Connecting to CICS
From WebSphere for z/OS

This document can be found on the web at:
www.ibm.com/support/techdocs
Search for document number WP100607 under the category of "White Papers"

Version Date: February 26, 2007

John Cowel
IBM Kansas City
816-556-6244
jrcowel@us.ibm.com
Many thanks to the following people:

- **Don Bagwell** for providing the template for this document.

- **John Hutchinson** for reviewing the document and getting it published on TechDocs.

- **Mike Cox** for reviewing the document for accuracy and pointing out additional information that needed to be included.

- **Phil Wakelin** for the Security Management Decision Tree and EXCI Pipe Considerations information.
Overview and Introduction

The purpose of this document is to present the prerequisites and steps required to facilitate connection from WebSphere V6 for z/OS to CICS using CICS Transaction Gateway V6.

The WebSphere configuration used in creation of this document looks like this:

Legend:
A - Deployment Manager Cell - j6cell
B - Deployment Manager Node - j6dmnode
C - Application Server node - j6noded
D - Java Native Interface (JNI) module libctgjni.so, which makes the EXCI call to CICS
E - CICS Transaction Server 1.3 Region - CTS130J

Note: The steps outlined in this document include the necessary definitions for connecting to an adjacent CICS region, i.e. a CICS region on the same image as WebSphere. By connecting to an adjacent CICS region, you gain full 2-Phase Commit as well as propagation of security context. This does not mean that the CICS application must be present in this adjacent region. The CICS programs that make up the application may be defined as “Remote” and actually reside in a region on a different image and would be accessed via some form of MRO - ISC or XCF. Ideally, your application will contain no run-time affinities and you will have multiple Application Owning Regions (AORs) that are pure clones of each other. In this case, every application program will be replicated and therefore be available in any AOR. Your workload
distribution scheme would be a simple one of balancing requests based on AOR availability and capacity. The recommendation in this case would be the use of CICSPlex Systems Manager’s (CPSM's) dynamic routing, since it is able to maintain awareness of which regions are active and how busy each of them is at any given moment in time.
Prerequisites and CICS Settings

Prerequisites:

- z/OS(.e) V1.4 or later, and the following APARs
  - LE   - PQ82905
  - RRS  - OA01707
- Java 2 Technology Edition Version 1.2 or higher (SDK 1.4.2)
- WebSphere Application Server for z/OS V6.0.1 or later
- CICS Transaction Gateway V6.0 or later
- CICS Transaction Server V1.3 or later with APAR PQ25809 for CICS TS V1.3

Connecting to CICS from WebSphere does not require the CTG Gateway address space to be running on z/OS if the CICS region you are connecting to resides on the same image as WebSphere.

CICS Region Settings

The CICS region to which you are going to connect must have the following System Initialization Table (SIT) parameters set:

- IRCSTRT=yes
- ISC=yes
- RRMS=yes

In addition to these settings you also require a Generic EXCI connection and session definitions in your region. Or, if you choose or require, a Specific EXCI connection and session definition. A sample of each of these types of connections is included in RDO Group DFH$EXCI, which ships with CICS. Included in this document are copies of the definitions of those EXCI connections.

Generic Connection definition:

CEDA  View Connection( EXCG )
Connection   : EXCG
Group        : DFH$EXCI
DEscription  : Sample EXCI Generic connection

CONNECTION IDENTIFIERS
Netname      :
INDsys       :

REMOTE ATTRIBUTES
REMOTESYstem :
REMOTEName   :
REMOTEYSNet :

CONNECTION PROPERTIES
Accessmethod : IRc Vtam | IRc | INdirect | Xm
Protocol     : Exci Appc | Lu61 | Exci
Conntype     : Generic Specific
Singlesess   : No No | Yes
Datastream   : User User | 3270 | SCs | STrfield | Lms
RECordefomat : U U | Vb
Generic Session Definition:

CEDA View Sessions( EXCG )
Sessions : EXCG
Group : DFH$EXCI
Description : Sample EXCI Generic sessions definition

SESSION IDENTIFIERS
Connection : EXCG
SESSName :
NETnameq :
Modename :

SESSION PROPERTIES
Protocol : Exci Appc | Lu61 | Exci
Maximum : 000 , 000 0-999
RECEIVEPfx : RG
RECEIVECount : 004 1-999
SENDPfx :
SENDCount : 1-999
SENDSize : 04096 1-30720
RECEIVESize : 04096 1-30720
SESSPriority : 000 0-255
Transaction :

OPERATOR DEFAULTS
OPERId :
OPERPriority : 000 0-255
OPERRsl : 0 0-24
OPERSecurity : 1 1-64

PRESET SECURITY
USERId :

OPERATIONAL PROPERTIES
Autoconnect : No No | Yes | All
INService : Yes No | Yes
Buildchain : Yes Yes | No
USERArea1en : 000 0-255
IOArea1en : 04096 , 04096 0-32767
RELreq : No No | Yes
DIScreq : No No | Yes
NEPclass : 000 0-255

RECOVERY
RECOVOpt1on : Sysdefault Sysdefault | Clearconv | Releasesess
| Uncondrel | None
RECOVNNotify : None None | Message | Transaction
Specific Connection definition:

CEDA  View Connection( EXCS )
Connection     : EXCS
Group          : DFH$EXCI
DEscription    : Sample EXCI Specific connection

CONNECTION IDENTIFIERS
Netname        : BATCHCLI
INDsys         :
REMOTE ATTRIBUTES
REMOTEYSystem : 
REMOTEName     :
REMOTEYSysNet : 

CONNECTION PROPERTIES
Accessmethod   : IRc  Vtam | IRc | INdirect | Xm
Protocol       : Exci  Appc | Lu61 | Exci
Conntype       : Specific  Generic | Specific
SInglesess     : No  No | Yes
DAtastream     : User  User | 3270 | SCs | STrfield | Lms
RECORDformat   : U  U | Vb
QueueLimit     : No  No | 0-9999
MaxQtime       : No  No | 0-9999

OPERATIONAL PROPERTIES
Autoconnect    : No  No | Yes | All
INService      : Yes  Yes | No

SECURITY
SECuritiname   :
ATtachsec      : Identify  Local | Identify | Verify | Persistent | Mixidpe
BINDPassword    :
BINDSecurity    : No  No | Yes
Usedfileuser    : No  No | Yes

RECOVERY
PSRecovery     :
Xlnaction      : Keep  Keep | Force

Specific Session Definition:

CEDA  View Sessions( EXCS )
Sessions       : EXCS
Group          : DFH$EXCI
DEscription    : Sample EXCI Specific sessions definition

SESSION IDENTIFIERS
Connection     : EXCS
SESSName       :
NETnameq       :

SESSION PROPERTIES
Protocol       : Exci  Appc | Lu61 | Exci
MAXimum        : 000 , 000  0-999
RECEIVEPfx     : RS
RECEIVECount   : 004  1-999
SENDPfx        :
SENDCount      : 1-999
SENDSize       : 04096  1-30720
RECEIVESize    : 04096  1-30720
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SESSPriority</td>
<td>000</td>
<td>0-255</td>
</tr>
<tr>
<td>Transaction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPERATOR DEFAULTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPERId</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPERPriority</td>
<td>000</td>
<td>0-255</td>
</tr>
<tr>
<td>OPERRsl</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>OPERSecurity</td>
<td>1</td>
<td>1-64</td>
</tr>
<tr>
<td>PRESET SECURITY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USERId</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPERATIONAL PROPERTIES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Autoconnect</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>INservice</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Buildchain</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>USERArealen</td>
<td>000</td>
<td>0-255</td>
</tr>
<tr>
<td>IOarealen</td>
<td>04096 , 04096</td>
<td>0-32767</td>
</tr>
<tr>
<td>RELreq</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DIScreq</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>NEPclass</td>
<td>000</td>
<td>0-255</td>
</tr>
<tr>
<td>RECOVERY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVOption</td>
<td>Sysdefault</td>
<td>Sysdefault</td>
</tr>
<tr>
<td>RECOVNotify</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: This sample Session Definition has RECEIVECount set to 4, which is the default. This means that only 4 pipes can be used by the IRC connection to CICS. Since most users have at least 6 servant threads, some as many as 40, The RECEIVECount should be set to at least the number of servant threads. Based on customer experience, it is recommended to set RECEIVECount to at least 50% higher than the maximum number of servant threads (999 is the maximum value for RECEIVECount).
The remainder of this document covers installing a Resource Adapter (RAR) file and defining a J2C Connection Factory to facilitate connection to an adjacent CICS region from WebSphere for z/OS Version 6.

The steps to connect to CICS from WebSphere are as follows:

Step1:  (Optional) Update the PROCs for servant address spaces to include the CICS library.
Step2:  Install cicseci.rar file.
Step3:  Define J2C Connection Factory.
Step4:  Define Custom Properties.
Step5:  (Optional) Define DFHJVPIPE environment variable to use Specific EXCI Pipe
Step6:  (Optional) Define J2C Authentication Alias(es)

Also included in this document are discussions about EXCI Pipe considerations and limitations, Security Management/Context, and pointers to additional/supporting documentation.
Step 1: (Optional) Update the server’s PROC

Installations have the option of placing the CICS libraries in the linklist or //STEPLIB DD concatenation for the WebSphere address spaces which will connect to CICS. In some installations a combination of techniques are used. If your installation does not have CICS library SDFHEXCI in the linklist, then you must update the //STEPLIB DD concatenation for the servant address space with the missing library.

On this system, the servant address space uses procedure J6ASRD which includes member J6ASRDZ. Member J6ASRDZ contains the //STEPLIB DD concatenation. Here is the update, in the gray area, which was made to include the SDFHEXCI library.

```c
//*
//* Output DDs
//*
//CEEDUMP   DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSOUT    DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSPRINT  DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//*
//*Steplib Setup
//*
//STEPLIB  DD DISP=SHR,DSN=WAS600.WAS.SBBOLD2
//           DD DISP=SHR,DSN=WAS600.WAS.SBBLOAD
//                   DD DISP=SHR,DSN=CTS130.CICS.SDFHEXCI
//*
```
Step 2  Install cicseci.rar file

To provide the ability to connect to CICS you need to install the resource adapter supplied by CICS Transaction Gateway into WebSphere.

From the admin console, click on Resources -> Resource Adapters. Select the Application Server Node, in this case “j6noded”, and click “Apply”. Next, click “Install RAR” as shown in the following picture:

Note: The RAR file MUST be installed at the Node level to begin with. If you wish to define it at a specific server level you may do so later, but this document does not provide details of performing that function.
Select “Server path:”, fill in the path to the cicseci.rar file. The full path and filename on this system is “/usr/lpp/cicstg/ctgv60/classes/cicseci.rar”

Click “Next” and you will see the following:
On the resulting screen enter the location of the libctgjni.so file in the “Native path” field:

/usr/lpp/cicstg/ctgv60/bin

Cick “OK”.

At this point you should “Save” the configuration and recycle the server.
Step 3: Define J2C Connection Factory

From the admin console, click on Resources -> Resource Adapters, then click on ECIResourceAdapter as shown.
Click on **J2C Connection Factories**.
From the subsequent screen, click the “New” button. You will see a panel which looks like the following:

![WebSphere Administrative Console - Microsoft Internet Explorer](image)

Fill in the “Name” and “JNDI name” fields. In this example, “Name” is “Local CTS130J” and “JNDI name” is “eis/local_cts130j”. You may optionally specify a Component-Managed Authentication Alias and/or a Container-Managed Authentication Alias.

**Note:** See “Security Management/Context Considerations” later in this document to assist your decision making process concerning “Authentication Alias” definitions and settings.

Click the “OK” button.
Step 4: Define Custom Properties

From the resulting panel at the end of Step 4, click on “Local CTS130J”, then click on “Custom Properties” on the right side of the panel.

The following variables should be set as shown:

- **ConnectionURL** Local:
- **PortNumber** Local: 0 (zero)
- **ServerName** “applid” (VTAM Applid of CICS region)

To make these changes, click on each parameter, fill in the value for the parameter and click on the “OK” button. The results should look like the following:

![WebSphere Administrative Console](image)

Click “Save” to save the configuration. Once the configuration has been saved, recycle the server to make sure the changes are picked up.
Once the server is reinitialized you are ready to deploy an application that makes use of CICS Transaction Gateway to connect to CICS.
Step 5: (Optional) Define DFHJVPIPE environment variable to use Specific EXCI Pipe

The process outlined in this document makes use of a Generic EXCI connection in CICS. The default is to use a Generic EXCI connection. If you need or choose to use a specific EXCI connection to connect to CICS from WebSphere, you must define the DFHJVPIPE environment variable to your servant. The “value” of the DFHJVPIPE variable must match the “NETNAME” parameter of an installed EXCI CONNECTION definition in CICS.

**Note:** If you wish to connect to multiple CICS regions from one servant, the "NETNAME" of the EXCI connection definition must be the same in each CICS region because you can only define one DFHJVPIPE variable per servant.

To define the DFHJVPIPE environment variable, from the admin console click on Environment -> WebSphere Variables. Set the scope to the specific servant and click on the “New” button. Fill in the spaces using the following as an example:
**EXCI Pipe Considerations**

The CICS J2EE Connectors, when run in conjunction with WebSphere on OS/390 or z/OS, create a cache of EXCI pipes instantiated inside the WebSphere Servant address space. This cache is created and controlled by the CICS Transaction Gateway. In addition to the EXCI pipe cache, WebSphere controls of pool of connections obtained from the J2C Connection Factory, the attributes of which are set during deployment of the CICS Resource Adapter. The most critical attribute, or property, of the CICS ECI connection factory is the ConnectionURL ("Local:" is the only one which provides RRS global transaction support (i.e. 2-phase commit function) and the ServerName (the CICS region's VTAM APPLID to which the EXCI pipe connects). You can also control the maximum number of connections that can be created by the WebSphere Pool Manager by setting a value in the 'Maximum Connections' field in the connection factory. This field should be set to 0 when using a local ConnectionURL to allow unlimited connections to be created.

If you choose to create more than one J2C Connection Factory in a given WebSphere Node, and if this Connection Factory specifies a different CICS APPLID, you will need to make sure that the aggregate number of threads multiplied by the number of J2C connections does not exceed the maximum of 100 EXCI pipes per servant address space.

Note that APAR PQ92943 for CICS TS v2 raised the 100 pipe limit to 250 using the new SYS1.PARMLIB LOGONLIM parameter.

Example:

Suppose that you have a server with a workload profile of LONGWAIT. There are application(s) in this server which access three different CICS address spaces via three defined J2C Connection Factories. Since the EXCI pipe is associated with a thread for the life of the thread, you are now in the situation where you potentially need 120 pipes (3 x 40)... but the limit per address space is 100 PIPES. This means that application requests will randomly fail. Once all 100 PIPES are in use... there will be some number of threads that cannot allocate a pipe for a connection and hence it will fail.

Note if using WebSphere Application Server v6 and the CICS Transaction Gateway v6.0 it is recommended that the EXCI cache is limited to one pipe per thread (of the servant region) to avoid pipes being allocated beyond the CICS EXCI pipe limit. This behavior is set by adding the environment variable CTG_PIPE_REUSE=ONE to the WebSphere runtime, and will prevent WebSphere allocating more pipes than there are threads in the servant region.
Step 6: (Optional) Define J2C Authentication Alias(es)

To define J2C Authentication Alias(es), from the admin console click on Security -> Global Security -> JAAS Configuration -> J2C Authentication Data as shown here:
Click on the “New” button and you will see the following panel where you can specify the “alias” name with the appropriate userid and password:
Specify an alias name, a userid and a password that can be passed to CICS (or any other J2C resource).

Then press the “OK” button.

Note: Add as many aliases as you require for access to CICS or other J2C resources.
Security Management/Context Considerations

The following diagram shows the various possibilities of what security principal will be used in CICS for the call being made:

![WAS z/OS Security Management Decision Tree](diagram)
There are several settings in your environment that have an affect on what security principal is sent to CICS.

**Global Security “OFF”:**

Resource Authorization setting (set via AAT or WSAD-IE, default “Container”)

- Application
  - Connection Spec in use on getConnection() method?
    - Yes
      - Userid specified on Connection Spec
    - No
      - Component Managed Authentication Alias specified?
        - Yes
          - Userid from Authentication Alias
        - No
          - Connection Factory Custom Properties (userid/password)?
            - Yes
              - Connection Factory Custom Properties  **
            - No
              - Servant userid

- Container
  - Container Managed Authentication Alias specified?
    - Yes
      - Userid from Authentication Alias
    - No
      - Servant userid

** - Requires environment variables AUTH_USERID_PASSWORD=YES and JAVA_PROPAGATE=NO
Global Security “ON”:

Resource Authorization setting (set via AAT or WSAD-IE, default “Container”)

- Application
  - Connection Spec in use on getConnection() method?
    - Yes
      - Userid specified on Connection Spec
    - No
      - Component Managed Authentication Alias specified?
        - Yes
          - Userid from Authentication Alias
        - No
          - Connection Factory Custom Properties (userid/password)?
            - Yes
              - Connection Factory Custom Properties **
            - No
              - WebSphere unauthenticated userid

- Container
  - Container Managed Authentication Alias specified?
    - Yes
      - Userid from Authentication Alias
    - No
      - RunAs security
        - Caller
          - If deployment descriptors are present to secure application, e.g. Basic-Auth:
            - Challenged for userid
        - Server
          - Unauthenticated userid (If neither Caller nor Role are present)
        - Role
          - Specified EJB Role

** - Requires environment variables AUTH_USERID_PASSWORD=YES and JAVA_PROPAGATE=NO
Additional/Supporting documentation

WebSphere Library


WebSphere V6 for z/OS InfoCenter

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp

CICS Transaction Server:

http://www.software.ibm.com/ts/cics

CICS External Interfaces Guide
   Version 1 - SC33-4933
   Version 2 - SC34-6006

CICS RACF Security Guide
   Version 1 - SC33-1701
   Version 2 - SC34-6011

CICS Transaction Gateway:

http://www.software.ibm.com/htp/cics/ctg/library/

CICS Transaction Gateway V6 InfoCenter:

http://publib.boulder.ibm.com/infocenter/cicstg60/index.jsp