Using CICS Transaction Gateway, High Availability, and the CICS Explorer

This IBM® Redpaper™ publication describes how the CICS® Explorer™ can be used to aid configuration and administration of the CICS Transaction Gateway for z/OS®. Starting from a basic system configuration, it takes you through the steps to create a high availability topology and implement request monitoring.

At a high level, the procedures described are:

- Configure CICS Transaction Gateway (CICS TG) for z/OS
- Set up the CICS TG plug-in for CICS Explorer
- Start up and administer the Gateway daemon on z/OS
- Configure CICS Transaction Server (CICS TS) connections to CICS TG
- Configure CICS TG connections to CICS TS
- Configure and run a test ECI workload
- Configure Dynamic Server Selection in the Gateway daemon
- Configure request monitoring in the Gateway daemon

It is assumed that the reader has CICS Explorer available with the CICS SM and z/OS perspectives operational.
Overview

IBM CICS Transaction Gateway (CICS TG) is IBM’s premier connector solution for CICS Transaction Server for z/OS and TXSeries. CICS TG provides access to CICS from a range of application programming environments, including JEE and .NET, across a range of platforms (z/OS, Linux, AIX®, Windows, HP-UX, Solaris), and with a proven track record for scalability, performance, robustness and security.

CICS Explorer is the central pillar of IBM’s strategy to modernize administration of CICS Transaction Server. Because CICS Transaction Gateway is a key integration component in many business solutions that leverage resources through CICS, system programmers need the ability to seamlessly monitor and administer CICS TG alongside CICS Transaction Server. Using the CICS Explorer as an integration point for administration, the same skills can be used to configure and administer components throughout the enterprise.

While this paper concentrates on the CICS TG for z/OS product, many of the concepts are also applicable to the Multiplatforms product. The CICS TG plug-in for CICS Explorer interoperates with the Gateway daemon regardless of the run time platform. The high availability configuration is also possible with CICS Transaction Gateway for Multiplatforms Version 8.1.

Introduction

This section shows how to obtain or verify access to systems through CICS Explorer, and how to locate the CICS TG installation and configuration files. In addition to providing a review of the topology, it describes how to:

- Identify CICS regions, user IDs, port numbers, product installation files
- Define the CICS TG tcp handler port
- Define the CICS TG statistics API handler port
- Configure and verify z/OS and CICS SM perspective connectivity
- Locate and review CICS TG configuration files

Topology review

This paper is based around use of the CICS TG in remote mode (3-tier), and running on z/OS. This topology is also known as Topology 2, described in the whitepaper “Integrating WebSphere® Application Server and CICS using CICS
Using CICS Transaction Gateway, High Availability, and the CICS Explorer

Transaction Gateway,” which can be found at:

While that paper uses WebSphere Application Server as the source of application requests, it should be noted that support is available for any J2EE 1.4, JEE 5, or JEE 6 certified application servers through the JCA Install Verification Test (IVT) program with CICS TG V7.1, or later.

Furthermore, application requests can originate from any Java 5 Standard Edition (or later) run time environment (for example, a Servlet engine such as Apache TomCat, a browser Applet, or a Java application). The ECIv2 and .NET APIs allow applications written in C, C++, COBOL, or .NET to connect to CICS through the Gateway daemon, in parallel with Java-based applications.

This paper uses a combination of the CICS Explorer and CICS Support Pacs to allow configuration, administration, and verification of the topology shown in Figure 1 on page 4.

Note: The topology covered by this paper describes a single Gateway daemon and a single LPAR system. However, the Gateway daemon instance could be replicated multiple times within the same LPAR, given appropriate naming conventions and resource allocation, with workload distribution to the tcp handler port through TCPIP port sharing. The solution might be scaled still further using multiple LPARs with workload distribution between LPARs through Sysplex Distributor, or equivalent.
Assumptions

Your CICS Explorer is already configured to either connect to a CICSPlex® SM WUI region (using a CICSPlex SM Data Interface connection), or directly to a stand alone CICS TS 4.1 (or later) region (using a CICS Management Interface connection). One such connection is required to create and install the CICS definitions for this paper, through the CICS Explorer CICS SM perspective, as well as to perform operations on CICS resources. These operations could be completed using CICS-supplied transactions CEDA and CEMT, but this paper demonstrates the CICS Explorer method. It can be seen from the screen captures within the paper that we used a CICSPlex SM Data Interface connection named CICSPLEX1.
There is a limited requirement for a 3270 TSO session to check the progress of running JOBs. Although this is useful it is not strictly necessary.

Support Pac CH50 (CTGPING) includes the COBOL CICS program PONG, which must be compiled, defined, and installed to the CICS region.

CICS TS 3.2 or later is required to complete the High Availability example since this uses an IPIC connection to CICS TS. A second EXCI connection could substitute for the IPIC connection if IPIC is not available. Both protocols are utilized in this paper purely for demonstration purposes, and not for any technical reason.

CICS Transaction Gateway for z/OS V8.0 is used throughout this paper. However, V8.1 can be used directly in place of V8.0, or V7.2 with Support Pac CA1T V1.1 in place of CA1T V2.0 (or later).

**System resources**

For simplicity, the same user ID is used to access z/OS from CICS Explorer and to run the Gateway daemon. This TSO/OMVS ID requires access to the UNIX System Services file system, FTP, CICS definitions, and CICS resources.

| Table 1  System resources

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSO / OMVS ID</td>
<td>RCJONES / rcjones</td>
</tr>
<tr>
<td>LPAR ID</td>
<td>LP01</td>
</tr>
<tr>
<td>LPAR dns address</td>
<td>lp01.redbooks.ibm.com</td>
</tr>
</tbody>
</table>

The user ID used to run the Gateway daemon must have an OMVS segment. The CICS TG Information Center describes the full requirements for the Gateway daemon user ID.

**CICS Transaction Server resources**

The CICS server will use two types of connections to communicate with CICS TG: EXCI and IPIC. Table 2 on page 6 summarizes the CICS Transaction Server resources and definition values that we used for the examples in this book.
### Table 2  CICS TS resources

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS APPLID</td>
<td>CICSDM01</td>
</tr>
<tr>
<td>IPIC TCPIP Service port number</td>
<td>14101</td>
</tr>
<tr>
<td>EXCl connection net name</td>
<td>CTGEXC01</td>
</tr>
</tbody>
</table>

### CICS Transaction Gateway resources

The CICS Transaction Gateway needs to communicate with its remote applications (tcp handler), the plug-in for CICS Explorer (stats handler), and the CICS server (EXCI). Table 3 summarizes the CICS Transaction Gateway resources and definition values that we used.

### Table 3  CICS TG resources

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLID</td>
<td>DMCTG01</td>
</tr>
<tr>
<td>APPLID Qualifier</td>
<td>LP01</td>
</tr>
<tr>
<td>tcp handler port number</td>
<td>14201</td>
</tr>
<tr>
<td>stats handler port number</td>
<td>14301</td>
</tr>
<tr>
<td>EXCI specific pipe name</td>
<td>CTGEXC01</td>
</tr>
</tbody>
</table>

**Note:** The last four characters of the CICS TG APPLID are used as the IPIC Connection name when using auto-installed connections. In this case “TG01”.

### Additional materials

All of the CICS TG configuration and Support Pac materials used in this paper are included in the Redpaper additional materials. The UNIX System Services files contained in the tar ball were uploaded and unarchived in the home folder of our Gateway daemon user ID. The MVS™ file resources were uploaded by FTP, received (using TSO RECEIVE INDS=xxx) and then untered under the Gateway daemon user id high level qualifier.

The locations of the Support Pac materials are identified in Table 4 on page 7.
Table 4  Support Pac materials

<table>
<thead>
<tr>
<th>Support Pac</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH50 / CTGPING</td>
<td>/u/rcjones/ctg800demo/ch50</td>
</tr>
<tr>
<td>CH50 / CTGPING</td>
<td>RCJONES.CTG800.DEMO(PONG)</td>
</tr>
<tr>
<td>CA1T / High Availability</td>
<td>/u/rcjones/ctg800demo/ca1t</td>
</tr>
<tr>
<td>CH51 / Request Monitoring</td>
<td>/u/rcjones/ctg800demo/ch51</td>
</tr>
</tbody>
</table>

Note: The CH50 CICS program PONG must be compiled, defined, and installed to the CICS region.

The Redpaper Additional Materials can be downloaded from the following address:

ftp://www.redbooks.ibm.com/redbooks/REDP4782/

Alternatively, you can go to the IBM Redbooks® Web site at:

http://ibm.com/redbooks

Select Additional materials and open the directory that corresponds with the Redpaper form number, REDP4782.

CICS TG product files

CICS Transaction Gateway for z/OS is SMP/E maintained and consists of several MVS data sets and a product folder on the UNIX System Services file system. The location of our CICS TG installation resources are shown in Table 5.

Table 5  CICS TG product files

<table>
<thead>
<tr>
<th>Product system files</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS</td>
<td>CICSTG.CTG800.*</td>
</tr>
<tr>
<td>UNIX System Services (HFS/zFS)</td>
<td>/usr/lpp/cicstg/ctg800</td>
</tr>
<tr>
<td>CICS TG plug-in for CICS Explorer</td>
<td>(Workstation)</td>
</tr>
<tr>
<td></td>
<td>cicstg_plugin_v1002.zip</td>
</tr>
</tbody>
</table>
Basic setup of CICS TG for z/OS

This section shows how to configure and start the Gateway daemon using the CICS Explorer. At a high level, setup includes the following procedures:

- Review the CICS TG JCL, STDENV, and INI file.
- Using the z/OS perspective, customize the CICS TG JCL and configuration files.
- Configure a tcp handler (for incoming application connections).
- Start and stop the Gateway daemon from CICS Explorer.

Detailed steps for each part of the setup process are provided in the following sections.

Review CICS TG configuration resources

Table 6 identifies the UNIX System Services file and PDS members in our Gateway daemon configuration.

<table>
<thead>
<tr>
<th>Location</th>
<th>Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>/u/rcjones/ctg800demo</td>
<td>ctg800.ini</td>
</tr>
<tr>
<td>RCJONES.CTG800.DEMO</td>
<td>CTGENV</td>
</tr>
<tr>
<td>RCJONES.CTG800.DEMO</td>
<td>CTGSTART</td>
</tr>
<tr>
<td>RCJONES.CTG800.DEMO</td>
<td>CTGSTOP</td>
</tr>
</tbody>
</table>

Note: With CICS TG for z/OS V8.1, the Gateway daemon configuration file can also be stored in an MVS data set, along with the JCL and environment variables.

Configure CICS TG

A JCL job is required to start the Gateway daemon address space. The template JCL job CTGSTART is customized with the install location of the CICS TG product files (refer to Table 5 on page 7). CICS TG for z/OS configuration consists of two parts: a set of environment variables and a text-based configuration file.
CTGBATCH is the CICS TG utility program for z/OS used to execute UNIX System Services programs or shell scripts. It also reads environment variables from the STDENV DD card.

Use the following steps to load, edit, and save the configuration resources for the Gateway daemon using the CICS Explorer z/OS perspective:

1. Locate and review MVS files CTGSTART, CTGSTOP, and CTGENV, and the UNIX System Services file ctg800.ini in the z/OS editor.

2. Open the CTGSTART JCL file in the z/OS editor (Figure 2 on page 10). It uses CTGBATCH to run the ctgstart UNIX Shell Script. The location of the MVS product files is defined by JCL variable CTGHLQ; the UNIX System Services product files are referenced by JCL variable CTGHOME.

   Verify that the CTGSTART JCL is using the correct product files.

3. Update the CTGSTART JCL to ensure that the CTGBATCH job step uses the intended set of environment variables. The STDENV DD card should be defined as <userid>.CTG800.DEMO PDS member CTGENV.

   Press Ctrl+S to save your changes.

**Note:** Always use lower case for UNIX System Services paths or file names unless otherwise specified.
4. Open the PDS member CTGENV in the z/OS editor. Update it to ensure that the Gateway daemon address space started by JCL job CTGSTART uses the intended Gateway daemon configuration file:

```
/u/<userid>/ctg800demo/ctg800.ini
```

Press Ctrl+S to save your changes.

**Explanation:** The CICSCLI environment variable is used to define the UNIX System Services path and file name for the Gateway daemon configuration file.

5. Switch to the **z/OS UNIX Files** tab on the z/OS Explorer view. Open the UNIX System Service file `ctg800.ini` in the z/OS editor (Figure 3 on page 11).

Make the following modifications using the values from Table 2 on page 6 and Table 3 on page 6:

- Update the Gateway daemon APPLID.
- Update the default server definition with the target CICS APPLID.
- Update the tcphandler definition port number.
Use Ctrl+S to save your changes.

**Figure 3  Editing the Gateway daemon configuration file**

**Explanation:** The Gateway daemon APPLID is used to identify a specific Gateway daemon instance.

The default server definition ensures there is at least one candidate CICS region to process requests that do not specify a CICS server. This is the simplest form of Dynamic Server Selection and allows applications to be agnostic of their deployment topology.

The tcphandler definition instructs the Gateway daemon to listen for connections from client applications on the specified tcpip port number.

**Start the Gateway daemon**

Use these steps to start the Gateway daemon:

1. Submit the CTGSTART job from the z/OS perspective of CICS Explorer. This can be done by either of these methods:
   a. Right-click the PDS member CTGSTART on the Data Sets tab of the z/OS Explorer view, or click anywhere in the open editor window for CTGSTART.
b. Select **Submit Job** from the context menu (highlighted in Figure 4) or click the **Submit Job** icon (circled in the figure).

![Figure 4  Submitting the Gateway daemon JCL job](image)

2. You should be able to see the Gateway daemon job running in the Jobs view of the z/OS perspective. If you do not see any entry for your Gateway daemon job, click the **Refresh** icon (Figure 5).

![Figure 5  Reviewing the Gateway daemon job status](image)

3. Logon to TSO using a 3270 terminal session, using the TSO user ID shown in Table 1 on page 5. Review the progress of the JOB through the SDSF interface. If the Gateway has started successfully, you will find the following message:

   CTG6512I CICS Transaction Gateway initialization complete
Stop the Gateway daemon

Assuming that the Gateway has started, you can now stop the Gateway daemon using the CTGSTOP job:

1. Modify the CTGSTOP JCL job through the z/OS perspective, such that the modify command issued by the job specifies your Gateway daemon JOB name. Save and submit the job.

2. The CTGSTOP JCL job issues the following modify command from the SDSF command:

   /F <jobname>,APPL=SHUT

3. Once the Gateway daemon has completed shutdown, the job output can be reviewed within the z/OS perspective. Refresh the Jobs view and if the CTGDMnn job icon has changed to Complete (RC=0), simply double-click the entry in the Jobs view and the output will appear in the main editor view.

Set up CICS TG plug-in for CICS Explorer

This section shows how to install the CICS TG plug-in for CICS Explorer, and how to configure a connection to a Gateway daemon. At a high level, setup includes the following procedures:

- Configure a stats API handler (for CICS TG plug-in for CICS Explorer).
- Install the CICS TG plug-in for CICS Explorer.
- Define a new CICS TG connection.
- Connect to the Gateway and review the Gateway daemon, CICS connections, and properties views.
- Verify connectivity to the tcp protocol handler using the Test connection feature.

Note: Using the CICS Explorer 1.1, it is not possible to review active jobs unless you have the CICS Deployment Assistant plug-in installed and configured. Since this paper does not presume anything more than CICS Explorer and the CICS TG plug-in, we suggest using a 3270 session to confirm our progress.
Configure the statistics API handler

The CICS TG statistics API allows remote users to obtain live statistical data from a Gateway daemon at run time. The CICS Transaction Gateway plug-in for CICS Explorer uses this facility to retrieve information on the Gateway daemon status. The statistics API listener port is not enabled by default. Therefore, you need to add a statistics API handler definition to the configuration file. The definition is very similar to that of the tcphandler described previously.

Return to the Gateway daemon configuration file, the UNIX System Service file ctg800.ini in the z/OS editor. Below the tcphandler definition, uncomment the statistics API handler and update the port number as per Table 3 on page 6:

```
#Listener port for statistics API/CICS Explorer
protocol@statsapi.handler=com.ibm.ctg.server.RestrictedTCPHandler
protocol@statsapi.parameters=port=14301;bind=;connecttimeout=2000;maxconnection=15;
```

Save your changes by pressing Ctrl+S, then start the Gateway daemon again by submitting the CTGSTART JCL job.

Install the CICS TG plug-in for CICS Explorer

The CICS TG plug-in for CICS Explorer “update site” has been provided as a zip file. Install the plug-in using these steps:

1. From the CICS Explorer Help menu, select Install New Software. The Install dialog window is displayed (Figure 6).
2. Click **Add**. From the Install dialog window, select **Archive**. Use the standard file select dialog to locate and select the CICS TG plug-in “update site” file `cicstg_plugin_v1002.zip` on your local hard disk.

The CICS TG plug-in should now appear in the list of Available Software (Figure 7 on page 16).

**Note:** There is a known issue in CICS Explorer V1 where plug-ins to be selected for install might not appear for selection unless the check box “Group items by category” is deselected.
3. Select the CICS TG plug-in for installation and select **Install**.

4. The Install dialogue summarizes the actions about to be taken. Check that CICS TG plug-in details appear as expected. Click **Finish** to complete the installation.

5. When prompted to restart the CICS Explorer, click **Yes**.

When CICS Explorer has re-started, a new CICS TG perspective is available. However, before you can use the CICS TG perspective, a connection to the Gateway daemon must be defined.

**Note:** If you are re-installing the CICS TG plug-in, it might be necessary to reset the perspective. Select **Reset perspective** from the Window menu.
Define a new CICS TG connection

You need to define a connection to your Gateway daemon to use the CICS TG perspective. The CICS TG plug-in connects to a Gateway daemon statistics handler port to retrieve information about that Gateway daemon. The CICS TG plug-in allows multiple Gateway daemons to be connected, each requiring a separate connection definition. Define the new connection using the following steps:

1. Select Preferences from the Window menu. The Preferences dialog window is displayed. Expand the CICS Explorer node in the attribute tree on the left and select Connections (Figure 8).

2. From the Type drop-down menu, select CICS Transaction Gateway and then New.

![Figure 8 Creating a CICS Explorer connection a Gateway daemon](image)

3. The Name field is the label by which CICS Explorer will identify this connection. An obvious candidate is the APPLID of the Gateway daemon, DMCTG01.

4. Enter the DNS host name of the z/OS system in the Host name field.

5. Enter the port number of your Gateway daemon statistics handler in the Port number field.
6. Click **Apply**, then **Connect**. If the Gateway daemon is running and the statistics handler initialized successfully, you should now have established connectivity between CICS Explorer and your Gateway daemon.

**Tour of CICS TG Perspective**

The CICS TG perspective (Figure 9 on page 19) consists of four parts:

- CICS TG Explorer view
- Gateway daemons view
- CICS connections view
- Properties and console views

**Note:** The Credentials field is not available for a CICS Transaction Gateway connection. The CICS TG statistics handler does not require authentication since it is only capable of supplying statistical data. No operational changes can be made to the Gateway daemon through this connection.
The CICS TG Explorer view provides a topological view of those connected Gateway daemons, arranged into Gateway groups by APPLID Qualifier, which is analogous to a VTAM® Network ID. Expanding a Gateway group node reveals the Gateway daemons within that group, and expanding a Gateway daemon node reveals the CICS connections known or available to that Gateway daemon.

The Explorer view controls the scope of information displayed in the Gateway daemons view and CICS connections view. It is possible to select all defined Gateway daemon and CICS connections by clicking within the Explorer view and pressing Ctrl+A.

The Gateway daemons view provides a summary of the connected Gateway daemons that are in the scope of the Explorer view. Many Gateway daemon attributes are available for inspection, and users can customize the view as they prefer.
The user can switch between three predefined sets of related Gateway attributes through the Column layout sub-menu of the Gateway daemons view menu. Choose from Gateway, Requests, and Resources.

The CICS connections view provides a summary of the CICS connections defined or in use by the Gateway daemons that are in the scope of the Explorer view. Many CICS connection attributes are available for inspection, and users can customise the view as they prefer.

The Properties view displays all of the available attributes associated with the entry selected in either the Gateway daemons or CICS connections view. Context-sensitive help, providing a detailed explanation of any particular attribute, can be found by selecting the attribute in the Properties view and pressing the F1 key.

The Console view displays output from operations such as Test Gateway connection.

Experiment with the predefined Gateway daemons view layouts, customizing columns and the layout of the perspective.

Reset the perspective back to the default layout before continuing. Select Reset perspective from the Window menu.

**Verify Gateway connectivity**

This might seem strange, since you are already connected to the Gateway daemon with the Explorer plug-in. However, the CICS TG plug-in utilizes the statistics API port, whereas applications utilize the tcp handler port. The CICS TG plug-in provides a test function that allows you to vary application connectivity to the tcp handler port.

Right-click your Gateway daemon node within the CICS TG Explorer view. Select **Test Gateway connection** from the context menu (Figure 10).

![Testing Gateway daemon connectivity from CICS Explorer](image-url)
If your tcp handler is correctly configured, you should receive a console message similar to the following:

```
Attempting to open connection to Gateway daemon using host LP01 on port 14,201
Success: Opened connection to LP01 on port 14,201
```

**Note:** Test Gateway connection depends on successful DNS resolution of the MVS system name LP01 by the local CICS Explorer machine. You might need to add a mapping to the local TCP/IP hosts file to define an alias. For example, running the CICS Explorer on a Windows machine, add the following line to the file C:\Windows\system32\drivers\etc\hosts:

```
97.32.217.20   LP01
```

Alternatively, you could add the appropriate domain name suffix in the workstation TCPIP configuration.

### Configure connections in CICS TS

This section shows how to create and install the minimal CICS definitions required for an EXCI connection and an IPIC connection. These procedures are completed using the CICS SM perspective:

- Create and install EXCI connection/session pair.
- Create and install IPIC TCPIPService.

Step-by-step details follow.

**Note:** The IPIC connection is required only for high availability, so you could skip this step if you want to fast track the configuration process and are not planning on implementing high availability.

### Create an EXCI connection in CICS TS

The traditional method of communication between a CICS Transaction Gateway daemon for z/OS and CICS Transaction Server is EXCI. This uses a cross memory transport better known to CICS systems programmers as MRO. To create an EXCI connection for CICS Transaction Gateway, you must create two CICS definitions: a connection and a session pair.
From the CICS SM perspective of the CICS Explorer use the following steps to create a new ISC/MRO connection definition:

1. Select the desired CICS region in the CICSpex Explorer view and select **ISC/MRO Connection Definitions** from the Administration menu (Figure 11).

2. The ISC/MRO Connection Definitions panel opens, showing the existing set of connection definitions available for installation on your CICS region. To create a new connection definition, right-click within the ISC/MRO Connection Definitions panel and select **New**.

3. Complete the New ISC/MRO Connection Definitions (Figure 12) by making the following entries and selections:
   - The Region CSD box should be selected and set to your assigned CICS region name (CICSDMnn)
   - Resource Group: CTGDMnn
   - Name: CXnn
– Description: Enter a suitable description for this connection.

**Note:** In the figures that follow, \( NN \) or \( nn \) is replaced by 01 in the worked example.

![New ISC/MRO Connection Definition](image)

*Figure 12  Create a new EXCI connection*

4. The ISC/MRO Connection Definition editor is displayed. Update the following attributes, leaving all other attributes with their default value:
   – Attachsecmethod: IRC
   – Conntype: SPECIFIC
   – Netname: CTGEXC01
   – Protocol: EXCI

5. Save the definition by pressing Ctrl+S. You should encounter the error shown in Figure 13 on page 24:
   
   Update Failed: CONNDEF_INV_DATA.
One of the connection attributes you have not set is marked with an **Update** icon. Select that field and press F1 to see context-sensitive help. This should help you select the appropriate value.

6. Save the definition again (Ctrl+S); there should not be any more errors.

**Create the sessions definition**

Create the sessions definition to be associated with your new EXCI connection definition. The session definition dictates the operational characteristics of the connection. For example, the number of concurrent requests that can be processed through the EXCI connection is managed by the RECEIVECOUNT parameter.

Create a new Sessions definition using the following steps:

1. From the CICS SM view, select CICS region **CICSDM01** in the CICSplex Explorer view (if it is not already selected). Select **Session Definitions** from the Administration menu.

2. The Session Definitions panel opens, showing the existing set of session definitions available for installation on your CICS region. To create a new sessions definition, right-click in the Session Definitions panel and select **New** from the context menu.

3. Complete the New Session Definition dialogue fields as follows:
   - The Region CSD box should be selected and set to CICS region **CICSDM01**.
   - Resource Group: **CTGDM01**
   - Name: **SX01**
   - Connection: **CX01** (This should match the name of the new connection created in the previous procedure.)
   - Description: Enter a suitable description.
4. The Session Definition editor is displayed. Update the attributes with the following values:
   - Ioarealen: 4096
   - Ioarealen 2: 4096
   - Maxingrp: 0
   - Protocol: EXCI
   - Receivecount: 100
   - Receivepfx: E

5. Save the definition by pressing Ctrl+S.

### Install the connection/session pair

Perform the following steps to install the connection/session pair:

1. Open the Resource Group Definitions view by selecting Administration → Resource Group Definitions. Scroll through the list of groups to find your new resource group CTGDM01.

2. Select the resource group. This has the effect of limiting the scope for the ISC/MRO Connection Definitions or Sessions Definitions views. They should now only show the EXCI connection or sessions definition contained within your CSD group.

3. Right-click your resource group within the Resource Group Definitions view and select Install. (Figure 14 on page 26).
4. In the Perform Operation dialog, select the check box under the Target list for CICS region CICSDM01. Review the Execution Queue to ensure you are installing the group you intend to! It should show CTGDM01.

5. If the resource group is installed successfully, the Perform Operation dialog disappears from view. Select the Operations → ISC/MRO Connections menu to open (or re-select) the ISC/MRO Connections panel. Your new connection should be listed with a Service Status of INSERVICE.

Create an IPIC connection in CICS TS

The strategic method of communication between a CICS Transaction Gateway daemon for z/OS and CICS Transaction Server is IP interconnectivity (IPIC). This method uses as a transport TCP/IP sockets, which has the flexibility to operate across LPAR or Sysplex boundaries, but also optimizes to Fast Local Sockets when CICS TS and CICS TG are co-located within the same LPAR, or HiperSockets™ between LPARs.
To create an IPIC connection for CICS Transaction Gateway, you need to create a TCPIP Service definition. No IPCONN definition is required because an IPCONN resource will be autoinstalled by CICS at run time.

Use the CICS SM perspective of the CICS Explorer to create and install the definition.

Follow these steps to create a new TCPIP Service definition:

1. From the CICS SM view, select CICS region **CICSDM01** in the CICSplex Explorer view (if not already selected), and select **Operations → TCP/IP Service Definitions**.

2. The TCP/IP Service Definitions panel opens showing the existing set of TCPIP Service definitions available for installation on your CICS region. To create a new TCP/IP Service definition, right-click within the Session Definitions panel and select **New**.

3. Complete the New TCP/IP Service Definition dialog fields by making the following entries and selections (Figure 15 on page 28):
   - Ensure that Region CSD is selected and CICS region name **CICSDM01** is specified.
   - Resource Group: **CTGDM01**
   - Name: **CTG14101**
   - Description: Enter a suitable description.

Click **Finish**.
4. The TCP/IP Service Definition editor is displayed. Update the following attributes, leaving all other attributes with the default values:
   - Authenticate: NOTAPPLIC
   - Maxdatalen: (Leave blank)
   - Protocol: IPIC
   - Transaction: CISS
   - Urm: DFHISAIP

**Explanation:** CISS is the CICS transaction used during establishment of IPIC connections. DFHISAIP is the CICS autoinstall user program for IPIC connections. Authenticate is set to NOTAPPLIC since that field only applies to HTTP or IIOP protocols. The Maxdatalen attribute is not applicable to the IPIC protocol.

5. Click **Finish**.
The TCPIP Service CTG14101 is now defined within the same CSD Resource Group as the EXCI connection and session pair. Use the following steps to install just the TCPIP Service definition rather than the whole group:

1. The TCP/IP Service Definition panel shows the SM scope CICSPXLX1 and the name of the TCP/IP Service definition currently displayed within the editor panel. Right-click **CTG14101** and select **Install** from the context menu (Figure 16).

![Figure 16 Installing a TCPIP Service definition](image)

2. In the Perform Operation dialog, select the check box under the Target list for CICS region CICSDM01. Review the Execution Queue to ensure you are installing the group you intend to! It should show **CTGDM01**.

3. If the TCPIP Service definition installed successfully, the Perform Operation dialog disappears from view. Select **Operations** → **TCPIP Services** to open (or re-select) the TCPIP Services panel. Your new TCPIP Service should be listed with a Service Status of OPEN.

You should now have one EXCI connection and one IPIC connection defined and installed in your assigned CICS region. The next task is to configure the CICS Transaction Gateway side of these two connections.

### Configure CICS connections in CICS TG

This section shows how to define connections between CICS TG for z/OS and CICS Transaction Server using EXCI and IPIC protocols.

**Note:** The IPIC connection is required only for high availability, so you could skip this step if you want to fast track the configuration process and are not planning on implementing high availability.
These procedures are done using the z/OS perspective:

- Configure CICS TG for EXCI.
- Define an IPIC Server definition.

**Configure CICS TG for EXCI connections**

Perform the following steps to configure CICS TG for EXCI connections:

1. From the z/OS perspective open PDS member `<userid>.CTG800.DEMO(CTGENV)`.

2. Set the EXCI net name for the Gateway daemon:
   
   ```
   DFHJVPIPE=CTGEXC01
   ```

   From the CICS viewpoint, the Gateway daemon address space is an *EXCI client*. As such, the Gateway must identify itself as an MRO end point with a net name. The DFHJVPIPE environment variable is used to define the Gateway daemon EXCI net name. This must match the Netname attribute of the EXCI connection definition you created previously in “Create an EXCI connection in CICS TS” step 4 on page 23.

3. Define the CICS APPLID for the EXCI connection to the Gateway daemon:
   
   ```
   DFHJVSYSTEM_00=CICSDM01
   ```

   **Note:** This value must be the CICS APPLID, which might not be the same as the CICS job name.

   **Explanation:** There is no need to define individual EXCI connections to the Gateway daemon. If a request for a given CICS APPLID is received by the Gateway daemon, then that name is assumed to be a CICS APPLID available through EXCI if it does not match an IPIC SERVER definition.

   However, it is possible to define a set of EXCI connections that will contribute to an ECI List Systems response. Such a call might be used by an application to allow the target CICS server to be determined by the application logic or user at run time.

4. Save the changes to CTGENV by pressing Ctrl+S.
Configure CICS TG with an IPIC connection

Perform the following steps to configure CICS TG with an IPIC connection:

1. Open the UNIX System Services file `/<userid>/ctg800demo/ctg800.ini`.

2. Add an IPIC SERVER definition for the IPIC connection to CICS region CICSDM01.

   Example 1 shows an IPIC connection definition with the CICS server name BAKDM01, which can support up to 100 concurrent requests.

```
Example 1  IPIC Connection

SECTION IPICSERVER = BAKDM01
  DESCRIPTION=Backup IPIC connection to CICSDM01
  SRVIDLETIMEOUT=0
  HOSTNAME=lp01.redbooks.ibm.com
  PORT=14101
  CONNECTTIMEOUT=60
  TCPKEEPALIVE=Y
  SENDSESSIONS=100
ENDSECTION
```

**Explanation:** Unlike EXCI connections, IPIC connections must be predefined to the Gateway daemon. An IPIC connection is defined in the Gateway daemon configuration file by a SECTION IPIC SERVER entry, and typically follows the GATEWAY section of the configuration file.

When an ECI request is received by the Gateway daemon for CICS server BAKDM01, the request is matched to this connection definition and the request is sent over this IPIC connection.

Save the changes to `ctg800.ini` by pressing Ctrl+S.

3. Restart the Gateway daemon using the CTGSTOP and CTGSTART jobs.

---

Set up and run workload

This section shows how to run the sample workload application CICS Support Pac CH50, otherwise known as CTGPING.
CH50 or CTGPING overview

To test your Gateway daemon and CICS connection definition use CICS Support Pac CH50, named CTGPING.

CTGPING is a Java ECI application that makes single shot (synconreturn) requests to CICS for the COMMAREA program PONG. It produces output that is similar to the TCPIP ping utility and is very useful for verifying basic configuration as well as network latency.

CTGPING can also be used to flow channel requests by specifying a data length of greater than 32K. In this case, it will call the channel program PONGLONG.

**Note:** As highlighted in “Assumptions” on page 4, at least the CICS program PONG from Support Pac CH50 is required.

Customize the CTGPING JCL

Use the following steps to customize the CTGPING JCL:

1. Switch to the z/OS perspective in CICS Explorer and open the PDS member <userid>.CTG800.DEMO(CTGPING).
   
   This job uses the CTGBATCH utility to execute UNIX System Services shell script /u/<userid>/ctg800demo/ch50/ctgping.

2. Set the JCL variables SCRIPTIN, GATEWAY, and PORT using details recorded in the first section:

   ```jcl
   // SET SCRIPTIN='/u/<userid>/ctg800demo/ch50/ctgping'
   // SET GATEWAY='tcp://<dns hostname address>'
   // SET PORT=<tcp port>
   ```

   **Note:** <tcp port> refers to the Gateway daemon tcp handler port number.

3. Update the CLASSPATH environment variable in the STDENV DD card data with the correct path for the ctgping.jar file:

   ```
   CLASSPATH=/u/<userid>/ctg800demo/ch50/ctgping.jar:
   /usr/lpp/cicstg/v8r0/classes/ctgclient.jar
   ```

Run CTGPING

Use the steps in this section to run and review CTGPING.
1. Submit your CTGPING job. This JCL includes CTGBATCH PARM data '-i=5', instructing CTGPING to run 5 iterations.

Review the output after your CTGPING job has run; it should be similar to the STDOUT shown in Example 2.

**Example 2  Output of CTGPING**

```
Pinging gateway url: tcp://lp01.redbooks.ibm.com:14201
  Sending Gateway daemon request
  Reply from tcp://lp01.redbooks.ibm.com:14201 open=21ms, request=12,5,5,8,6ms, close=2ms
  Reply from tcp://lp01.redbooks.ibm.com:14201 open=11ms, request=5,6,4,5,5ms, close=2ms
  Reply from tcp://lp01.redbooks.ibm.com:14201 open=5ms, request=5,4,6,10,7ms, close=2ms
  Reply from tcp://lp01.redbooks.ibm.com:14201 open=8ms, request=6,6,4,5,4ms, close=2ms
  Reply from tcp://lp01.redbooks.ibm.com:14201 open=4ms, request=4,4,4,9,4ms, close=2ms
  ----ctgping statistics----
  Opens issued=5, min=4ms, max=21ms, avg=9.8ms, errors=0
  Requests issued=25, min=4ms, max=12ms, avg=5.7ms, errors=0
  Closes issued=5, min=2ms, max=2ms, avg=2ms, errors=0
```

**Explanation:** By default, each CTGPING iteration opens a connection to the Gateway daemon, issues five ECI requests, and then closes the connection to the Gateway daemon.

2. Switch to the CICS TG perspective and check the Gateway daemon and CICS connection request/response statistics. Hopefully they are not all zero!

**Explanation:** You might be wondering which CICS server the PING requests have been sent to. The CTGPING parameter -c, when not qualified with a value, allows the Gateway daemon to select the target CICS server dynamically.

This deferral of the CICS server choice to the Gateway daemon is known as Dynamic Server Selection (DSS). In this case, we are using the most simple form of DSS, that is, applying the Gateway daemon DEFAULT SERVER definition. (This was shown in Figure 3 on page 11).

3. Right-click your Gateway daemon node within the CICS TG Explorer view and select **Refresh status** from the context menu. You should now have an entry in the CICS connections view for the EXCI connection, and that the Requests, Response, Sent and Received fields now contain interesting information (Figure 17 on page 34).
4. Switch back to the z/OS perspective and modify the CTGPING job PINGPARMS variable to specify your IPIC server connection as the target CICS server:

```
// SET PINGOPTS='-i=5 -d=1 -c=BAKDM01
```

Submit the CTGPING job.

5. Hopefully CTGPING ran successfully again! Check the output.

6. Switch back to the CICS TG perspective, and click the refresh icon on the CICS connections view.

The details for the IPIC connection are now visible and the Requests, Response, Sent and Received fields now contain interesting information.

You have now demonstrated that your Gateway daemon and CICS connections are operational.

**Note:** You can skip ahead to “Set up Request Monitoring for CICS TG” on page 40 if you are not implementing high availability.

## Setting up High Availability for CICS TG

This section shows how to configure CICS TG to use Dynamic Server Selection, providing High Availability for connectivity between CICS TG and CICS TS.

High Availability of the connection between the application and the Gateway daemon can be achieved within an LPAR using TCPIP port sharing, or at the Sysplex level using Sysplex Distributor. These are technologies independent of CICS TG and as such are not covered here.

## CICS Request Exit Overview

In the previous section, we used a form of Dynamic Server Selection where the choice of target CICS server for an ECI request was delegated by the application to the Gateway daemon DEFAULT SERVER definition. CICS Transaction Gateway daemon for z/OS has a further DSS feature called the CICS Request Exit (Figure 18).
The CICS Request Exit allows a user-written Java class to decide the target CICS server at the time the request is processed by the Gateway daemon. Depending on the logic within the CICS request exit, it might simply use the CICS server specified by the application, override the application, or reject the request.

Furthermore, the CICS Request Exit can be used to retry requests which encounter errors when flowed to a given CICS server. The combination of Dynamic Server Selection and the ability to retry failing requests form a large part of the CICS Transaction Gateway High Availability story.

You could attempt to write your own CICS Request Exit, but in this case we opted to use Support Pac CA1T.

CA1T provides a pre-built CICS Request Exit that can be configured through simple text files, much like the CICS Transaction Gateway itself. There are actually two exit classes provided by CA1T; one implements a round robin algorithm and the other implements a failover algorithm.

**Configure CICS TG to use Support Pac CA1T**

A Gateway daemon is configured to use a CICS Request Exit class with a single entry to the GATEWAY section of the configuration file. The steps described here use the CA1T failover exit and configure it to use our EXCI connection as the primary CICS server, with the IPIC connection as a backup.
1. Switch to the z/OS perspective and open your Gateway daemon configuration file (ctg800.ini). Add the following line to the GATEWAY section:

   cicsrequestexit=com.ibm.ctg.samples.ha.FailOverExit

   Save your changes.

2. Open your STDENV file (CTGENV) and update the CLASSPATH environment variable to include the CA1T Java Archive (JAR). The UNIX System Services path for the CA1T materials is shown in Table 4 on page 7.

   CLASSPATH=/usr/lpp/cicstg/v8r0/classes/ctgsamples.jar:
   /u/<userid>/ctg800demo/ca1t/ctghav2.jar

3. Add an entry for environment variable CTG_HACONFIG. This environment variable is recognized only by CA1T, and defines the UNIX System Services path for the CA1T configuration file.

   The CA1T configuration file is located alongside the previously mentioned JAR file.

   CTG_HACONFIG=/u/<userid>/ctg800demo/ca1t/policy1.ini

4. Add an entry for environment variable CTG_HAINTERVAL. This environment variable is recognized only by CA1T, and defines the time (in seconds) that a failing CICS server connection is suspended from the selection list following a request failure. For the purpose of demonstration, 15 seconds is a reasonable value.

   CTG_HAINTERVAL=15

5. Save your changes.

**Configure Support Pac CA1T**

Follow the steps in this section to configure Support Pac CA1T.

1. Locate and open the UNIX System Services file policy1.ini. This is the CA1T configuration file; it is located alongside the previously mentioned JAR file. The CA1T configuration (or policy) file defines a set of rules by which the CICS Request Exit class will choose a target CICS server.

2. Edit policy1.ini to specify your EXCI connection followed by your IPIC connection on the generic server rule:

   *=<CICS server 1>,<CICS server 2>

   **Explanation:** For the purposes of demonstration, we only make use of the CA1T generic server rule (denoted by an asterisk *). This means that regardless of the CICS server specified by the application, the CA1T exit code will chose from the list of CICS servers specified against the generic server rule. The CICS server chosen from the list depends on the CA1T exit
in use, that is, failover or round robin. Further documentation on CA1T rule definitions is available with the Support Pac. Support Pac CA1T and its documentation can be found at:


3. Save your changes.

4. Restart the Gateway daemon. Successful configuration of the CICS Request Exit can be verified in the Gateway daemon job log (TSO SDSF), by the presence of the following messages:

From CA1T itself:

CA1T Initialising CICS request exit FailOverExit V2.0 retry interval(15s) logging level(1)
CA1T Reading HFS configuration file
/u/suci101/ctg800demo/ca1t/policy1.ini
CA1T Server rule * using actual servers [CICSDM01, BAKDM01]
CA1T Set maximum request count (2)

From the Gateway daemon:

CTG8447I CICS Request exit com.ibm.ctg.samples.ha.FailOverExit installed successfully

Test the HA configuration

Use CTGPING to test the High Availability topology. Increase the number of requests such that CTGPING provides a longer, steady workload, allowing you time to disrupt the topology and observe failover in action. Then restore the topology to observe that the workload has been restored to its primary path. The detailed steps to do this are:

1. Open the CTGPING JCL job and update the PINGOPTS variable to specify 75 iterations (-i switch).

   // SET PINGOPTS=' -i=75 -d=1 -c'

   **Explanation:** By default, CTGPING issues 5 ECI requests per iteration, and we have set the delay between requests to 1 second (-d=1). Setting -i=75 causes CTGPING to steadily generate requests over 5 minutes, which should be enough time to demonstrate our High Availability topology.

2. Switch to the CICS SM view, and open the ISC/MRO Connections view by selecting **Operations → ISC/MRO Connections**.

3. Verify that your CICS region is selected within the CICSplex Explorer view and that your EXCI connection CX01 is visible in the ISC/MRO Connections view.

4. Switch to the z/OS perspective and submit the CTGPING JCL job.
5. Switch to the CICS TG perspective. Right-click the Gateway daemon in the CICS TG Explorer view and select Refresh status. Clicking the refresh icon on the CICS connections view (Figure 19) should reveal that the Gateway is sending work to your CICS region using the EXCI connection.

Figure 19  EXCI connection processing the initial workload

6. Switch to the CICS SM perspective and set your EXCI connection Out of Service.

Right-click the CX01 entry and select Out of Service from the context menu. Review the Perform OUTSERVICE operation dialog (Figure 20) and click OK.

Figure 20  Setting the EXCI connection Out of Service

The EXCI connection should now show a Service Status of OUTSERVICE. Requests sent from the Gateway daemon to this CICS connection will now fail.
7. Switch to the CICS TG perspective. Refreshing the CICS connections view should reveal that the workload is now being completed by the IPIC connection (Figure 21).

The CA1T failover algorithm has detected that the EXCI connection is unavailable, and automatically switched over to the backup connection, as defined by the policy file.

![Figure 21: Automatic fail-over of the workload to the IPIC connection](image)

8. Switch to the CICS SM perspective and set your EXCI connection CX01 back to INSERVICE.

9. Switch to the CICS TG perspective. Refreshing the CICS connections view should reveal that the EXCI connection has automatically resumed responsibility for handling the workload (Figure 22). You might observe a slight delay in the failback to the primary CICS server due to the CTG_HAINTERVAL value of 15 seconds.

![Figure 22: Automatic fail-back of the workload to the EXCI connection](image)

This section has shown the high availability failover and fail-back capability of CA1T in action.
Switch from failover to round robin

Without changing the CA1T policy file, it is possible to switch from the failover to the round robin algorithm. To do so follow these steps:

1. Open your Gateway daemon configuration file (ctg800.ini), and change the cicsrequestexit keyword in the GATEWAY section to:
   
   cicsrequestexit=com.ibm.ctg.samples.ha.RoundRobinExit

2. Restart your Gateway daemon and run the CTGPING workload again. Using the CICS TG perspective, verify that the request counts are increasing on both CICS server entries.

With the round robin algorithm, work is shared between the two connections. If the EXCI connection is set out of service, then the IPIC connection will process all requests. When the EXCI connection is restored, work will again be shared between the two connections.

The Support Pac has more capabilities than we have been able to demonstrate here. For instance, the systems programmer can interact with the CICS Request exit via the CREXIT modify command in TSO. There are further rule types that can provide a comprehensive policy for dynamically routing requests, or rejecting requests. For further details, refer to the support document.

Set up Request Monitoring for CICS TG

This section shows how to configure and use the Request Monitoring exit for the Gateway daemon using the Support Pac CH51. The steps are summarized as follows:

- Configure CICS TG to use Request Monitoring support pac CH51.
- Run the application workload with monitoring.
- Review monitoring output.

Configure CICS TG to use Support Pac CH51

A Gateway daemon is configured to use a Request Monitoring Exit class with a single entry to the GATEWAY section of the configuration file. To use the CH51 request monitoring exit and configure it to write details to an HFS file follow these steps:

1. Switch to the z/OS perspective and open your Gateway daemon configuration file (ctg800.ini). Add the following line to the GATEWAY section:
requestexits=com.ibm.ctg.samples.requestexit.SimpleMonitor
Save your changes.

2. Open your STDENV file (CTGENV) and update the CLASSPATH environment variable to include the CH51 Java Archive (JAR). The UNIX System Services path for the CH51 materials is shown in the first section.

CLASSPATH=/usr/lpp/cicstg/v8r0/classes/ctgsamples.jar:
/u/<userid>/ctg800demo/ca1t/ctghav2.jar:
/u/<userid>/ctg800demo/ch51/ctgmonitor.jar

3. Update the entry for environment variable CTGSTART_OPTS to export the two Java properties:

   com.ibm.ctg.samples.requestexit.out
   This sets the output destination for CH51.

   com.ibm.ctg.samples.requestexit.ch51
   This sets the level detail produced by CH51

Java properties are defined for the Gateway daemon via the CTG_START_OPTS using the prefix -j-D. To configure CH51 to use logging level 2 and output to a UNIX System Services file, use the following pattern:

CTGSTART_OPTS=-j-Dcom.ibm.ctg.samples.requestexit.out=
/u/<userid>/ctg800demo/ch51.log 
-j-Dcom.ibm.ctg.samples.requestexit.ch51=2

**Note:** The STDENV data set has a LRECL of 80 characters. Long STDENV entries can be continued using the back slash character (\). Be sure to leave a space character between property entries of the form -j-Dproperty=value'.

4. Save your changes.

**Run workload with monitoring**

Use these steps to run a workload with monitoring:

1. Restart the Gateway daemon. Check for the following message in the Gateway daemon job log (SDSF) to confirm that CH51 has initialized correctly:

   CTG8402I Request Monitoring Exit
   com.ibm.ctg.samples.requestexit.SimpleMonitor is enabled
   CTG8469I Request monitoring is active

   The CH51 output file is also created during initialization, and contains a header message, for example:
Using CICS Transaction Gateway, High Availability, and the CICS Explorer

Review monitoring output

Our CTGPING job starts an ECI application which loops, submitting requests for the CICS COMMAREA program PONG. The requests and responses are captured in the CH51 output:

(Request)
02/02/11 16:11:08:231 CH51 -> Rqst(2) Flow(EciSynconreturn) Pgm(PONG) ClientIP(9.20.35.253) commarea(140)bytes SocketData(263)bytes

(Response)
02/02/11 16:11:08:339 CH51 <- Rqst(2) Flow(EciSynconreturn) Srv(null/IY24CTGU) Pgm(PONG) ClientIP(9.20.35.253) commarea(140)bytes SocketData(198)bytes RespTime(108)ms CICScall(78)ms ctg_rc(-7) Abend(AEIO)

In this example, the target CICS region did not have the PONG program installed, and so the request failed with ABEND AEIO. Of particular interest here are details of the client application IP address, time spent to complete the CICS call, and request data size.

As the name suggests, the request monitoring exits provide a much greater level of detail at the request level, compared to the Gateway daemon statistics which provide an overall status.

Use the following steps to set up a rolling monitor window:

1. Log on to UNIX System Services with a VT100 terminal emulator such as putty. Use the UNIX tail command to set up a rolling view of the CH51 log output.

2. Start the putty terminal emulator. Connect to the z/OS machine using SSH and log on the UNIX System Services using your TSO user ID.

3. Use tail -f to start a rolling view of the CH51 output, for example:
   
tail -f /u/<userid>/ctg800demo/ch51.log
4. Start the CTGPING workload again, and observe the CH51 output echoed to the putty terminal.

The Support Pac CH51 has more capabilities than we have been able to demonstrate here. For instance, the systems programmer can interact with the Request Monitor exit via the RMEXIT modify command in TSO. For further details, refer to the Support Pac CH51 documentation at:


The author who wrote this paper

This paper was produced by a technical specialist working at the International Technical Support Organization, Hursley Center.

Rob Jones is a Software Engineer at IBM Hursley Software Laboratory in the United Kingdom and is responsible for the strategy and technical planning of the CICS Transaction Gateway product suite. He has worked on the CICS Transaction Gateway products since 2002, specializing on the z/OS platform in development. Rob also produced the initial CICS TG plug-in for CICS Explorer. He holds a degree in Computer Science and Mathematics from Durham University.

Now you can become a published author, too!

Here’s an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html
Stay connected to IBM Redbooks

- Find us on Facebook:
  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:
  http://twitter.com/ibmredbooks
- Look for us on LinkedIn:
  http://www.linkedin.com/groups?home=&gid=2130806
- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
- Stay current on recent Redbooks publications with RSS Feeds:
  http://www.redbooks.ibm.com/rss.html
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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates 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. 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 IBM's application programming interfaces.

© Copyright International Business Machines Corporation 2011. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.