CICS Transaction Server Application Architecture

IBM Redbooks Solution Guide

IBM® CICS® Transaction Server is a transaction server that primarily runs on IBM System z® mainframes under IBM z/OS®. It controls the interactions between applications and users (from a small number of users to thousands of them). CICS applications offer high availability and easy scalability. CICS applications have built-in redundancy, which uses various client interfaces that range from terminals and web browsers to web services.

CICS Transaction Server has a wide range of capabilities that can be used readily by applications that are deployed into its run time. Application architects can rely on the runtime qualities of service that are provided by CICS, without needing to design these qualities of service into the application code. Such qualities of service include security, transactional integrity, and the dispatching of work for multiple processes or hardware clusters.

As this IBM Redbooks® Solution Guide describes, the CICS application architecture (Figure 1) has several business and technical benefits.

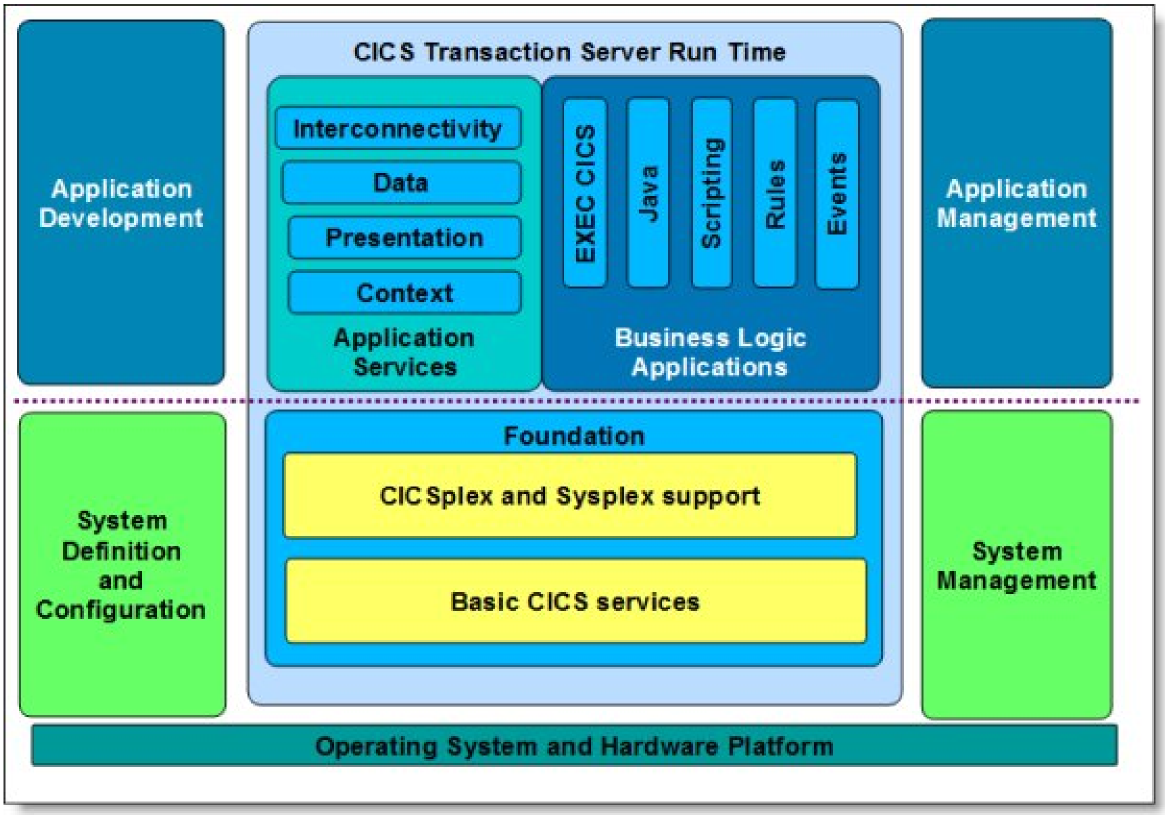

Figure 1. CICS application architecture

Did you know?

CICS was first released in 1969 and remains highly popular today. A significant number of commercial electronic transactions are processed by CICS Transaction Server:

- More than 30 billion transactions are processed by CICS every day.
- At least three CICS customers are achieving transaction volumes in excess of one billion a day.
- CICS processes one trillion dollars in transactions every week.

Business value

CICS Transaction Server is a managing environment for hosting high-volume, highly efficient, transactional applications. Instead of requiring developers to write system functions in their applications, CICS can perform these services automatically or upon request from an application.

By using the capabilities of CICS, developers can focus on solving business problems and coding business logic into their applications rather than worrying about system functions such as database integration and security control points. This approach offers several advantages, including shorter development cycles, easier maintenance, and easier incorporation of new functions. Applications do not have to be changed to work with the new operating system releases or databases because CICS Transaction Server handles the interfaces with these systems.

CICS provides a fast, low-cost, and low-risk way to evolve business applications. It provides significant opportunities for cost savings and improves productivity. Because applications can be adjusted by using CICS capabilities, and because CICS can support multiple programming languages, the savings can be substantial.

Solution overview

CICS provides a set of application programming interfaces (APIs) that enable simplified access to CICS-controlled resources. The APIs also allow CICS container services to provide runtime qualities of service such as transactionality, security, and workload management.

The CICS APIs significantly simplify application development by providing a neutral, standard language to access resources such as files, queues, or other programs. And the APIs enable a modular development model by separating the business logic from the system management logic that supports it. You can read, write, and rewrite a file without using the syntax of the selected program language. You reference the alias of the resource and perform the CICS command by using the appropriate API.

An additional benefit of CICS APIs is that load module compatibility is maintained across CICS releases. Even when APIs are deprecated and removed, modules can still be used in newer CICS releases if they are coded to expect the relevant errors.

Businesses use various application development languages that are based on their requirements and existing skill sets. Because CICS supports many language options, applications that run on CICS can exist in a truly heterogeneous environment that consists of procedural modules (such as COBOL) and object-oriented languages such as Java or C++.

Figure 2 summarizes the languages that are supported by CICS.

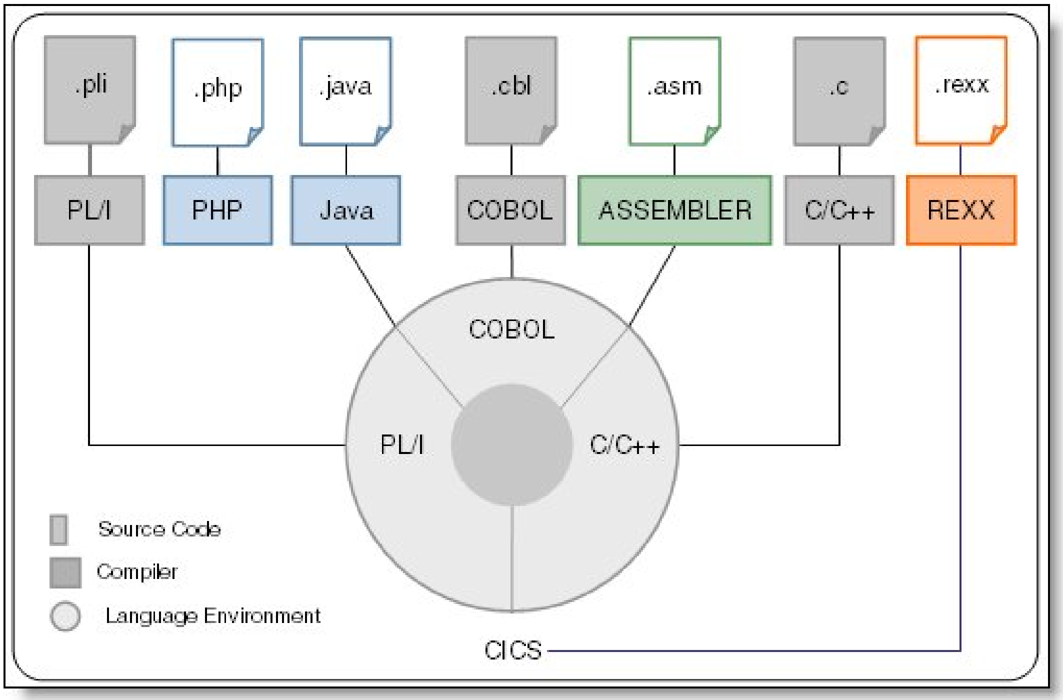

Figure 2. The CICS programming environment

Central to CICS language support is the IBM z/OS Language Environment®, which CICS uses to integrate the different programming language run times. Each programming language (for example, PL/I, COBOL, and C/C++) has its own library within the language environment run time so that CICS can provide simple communication between different programs at the language level. REXX uses the CICS services directly and has no link with the language environment libraries.

The CICS multilanguage run time includes the following benefits:

- *Interlanguage communication*, which provides flexibility and efficiency in the design and development processes. It also allows developers to use a fit-for-purpose method to select the best programming language for each project.
- *Standardized set of interfaces* in the CICS APIs for interacting with any resources that are controlled by CICS. CICS also provides an API for Java, which, although implemented in Java, behaves identically to a standard CICS API. This API provides simplicity of development, consistency between languages, and an easy way to exploit the CICS runtime qualities of service such as security, workload management, and transactionality.
- *Central diagnosis*, which is a common layout for the information that is provided by the system when errors are thrown. It reduces the time and effort that are needed to determine the source of problems, which can improve application quality.

Solution architecture

CICS offers an extensive set of programming commands that an application program can use to request CICS services from the CICS run time. CICS also uses the full set of functions and procedures that are provided by the native language run time. The set of CICS commands is known as the API. The commands are macro statements that are supported in a range of languages and that are translated by the CICS translator before compilation. They provide the key to taking advantage of the qualities of service, such as transactionality and security, from the CICS run time.

CICS commands

The commands that are provided through the API are categorized into the following areas:

- *Presentation services* are used for communication between the user, directly or with an intermediate system, and the transaction processing server. Presentation services work with the presentation management facilities of the system, such as a web browser or 3270 display device, which might be external to the CICS system.
- *Data services* retrieve and update data and provide access to files, temporary storage or transient data queues, and CICS journals.
- *Business services* manipulate data from the time that it is retrieved from storage to the time that it is either presented or updated and cover many functions.

By using the CICS API, the application developer can maintain independence between the services of the business application and the transaction run time. This independence allows the creation of business applications that are focused on solving business processing.

Reusable modular model

The modular design model standardizes components that can be assembled to build a bigger component. This model is often referred to as *component-based software engineering*. This design decomposes the process into simpler elements that can be arranged in many patterns to benefit from synergies between the components. Figure 3 illustrates the CICS modular application architecture layers.

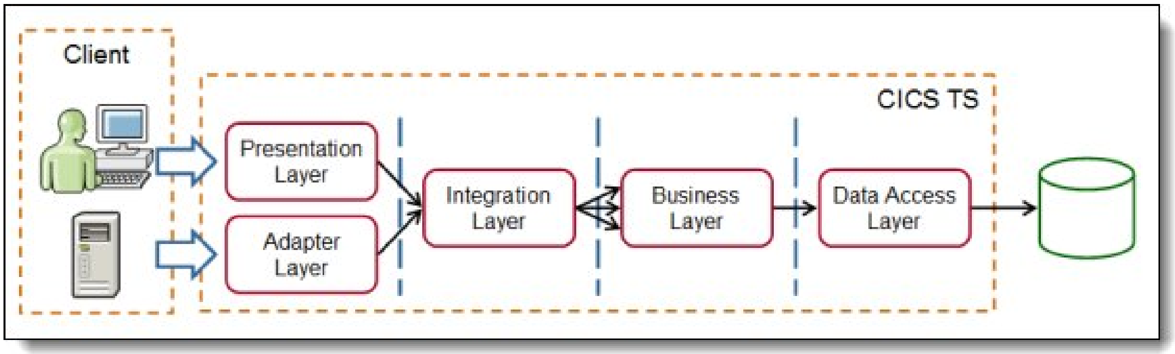

Figure 3. CICS modular application layers

The client, or service consumer, initiates the request. Examples include a web service requester, web browser, IBM WebSphere® MQ client, TCP/IP socket client, 3270 device, z/OS batch program, or other CICS applications. Then, the request flows through the following architecture layers:

- The *adapter layer* processes the protocols and data with the client, establishes the transaction and security context in CICS, and works with the integration or business layer.
- The *presentation layer* is a special case for web or 3270 applications, where the user interface is provided directly within the CICS application.
- Optionally, when required, the *integration layer* implements a sequence of calls to business logic for situations where it is more efficient or offers better encapsulation. Another situation is where it makes the services easier to use if the calls are done in CICS rather than the client making several calls directly to the business layer.
- The *business layer* implements the business functions of the service and is often the most extensive layer that has the most processing requirements.
- The *data access layer* provides the logic to access the data store in use, such as IBM DB2®, IMS™, Virtual Storage Access Method (VSAM), or other resources. Separating the data access from the

business layer facilitates reuse and allows changes to the data store type, without affecting other modules.

Usage scenarios

Enterprises in nearly every area of industry and commerce rely on CICS transaction processing. CICS is widely used in the following industries among others:

- Agriculture
- Architecture
- Automotive
- Banking
- Chemical
- Construction
- Education
- Financial services
- Government
- Insurance
- Manufacturing
- Media and broadcasting
- Medical and pharmaceutical
- Military
- Oil, gas, and mining
- Real estate
- Research and development
- Retail
- Shipping and transport
- Telecommunications
- Travel

As shown in Figure 4, more than half of all CICS installations are used by financial and government industries.

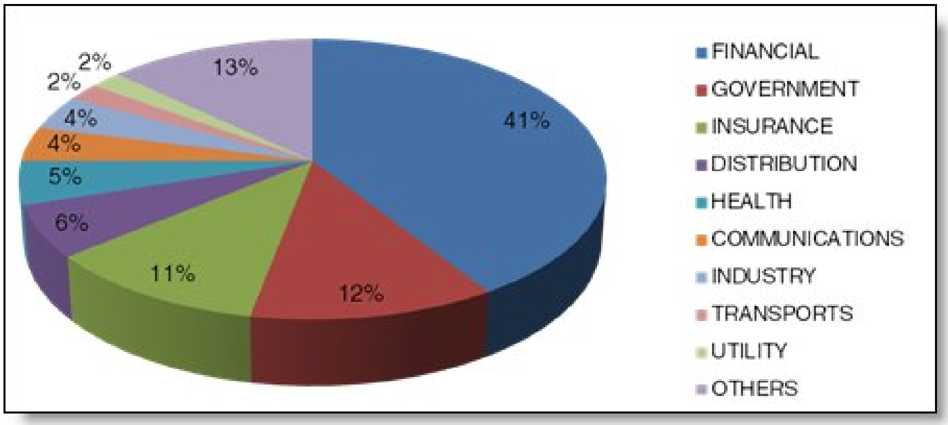

Figure 4. CICS usage by industry

Integration

Integration is core to CICS and allows applications that are built for one purpose to be modernized so that they can continue to be used today. This section examines a few of the integration options for reusing and extending existing applications.

CICS web services

Application programs that run in CICS can participate in a heterogeneous web services environment as service requesters, service providers, or both. CICS support for web services conforms to open standards, including SOAP 1.1 and 1.2, HTTP 1.1, and Web Services Description Language (WSDL) 1.1 and 2.0. CICS supports the most common type of communication between service requester and service provider, which is SOAP over HTTP and SOAP messages to WebSphere MQ. Tools support for CICS web services is provided by the CICS web service assistant and IBM Rational® Developer for System z. Figure 5 shows an overview of CICS web services support.

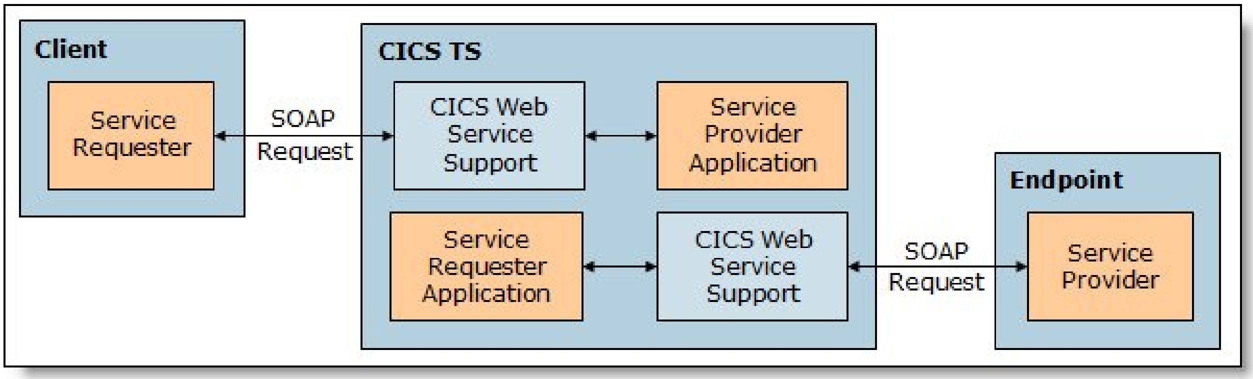

Figure 5. Web services architecture inside CICS

Java Connector Architecture

The Java EE Connector Architecture (JCA) defines a standard set of APIs and interfaces to connect from the Java EE platform to heterogeneous enterprise information systems (EISs). By using the JCA standards, such vendors as IBM can provide a JCA resource adapter to connect and call services in CICS. This support is enabled with the CICS Transaction Gateway, which provides resource adapters to connect with CICS applications. These resource adapters run in a Java EE application server. Figure 6 shows a z/OS solution, where JCA is provided by using a WebSphere z/OS Optimized Local Adapter.

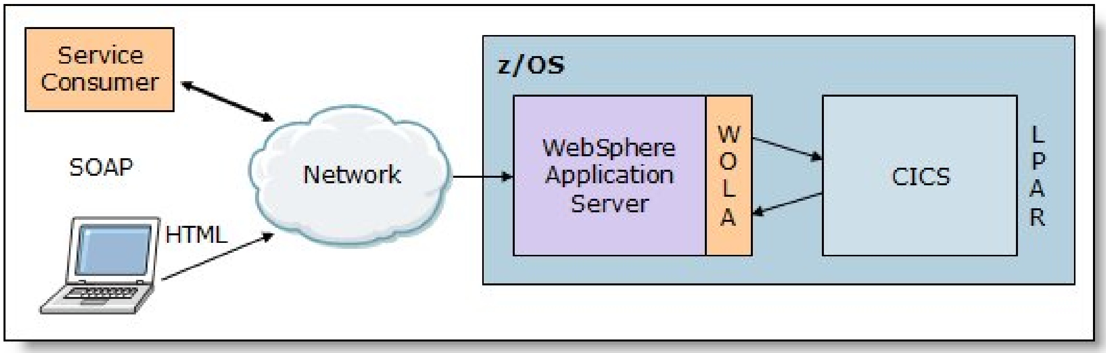

Figure 6. JCA connecting to CICS

CICS web support

CICS web support provides an HTTP listener and a message adapter program that can be written by using CICS WEB APIs. Figure 7 shows how HTTP is used directly with CICS. With CICS web support, a CICS application can initiate an HTTP request and receive the response from an HTTP server program, providing bidirectional support for the HTTP protocol.

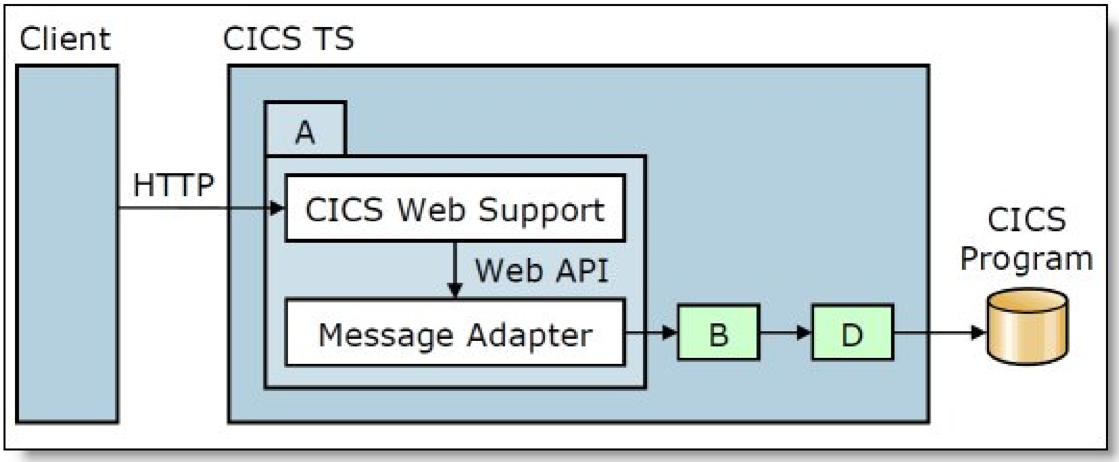

Figure 7. CICS web support

Messaging with WebSphere MQ

By using WebSphere MQ, you can easily exchange information across different platforms, integrating existing business applications in the process. WebSphere MQ ensures reliable delivery of messages, dynamically distributes workload across available resources, and helps to make programs portable. WebSphere MQ provides Java Message Service (JMS) APIs and basic WebSphere MQ APIs for use by service requesters on various platforms, with many options for routing and encrypting messages before they arrive on WebSphere MQ for z/OS.

Figure 8 shows the WebSphere MQ trigger monitor program that is provided by CICS. You can use this program to automatically start an appropriate message adapter program when messages arrive. The message adapter uses WebSphere MQ basic APIs to receive the message, transform it if required, and call the business logic program. A reply message can be sent by using the reply-to queue that is defined in the message.

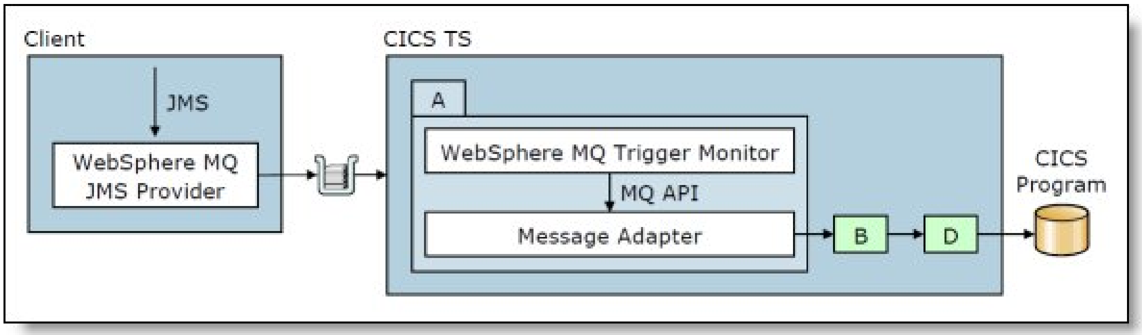

Figure 8. Connecting to CICS by using WebSphere MQ

Supported platforms

For detailed system requirements for CICS Transaction Server for z/OS V5.1, see "CICS Transaction Server for z/OS 5.1 detailed system requirements" at:

http://www.ibm.com/support/docview.wss?uid=swg27035673

Ordering information

When you order a new license, use the Program ID Type: 5655 and Model: Y04. Table 1 summarizes the ordering information.

Table 1. Ordering information

License	Entitlement identifier	Description	License option or pricing metric
Parallel Sysplex license charge (PSLC)	S0172DF	CICS TS V5.1	Basic MLC, PSLC below 3 MSU Basic MLC, PSLC AD SYSUSGREG NC, PSLC AD
Workload License Charge (WLC)	S0172DF	CICS TS V5.1	Basic MLC, Variable WLC Workload Registration, Variable WLC
Entry Workload License Charge (EWLC)	S0172DF	CICS TS V5.1	Basic MLC, Entry WLC
Advanced Workload License Charge (AWLC)	S0172DF	CICS TS V5.1	Basic MLC, AWLC
Advanced Entry Workload License Charge (AEWLC)	S0172DF	CICS TS V5.1	Basic MLC, AEWLC
S/390 and System z Usage License Charge, basic license	S0172DF	CICS TS V5.1	0 to 0.25 MSU Base 0.26 to 0.5 MSU Base 0.51 to 1.0 MSU Base Level A Chg/MSU (2 to 11 MSUs) Level B Chg/MSU (12 to 44 MSUs) Level C Chg/MSU (45 to 78 MSUs) Level D Chg/MSU (Above 78 MSUs) Level D Chg/MSU (Above 78 MSUs), per 50 MSUs
System z entry license charge (zELC)	S0172DF	CICS TS V5.1	Basic MLC, zELC

Related information

For more information, see the following documents:

- CICS Transaction Server for z/OS V5.1 Sales Manual
  http://ibm.co/T6ARi4
- CICS Transaction Server product page
  http://www.ibm.com/software/htp/cics/tserver
- CICS Transaction Server for z/OS V5.1 announcement letter
  http://www.ibm.com/support/docview.wss?uid=swg21612470
- IBM Software support lifecycle for CICS Transaction Server
  http://ibm.co/TUKe8t

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document was created or updated on October 31, 2012.

Send us your comments in one of the following ways:

- Use the online **Contact us** review form found at:
  ibm.com/redbooks
- Send your comments in an e-mail to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

This document is available online at http://www.ibm.com/redbooks/abstracts/tips0922.html .

Trademarks