

CICS Transaction Gateway

1. Overview

The IBM CICS Transaction Gateway (CICS TG) software product permits the use of modern Java tools for interfacing to CICS. It provides the means to access existing and new CICS applications from a web browser using Java and HTML. It combines the capabilities of two previous products, the Internet Gateway and the Java Gateway. CICS TG performs the task of accepting a CICS request from a (possibly remote) Java-client program and passing it up to the CICS server. The CICS TG runs on a variety of platforms including NT, AIX and OS/390, and accepts CICS requests over a variety of network protocols including HTTP and industry-standard SSL.

The CICS Transaction Gateway is available in two different flavours. The Unix or NT version runs on a middle tier computer. The OS/390 version runs on the same computer as CICS, however in a separate virtual address space, or on a separate S/390 computer within a Sysplex. It is also included as a part of the CICS Transaction Server (for OS/390 and VSE), WebSphere, VisualAge for Java and other application servers. It contains the CICS Universal Client and a range of technologies that provide Java and web interfaces into CICS.

CICS Transaction Gateway provides a set of Java-based web server facilities for access to CICS applications from a web browser. These include Java classes and Java beans for writing application-specific server programs (servlets) and browser programs (applets). There are classes for access to both traditional and object-oriented CICS applications. It provides an API for programming the presentation interface for new applications (or new front-ends to old ones). No programming is required for unmodified ("green screen") access to 3270-interface CICS transactions with the CICS Transaction Gateway terminal servlet. All other uses of the CICS Transaction Gateway require programming in Java, using the classes and beans provided.

2. Components

The CICS Transaction Gateway contains the following components:

- A **Java gateway application** that resides on a web server. On non-S/390 platforms, it communicates with CICS applications through facilities provided by the CICS Universal Client. On S/390 systems, it communicates inbetween virtual memory spaces with CICS applications via the CICS External Communication Interface (EXCI).
- A **CICS Universal Client** that provides the External Call Interface (ECI) and External Presentation Interface (EPI), as well as a terminal emulation function. The **ECI interface** enables a non-CICS client application to call a CICS program as a subroutine using distributed program link calls. The **EPI interface** enables a non-CICS client application to act as a logical 3270 terminal and so control a CICS 3270-based application. It uses transaction routing. The CICS Universal Client is the same program that is used on workstations to access CICS. It runs only on non-S/390 platforms.
- A **CICS Java class library** that provides an API for communication between the Java gateway application and a user-written Java application, either applet or servlet. The class **JavaGateway** is used to establish communication with the gateway process. The **ECIRequest** class is used to specify ECI calls and the **EPIRequest** class for EPI calls (through the CICS Universal Client).
- A set of **Java EPI beans** for creating Java front-ends to existing CICS 3270-based applications.
- A **terminal servlet** that allows to use a web browser as an emulator for a 3270 CICS application.

There are 5 alternative ways to use the CICS Transaction Gateway facilities:

- **Writing servlets** (Java web server programs): The CICS Transaction Gateway supplies Java classes and Java beans that allows to write Java web server code to handle a browser request that requires services from CICS. They may be used to provide access to existing CICS applications, to entirely new function, or a combination of both. They include all of the data conversion and manipulation functions that are ordinarily needed to invoke S/390 CICS transactions with a 3270 or COMMAREA interface, and to communicate with the browser.
- **Writing applets:** The same facilities may be used to code an applet. An applet has the same function as a servlet, but executes on the end-user workstation after download by a server. It requires a Java-enabled browser on the workstation.
- **Turnkey access to 3270 transactions:** The CICS Transaction Gateway supplies one pre-coded servlet, called the "terminal servlet", that provides turnkey browser access to CICS 3270-interface transactions. The servlet translates browser requests to CICS interactions on a one-for-one basis, so that the user sees the browser equivalent of the "green" 24x80 screens that would appear on a real 3270 terminal. It uses the Java classes and beans described above to convert URL-encoded HTTP browser input to 3270-format data inbound, and to convert 3270-format transaction output to HTML for return to the browser.
- **Tailored access to 3270 transactions:** These facilities permit modifying, without reprogramming, the green-screen interface produced by the terminal servlet. You can change the HTML on a screen-for-screen basis with CICS facilities, and you can use variable substitution with web server server-side includes to map one browser request to several 3270 interactions, or to invoke a 3270 transaction and extract data from the returned screen.
- **CORBA client support:** ORB-enabled browsers can run Java beans which interoperate with server-side Java beans running in a CORBA server (such as WebSphere Application Server) via the CORBA IIOP protocol. The server-side beans can then invoke CICS Transaction Gateway Java methods to execute 3270- or COMMAREA-based CICS applications.

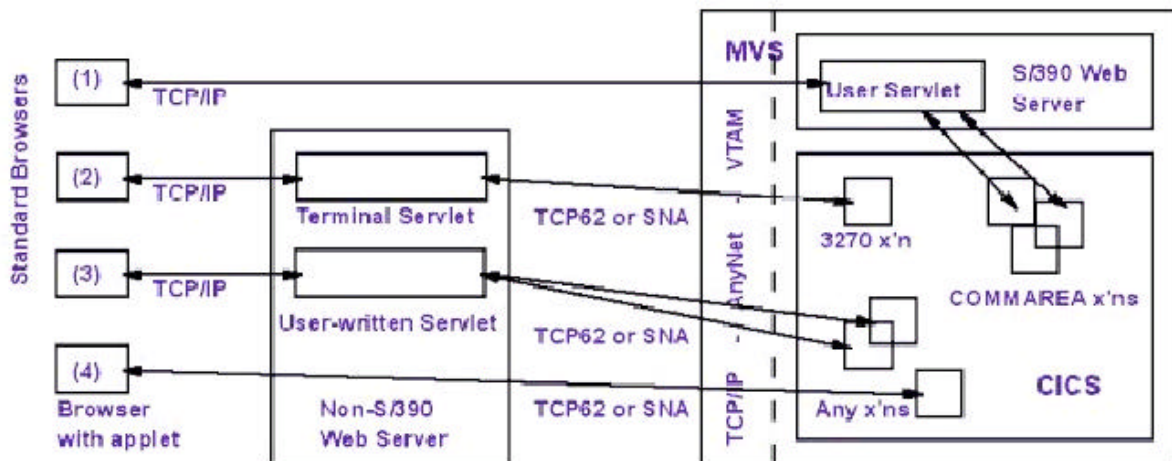


Figure 1: Examples of web access to CICS applications using the CICS Transaction Gateway:

- (1) User-written servlet on a S/390 web server invoking multiple COMMAREA-based transactions for one browser request;
- (2) turnkey 3270 access through the Terminal Servlet;
- (3) a user-written servlet on a non-S/390 web server invoking multiple transactions (3270 or COMMAREA) for one browser request;
- (4) applet counterpart for (3).

The CICS Transaction Gateway permits to drive existing or new applications from a set of provided Java classes. The interface could be a workstation, web browser or NC. The classes (which are representations of the ECI and EPI APIs) communicate with a Java application named the "CICS Java Gateway" that is part of CICS TG. In a middle tier configuration it uses the CICS Universal Client to communicate with the CICS server. If CICS TG and its Java Gateway runs directly on OS/390, it uses the EXCI (External CICS Interface) to communicate with CICS. The EXCI is an OS/390 cross-memory interface packaged with CICS/390. It allows an OS/390 application to call a CICS program in an RPC style. SSL security is supported for this interface.

3. e-business connectors

Assume a company has existing CICS applications written in COBOL, which are to be leveraged in a modern Web environment. The end-user GUI may be implemented via a Java Applet or a Java Servlet and HTML. In all cases, there is a requirement to efficiently access CICS applications, which provide the core business logic, from the Java code, which provides the user interface. Assume the CICS applications are callable using the CICS External Call Interface (ECI). This may be done using the CICS Transaction Gateway (CICS TG).

The need to access existing logic and data from Java is not unique to CICS. Companies often have valuable applications that exist in other enterprise servers such as IMS or which can be accessed via MQSeries. Rather than accessing these resources using disparate APIs, a common technology has been adopted by several IBM products. Called the Common Connector Framework (CCF), it provides a consistent means of connecting to, and interacting with, enterprise resources from any Java execution environment.

A set of IBM e-business Connectors have been written to fit into the CCF model. A CICS Connector is included as part of the CICS Transaction Gateway. Connectors also exist for IMS, MQSeries and Encina/DCE.

The CCF provides consistency in two ways :

- It provides a consistent client-application view of enterprise resources, whatever Connector is being used. This consistent view also makes it easier for AD tools such as VisualAge for Java to generate client code to use the CCF.
- It provides a consistent run-time view to a Connector, so that the Connector can be hosted in any run-time environment that supports CCF. As a result, any client code that uses the Connectors should be applicable anywhere that supports CCF.

The CCF is currently key in providing access to enterprise resources from within VisualAge for Java, WebSphere and IBM's Enterprise Server for Java implementations.

4. Building applications using VisualAge for Java

A Web application can use the CICS Connector to access CICS applications via the CICS Transaction Gateway.

One approach is VisualAge for Java Version 2.0 (VAJ), Enterprise Edition. This includes the Enterprise Access Builder (EAB), which comprises of the framework and tools needed to easily build code that uses the Connectors.

The use of EAB requires three steps. To start with, you need to build an EAB Command bean which encapsulates a single interaction with CICS. To do this you must use the JavaBeans that make up the CICS Connector. You take a CICSConnectionSpec and set its properties to describe the CICS server it should access, along with details of the CICS Transaction Gateway used to get to it. Then take an ECInteractionSpec and set its properties to describe the CICS application that you want to invoke. The other thing required to build the Command bean are Records that describe the input and output data for the interaction. If the CICS applications are written in COBOL you can import your source, using an EAB tool that parses COBOL, to create the Record definitions for the CICS COMMAREA

used as input/output to your CICS applications. You 'wire' all these bits together visually in VAJ to create your Command bean.

It is then possible to create several Commands that can be linked together to create one logical interaction. This can be done by creating an EAB Navigator bean which encapsulates the sequence of Commands required.

The next part of the process is to create an EAB business object to provide the public interface to your CICS resources. This contains a number of Java properties that represent the information from the interactions with CICS that you want to expose. EAB mappers should be used to define the relationships between your Commands/Navigators and these business object properties.

Finally, VAJ's GUI building tools allow you to assemble the desired user interface and to link it to the interfaces of your business object.

In addition to the VAJ IDE and the EAB, VisualAge for Java, Enterprise Edition also includes a CICS Transaction Gateway and the CICS Connector.

5. Environment

5.1 Hardware and Software Requirements

The CICS Transaction Gateway runs on the OS/390, OS/2, Windows NT, AIX, and Solaris. The terminal servlet requires a web server (or servlet engine) that provides support equivalent to Java Servlet Development Kit (JSDK) V1.1 or later (e.g the IBM WebSphere Application Server).

5.2 Connectivity

For all platforms except Sun Solaris, the CICS Universal Client can connect to CICS using SNA. On Intel platforms (Windows NT/95/98 and OS/2), the Client contains TCP62 support providing transport of LU6.2 data over TCP/IP connections. Function equivalent to TCP62 is provided by AnyNet in IBM Communication Server products for Windows/NT, OS/2, AIX, and OS/390. In addition, all platforms support TCP/IP connection of the Client to an intermediate CICS system running on Open Systems platforms (AIX, HPUX, Sun Solaris) and Windows NT. The intermediate CICS system can then connect to CICS on S/390 using SNA.

6. Literature

6.1 Library

1. CICS Family: Client/Server Programming, SC33-1435
2. CICS Family: Communicating from CICS on System/390, SC33-1697
3. CICS Transaction Gateway Administration Reference, SC34-5448
4. CICS Internet Guide, SC34-5445 (CICS TS for OS/390, V1.3)
5. CICS Internet Guide, [SC34-5445](#) (CICS TS for OS/390, V1.3)
6. CICS Internet and External Interfaces Guide, [SC33-1944](#) (CICS TS for OS/390 V1.2)
7. CICS Web Interface Guide, [SC33-1825](#) (CICS TS for OS/390 V1.1) and [SC33-1892](#) (CICS/ESA V4.1)

6.2 Redbooks

1. Revealed! Architecting Web Access to CICS, SG24-5466
2. OS/390 Version 2 Release 4 Performance Figures for CICS Web-Enabled Applications, SG24-5612, an update and republication of Capacity Planning for CICS Web-Enabled Applications on OS/390, SG24-5168
3. e-Business Application Solutions on OS/390 Using Java, SG24-5342
4. Revealed! CICS Transaction Gateway with More CICS Clients, SG24-5277
5. Integrating Java with Existing Data and Applications on OS/390, SG24-5142
6. Application Development for VisualAge* for Java Enterprise, SG24-5081
7. Accessing CICS Business Applications from the World Wide Web, [SG24-4547-02](#)