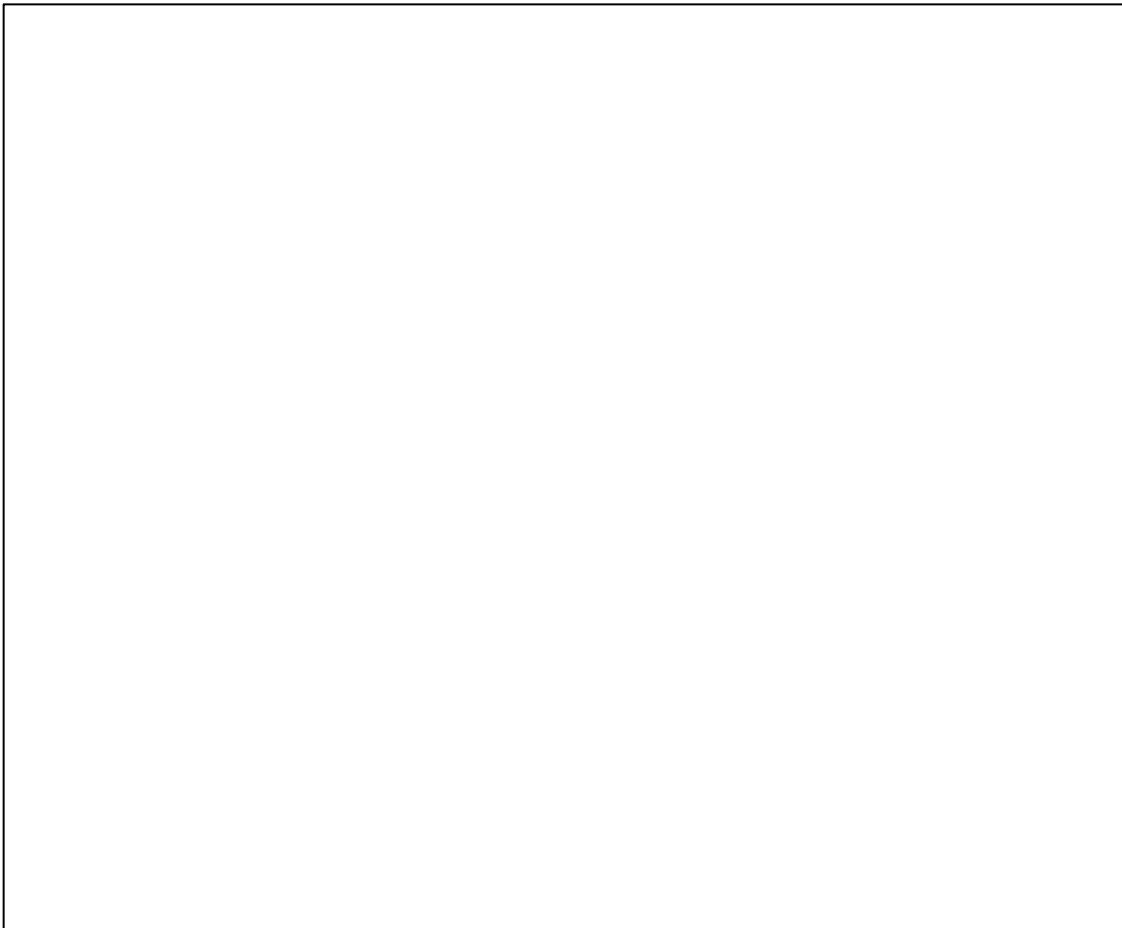
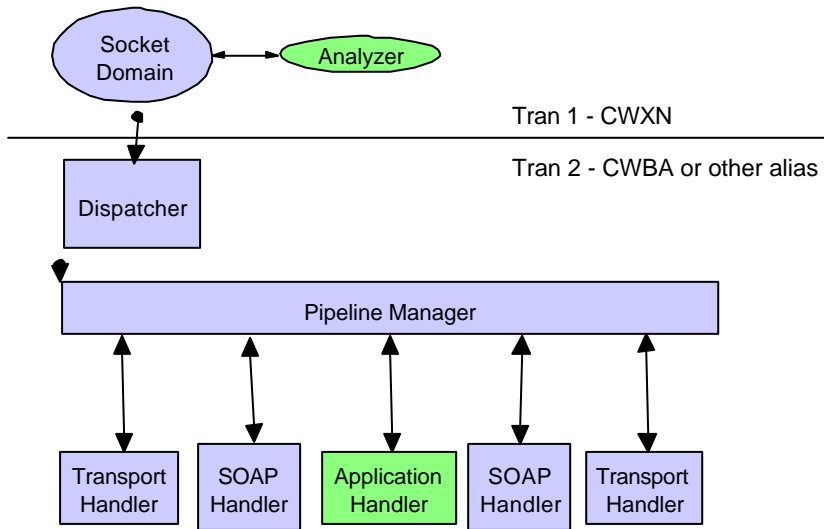
IBM Software Group

# Programming with SOAP in CICS



@business on demand software

# Design overview - Inbound SOAP / HTTP



## Service provider application

- Application Handler will be presented with a set of named containers.
- Received as Input
  - ▶ INPUT - Contains the Body element in 'raw' xml
  - ▶ NAMESPACES - Space separated list of xmlns definitions from the envelope tag
- Produces as Output
  - ▶ OUTPUT - Should be set 'raw' xml representing Body to be returned to the client
  - ▶ NAMESPACES - Space separated list of xmlns definitions to be added to the envelope tag.
- Dealing with namespaces is optional.

**Service provider application** A service provider application receives an inbound SOAP message, processes the contents and, sends a response.

## Programming the service provider application

1. Retrieve the attach event, with the RETRIEVE REATTACH EVENT command
2. (Optional) Use the GET CONTAINER command to retrieve the namespace information from the NAMESPACES container
3. Use the GET CONTAINER command to retrieve the request body from the INPUT container
4. Parse the request body
5. Invoke the business logic
  - ▶ Might be existing application program
6. Construct the body of the SOAP response
7. (Optional) Use the PUT CONTAINER command to place namespace information in the NAMESPACES container
8. Use the PUT CONTAINER command to return the response body to the pipeline in the OUTPUT container

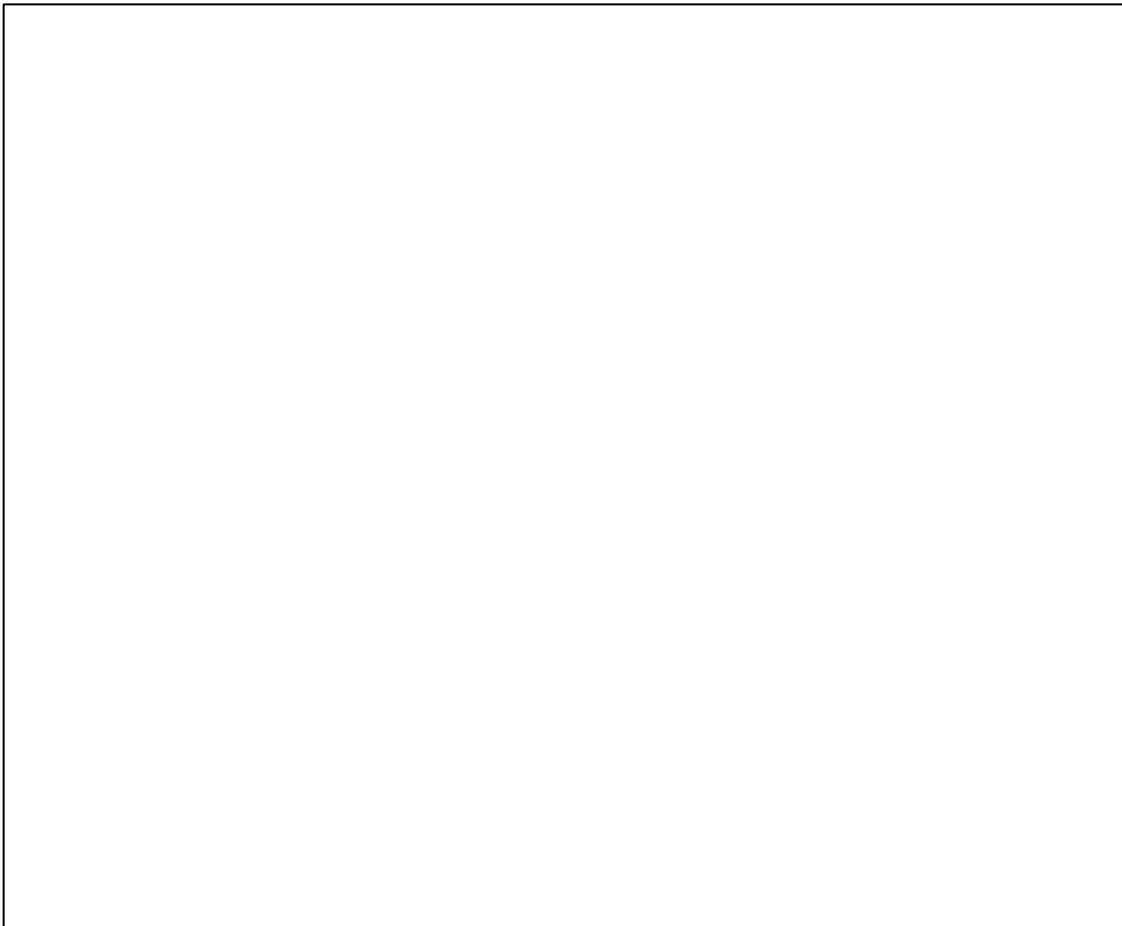
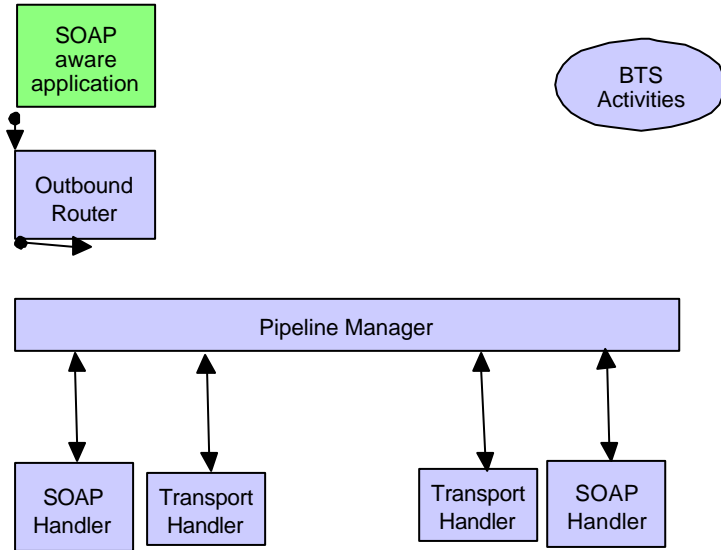
These are the main processing steps that your service provider application should perform. Depending upon the purpose for which your application program is written, the number, and sequence, of steps may vary.

1. Retrieve the attach event, with the RETRIEVE REATTACH EVENT command. The application program is a BTS activity program, which is attached one time by the pipeline manager. All BTS activity programs must deal with their reattachment events.
2. Optional: If your application expects the inbound SOAP body to contain namespace-qualified element and attribute names, use the GET CONTAINER command to retrieve the namespace information from the NAMESPACES container. The namespace information consists of a list of space-separated namespace definitions.
3. Use the GET CONTAINER command to retrieve the request body from the INPUT container.
4. Parse the request body. You can use the language statements which some compilers provide for parsing XML, you can LINK to a parsing program, or you can provide your own parsing code in the body of your program.
5. Using the relevant data from the request body, invoke the business logic. It is advisable to keep the business logic and the manipulation of the SOAP message separated. To do this, put the business logic in a separate program, and LINK to it.
6. Using the response from the business logic, construct the body of the SOAP response. You can use tools such as WebSphere Studio to construct the SOAP body. Alternatively, you can use the CICS DOCUMENT API.
7. Optional: If your application constructs a SOAP body that contains namespace-qualified element and attribute names, and you want the namespace prefix to be declared, and bound to a namespace URI, in the SOAP envelope, pass the information with the PUT CONTAINER command. The namespace information consists of a list of space-separated namespace definitions. Each definition has the form *symbol="URI"*.
8. Return the response body to the pipeline in the OUTPUT container, using the PUT CONTAINER command. For example: EXEC CICS PUT CONTAINER('OUTPUT') FROM(OUT-BODY) FLENGTH(OUT-BODY-LEN)

### Notes:

- In a SOAP message, the namespace prefix may be declared, and bound to a namespace URI, in the SOAP envelope, or in the SOAP body. If you do not declare the namespace information in the SOAP envelope, your application is responsible for providing the namespace information within the SOAP body.
- Your application can assume that the prefix SOAP:ENV will be bound to <http://schemas.xmlsoap.org/soap/envelope> in the outbound SOAP envelope; you do not need to pass this declaration in the container.

## Design overview - Outbound SOAP



## Service requester application

- A router activity is called with a set of containers
- Passed to router
  - ▶ TARGET-URI - The URI of the SOAP destination
  - ▶ REQUEST-BODY - Contains the Body element in 'raw' xml
  - ▶ APP-NAMESPACES - Space separated list of xmlns definitions to be added to the envelope tag.
- Received (response) from router
  - ▶ RESPONSE-BODY - The body of the SOAP reply message
  - ▶ APP-NAMESPACES - Space separated list of xmlns definitions from the reply envelope tag.
  - ▶ FAULT - Contains a CICS generated fault if the reply message is invalid in some way.
- As with input programs, dealing with namespaces is optional.

**Service requester application** A service requester application sends an outbound SOAP message, receives the response, and processes the contents of the response

Before you write a service provider application, you should be familiar with CICS business transaction services (BTS).

These are the main processing steps that a service requestor application should perform.

1. Perform the business logic that is necessary in order to construct the SOAP request. It is advisable to keep the business logic and the manipulation of the SOAP message separated. For example, you could put the business logic in a separate program, and LINK to it.
2. Construct the outbound SOAP request body
3. Invoke the SOAP outbound pipeline manager
4. Process the inbound SOAP response

## Programming the service requester application

1. Perform the business logic that is necessary in order to construct the SOAP request
2. Use the DEFINE PROCESS command to add a new CICS business transaction services (BTS) process and create the root activity.
3. Use the PUT CONTAINER command to pass the URI of the target SOAP processing code in container TARGET-URI
4. Construct the outbound SOAP request body
5. Use the PUT CONTAINER command to pass the SOAP request body in container REQUEST-BODY
6. (Optional) Use the PUT CONTAINER command to pass the namespace information in container APP-NAMESPACES
7. Link to the SOAP router process to invoke the SOAP outbound pipeline manager
8. Wait for the process to complete
9. Use the GET CONTAINER command to retrieve the response body
10. Parse the SOAP response body
11. Invoke the business logic to process the response

Depending upon the requirements of your application, you may need to construct a root activity, and child activities. The steps described assume that you will perform all parts of your application in the root activity.

1. Optional: Use the DEFINE PROCESS command to add a new CICS business transaction services (BTS) process, and create the root activity.
2. Pass the URI of the target SOAP processing code to the SOAP outbound pipeline manager in container TARGET-URI, using the PUT CONTAINER command.
  - For the HTTP transport, the target is specified as the URI of the target process. For example: `http://ip-address/CICS/CWBA/WSSOAPHT/SOAPSAMP`
  - For the WebSphere MQ transport, the target is specified in the following form: `MQ://queuenamel/queue_manager` where the queue manager name is optional if the queue is local, or a remote queue definition exists.
3. Using information obtained from the business logic, construct the body of the SOAP request. The message body must include the <SOAP-ENV:body> element.
4. Pass the SOAP request body to the SOAP outbound pipeline manager in container REQUEST-BODY, using the PUT CONTAINER command.
5. Optional: If your application constructs a SOAP request body that contains namespace-qualified element and attribute names, and you want the namespace prefix to be declared, and bound to a namespace URI, in the SOAP envelope, pass the namespace information to the SOAP outbound pipeline manager in container APP-NAMESPACES, using the PUT CONTAINER command. For example: `EXEC CICS PUT CONTAINER('APP-NAMESPACES') ACQACTIVITY FROM(OUT-NSP) FLENGTH(OUT-NSP-LEN)` The namespace information consists of a list of space-separated namespace definitions. Each definition has the form *symbol="URI"*.

### Notes:

- In a SOAP message, the namespace prefix may be declared, and bound to a namespace URI, in the SOAP envelope, or in the SOAP body. If you do not declare the namespace information in the SOAP envelope, your application is responsible for providing the namespace information within the SOAP body.
- Your application can assume that the prefix SOAP-ENV will be bound to `http://schemas.xmlsoap.org/soap/envelope` in the outbound SOAP envelope; you do not need to pass this declaration in the container. Your application should now invoke the SOAP outbound pipeline manager.



## XML parsing with COBOL & PL/1

- Event-based interface (SAX-like)
- 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
- High performance, but non-standard interfaces
  - Transform document contents to COBOL (PL/I) data structures
- Multiple code pages supported
- Inbound XML only – no compiler facilities for XML creation at this time

Both the new COBOL and PL/I compilers provide a parser for XML. They use the same basic parser, but the application programming interface provided is different in COBOL from PL/I, because of the differences in the two languages.

There are two basic types of XML parsers. A tree-based parser processes the entire XML document into a tree structure in a single operation and provides interfaces to navigate the tree. An event-based parser processes the document one XML-defined fragment at a time, reporting each one to the calling application with a callback. The application provides handlers for various types of fragments (events), which include the start of the document, the start of an element, the contents of an element, etc. Control goes back and forth from the calling application to the parser as the analysis of the document into fragments proceeds.

This parser is event-based or “SAX-like”. SAX stands for Simple API for XML, an example of an industry standard event-based API. It is quite fast, and handles many different code pages, but does not conform fully to the XML specification. The principal reason for non-conformance is that it does not do all of the checking required for “well-formedness”. Specifically, it does not process the internal document type definition (the DTD internal subset) and therefore does not normalize attribute values or supply defaults for them, nor include replacement text of internal entities except for pre-defined entities. Rather it simply returns the entire DTD contents for the DTD event.

This parser is also non-validating, which means that it does not attempt to check documents for validity. (A document is valid if it has an associated document type declaration (DTD) and it complies with the constraints of that DTD.)

The second reason that the parser is non-conforming is that it allows programs to continue normal parsing after some errors classed as “fatal” in the XML specification. The reason for doing this is to assist in debugging, but continuing after such an error renders the parser non-conforming.

Note that neither language provides new API elements for creating XML, but you can take advantage of the CICS DOCUMENT API to build outbound XML.

## 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

COBOL API support for XML parsing consists of the new verb 'XML PARSE' and four new special registers for exchanging information between the parser and the application program. The XML PARSE command specifies the paragraph or paragraphs that should be performed each time the parser extracts a new XML fragment (the PROCESSING PROCEDURE option), the statement(s) to execute if there is an exception (the ON EXCEPTION clause), and the statement(s) to execute if the whole document gets parsed without error (the NOT ON EXCEPTION clause).

The box on the foil shows an example of the syntax, the required words in blue and the application-supplied values in purple (italics). The example is entirely trivial, but the XML PARSE statement for a real case might not be much more complex; the complexity resides in the PROCESSING PROCEDURE code.

At each step in the parse process, the parser returns results in new special registers:

XML-EVENT identifies the event or fragment type (start-of-document, attribute-characters, etc.).

XML-TEXT or XML-NTEXT contains the data associated with the event, if any. If the data is in NATIONAL format, it is in XML-NTEXT; otherwise it is in XML-TEXT.

XML-CODE is the numeric identifier of the error when the parser encounters an exception. It is zero otherwise.

The application can set XML-CODE to -1 if it wants to indicate an exception based on its own analysis. Doing so ends the parsing and sends control to the ON EXCEPTION clause.

## 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);
```

- PLISAXB for parsing XML document in a file

In PL/I, parsing is implemented with two new built-in functions, PLISAXA and PLISAXB. They are essentially the same, except that PLISAXA parses an XML buffer in memory and PLISAXB parses an XML document in a file. To use either version, you define an event structure with code to handle each of the 24 possible event types that may occur in parsing (corresponding to the 24 XML fragment types). As in COBOL, control passes back and forth between the parser and the appropriate event code in the event structure as the parser presents each XML fragment to the program.

## CICS TS V2.3 new features

### ▪ e-business Access to CICS

- f* Java evolution including the SDK 1.4.1
- f* EJB hardening and performance
- f* Enhancements to IOP authentication and encryption
- f* SOAP for CICS feature



### ▪ Application Modernization

- f* New JCICS classes for the CICS APIs WEB, DOCUMENT, and EXTRACT
- f* New CCI Connector for CICS TS
- f* Interactive end-to-end debugging enhancements
- f* DB2 restart-light and JDBC support

### ▪ Enterprise Management

- f* CICSplex SM enhancements
  - Definitional web user interface
  - Workload management for the LINK3270 Bridge

Let us now consider application modernisation

## New CICS Java classes for Web support

- **Enables CICS Web support applications to be written in Java**

- f* Import JCICS class library onto any Java workstation IDE
  - Javadoc included
  - Growing set of CICS APIs now available in Java

- **New JCICS classes**

- f* WEB
  - WEB READ / WRITE
  - WEB SEND / RECEIVE
  - WEB RETRIEVE
  - WEB STARTBROWSE / READNEXT / ENDBROWSE
  - WEB EXTRACT
- f* DOCUMENT
  - DOCUMENT CREATE / INSERT / RETRIEVE / SET
- f* EXTRACT
  - EXTRACT CERTIFICATE / TCPIP / WEB

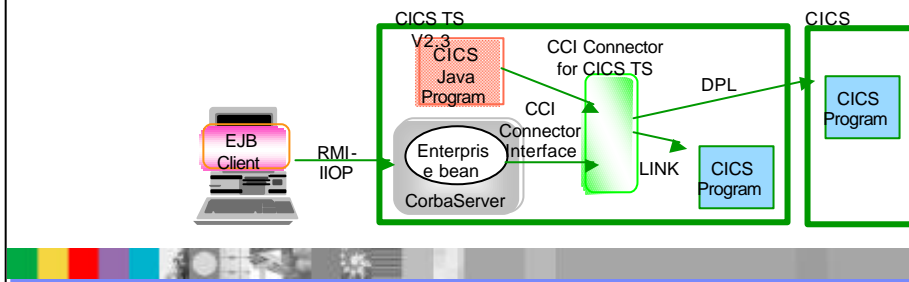
Java applications running under CICS have direct access to CICS services and resources via a set of Java classes known as JCICS, which provide equivalent function to the EXEC CICS interface used by applications in other languages. In CICS TS V2.3, new JCICS classes are provided that are equivalent to the EXEC commands for CICS Web Support. The extra classes provide Java applications with the ability to use the CICS Web, Document, and Extract APIs. This enables programs written in Java to be driven by CICS Web Support. The addition of the JCICS classes for CICS Web support allows Java application programmers to access CICS Web support data and develop CICS Web support applications. This broadens the appeal of CICS application programming to a larger number of application programmers, particularly those with Java programming skills.

## CCI Connector for CICS TS

- **Common Client Interface (CCI) part of J2EE 1.3 Connector Architecture (JCA) standard**

- **Strategic API for synchronus access to CICS programs from Java**

- ƒ Migrate from the Common Connector Framework (CCF) to CCI
- ƒ Supports the calling of CICS programs only
- ƒ Optimized for execution within CICS resulting in a standard LINK / DPL
  - 2 Phase Commit
- ƒ Supported by WebSphere Studio family
  - note Enterprise Services Toolkit is not supported in CICS



CICS TS V2.3 provides a new method for invoking CICS function from a Java client. This function uses the standard interface defined in the J2EE Connector Architecture (JCA) specification 1.0 - JSR016. This interface is called the Common Client Interface for CICS TS (CCI for CICS TS). It replaces the CICS Connector for CICS TS, introduced in CICS TS V2.1.

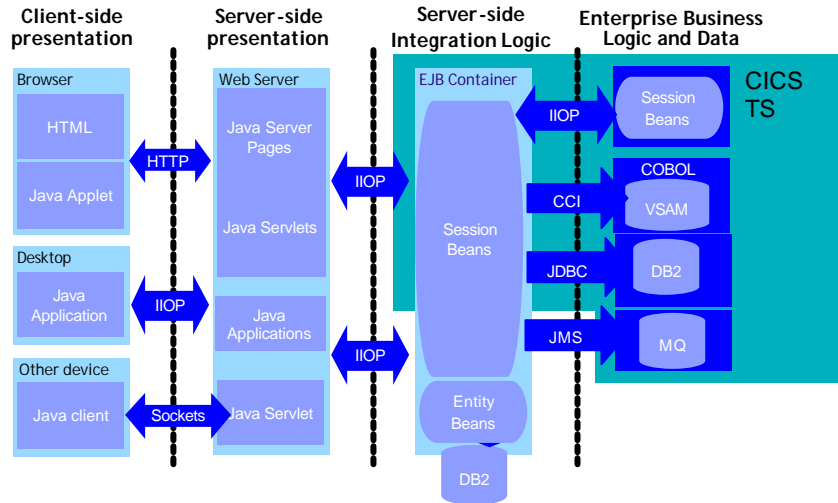
The CCI for CICS TS enables a Java program or an Enterprise Bean running under CICS TS 2.3 to communicate with any (Java or non-Java) COMMAREA application program, wherever that application program resides, whether in the same CICS region as the invoking Java class, or in another CICS region.

The CCI Connector for CICS TS enables customers to build powerful server components that exploit existing CICS programs. It enables Java programmers with little or no knowledge of CICS to reuse CICS applications. Java client applications should be portable between Java-enabled platforms with little or no modification, and, in particular, should be portable from a non-CICS JVM into the CICS environment.

Because the Common Client Interface is a non-proprietary standard, the CCI code that calls the server program should be portable, with little or no modification, to and from most Java-enabled platforms.

Because the CCI Connector for CICS TS runs *inside* CICS, no network flows are required between the connector and CICS. Thus, the connector's performance is better than that of CCI connectors that use the ECI resource adapter to access CICS programs from outside CICS.

## Connectors in context



The Software Strategy for e-business strongly separates different roles into different logical tiers. This separation promotes flexibility since, given the use of open and portable standards, a solution can be deployed and re-deployed across a number of different physical tiers without the need for mass code change.

A thin-client tier provides client-side presentation, using portable standards such as HTML, XML and Java applets, with communication via protocols such as HTTP.

A middle-tier provides server-side presentation logic to generate, for example HTML, presentation data. This presentation along with the control logic is written using Java servlets and JSP pages.

A distinct server-side business logic tier contains the core business logic code, written as Enterprise JavaBeans. This strong separation between presentation and business logic, ensures that new presentation code for a new presentation channel can be added without affecting the business logic. Finally, the business logic tier accesses data in Enterprise Data Systems. These not only include databases, but also existing non-java applications servers using a range of connectors.

## Interactive debugging enhancements

### ▪ Debug CICS application end-to-end using tools

#### f Debugger client

- GUI tool that runs on the workstation
  - WebSphere Studio integrated debugger
  - IBM Distributed Debugger
- 3270 client
  - IBM Distributed Debugger

#### f Debugger server

- Runs on the same system as the application program
- Communicates with the debugger client
- For compiled languages
  - IBM Debug Tool
  - User-written or vendor-written debugging tools
- For Java programs
  - Executing in the JVM in debug mode

You can debug a CICS application using debugging tools that run on a workstation. There are two components to the debugging tools in this environment:

The *debugger client*, which runs on the workstation. It is through the graphical user interface (GUI) provided by the debugger client that you interact with the application program. For example, you can use the debugger client to set breakpoints, to step through your program, and to examine the variables used by your program.

The *debugger server*, which runs on the same system as the application program, and communicates with the debugger client.

You can debug the following sorts of CICS applications using a debugger client on a workstation:

Applications written in a compiled language (COBOL, PL/I, C, C++)

Language Environment-enabled Assembler subroutines

Java applications running in a JVM

Applications that use a combination of compiled language programs and Java programs

You cannot debug PLT programs using a debugger client on a workstation.

You can use the following as your debugger client:

WebSphere Studio Enterprise Developer

WebSphere Studio Application Developer

For compiled languages and Language Environment-enabled Assembler subroutines, you can use the Debug Tool as your debugger server. For Java programs, the debugger server is the Java Virtual Machine (JVM) executing in debug mode.



## Interactive debugging enhancements

### ▪ Simplified setup for Debug Tool

- f Removed requirement for CICS TCP/IP Socket Interface
- f Removed requirement for LE use exit CEEBXITA

### ▪ Improved management of debugging profiles

- f In earlier releases, profiles are managed by Debug Tool
- f In CICS TS V2.3, CICS manages debugging profiles
  - Store profiles in a VSAM file - can be shared and not discarded on CICS shutdown
  - Supports Java programs as well as other languages
  - Multiple profiles can be active at one time
  - Web-based and 3270 interfaces

### ▪ Greater integration when debugging mixed languages

- f Compiled languages incl. COBOL, PL/I, C, C++ and Assembler
- f Java programs
- f Combination of compiled languages Java

### ▪ Dynamic activation of JVM profiles

- f Debugging on and off for a Java program dynamically

When earlier versions of CICS are used with Debug Tool, Debug Tool uses the TCP/IP Socket Interface for CICS feature of TCP/IP for MVS to communicate with a debugging client on a workstation. When it is used with CICS TS for z/OS, Version 2.3, Debug Tool uses the CICS sockets domain, and you do not need to install and configure the TCP/IP Socket Interface.

Earlier versions required you to link-edit the Language Environment user exit (CEEBXITA) with programs that you intended to debug. In CICS TS for z/OS, Version 2.3, this is not necessary.

When earlier versions of CICS are used with Debug Tool, the debugging requirements are saved in a profile which is managed by Debug Tool. In CICS Transaction Server for z/OS, Version 2 Release 3, debugging profiles are managed by CICS, and their function, and management, is enhanced:

Each debugging profile is associated with a user. Profiles are stored in a data set, and persist when CICS shuts down. They are not discarded except when a user explicitly requests it and can be shared between several CICS regions. Debugging profiles can specify the debugging requirements for Java programs as well as languages supported by Debug Tool. Debugging profiles can specify generic attributes, and a number of profiles can be active, and associated with a debugging session, at one time. This makes it possible to specify complex selection criteria for which programs are to run under the control of a debugging tool.

Because CICS provides a Web based tool for managing debugging profiles, and because debugging profiles support both Java and compiled language programs, programmers can use the same set of tools for debugging programs from a workstation. The tools support Java programs, compiled language programs and applications that use a combination of both

To turn debugging on and off for a Java program, you must change the JVM profile associated with the program. In CICS TS for z/OS, Version 2.2, you cannot do this dynamically; you must either edit the profile, or change your resource definitions to select a different JVM profile. In CICS TS for z/OS, Version 2.3, you can use the application debugging profile manager to associate a new JVM profile with a Java program when you need it.

## DB2 restart-light support

### ▪ Enhanced DB2 restart-light for improved recovery

- f* Introduced in DB2 V7
- f* Provides the ability to cross-system restart a DB2 data sharing member quickly, with a minimal storage footprint
- f* Release the retained locks quickly, and then terminate
  - No new work will be accepted

### ▪ Enhanced DB2 restart-light

- f* DB2 V8 supports indoubt UOW, not only inflight UOW
- f* If indoubts are outstanding then it stays up long enough for CICS to connect to it and resynchronize indoubts before terminating
  - Stop new work being flowed into DB2

CICS now supports the enhanced DB2 restart-light capability provided in DB2 Version 8. Restart-light mode is intended for a cross-system restart in the event of an MVS system failure, where the CICS systems are configured to use group attach. The reduced storage requirement for restart-light mode makes it possible to restart a DB2 data sharing member temporarily on a system that might not have enough resources to start and stop DB2 in normal mode. The DB2 restart-light subsystem is able to release retained locks and resynchronize indoubt units of work, and then terminate. This process promotes faster recovery and data availability.

In DB2 Version 7, the original DB2 restart-light capability allowed a DB2 data-sharing member to restart with a minimal storage footprint, free retained locks, and then terminate normally. However, this only applied to in-flight units of work, and not indoubt units of work. For DB2 Version 8, this capability was extended to encompass indoubt units of work. This enables a CICS system to connect to a DB2 restart-light subsystem for the purpose of resynchronizing indoubt units of work. This capability is especially useful because DB2 does not support peer recovery; that is, one DB2 subsystem cannot resynchronize indoubt units of work on behalf of another DB2 subsystem.

## DB2 JDBC V2 driver

- **Delivered with DB2**

- ƒ Many performance improvements within the driver
- ƒ Downward compatible with JDBC 1.2
- ƒ DB Beans supported and conform more closely to the JDBC V2.0 rowset specification

- **CICS TS V2.2 and V2.3 JVM environment only**

- ƒ HPJ supports JDBC 1.2 only

- **Exploit CICS TS V2.3 shared class cache support feature of prJVM**

DB2 Version 7 provides two levels of the JDBC driver, the 1.2 level and the 2.0 level. The JDBC 1.2 driver provided by DB2 can be used by Java program objects created using VisualAge for Java Enterprise Edition for OS/390, and by Java programs and enterprise beans that run in a Java Virtual Machine (JVM). However, the JDBC 2.0 driver provided by DB2 only supports Java programs and enterprise beans that run in a JVM, and does not support Java program objects.

The JDBC 2.0 driver supports a selected subset of the JDBC 2.0 application programming interface, and it is downward compatible, so it supports the JDBC 1.2 API as well. Existing Java applications and enterprise beans that were written using the JDBC 1.2 API and that run in a JVM, can run using the JDBC 2.0 driver, and can benefit from performance improvements made in the JDBC 2.0 driver. If you create new Java applications and enterprise beans that use the JDBC 2.0 API, you need to use the JDBC 2.0 driver.

The use of the JDBC V2.0 API in DB2 V7 provides significant performance improvements when compared with the use of the JDBC V1.2 API. Improved handling of data passed to DB2 from the JDBC driver results in reduced use of the Java Native Interface (JNI), minimizing associated costs.

## CICS TS V2.3 new features

### ▪ e-business Access to CICS

- f* Java evolution including the SDK 1.4.1
- f* EJB hardening and performance
- f* Enhancements to IOP authentication and encryption
- f* SOAP for CICS feature

### ▪ Application Modernization

- f* New JCICS classes for the CICS APIs WEB, DOCUMENT, and EXTRACT
- f* New CCI Connector for CICS TS
- f* Interactive end-to-end debugging enhancements
- f* DB2 restart-light and JDBC support



### ▪ Enterprise Management

- f* CICSplex SM enhancements
  - Definitional web user interface
  - Workload management for the LINK3270 Bridge

The CICSplex System Manager is an integral part of CICS TS, whose role is to reduce the complexity of operating multiple CICS systems by presenting them as a simple and integrated plex. It combines all the major CICS management functions within a single user interface. It also cooperates with Tivoli products to meet the need for end-to-end management and automation of CICS with z/OS and the customer's network. This release continues the themes for systems management of integration, simplification, monitoring, and automation. Through the CICSplex SM Web User Interface, CICS has a modern intuitive interface for all aspects of system management.

## CICSplex SM enhancements

### ■ CPSM Definitional WUI

- f Manage CICS resource definitions from the web
- f Supports CREATE, UPDATE and REMOVE
- f Improved starter set of views for definitions
- f Improved security for WUI view and menus
  - RACF profile to READ/UPDATE custom views
  - Audit trail of updates
- f Usability and accessibility improvements
  - Accessibility, tabular views, multiple record selection
- f Separate CICS region (LMAS) as WUI server

### ■ CPSM Operations WUI

- f NLS enabled
- f Customizable
  - By resource category / user task / application (dynamic filters)
  - Add explanatory annotations, add hyperlinks
  - Control updates (by restricting views)
  - Customizable help referring to e.g. documentation about procedures



CICS Transaction Server for z/OS, Version 2 Release 3 introduces a major extension of the tasks you can carry out using the Web User Interface with the introduction of the Create, Update and Remove actions for definitions.

The Web User Interface starter set of views and menus includes a set of new administration views you can use to create and update CICS resource definitions. You can use these as provided, or as a basis for creating your own using the Web User Interface view editor.

The Web User Interface has also been extended to include, enhanced security for Web User Interface views and menus, including an audit trail, general usability and accessibility enhancements, such as mixed case field hints and better data formatting and an enhanced starter set of views and menus, offering definitional views and a default environment designed to be more familiar to users of the ISPF end user interface.

The Web User Interface enhancements in CICS Transaction Server for z/OS, Version 2 Release 3 significantly extend its power and usability. You can now perform more tasks from this platform-independent interface. Improved security allows you to protect views and menus from unauthorized access and to trace users who make updates. The usability improvements make the interface easier to use and more accessible to those with visual impairments. The improved starter set of views and menus provides a better starting point from which you can customize the interface to match the requirements of your business.

## CICSplex SM enhancements

### ▪ New CPSM APIs

*f* New add actions for certain resource tables

### ▪ CPSM WUI support for code page GB18030

*f* Enhancement of support for simplified Chinese code page

*f* WUI supports web clients using :

–GB2312 (CHS1)

• Server code page(EBCDIC) : 935

–GB18030(CHS2)

• Extended code page of GB2312

• Server code page(EBCDIC) : 1388, superset of code page 935

### ▪ Synchronization of CPSM and CICS releases

*f* Remove requirement for running CMAS in older releases of CICS

CICS Transaction Server for z/OS, Version 2 Release 3 introduces several new add actions for certain resource tables. These allow you to perform an add action against one of the records to be linked, naming the record to be linked to. In order to make them more intuitive the names of these actions indicate the type of resource involved.

The CICSplex SM Web User Interface has been modified to support web clients using the GB18030 ASCII encoding standard for simplified Chinese. This encoding standard extends the range of characters defined in the existing simplified Chinese GB2312 code page.

A simplified Chinese Web User Interface server can support client browsers using either GB2312 (CHS1) or GB18030 (CHS2). For GB2312 clients, the server uses the EBCDIC code page, 935. For GB18030 clients, the server assumes a second EBCDIC code page, 1388, which is a superset of code page 935. View sets and menus edited using a GB2312 client browser are stored in 935 and can be used in either client code page. However, if a view set or menu is edited using a GB18030 client and characters not available in 935 are used, the resulting views set or menu are not displayed correctly on GB2312 clients. Note that all supplied messages and starter set views and menus, and view sets and menus customized using CICS TS 2.2 or earlier, use the 935 code page and can be displayed on either client code page.

In CICS Transaction Server for z/OS, Version 2 Release 3 a CICSplex SM CMAS will run only in a CICS system at the same release level. The ability to run the latest release of CICSplex SM in older releases of CICS in the CMAS is no longer necessary and has been removed. This removes unnecessary complications in CICSplex SM setup procedures.

## CICSplex SM enhancements

### ▪ Workload management for LINK3270 Bridge

*f* with CPSM dynamic routing program, EYU9XLOP

- Dynamic routing
- Workload balancing
- Workload separation

### ▪ Architectural improvements

*f* Removal of internal architectural constraints

- Shutdown entire CICSplex for some service operations
- Coexistence of multiple levels of CICS and CPSM in the same CICSplex

Link3270 bridge dynamic workload management provides CICSplex SM dynamic routing and workload management support for the CICS Link3270 bridge, which was introduced in CICS Transaction Server for z/OS, Version 2 Release 2. It makes use of CICSplex SM workload management services to provide workload balancing and workload separation for Link3270 bridge workloads, making it unnecessary to program an exit to select target regions for dynamic routing. CICSplex SM dynamic routing and workload management support for the Link3270 bridge allows you to make use of CICSplex SM facilities for dynamic routing and workload separation of Link3270 bridge workloads, and removes the need for you to program a CICS exit to select a target region.

CICSplex SM architectural limits removes a number of internal architectural constraints which limit the ability to support additional function. The constraints that have been removed include the frequent need for a simultaneous shutdown and upgrade to all CICS systems within the CICSplex when applying service to resource tables, the coexistence of multiple levels of CICS and CICSplex SM within the same CICSplex, the types and size of data that can be handled. Although directed primarily at improving the reliability, availability and serviceability of CICSplex SM, architectural limits delivers a number of immediate benefits as well as providing a platform that will enable CICSplex SM to grow in the future.

## Additional enhancements

### ▪ Additional CICS APIs made threadsafe

- f EXEC CICS DOCUMENT + ASKTIME + FORMATTIME
- f Change task commands

### ▪ Display TN3270 IP address

- f CICS supplies the IP address of terminals in message
- f For autoinstalled terminals connecting through TN3270

### ▪ Outbound HTTP

- f Program LINK interface (also available on CICS TS V2.2 with APAR)
- f Will be replaced with equivalent function in a future CICS TS release

### ▪ Sympathy Sickness prevention for improved availability

- f Read time-out for MRO
  - Purge waiting task if waiting too long
  - As specified on corresponding PROFILE definition
    - 1 second granularity
  - Applies to all MRO and SNA connections

When CICS is connected to DB2 Version 6 or later, the CICS DB2 attachment facility exploits the open transaction environment (OTE), to enable the CICS DB2 task-related user exit to invoke and return from DB2 without switching TCBS. In the open transaction environment, the CICS DB2 task-related user exit runs on an open L8 mode TCB, and requests to DB2 are issued on this TCB. Application programs that are defined to CICS as threadsafe and conform to threadsafecoding conventions can also execute on the open TCB, which means that no switching back to the QR TCB is required until the task terminates. In these conditions, there could be a considerable increase in the throughput of a CICS region.

If an application program that is executing on an open TCB issues a non-threadsafe CICS request, the request forces a switch back to the QR TCB, reducing the performance benefits of OTE exploitation. To avoid this, ensure that all the EXEC CICS commands used by the application program are threadsafe. The commands that are threadsafe are indicated in the command syntax diagrams in *the CICS Application Programming Reference* with the statement "This command is threadsafe". For CICS TS for z/OS, Version 2.3, the following EXEC CICS commands have been converted to be threadsafe: Asktime, Document and Formattime

In earlier releases, it has not been possible to determine the IP address of autoinstalled VTAM terminals connecting into CICS through TELNET 3270. The VTAM CINIT contains the terminal TCP/IP client address, IP port and host name, known as the TNADDR, to the VTAM CINIT. CICS now supplies the TNADDR in messages, dump and trace as an aid to problem diagnosis. CICS also provides code in the sample user-replaceable programs to extract this information.

A new CICS-supplied program DFHWBCLI, is provided that can be invoked via EXEC CICS LINK to establish an outbound HTTP connection. The utility permits the specification of a target URL, from which an HTTP response is obtained.

An enhancement has been introduced whereby tasks waiting on an MRO connection are purged if the wait exceeds a specified time-out value. The purpose of this is to prevent a 'sympathy sickness' situation whereby if transactions in an AOR for some reason stall or deadlock, the corresponding relay transactions in an MRO-connected TOR are left hanging and can eventually cause the TOR to stall completely



## Additional enhancements

- **CICS samples are supplied compiled**

- **Case folding in CEDA**

- f* Case-sensitive fields are marked (Mixed Case) on the CEDA input panels
  - eg. JVMCLASS and JVMPROFILE on the PROGRAM definition
- f* Not applied to the CEDA command line

- **Changes to data conversion**

- f* Generic templates for data conversion
  - Use extended DFHCNV TYPE=ENTRY macro
  - Apply to multiple resources

- **Statistics enhancements**

- f* DFHSTUP provides a new extract reporting function
- f* Three event types to be reported :
  - Exception, Warning and Information

The CEDA transaction is used to define CICS resources. Some of these items have to be entered in mixed case as they apply to Hierarchical File System (HFS) or Java/EJB class/method names. An enhancement has been introduced to the CEDA (for Resource Definition), CADP (Debugging definition), and CREA (RequestModel configuration) transactions which greatly increases the usability of CICS system definition when mixed case entries are required. Terminal input is BY DEFAULT taken to be in mixed case. Items which have to be in upper case are automatically folded. Other items are taken to be case dependent.

The DFHCNV TYPE=ENTRY macro defines the data conversion for resources in the conversion table. The macro has been extended so that you can use it to create generic templates that apply to multiple resources, rather than creating an entry in the conversion table for each individual resource. You can do this by using a prefix in an entry rather than specifying the full resource name. This groups together all resources which match the prefix and resource type. The matching resources are all converted in the same way.

The CICS statistics utility program, DFHSTUP, has been improved with the addition of a new extract reporting function very similar in concept to the RDO utility DFHCSDUP extract function. This provides a method of writing a user program that can process statistics records to create reports of statistics data. This allows you to design statistics reports tailored to the needs of your installation.

## Additional enhancements

### ▪ Remove non-LE runtime

- f Remove Interfaces - VS COBOL II, OS PL/I and C/370 runtimes
  - Support for compilers was removed in CICS TS V2.2
- f Existing non-LE Applications can be executed in LE compatibility mode

### ▪ Announce future withdrawal of support for Java HPJ program objects

- f In any future release of CICS
- f Migration path to the Java SDK

In CICS Transaction Server for z/OS, Version 2 Release 3, interfaces to the VS COBOL II, OS PL/I and C/370(TM) runtimes are removed. Applications compiled and linked with these non Language Environment conforming products usually execute successfully under Language Environment in *compatibility mode*. CICS maintains its interfaces to the OS/VS COBOL runtime, and applications compiled at this level of COBOL usually execute successfully provided that your CICS system is executing with a Language Environment which includes support for the OS/VS COBOL runtime.

Java Program objects are programs compiled with VisualAge for Java Enterprise Toolkit for OS/390 (ET/390) byte code binder. They are sometimes referred to as compiled Java programs or as HPJ programs. To assist in migration, support is continued in CICS TS V2.3, but this will be the last release that provides support.

## Information Center enhancements

### ■ CICS documentation organized into



Task guides



Concepts



References

### ■ Task Guides

- ƒ CPSM configuration
- ƒ Migration scenarios
  - CICS/ESA V4.1 to CICS TS V2.3
  - CICS TS V2.2 to CICS TS V2.3
- ƒ Updated EJB task guides

### ■ Tables now interactive

- ƒ Sort order + limit lines displayed

### ■ Searches saved across sessions

### ■ Supported on Windows 2000 / XP, AIX

### ■ (soon to be) Available online

- ƒ <http://publib.boulder.ibm.com>



The CICS Transaction Server for z/OS Information Center is the primary source of user information for CICS Transaction Server. One copy of the CICS Information Center, on a CD-ROM, is provided automatically with the product. Further copies can be ordered, at no additional charge. Licensed documentation is available only to licensees of the product. A version of the Information Center that contains only unlicensed information is available through the publications ordering system, order number SK3T-6945.

In CICS TS for z/OS, Version 2.3, the CICS Information Center has been improved. In order to make the information center more widely available and flexible for customers, it is now available on an extended range of platforms. This release of the information center is fully supported on the following platforms:

Windows NT  
 Windows 2000  
 Windows XP  
 AIX(R) 5.1

The CICS TS 2.3 Information Center can be installed to coexist with other versions of the information center on the Windows platform. For example, you can have the information centers for CICS TS 2.2 and CICS TS 2.3 both available on your workstation. You can identify the latest release of the information center by looking at the desktop icon, the directory structure and the introduction page of the information center itself.

Most tables in the information center are now interactive and can be sorted and searched by column. Only the rows that match the specified string or regular expression are displayed.

New lookup tables have been provided for CEMT commands and parameters, figure titles and table titles. These lookups can be found in the search bar in the masthead.

When you save searches, the results as well as the search term are now listed in the search navigation panel. You can click on the appropriate link in the navigation panel to select a topic from the list of search results. These results are saved between information center sessions, and will remain listed until you delete them.

Migration information has been restructured to assist customers who are migrating to CICS TS for z/OS, Version 2.3 from releases other than CICS TS for z/OS, Version 2.2. Separate migration information is provided for:

Migration from CICS TS for z/OS, Version 2.2  
 Migration from CICS TS OS/390, Version 1 Release 3  
 Migration from CICS TS OS/390, Version 1 Release 2  
 Migration from CICS/ESA 4.1



IBM Software Group

# Installation & Migration



@business on demand software

## CICS TS V2.3 packaging



### ▪ CICS TS V2.3 product components

- f CICS MVS/ESA
- f CICSplex System Manager
- f REXX Development and runtime for CICS/ESA
- f Tivoli Global Enterprise Manager CICSplex SM Instrumentation
- f CICS Application Migration Aid

### ▪ SOAP for CICS Feature

- f Free optional order

### ▪ WebSphere Application Server V5

- f 5 x restricted entitlements for use with CICS for development, deployment + production

### ▪ WebSphere Studio Enterprise Developer V5

- f Integrated development environment for CICS and WebSphere
  - COBOL, PL/I, Java incl. CICS and J2EE applications

The elements included in CICS TS V2.3 are shown here.

The SOAP for CICS feature can be ordered at no cost.

Also included are WebSphere Application Server and WebSphere Studio Enterprise Developer

## CICS TS V2.3 requirements

- **z/OS V1.4 or later**
- **IBM Developer Kit for OS/390, Java 2 Technology Edition V1.4.1 - Service Refresh 1**
- **Optional IBM CICS Tools - recommended versions**
  - f* IBM CICS Transaction Gateway V5.01
  - f* IBM CICS Performance Monitor V1.2
  - f* IBM CICS Performance Analyzer V1.3
  - f* IBM CICS VSAM Recovery V3.2
  - f* IBM CICS Business Event Publisher for MQSeries V1.1
  - f* IBM CICS Online Transmission Time Optimizer V1.1
  - f* IBM CICS Interdependency Analyzer V1.2
  - f* IBM Session Manager for z/OS V1.1
  - f* IBM Debug Tool V3.1
  - f* Fault Analyzer for z/OS and OS/390 V3.1

The minimum required level of operating system is z/OS V1.4 or later. For application Java program the IBM SDK for z/OS, Java 2 Technology Edition 1.4 is required and is available on the installation tape.

IBM also provides a range of CICS Tools and other products which support CICS TS V2.3 and are listed here for your convenience.

## CICS TS V2.3 requirements

### Optional IBM products

- **WebSphere Application Server V4 or later**
- **IMS/ESA Database Manager V7 or later**
- **DB2 for z/OS and OS/390 V6 later**
  - For using the JDBC/SQLJ support :
    - PTF UQ73040 for APAR PQ66125 for DB2 Version 6
    - PTF UQ73041 for APAR PQ66126 for DB2 Version 7
- **WebSphere MQ for z/OS and OS/390 V2.1 or later**
  - with the PTF for APAR PQ35501, or later
- **Tivoli Decision Support for OS/390 V1.6 or later**

We also have a list of optional products and there minimum levels shown here

## 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





## Supported compilers

- COBOL
  - ▶ SAA AD/Cycle COBOL/370(TM) (5688-197)
  - ▶ IBM COBOL for MVS & VM, Version 1, (5688-197)
  - ▶ IBM COBOL for OS/390 & VM, Version 2, (5648-A25)
  - ▶ IBM Enterprise COBOL for z/OS and OS/390 Version 3 (5655-G53)
- PL/I
  - ▶ SAA AD/Cycle PL/I for MVS & VM(5688-235)
  - ▶ IBM PL/I for MVS & VM Version 1 (5688-235)
  - ▶ IBM VisualAge PL/I for OS/390, Version 2 (5655-B22)
  - ▶ IBM Enterprise PL/I for z/OS and OS/390 Version 3 (5655-H31)
- C/C++
  - ▶ AD/CYCLE C/370(5688-216)
  - ▶ C/C++ for MVS/ESA (5655-121)
  - ▶ OS/390 C/C++ (5647-A01)
- Assembler
  - ▶ High Level Assembler/MVS & VM & VSE Version 1.1 (5696-234)



## Removal of non-LE runtimes

- Remove all compiler run-time libraries from CICS JCL
  - ▶ xxx.COB2CICS
  - ▶ xxx.COB2LIB
  - ▶ xxx.PLILINK
  - ▶ xxx.SEDCLINK

### OS/VS COBOL

Language Environment provides a set of compatibility library routines that permit you to run OS/VS COBOL applications on CICS in compatibility mode.

When you run an OS/VS COBOL application on CICS, the environment that is established for a run unit by the compatibility library routines supports only OS/VS COBOL.

If you have any VS COBOL II programs that use the COBOL CALL statement to call OS/VS COBOL programs, you must compile the OS/VS COBOL program with either VS COBOL II or IBM COBOL.

### VS COBOL II

When running under Language Environment, IGZEOPT is ignored on CICS. Language Environment issues a warning message if IGZEOPT is specified in applications running on CICS.

The timing of abends between Language Environment and VS COBOL II is different for run-time detected errors (such as IGZ0061S division by zero or IGZ0006S subscript out of range). This difference in the timing of abends affects the behavior of CICS HANDLE ABEND.

## Changes for Java support

- Changes to JVM initialization
  - ▶ JVMProfiles moved to HFS (from PDS members)
- Heap sizing
  - ▶ Changes due to use of shared class cache
  - ▶ Sizing recommendations are not available yet
- Application migration from JDK 1.3.1s to JDK 1.4.1s
  - ▶ Possibly Java language differences
    - consult [java.sun.com](http://java.sun.com)



## Changes for Enterprise JavaBeans & CORBA applications

- May need to republish IORs in JNDI server
  - ▶ upgrading from IOP 1.1 to 1.2
  - ▶ `com.ibm.cics.iop.MaxGIOPMinorVersion=2`
- DFHXOPUS
  - ▶ GIOP 1.2 does not support IOP Principal in request headers
  - ▶ DFHXOPUS can only return userID derived from IOP when GIOP 1.1 is used.



## Migrating to CICSplex SM V2.3

- CPSM migration as in previous release-to-release changes
- CPSM must be upgraded to TS 2.3 level before managed regions are upgraded
  - ▶ CMAS must be running CICS V2.3 code
- Maintenance point CMAS must be first CMAS upgraded
- Agent code in CICS region must be at same level as CMAS to which it is attached



## Migrating the CPSM WUI

- Migrate customized view set and menu definitions
  - ▶ At previous release, export view set and menu definitions
  - ▶ Create a new WUI server repository for Version 2.3
  - ▶ Start the WUI server at the new release
  - ▶ Import the new starter set definitions
  - ▶ Import previous release view set and menu definitions
    - SKIP on Duplicate Names field



## Summary

- Installation
  - ▶ Much like previous releases
  - ▶ Some additional USS/hfs components
- Migration
  - ▶ Most of the impact is to systems-programmer support areas
  - ▶ Monitoring, statistics, messages, URMs
  - ▶ Applications may be affected by JDK level upgrade
  - ▶ Outdated COBOL & PL/1 applications may be affected by LE runtime



## Further reading

- **Announcement letter, Release Guide, Migration Guide, Info Center, Technical papers**

[f ibm.com/cics](http://ibm.com/cics)

- **SOAP for CICS**

[f ibm.com/cics/soap](http://ibm.com/cics/soap)

- **IBM Developer Kit for z/OS, Java 2 Technology Edition V1.4.1**

[f ibm.com/servers/eserver/zseries/software/java](http://ibm.com/servers/eserver/zseries/software/java)

- **z/OS**

[f ibm.com/servers/eserver/zseries/zos](http://ibm.com/servers/eserver/zseries/zos)

Further information can be found here..