

CICS® Transaction Gateway Configuration



Configuring CICS Transaction Gateway for Terminal Servlets (WebSphere Application Server for AIX Version 2.0.2)

CICS® Transaction Gateway Configuration



Configuring CICS Transaction Gateway for Terminal Servlets (WebSphere Application Server for AIX Version 2.0.2)

Contents

Chapter 1. Overview	1	Starting CICS Transaction Gateway	13
Chapter 2. Software checklist	3	Stopping CICS Transaction Gateway	13
Installing the workstation software.	3	Starting HTTP Server	13
Chapter 3. Installing the Gateway Server software	5	Stopping HTTP Server and WebSphere	14
HTTP Server	5	Chapter 6. Configuring the Terminal Servlet	15
JDK	5	Loading the TerminalServlet	15
WebSphere Application Server	6	Configuring the parameters.	18
CICS Transaction Gateway	6	Chapter 7. Testing using the supplied samples	21
Chapter 4. Configuring the Gateway Server software	9	Chapter 8. Troubleshooting	25
HTTP Server Plug-in for WebSphere Application Server	9	HTTP Server and WebSphere Application Server log files	25
Shell script for stopping WebSphere Application Server	9	TerminalServlet logging	25
Configuring CICS Transaction Gateway	10	CICS Universal Client log	25
Chapter 5. Starting and stopping the Gateway Server software	13	CICS Transaction Gateway tracing	26
		Appendix. Trademarks	27

Chapter 1. Overview

The sample configuration in Figure 1 on page 2 shows how you can use Java servlets form a workstation to converse with a remote CICS across HTTP. The browser on the workstation uses only HTML pages.

The servlets are managed by WebSphere Application Server and run by the HTTP Server plug-in for WebSphere Application Server on the gateway machine.

The servlets issue ECI or EPI requests to the CICS Transaction Gateway for AIX, which makes the relevant calls to the coupled CICS Universal Client. The CICS Universal Client then passes the calls to the remote CICS server, using a TCP/IP connection.

Although the sample configuration uses a TCP/IP connection to the remote CICS server, you can configure your CICS Universal Client for AIX to communicate with any CICS server that supports ECI or EPI.

The servlet used in this configuration is the Terminal Servlet provided with CICS Transaction Gateway. The Terminal Servlet allows you to access CICS 3270 applications using a Web browser. You create Web pages that invoke the Terminal Servlet to start a CICS transaction, display screen information sent from a CICS server, and send screens back to CICS.

For full information on the Terminal Servlet, including how it maps 3270 screens to HTML, refer to your *CICS Transaction Gateway Administration* book.

In this configuration document we cover the following topics:

- “Chapter 2. Software checklist” on page 3
- “Installing the workstation software” on page 3
- “Chapter 3. Installing the Gateway Server software” on page 5
- “Chapter 4. Configuring the Gateway Server software” on page 9
- “Chapter 5. Starting and stopping the Gateway Server software” on page 13
- “Chapter 6. Configuring the Terminal Servlet” on page 15
- “Chapter 7. Testing using the supplied samples” on page 21
- “Chapter 8. Troubleshooting” on page 25

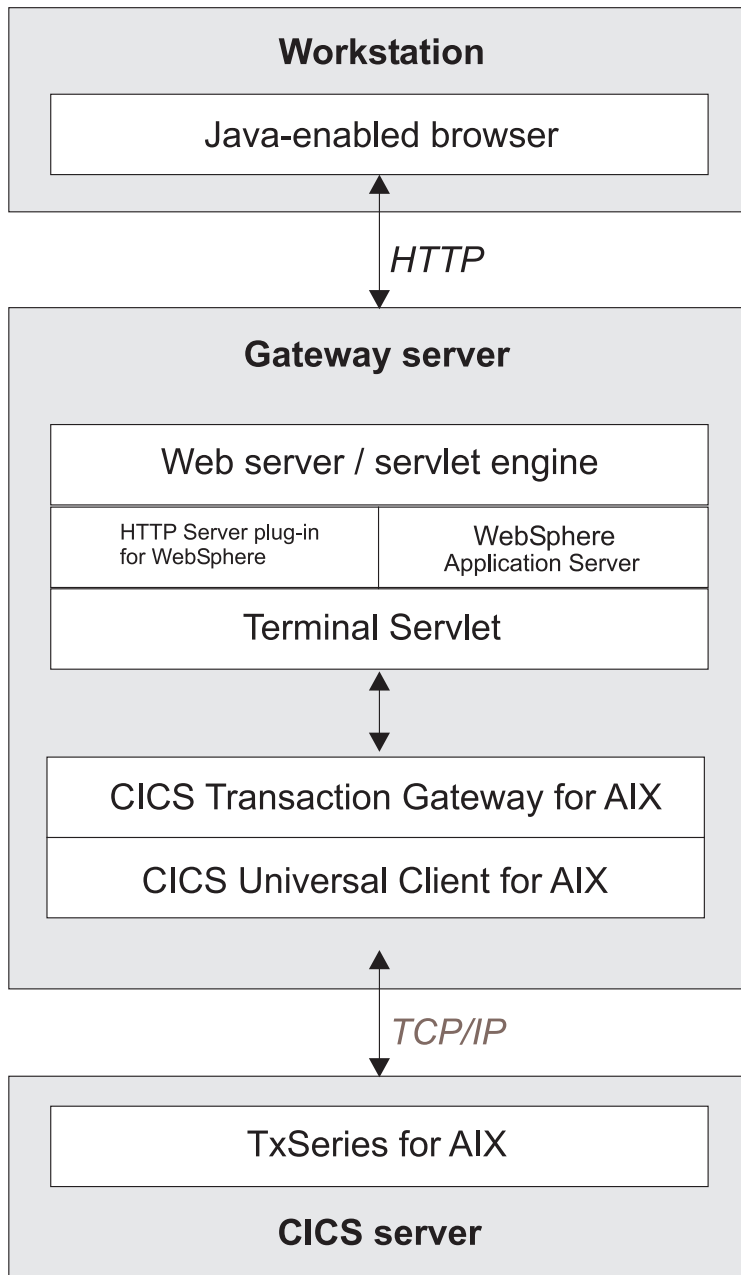


Figure 1. CICS Transaction Gateway using servlets across HTTP

Chapter 2. Software checklist

The levels of software we used in the sample configuration are not necessarily the latest levels available. Check the relevant products for levels of compatible software.

We used the following software on the workstation:

- Windows NT Workstation Version 4.0 (service level 5)
- Netscape Navigator for Windows NT Version 4.7

We used the following software on the gateway host:

- AIX Version 4.3.2
- Java Development Kit (JDK) Version 1.1.8 for AIX
- IBM HTTP Server Version 1.3.2 (formerly Apache Web Server)
- WebSphere Application Server Version 2.0.2 for AIX (includes HTTP Server plug-in)
- CICS Transaction Gateway for AIX Version 3.1 (includes CICS Universal Client for AIX Version 3.1)

We used the following software on the server host:

- TXSeries for AIX Version 4.3, which supports CICS access through TCP/IP
- CICS Transaction Gateway supplied EPIC transaction. See “Chapter 7. Testing using the supplied samples” on page 21 for details of how to prepare this transaction for use.

Installing the workstation software

This sample configuration assumes that Windows NT Version 4.0 is already installed and service level 5 has been applied. It also assumes Netscape Navigator for NT is installed and any available updates have been obtained from the Netscape Web site and applied.

No special configuration is needed for the workstation.

Chapter 3. Installing the Gateway Server software

Although this section describes the installation of all necessary gateway server software for this CICS Transaction Gateway scenario, the installation cannot be considered complete until you perform the configuration steps described in “Chapter 4. Configuring the Gateway Server software” on page 9.

You should install the software in the order given in this section, to avoid dependency problems.

HTTP Server

Follow the instructions for installing HTTP Server Version 1.3.2 and when complete add the following to your system PATH where appropriate:

```
/usr/lpp/HTTPServer/sbin
```

or enter the command:

```
ln -s /usr/lpp/HTTPServer/sbin/* /usr/sbin/
```

Now enter the command `apachectl start` to check that you can connect to the Web server, for example, <http://testc.hursley.ibm.com>.

Enter the command `apachectl stop` to stop the Web server.

JDK

You can install a JDK as part of the installation process for WebSphere. However, for the purposes of this configuration we used a pre-installed version of the JDK, Version 1.1.8 .

You can verify that the JDK is installed and working with the `java -version` command. Figure 2 shows the execution of this command on our system.

```
root@azov > java -version
java version "1.1.8"
root@azov >
```

Figure 2. Verifying a Correct JDK Installation

Installing the Gateway Server software

WebSphere Application Server

To install WebSphere Application Server for AIX:

1. Insert the installation CD into your CD-ROM drive.
2. Go to the subdirectory for the AIX operating system.
3. Run the setup file, `install.sh`. When asked whether you wish to install another version of the JDK, enter `n` for NO, and then enter the location for the `JAVA_HOME` environment variable (`/usr/jdk_base`), see Figure 3.
4. A series of panels guides you through the installation. Select the default installation path for WebSphere, that is, `/usr/WebSphere/AppServer`.

```
hydra /home/cicstest/poo # ./install.sh
java full version "JDK 1.1.8 IBM build a118ga-19990806 (JIT enabled; jtc V3.1-JDK1
,1-)"
Version JDK 1.1.8 IBM build a118ga-19990806 (JIT enabled; jtc V3.1-JDK1.1-) of JDK
 was found on the machine
The preferred version is JDK Version 1.1.6
Do you want to install JDK Version 1.1.6 [y/n]
n
Please enter your JAVA_HOME path:
/usr/jdk_base
```

Figure 3. WebSphere Application Server installation

CICS Transaction Gateway

The CICS Transaction Gateway is distributed as a compressed tar file containing the library files, commands, messages, and customization files, in a single directory tree.

To install CICS Transaction Gateway for AIX:

1. Log in as root.
2. Locate the `ctg-310x.tar` tar file on the distribution medium.
3. Copy the compressed tar file, `ctg-310x.tar.Z`, to a working directory on your hard disk.
4. Uncompress the file and extract the files, using the following commands:
5. `uncompress ctg-310x.tar`
6. `tar -xvf ctg-310x.tar`
7. Set up the CICS Transaction Gateway by executing the following scripts:
8. `mkcicscli`
9. `mkclimsgs us`

The `mkcicscli` shell script partially removes any previous CICS Universal Client installation and then installs the CICS Universal Client supplied with CICS Transaction Gateway.

Figure 4 shows the commands we used to install CICS Transaction Gateway for AIX.

```
root@azov > cp /downloads/ctg-310x.tar.Z /temp/  
root@azov > cd /temp  
root@azov > uncompress ctg-310x.tar  
root@azov > tar -xvf ctg-310x.tar  
...  
root@azov > cd /usr/lpp/ctg  
root@azov > ./mkcicscli  
...  
root@azov > ./mkclimsgs us  
...
```

Figure 4. Installing CICS Transaction Gateway for AIX

Chapter 4. Configuring the Gateway Server software

Start with the configuration steps in this section only after completing all installation steps described in “Chapter 3. Installing the Gateway Server software” on page 5.

Perform the configuration steps in the order in which they are given.

HTTP Server Plug-in for WebSphere Application Server

To enable WebSphere Application Server to work together with HTTP Server make sure that the file `/usr/lpp/HTTPServer/etc/httpd.conf` contains the following:

```
LoadModule ibm_app_server_module /usr/WebSphere/AppServer/plugins/aix/mod_ibm_app_server.so
:
:
Alias /IBMWebAS/samples/ /usr/WebSphere/AppServer/samples/
Alias /IBMWebAS/ /usr/WebSphere/AppServer/web/
NcfAppServerConfig BootFile /usr/WebSphere/AppServer/properties/bootstrap.properties
NcfAppServerConfig LogFile /usr/WebSphere/AppServer/logs/apache.log
NcfAppServerConfig LogLevel TRACE|INFORM|ERROR
```

Figure 5. Configuring HTTP Server file httpd.conf

Add the following line at the end of the file:

```
Alias /ctg/ /usr/lpp/ctg/
```

Shell script for stopping WebSphere Application Server

If WebSphere Application Server is not cleanly stopped, it will not start successfully the next time. Although you can stop WebSphere Application Server by killing processes manually, we recommend using a shell script.

Figure 6 on page 10 shows the shell script, `websphere_stop`, we used for stopping WebSphere Application Server.

Configuring the Gateway Server software

```
#!/bin/ksh
# userid of HTTP Server
# NOTE: Please edit this value to correspond to the userid
# that the Web server runs as specified in the User directive of
# httpd.conf
user=nobody

os='uname -a|awk '{print $1}''

# clean-up Application Server out-of-process procs
echo "bypass fix for Application Server residual OutOfProc"

if [[ $os = "AIX" ]]; then
    pid='ps -eaf|grep -v grep|grep "^ *${user}"|grep "OutOfProcEng"|awk '{print $2}''
fi

if [[ $pid -ne 0 ]]; then
    echo "killing Application Server OutOfProcessEng pid $pid";
    kill $pid;
fi
```

Figure 6. Shell script `websphere_stop`

We placed the `websphere_stop` shell script in the HTTP Server `/usr/lpp/HTTPServer/sbin` directory.

Configuring CICS Transaction Gateway

You use the CICS Transaction Gateway's configuration tool to define the CICS servers and make the servers reachable by the CICS Transaction Gateway. The configuration tool generates the CTG.INI file, which is located in the `/usr/lpp/ctg/bin` directory. The CICS Transaction Gateway uses the CICS Universal Client and the CTG.INI file to establish a connection to a CICS server.

For information on using the configuration tool, refer to your *CICS Transaction Gateway Administration* book.

You may need to change the following **Server** configuration settings (see Figure 7 on page 11):

Server name

An arbitrary name for a particular CICS server.

Description

An arbitrary description for the CICS server.

Network protocol

A protocol for communication with the CICS server. We use TCPIP in our sample.

Hostname or IP address

The TCP/IP address of the CICS server.

Configuring the Gateway Server software

Port The TCP/IP port number of the CICS server. Set this to zero to use the default TCP/IP port number of the CICS server.

The *CICS Universal Client Administration* book and the configuration tool's online help provide descriptions of the configuration settings for CICS Universal Client.

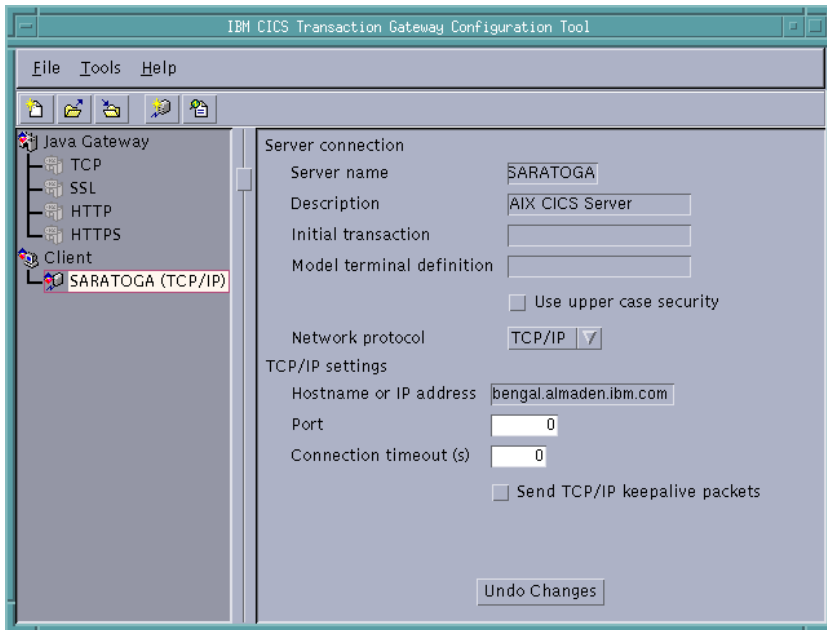


Figure 7. Configuration tool settings for using servlets

Figure 8 on page 12 shows an excerpt from the resultant CTG.INI file that would make the CICS AIX server known to the CICS Transaction Gateway.

Configuring the Gateway Server software

```
SECTION CLIENT = *
:::
ENDSECTION
:::
SECTION SERVER = SARATOGA
DESCRIPTION=AIX CICS Server
:::
PROTOCOL=TCPIP
NETNAME=bengal.almaden.ibm.com
PORT=0
ENDSECTION
:::
SECTION DRIVER = TCPIP
DRIVERNAME=CCLIBMIP
ENDSECTION
```

Figure 8. CTG.INI file definitions

Chapter 5. Starting and stopping the Gateway Server software

This section describes how to start and stop the software products installed and configured in the preceding sections.

Starting CICS Transaction Gateway

To start CICS Transaction Gateway, issue the following commands:

1. `cd /usr/lpp/ctg/bin`
2. `cicscli -s`
3. `./ctgstart`

Figure 9 shows the CICS Transaction Gateway being started.

```
root@azov > cd /usr/lpp/ctg/bin
root@azov > ./ctgstart
CICS Transaction Gateway, Version 3.1.0, 5648-B43.
(C) Copyright IBM Corporation 1999. All rights reserved.
CCL6500I: Starting the CICS Transaction Gateway with default values.
CCL6502I: [ Initial ConnectionManagers = 1, Maximum ConnectionManagers = 100,
CCL6502I:   Initial Workers = 1, Maximum Workers = 100, tcp: Port = 2006 ]
CCL6505I: Successfully created the initial ConnectionManager and Worker threads.
CCL6524I: Successfully started handler for the tcp: protocol.
```

Figure 9. Starting the CICS Transaction Gateway

Stopping CICS Transaction Gateway

Enter the character `q` in the CICS Transaction Gateway startup session console to terminate the program.

Starting HTTP Server

When HTTP Server is started, WebSphere Application Server is also automatically started.

To start HTTP Server, issue the following command:

```
/usr/lpp/HTTPServer/sbin/apachectl start
```

Figure 10 on page 14 shows the starting of HTTP Server and WebSphere Application Server.

Starting and stopping the Gateway Server software

```
root@azov > /usr/lpp/HTTPServer/sbin/apachectl start
/usr/lpp/HTTPServer/sbin/apachectl start: httpd started
root@azov
```

Figure 10. Starting HTTP Server and WebSphere Application Server

Stopping HTTP Server and WebSphere

We assume you have created the `websphere_stop` shell script as described in see “Shell script for stopping WebSphere Application Server” on page 9.

To stop HTTP Server, issue the following command:

```
/usr/lpp/HTTPServer/sbin/apachectl stop
```

To stop WebSphere Application Server, issue the following command:

```
/usr/lpp/HTTPServer/sbin/websphere_stop
```

Figure 11 shows the stopping of HTTP Server and WebSphere Application Server.

```
root@azov > cd /usr/lpp/HTTPServer/sbin
root@azov > ./apachectl stop
./apachectl stop: httpd stopped
root@azov > ./websphere_stop
bypass fix for Application Server residual OutOfProc
killing Application Server OutOfProcEng pid 33630
root@azov >
```

Figure 11. Stopping HTTP Server and WebSphere Application Server

Failure to execute `websphere_stop` after you stop HTTP Server can lead to problems when you restart HTTP Server and WebSphere Application Server.

Chapter 6. Configuring the Terminal Servlet

This section describes how to use the WebSphere Application Server to prepare the sample servlet, TerminalServlet, supplied with the CICS Transaction Gateway.

Loading the TerminalServlet

You load and configure TerminalServlet from the WebSphere Application Server Manager, the administrative interface, using the Netscape browser on the Windows NT workstation.

To start the WebSphere Application Server Manager, point the browser to a URL like the following:

```
http://azov.almaden.ibm.com:9527
```

Substitute your own gateway server host name in the above URL.

After the WebSphere Application Server Manager applet loads and initializes, you see the logon screen shown in Figure 12 on page 16. Log on with user name admin and password admin.

Configuring the Terminal Servlet



Figure 12. WebSphere Application Server Administration Tool Logon screen

After logging on, following these steps to load the TerminalServlet.

1. Select **Setup**.
2. Select **Java Engine**.
3. Add the following to the end of the **Application Server Classpath** field:
`/usr/lpp/ctg/classes/ctgclient.jar:/usr/lpp/ctg/classes/ctgserver.jar`
4. Add the following to the end of the **User Libpath** field:
`/usr/lpp/ctg/bin`

Then select **Save**.

5. Select **Servlets**.
6. Select **Configuration**.
7. Select **Add**, located on the left of the screen. The Add a New Servlet dialog is displayed, see Figure 13 on page 17

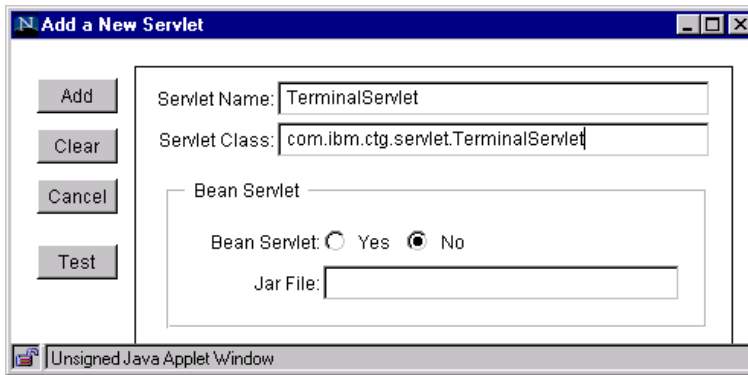


Figure 13. Add a New Servlet dialog

8. In the **Servlet Name** field, enter TerminalServlet.
9. In the **Servlet Class** field, enter com.ibm.ctg.servlet.TerminalServlet.
10. Select **Add**, located on the left-hand side of the panel.
The TerminalServlet configuration dialog is displayed (see Figure 14 on page 18).
11. In the **Description** field, enter an arbitrary description.
12. Select **Yes** for the Load at Startup radio button.
13. Select **Loaded Now**.

You must now close down the Web server and HTTP Server so that the new path settings can be used:

1. apachectl stop
2. webserver_stop
3. apachectl start

Configuring the Terminal Servlet

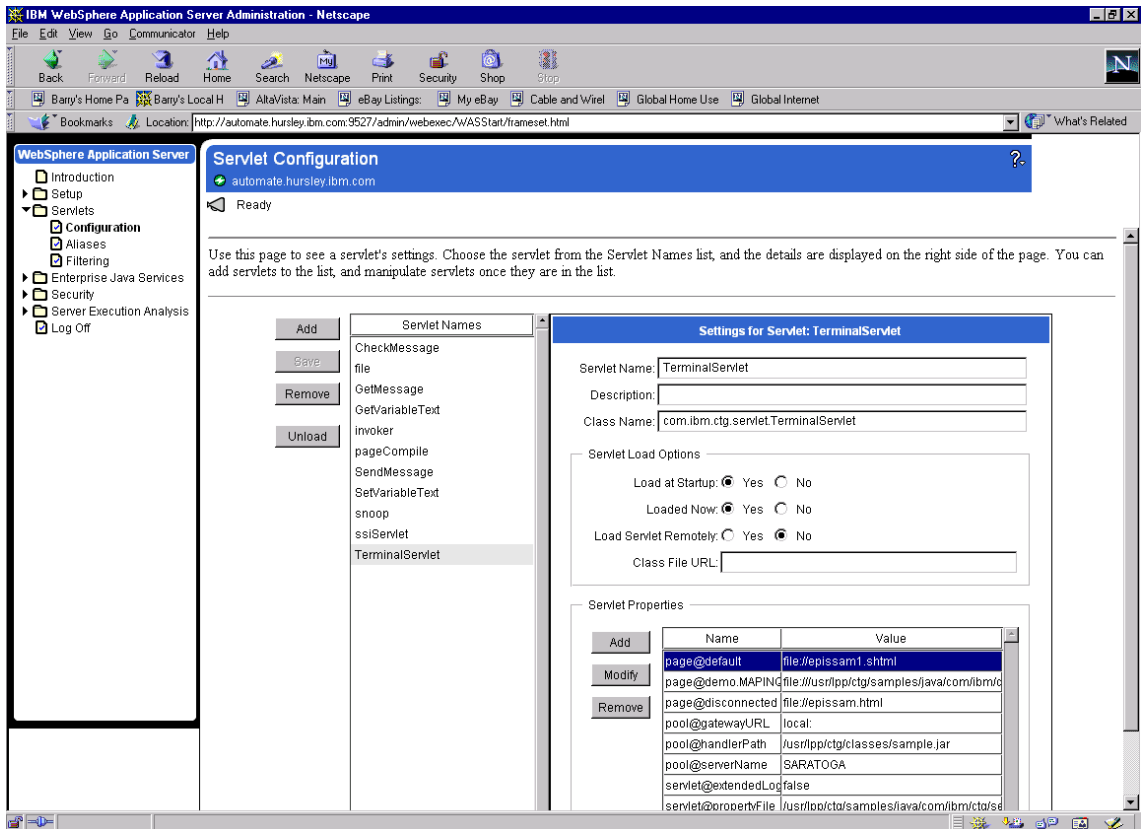


Figure 14. The TerminalServlet Configuration dialog

Configuring the parameters

Use the WebSphere Application Server Manager servlet properties dialog to configure the TerminalServlet parameters for use with the samples provided with the CICS Transaction Gateway server.

Use the following parameters and values:

page@default

Example: `file://epissam1.shtml`

This parameter specifies the default server-side include file that is used for returning CICS HTML to the browser.

Page mapping properties can be set either to a URL, or to point to a template file. This must start with `file://`, and if you set the **servlet@templateDir** property, you can set template filenames relative to a template directory, as in our example.

page@demo.MAPINQ1ScreenHandler

Example:

```
file:///usr/lpp/ctg/samples/java/com/ibm/ctg/servlet/epissam3.html
```

This parameter specifies the HTML page that is used as a template for mapping CICS 3270 map fields to HTML tags.

page@disconnected

Example: `file://epissam.html`

This parameter specifies the HTML file that will be displayed when the user clicks on the disconnect button from a CICS HTML page. File `epissam.html` contains the URL links that run the CICS Transaction Gateway servlet samples.

pool@gatewayURL

Example: `local:`

This parameter specifies the location of the CICS Transaction Gateway. For our scenario, the CICS Transaction Gateway is located on the same gateway server host as the WebSphere Application Server, so we use the value `local:`.

pool@handlerPath

Example: `/usr/lpp/ctg/classes/sample.jar`

This parameter specifies the archive containing the Java classes used by the samples.

pool@serverName

Example: `SARATOGA`

This parameter value corresponds to the arbitrary server name defined using the configuration tool, see “Configuring CICS Transaction Gateway” on page 10. Use the CICS server name you specified there.

servlet@extendedLogging

Example: `false`

Set this parameter to true to enable logging. See “Chapter 8. Troubleshooting” on page 25 for more information about using this parameter.

servlet@propertyFile

Example:

```
/usr/lpp/ctg/samples/java/com/ibm/ctg/servlet/servlet.properties
```

This parameter specifies the file containing default parameters for the TerminalServlet. The parameters described in this list are also described in the `servlet.properties` file.

Configuring the Terminal Servlet

servlet@templateDir

Example: /usr/lpp/ctg/samples/java/com/ibm/ctg/servlet

This parameter specifies the directory where the HTML pages and server-side includes are located.

Refer to see Figure 14 on page 18, and follow these steps:

1. Select **Add** in the Servlet Properties section.
2. Type in a parameter name in the **Parameter Name** field.
3. Use the tab key to move to the **Value text** field and type in a parameter value.
4. Press Enter.
5. Repeat steps 2 through 5 for each parameter listed in the preceding paragraph.
6. Click on the **Save** button.
7. Click on the **Unload** button.
8. Click on the **Load** button to activate the changes you have made.

The TerminalServlet is now loaded, configured, and ready for use with the samples provided with CICS Transaction Gateway.

Full information on Terminal Servlet parameters is given in the *CICS Transaction Gateway Administration* book.

Chapter 7. Testing using the supplied samples

After you have installed and configured all relevant products for the sample configuration, we recommend that you:

Test your configuration, using the samples provided with the CICS Transaction Gateway. The samples include the EPIC transaction, which you must place on your remote CICS server. The CICS Universal Client includes several files needed for the sample CICS server programs. These files are located in the /samples/Server directory and include:

- Epiinq.bms
- Epiinq.ccs

Copy the files to your CICS server and use them to build the CICS EPIC transaction. Refer to the documentation supplied with your CICS server for information about building CICS transactions.

To display the CICS Transaction Gateway Terminal Servlet Samples page, point your browser to a URL like the following:

`http://azov.almaden.ibm.com/servlet/TerminalServlet?request=disconnect`

Substitute the host name of your own gateway server host in the URL.

Figure 15 on page 22 shows the CICS Transaction Gateway Terminal Servlet Samples page. The page contains URL links to the samples supplied by the CICS Transaction Gateway and instructions on using the samples.

Testing using the supplied samples

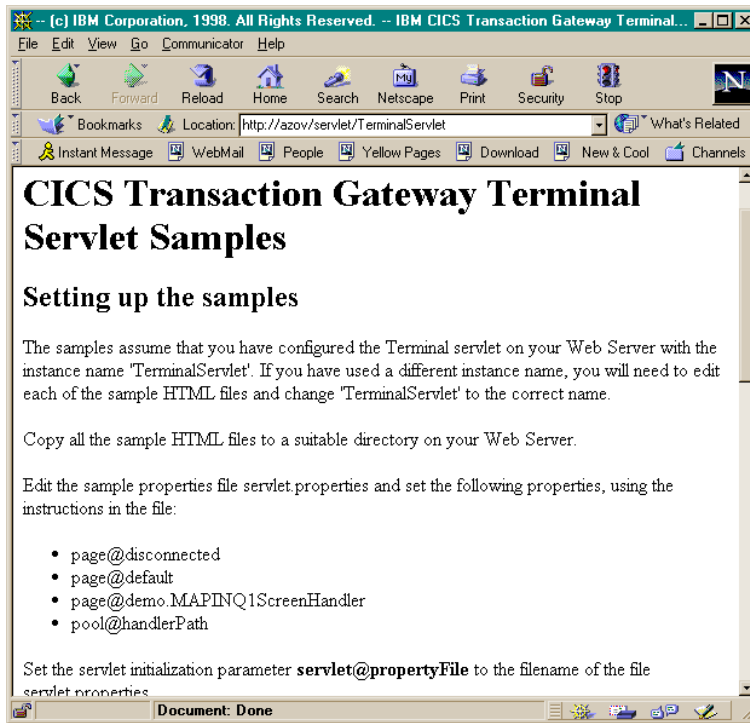


Figure 15. CICS Transaction Gateway Terminal Servlet Samples page

Select the link **Basic Terminal Emulation** on the CICS Transaction Gateway Terminal Servlet Samples page to display the CICS Transaction Gateway Terminal Emulation page. Figure 16 on page 23 shows the CICS Transaction Gateway Terminal Emulation page displaying the CICS CECI transaction.

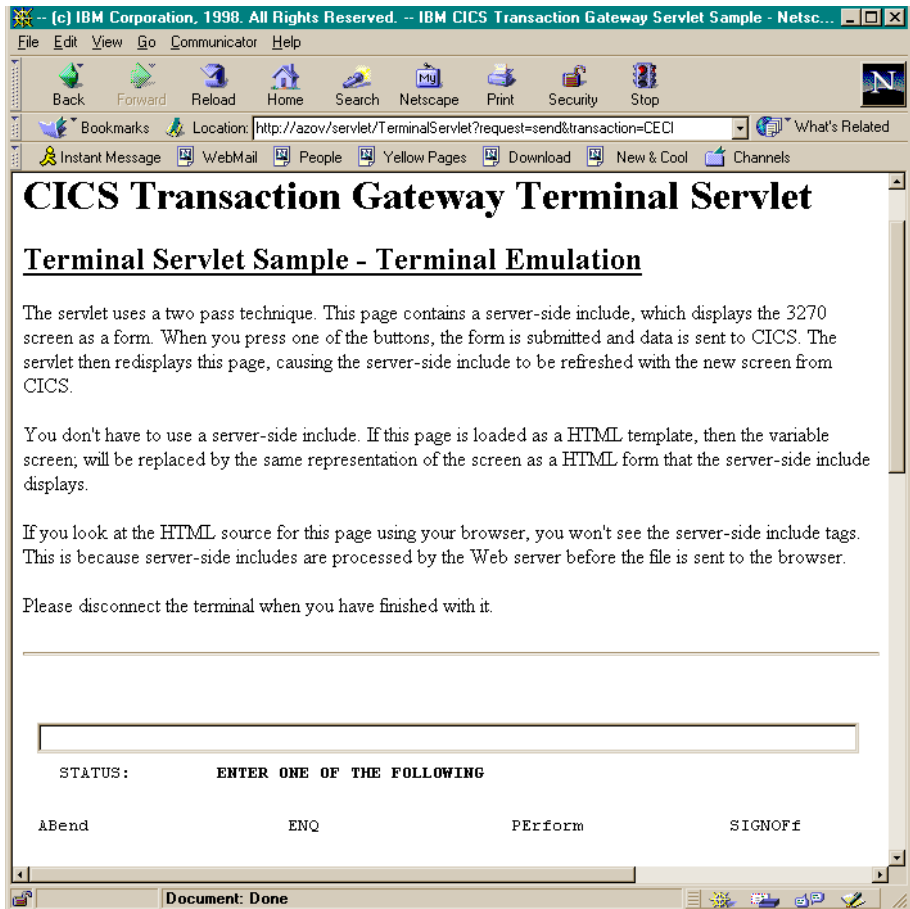


Figure 16. CICS Transaction Gateway Terminal Emulation page

Exit from the terminal emulation screen. Use the **disconnect** button to terminate the CICS session and return to the CICS Transaction Gateway Terminal Servlet Samples page.

Select the link **HTML Template Sample** on the CICS Transaction Gateway Terminal Servlet Samples page to start the EPIC transaction on the CICS server and display the EPIC Transaction page (see Figure 17 on page 24).

Testing using the supplied samples

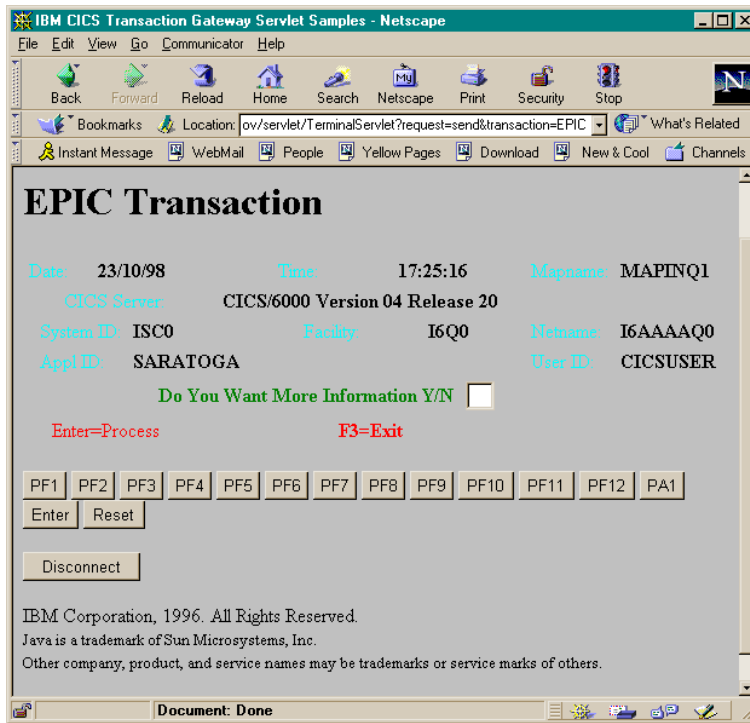


Figure 17. CICS EPIC Transaction page

Exit from the EPIC transaction. Use the **disconnect** button to terminate the session on the CICS server and return to the CICS Transaction Gateway Terminal Servlet Samples page.

Chapter 8. Troubleshooting

In this section we describe some of the facilities available for troubleshooting your configuration.

HTTP Server and WebSphere Application Server log files

Both HTTP Server and WebSphere Application Server write errors and other useful information to the log files, `error_log` and `access_log`.

For HTTP Server these files are located in:

```
/usr/lpp/HTTPserver/var/log
```

and for WebSphere Application Server Version 2.0.2 in:

```
/usr/WebSphere/AppServer/Logs/servlet/servlet/servletservice
```

WebSphere Application Server also writes to the `event_log` file in this directory. This contains information about the events associated with the errors.

You should examine these files to find the causes of errors and configuration problems.

TerminalServlet logging

The TerminalServlet is usually configured for minimal logging and maximum performance. You can increase the amount of TerminalServlet logging written to the Apache Web Server log file by modifying the TerminalServlet parameter, **`servlet@extendedLogging`**.

See “Configuring the parameters” on page 18 for more information on setting the **`servlet@extendedLogging`** parameter. Set the parameter to true to increase the level of logging. Reset the parameter to false when logging is no longer required.

CICS Universal Client log

The CICS Universal Client writes error messages to the file `CICSCLI.LOG` in the `/var/cicscli` directory.

These messages are explained in the *CICS Universal Client Messages* book.

CICS Transaction Gateway tracing

You can display trace information to the CICS Transaction Gateway console by starting the product with a trace parameter. Use the following start command to initiate tracing:

```
./ctgstart -trace
```

See “Chapter 5. Starting and stopping the Gateway Server software” on page 13 for details on how to use the ctgstart command.

Appendix. Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

Anynet	CICS
IBM	OS/390
VTAM	

Java, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, or other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.