



# Cisco Transaction Connection

---

This feature module describes the Cisco Transaction Connection (CTRC) software feature. It includes information on the benefits of using CTRC, supported platforms, configuration instructions, commands, error messages, and additional information about CTRC.

This document contains the following sections:

- Feature Overview, page 2
- CTRC and CICS, page 2
- CTRC and DB2, page 3
- Benefits, page 4
- Related Documents, page 5
- Supported Platforms, page 5
- Supported Standards, MIBs, and RFCs, page 5
- Configuration Tasks, page 6
- Elapsed time was 00:00:00Monitoring and Maintaining CTRC, page 13
- Configuration Examples, page 15
- Command Reference, page 25
- Debug Commands, page 79
- Error Messages, page 102
- Glossary, page 105

## Feature Overview

The Cisco Transaction Connection (CTRC) software feature provides the following functionality:

- CTRC allows Cisco routers to use the intersystem communication (ISC) protocol to provide a gateway between Customer Information Control System (CICS) clients (also known as common clients) running under Windows or UNIX on Transmission Control Protocol/Internet Protocol (TCP/IP) networks and CICS online transaction monitoring systems on IBM hosts.
- CTRC supports two interfaces to common clients: the Extended Call Interface (ECI), which lets non-CICS client programs call CICS transactions, and the Extended Presentation Interface (EPI), which lets distributed applications call CICS transactions that were originally accessed via 3270 terminals.
- CTRC supports the ability to configure routes for CICS transaction. Each transaction can be routed to a specific CICS region.
- In addition to its CICS-related functionality, CTRC includes the feature previously known as Cisco Database Connection (CDBC), which allows Cisco routers to use IBM's distributed relational database architecture (DRDA) protocol to provide a gateway between client workstations running ODBC-compliant applications on TCP/IP networks and IBM DB2 databases on Systems Network Architecture (SNA) networks. ODBC (Open Database Connectivity) is a call-level interface developed by Microsoft Corporation that allows a single application to access database management systems from different vendors using a single interface. SNA is a large, complex, feature-rich network architecture developed by IBM.
- CTRC adds support for TCP/IP passthrough, allowing the use of a TCP/IP network, rather than a SNA network, between a Cisco router and a DB2 database if the database version supports direct TCP/IP access.
- To match functionality provided in DRDA over TCP/IP, CTRC adds support for Password Expiration Management (PEM) in SNA networks where PEM is supported.

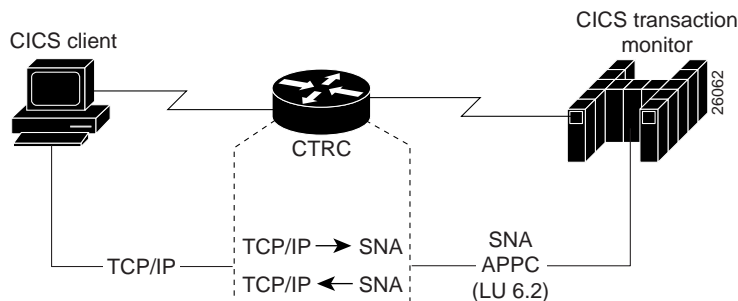
## CTRC and CICS

When a router is configured to use CTRC for communications with CICS systems, the router converts ISC packets over TCP/IP to ISC packets over Advanced Program-to-Program Communications (APPC LU6.2) and then routes them to the appropriate CICS region. CTRC converts CICS client messages received via TCP/IP to SNA messages and uses Cisco SNA Switching Services to transmit them to the host.

CTRC runs as a TCP/IP daemon on the router, accepting ISC client connections over TCP/IP. When a client connects to a CICS region on an IBM mainframe host, CTRC allocates an APPC conversation over SNA to an IBM server and acts as a gateway between ISC over TCP/IP and ISC over APPC.

Figure 1 illustrates how CTRC lets CICS client applications on TCP/IP networks interact with CICS transaction monitoring systems on IBM hosts.

**Figure 1 Cisco Router Configured with the CTRC Feature for CICS Communications**



## CTRC and DB2

CTRC enables Cisco routers to implement IBM’s DRDA over TCP/IP. The Cisco router with CTRC exists in the TCP/IP network, and clients use a CTRC IP address and port on the router to connect to the IBM host system that exists in either an SNA network or a TCP/IP network.

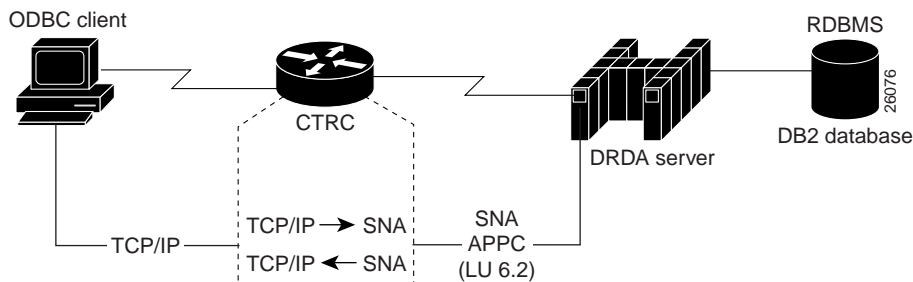
When CTRC is appropriately configured on a router, client-based ODBC applications can connect to the following IBM D2 relational databases:

- DB2 for OS/390 (MVS)
- DB2 for Virtual Machine (VM) (SQL/DS)
- DB2 for Virtual Storage Extended (VSE) (SQL/DS)
- DB2 for OS/400
- DB2 Universal Database (UNIX, Windows, OS/2)

For an SNA host connection, the router with CTRC converts DRDA packets over TCP/IP to DRDA packets over (APPC LU 6.2) and then routes them to DB2 databases. CTRC runs as a TCP/IP daemon on the router, accepting DRDA client connections over TCP/IP. When a client connects to the database on an IBM mainframe host, CTRC allocates an APPC conversation over SNA to an IBM server, and acts as a gateway between DRDA over TCP/IP and DRDA over APPC.

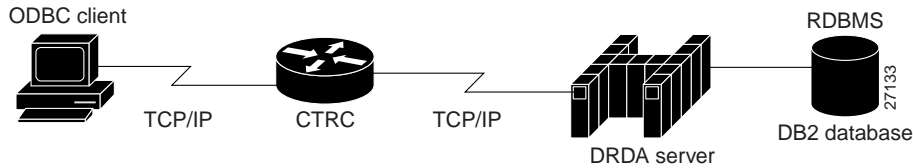
Figure 2 illustrates how the Cisco router configured with the CTRC feature enables the exchange of database information between ODBC client applications running DRDA in a TCP/IP network and a DRDA-based IBM system that accesses DB2 relational data.

**Figure 2 Cisco Router Configured with the CTRC Feature for DB2 Communications (SNA Host Network)**



For a TCP/IP host connection, the router with CTRC routes the DRDA packets over TCP/IP without protocol changes. To use this TCP/IP passthrough feature of CTRC, the host database version must support direct TCP/IP access. Figure 3 illustrates such a configuration.

**Figure 3 Cisco Router Configured with the CTRC Feature for DB2 Communications (TCP/IP Host Network)**



When configured for DB2 communications on a router, the CTRC feature enables desktop applications to access data in remote databases located on IBM hosts. CTRC receives database access messages from the client over a TCP/IP link. CTRC either converts the messages to SNA and transmits them to the host using APPC services provided by the Cisco SNA Switching Services, or routes the client messages to the TCP/IP-enabled host without protocol changes.

## Benefits

Cisco Transaction Connection (CTRC) provides TCP/IP end-users and servers with fast, reliable, and secure access to IBM DB2 databases using the Systems Network Architecture (SNA) protocol. CTRC, a cost-effective software and router solution based on Cisco IOS software, replaces expensive and hard to manage UNIX and NT gateways for database access.

CTRC lets Windows or UNIX client applications call CICS transactions without requiring changes to the client or host software. CTRC provides improved performance, lower cost of ownership, and greater reliability than existing solutions from other vendors.

In addition, CTRC provides Cisco 7200 and 7500 series routers with the functionality previously available in CDBC, which gives ODBC client applications access to data in DB2 databases.

## Related Documents

For information related to this feature, refer to the following document:

- Cisco IOS Release 12.0(5)XN *SNA Switching Services*
- *Revealed! CICS Transaction Gateway with More CICS Clients Unmasked* (IBM publication SG24-5277)

See Chapter 15.4 (Configuration for VTAM & CICS TS for OS/390 when connecting with Microsoft SNA Server)

- *CICS Transaction Server for OS/390 VIR2 CICS Intercommunication Guide* (IBM publication SC33-1695)
- *DRDA Connectivity Guide* (IBM publication SC26-4783)
- *Distributed Relational Database Architecture (DRDA)* (Open Group publication C812, ISBN 1-85912-295-7)

## Supported Platforms

- Cisco 7200 series routers
- Cisco 7500 series routers

## Supported Standards, MIBs, and RFCs

### MIBs

- CISCO-DATABASE-CONNECTION
- CISCO-TRANSACTION-CONNECTION

For descriptions of supported MIBs and how to use MIBs, see the Cisco MIB web site on CCO at <http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>.

### RFCs

No new or modified RFCs are supported by this feature.

### Standards

No new or modified MIBs are supported by this feature.

## Configuration Tasks

CTRC can be configured for use with CICS, with DB2, or both. Both CICS and DB2 configurations require Cisco SNA Switching Services. A few basic instructions for configuring SNA Switching Services are included in the following sections. For more extensive information, consult the SNA Switching Services documentation.

---

**Note** CTRC commands related to communications with CICS contain the word **txconn**. CTRC commands related to communications with DB2 contain the word **dbconn**. With the exception of commands related to licensing, **dbconn** and **txconn** commands act independently of each other and are discussed separately in this document. To configure CTRC for use with both CICS and DB2, complete all the configuration tasks listed below. To configure CTRC for use with CICS only, complete all the configuration tasks *except* those marked (DB2). To configure CTRC for use with DB2 only, complete all the configuration tasks *except* those marked (CICS). See the “Configuration Examples” section of this document for additional information.

---

Perform the following tasks to configure CTRC, in the order that they are listed here:

- Configuring CICS on the Host
- Configuring a CICS Client
- Configuring DB2 on the Host
- Configuring SNA Switching Services
- Configuring a CTRC Destination (CICS)
- Configuring a CTRC Server (CICS)
- Configuring a CTRC Route (CICS)
- Configuring a CTRC Server (DB2)
- Configuring the CTRC License
- Verifying SNA Connectivity
- Verifying CTRC Configuration

## Configuring CICS on the Host

If you plan to use CTRC to communicate with CICS, you must configure CICS for APPC connections. Refer to the “Related Documents” section for additional information.

To configure CICS on the host, complete the following tasks:

- In VTAM, define the PU and LU for the Cisco router that will run CTRC.
- Choose a mode name that will be used by both the Cisco router and VTAM, such as #INTER, and make a note of it.
- Make a note of the VTAM APPLID value. You will need this value in later steps.
- In the CICS System Initialization Table (SIT), set ISC to YES, and set APPLID to the VTAM APPLID value.

- In CEDA (a CICS resource definition transaction), create a SNA Connection Definition with a unique four-character name (such as CTRC). Set NETNAME to the router's LU name (control point name). Set the following additional values:
  - Accessmethod: VTAM
  - Protocol: APPC
  - Attachsec: Verify
- Also in CEDA, create sessions related to the Connection Definition. Set the mode name to the value you chose earlier.
- Make sure that your CICS system includes a DFHCNV table for data conversion between ASCII and EBCDIC. If this table is not present, attempts to install and use CTRC may encounter APPC errors on the host that report failure to transform data received from the remote system. A DFHCNV table may contain lines such as are shown below. For more information about DFHCNV tables, see your CICS documentation.

```
PRINT NOGEN
DFHCNV TYPE=INITIAL, SRVERCP=037, CLINTCP=437
DFHCNV TYPE=FINAL
END DFHCNVBA
```

The “Configuration Examples” section of this document illustrates many of the host configuration tasks described above.

## Configuring a CICS Client

To use IBM's CICS Common Client application with CTRC, complete the following tasks:

- Install the CICS client. You do not need to install TCP62.
- Edit the file CICSCLI\BIN\CICSLI.INI as follows:
  - If you want to use multiple servers, increase the value of MaxServers from its default value of 1.
  - Create one or more server descriptions, supplying the following information and making a note:
    - Server = Server name
    - Protocol = TCPIP
    - NetName = Host name or IP address for the CTRC router
    - Port = TCP/IP port number for the CTRC router (specify 0 to use the default of 1435)
  - Include the following driver statement:
    - Driver = TCPIP
    - Drivername = CCLWNTIP

## Configuring DB2 on the Host

For information about preparing a DB2 system to receive communications from CTRC, see the *CTRC Planning Guide* available from StarQuest Software, Inc.:

StarQuest Software, Inc.  
 1288 Ninth Street  
 Berkeley, CA 94710-1501

1-800-763-0050 in the US and Canada  
 1-510-528-2900 Worldwide, ask for extension 170  
 1-510-528-2986 FAX

sales@starquest.com

## Configuring SNA Switching Services

To configure basic SNA Switching Services, complete the following steps:

Step	Command	Purpose
1	Router#snasw cpname netid.cpname [hostname] [ip-address interface-name]	Defines a SNA Switching Services control point name.
2	Router#snasw port portname [hpr-ip   vdlc vring mac mac-address] [interfacename] [conntype nohpr   len   dyncplen] [dlus-required] [hpr-sap hpr-sap-value] [max-links link-limit-value] [sap sap-value] [vnname virtual-node-name] [nostart]	Associates a SNA Switching Services port with an interface.
3	Router#snasw link linkname port portname [rmac mac-address   ip-dest ip-address] [rsap sap-value] [nns] [nostart]	Associates a SNA Switching Services link with a SNA Switching Services port. <sup>1</sup>

<sup>1</sup> If the partner node is initiating the connection, a link definition is not necessary. It is built dynamically when the partner node initiates the connection. Links are typically defined for upstream connectivity.

---

**Note** For a LEN level connection between SNA Switching Services and the host, you will also need to configure the **snasw location** configuration command for the specific resource names to be contacted on the host. Do not define locations if APPN connectivity is being used between SNA Switching Services and the host.

---

For additional information about configuring SNA Switching Services, consult the SNA Switching Services documentation.

## Configuring a CTRC Destination (CICS)

To configure CTRC to communicate with CICS, the first step is to configure a CTRC destination. If you do not need to communicate with CICS, skip to the “Configuring a CTRC Server (DB2)” section. A CTRC destination is ordinarily a single CICS system defined in terms of its remote LU name and APPC mode. Use the following global configuration command to configure a destination:

Command	Purpose
Router(config)#txconn destination destination-name rlu rlu-name mode mode-name	Specifies a CICS system with which CTRC will communicate.

For load-sharing purposes, the **txconn destination** command can be repeated with the same destination name but new remote LU and mode values to assign more than one CICS system or region to a single CTRC destination name. If a CTRC destination is configured in this way, the CTRC server sends traffic to the destination's defined CICS regions on a rotating basis. A Cisco router can be configured to communicate with multiple CTRC destinations, whether each of those destinations is defined as an individual pair of remote LU and mode values or as a set of such values.

## Configuring a CTRC Server (CICS)

After you have configured a CICS destination, you can configure a CTRC server process that will handle communications with that CICS system. Additional CTRC servers can be configured on the same router for communications with other CICS destinations. Use the following global configuration command to configure a CTRC server process to communicate with CICS:

Command	Purpose
Router(config)# <b>txconn server</b> <i>server-name</i> <b>destination</b> <i>destination-name</i> [ <b>ipaddress</b> <i>ip-address</i> ] [ <b>port</b> <i>port-number</i> ] [ <b>client-timeout</b> <i>minutes</i> ] [ <b>host-timeout</b> <i>minutes</i> ] [ <b>window-size</b> <i>bytes</i> ] [ <b>fold</b> [ <b>off</b>   <b>on</b> ]]	Configures a CTRC server process for communicating with CICS. Note that if you do not supply a port number, CTRC uses the default value of 1435.

When a client attempts to connect to a CTRC server for CICS, the server's port and IP address determine whether that connection is accepted or not. By default, the CTRC server port for CICS client communications is 1435. You can create multiple CTRC txconn and dbconn server processes on one router.

## Configuring a CTRC Route (CICS)

After you have configured one or more destinations and server processes for communicating with CICS, you have the option of explicitly configuring CTRC routes that will direct traffic to the appropriate destination based on a transaction ID. If you do not explicitly configure CTRC routes, the CTRC server routes traffic to its own default destination. To configure a CTRC route, use the following global configuration command:

Command	Purpose
Router(config)# <b>txconn route</b> [ <b>server</b> <i>server-name</i> ] <b>transid</b> <i>transaction-id</i> <b>destination</b> <i>destination-name</i>	Configures a particular route for traffic with the specified transaction ID.

## Configuring a CTRC Server (DB2)

To configure a CTRC server process for communications with DB2, use the following command in global configuration mode:

Command	Purpose
Router(config)# <b>dbconn server</b> <i>server-name</i> [ <b>ipaddress</b> <i>ip-address</i> ] [ <b>port</b> <i>port-number</i> ] [ <b>rdbname</b> <i>rdbname</i> ] [ <b>rlu</b> <i>remote-lu</i> ] [ <b>mode</b> <i>mode</i> ] [ <b>tpname</b> <i>tp-name</i> ] [ <b>idle-timeout</b> <i>minutes</i> ] [ <b>window-size</b> <i>bytes</i> ] [ <b>wlm</b> [ <b>off</b>   <b>on</b> ]]	Configures a CTRC server for communications with DB2.

When a client attempts to connect to a CTRC server for DB2, the server's port, IP address, and remote database name (RDB name) determine if that connection is accepted. By default, the CTRC server port for client requests for DB2 communications is 446. You can create multiple CTRC dbconn and txconn server processes on one router.

## Configuring the CTRC License

An unlicensed installation of CTRC allows you to use up to two DB2 connections, two CICS conversations, or one DB2 connection and one CICS conversation for testing purposes. For communications with DB2, CTRC checks the number of connections against the licensed number of users. For communications with CICS, CTRC checks the number of concurrent and queued conversations. To use additional connections or conversations, you must configure the CTRC license. One license key is used for both CICS and DB2 communications.

To obtain your CTRC license key, use the **show config | include cpname** command to determine the SNA Switching Services cpname for the router you will be licensing. (The CTRC license key is locked to one node and is based on the cpname.) Next, contact your Cisco representative, request a CTRC license key, and provide the cpname value. You should receive information about the number of connections you are licensing along with the license key. If the key is temporary, you should also receive information about the expiration date. After obtaining your license key, use either of the following global configuration commands to configure the CTRC license:

Command	Purpose
Router(config)# <b>dbconn license</b> <i>license-key</i> <b>connections</b> <i>licensed-connections</i> <b>[expiration-date</b> <i>yyyymmdd</i> ]	Configures a CTRC license key for DB2 or CICS.
Router(config)# <b>txconn license</b> <i>license-key</i> <b>[connections</b> <i>licensed-connections</i> ] <b>[expiration-date</b> <i>yyyymmdd</i> ]	Configures a CTRC license key for CICS or DB2.

## Verifying SNA Connectivity

To verify that you have SNA connectivity between the router and each host system, complete the following task:

- Use the **ping sna** command, specifying the mode and the remote LU name:

```
ping sna -m IBMRDB STARW.BUDDY
```

## Verifying CTRC Configuration

To verify the CTRC configuration, complete the following tasks:

- Step 1** If you did not configure CTRC for communications with CICS, go to Step 6.

- Step 2** If you configured CTRC for communications with CICS, enter the **show txconn destination** command in EXEC or privileged EXEC mode. Make sure that any CICS destinations you configured are listed with the RLU and mode values you specified.

```
Router>show txconn destination
Name          Remote LU          Mode          Hits
-----
CICSB         CICSB             IBMRDB        31
GEN           CICSB             IBMRDB        50
              CICSC             IBMRDB        51
GUAVA         GUAVA             IBMRDB        0
CICSC         CICSC             IBMRDB        7
```

- Step 3** Enter the **show txconn server** command. Make sure that any CTRC servers you defined for communications with CICS are listed with the configuration values you specified.

```
Router>show txconn server
Server      Port  IP Address  Dest      State      NumConn
-----
CICSB      1435  0.0.0.0    CICSB     enabled    1
CICSB&C    1436  0.0.0.0    GEN       enabled    0
CICSC      1434  0.0.0.0    CICSC     enabled    0
GUAVA      1437  0.0.0.0    GUAVA     enabled    0
```

If necessary, use the **show txconn server server-name** form of the command to display more information for an individual server.

```
Router>show txconn server CICSB
server: CICSB
destination: CICSB
server state: enabled (accepting connections)
ip address: 0.0.0.0
port: 1435
client timeout: 0 (none)
host timeout: 0 (none)
window size: 4096 bytes
fold program name: on
number of connections: 1
number of transactions: 2
```

- Step 4** If you defined any routes for specific transaction IDs to take to CICS destinations, enter the **show txconn route** command. Make sure that any CTRC routes you defined are listed with the configuration values you specified. A <default> in the SERVER column indicates a global route that is used by all txconn servers on the router. A <default> in the TRANID column indicates the default route for the listed txconn server.

```
Router>show txconn route
Server      TranID          Destination
-----
CICSC       <default>       CICSC
CICSB       <default>       CICSB
CICSB&C     <default>       GEN
GUAVA       <default>       GUAVA
<default>   CPMI            CICSC
CICSB       CPMI            CICSB
```

- Step 5** If you did not configure CTRC communications with DB2, go to Step 7.

- Step 6** If you configured CTRC for communications with DB2, enter the **show dbconn server** command. Make sure that the servers you defined are listed with the configuration values you specified. If necessary, use the **show dbconn server server-name** form of the command to display more information for an individual server.

```
Router>show dbconn server
Server      Port  IPAddress      RDBName      State      NumConn
SERVERA    446   0.0.0.0        MATTY        enabled    0
SERVERB    446   0.0.0.0        SCU_DSNM     enabled    4
SERVERC    446   0.0.0.0        DSN4         enabled    0
SERVERD    446   0.0.0.0        MKTG         enabled    0
SERVERE    446   0.0.0.0        ABBY         enabled    6
SERVERF    446   0.0.0.0        DB2510       enabled    0
SERVERG    446   0.0.0.0        ELLE         enabled    3
SERVERH    446   0.0.0.0        SUNSET       enabled    0
SERVERI    446   0.0.0.0        NELL         enabled    1
SERVERJ    446   198.989.999.32 SAMPLE       enabled    0
SERVERK    446   0.0.0.0        DB2410       enabled    0
SERVERL    446   0.0.0.0        SQLDS        enabled    0
SERVERM    446   0.0.0.0        STELLA       enabled    1
SERVERN    446   10.10.19.4     OAK          enabled    2
SERVERO    447   0.0.0.0        DB2510       enabled    0
```

```
Router>show dbconn server BUDDY
server: BUDDY
server state: enabled (accepting connections)
ip-address: 0.0.0.0
port: 446
rdbname: DB2510
connection type: SNA
rlu: STARW.DSNV50
mode: IBMRDB
tpname: \x076DB
idle-timeout: 0 (none)
window-size: 4096 bytes
database server name: (unknown)
database product id: (unknown)
PEM: not configured
number of connections: 0
RDB server: active
WLM: inactive-enabled
```

- Step 7** Enter either the **show dbconn license** command or the **show txconn license** command. Make sure that the reported number of licensed connections matches the number you purchased.

```
Router>show txconn license
CTRC is licensed for 4990 connections, 20 licensed connections in use
This is a permanent license
```

```
Router>show dbconn license
CTRC is licensed for 1000 connections, no licensed connections in use
Expires on 1-1-2001.
```

- Step 8** For each CICS destination you displayed in Step 1, enter the **txconn ping** command to verify that the router can communicate with that destination.

```
Router>txconn ping CICSB
Trying CICSB CICSB:IBMRDB
Destination CICSB successfully contacted!
Elapsed time was 00:00:00.600
```

- Step 9** For each dbconn server you displayed in Step 6, enter the **dbconn ping** command to verify that the router can communicate with the DB2 systems associated with that server.

```
Router>dbconn ping BUDDY
.....
RDB named DB2510 on database server successfully contacted!
Elapsed time was 00:00:00Monitoring and Maintaining CTRC
```

This section describes commands used to monitor and maintain CTRC. Commands for CICS communications and DB2 communications are shown separately.

## Monitoring and Maintaining CTRC Communications with CICS

Command	Purpose
Router#show txconn server	Displays a list of CTRC servers for CICS communications for the current router.
Router#show txconn server <i>server-name</i>	Displays detailed status information for the specified CTRC server.
Router#show txconn connection [ <i>server server-name</i> ]	Displays a list of all CTRC connections to CICS clients from the current router or a particular server's CICS client connections.
Router#show txconn connection <i>connection-id</i>	Displays detailed status information for the specified CTRC connection to a CICS client.
Router#show txconn transaction [ <i>server server-name connection connection-id</i> ]	Displays a list of the current router's CTRC transactions involving CICS, or the transactions of a particular server or CICS client connection.
Router#show txconn transaction <i>transaction-id</i>	Displays detailed status information for the specified CTRC transaction.
Router#show txconn route [ <i>server server-name</i> ]	Displays a list of CTRC routes to CICS for the current router or a particular server.
Router#show txconn destination	Displays a list of all the current router's CTRC destinations for CICS communications.
Router#show txconn destination <i>destination-name</i>	Displays detailed status information for the specified CTRC destination.
Router#clear txconn transaction <i>transaction-id</i>	Kills the specified CTRC transaction.
Router#clear txconn connection <i>connection-id</i>	Kills the specified CTRC connection to a CICS client and all associated transactions.
Router#clear txconn statistics [ <i>kind histogram   kind summary</i> ] <i>name</i> { <i>activeconnections   activetransactions   allocatettime   clientreceived   clientsent   clientturnaround   dump   hostreceived   hostresponse   hostsent   latency   maxconnections   maxtransactions   totalconnections   totaltransactions</i> }	Clears specific statistics or all statistics related to CTRC communications with CICS.
Router#debug txconn { <i>appc   config   data   event   tcp   all</i> }	Enables debugging of CTRC communications with CICS.
Router#show debugging	Displays current status of debugging for the router.
Router#show txconn license or show dbconn license	Shows the status of the CTRC license.
Router#txconn ping <i>destination-name</i>	Verifies connectivity with a specific CICS destination.

## Monitoring and Maintaining CTRC Communications with DB2

---

Command	Purpose
<code>Router#show txconn statistics [kind histogram   kind summary] name {activeconnections   activetransactions   allocatetime   clientreceived   clientsent   clientturnaround   dump   hostreceived   hostresponse   hostsent   latency   maxconnections   maxtransactions   totalconnections   totaltransactions}</code>	Displays statistics related to CTRC communications with CICS.

---

## Monitoring and Maintaining CTRC Communications with DB2

---

Command	Purpose
<code>Router#show dbconn server</code>	Displays a summary of information about each CTRC server configured to communicate with DB2.
<code>Router#show dbconn server server-name</code>	Displays a detailed status of the specified CTRC server for DB2 communications.
<code>Router#show dbconn connection</code>	Displays the status of each CTRC connection to DB2.
<code>Router#show dbconn connection connection-id</code>	Displays a detailed status of the specified CTRC connection to DB2.
<code>Router#show dbconn connection server server-name</code>	Displays the status of CTRC connections to DB2 for the specified server.
<code>Router#show dbconn connection user userid</code>	Displays the status of a user connected to the CTRC for DB2 communications.
<code>Router#show dbconn connection rdbname rdb-name</code>	Displays a status of each connection to DB2 that matches the specified RDB name.
<code>Router#show dbconn ports</code>	Displays information on all ports through which CTRC servers are accepting connections to DB2.
<code>Router#show dbconn license or show txconn license</code>	Displays the status of the CTRC license for both DB2 and CICS.
<code>Router#clear dbconn connection connection-id</code>	Breaks the specified client connection to the server.
<code>Router#dbconn ping server-name [rdbname rdbname] [userid userid] [password password]</code>	Connects to the DB2 relational database on the IBM system for troubleshooting.
<code>Router#debug dbconn {appc   config   drda   event   tcp   all}</code>	Enables debugging of CTRC communications with DB2.
<code>Router#show debugging</code>	Displays current status of debugging for CTRC.
<code>Router#show dbconn statistics [chains clientturnaround connectionsdown connectionsup dump hostreceived hostresponse hostsent latency max connections]</code>	Displays current statistics related to CTRC communications with DB2.
<code>Router#clear dbconn statistics [chains clientturnaround connectionsdown connectionsup dump hostreceived hostresponse hostsent latency max connections]</code>	Clears statistics related to CTRC communications with DB2.

---

## Configuration Examples

The following sections provide CTRC configuration examples:

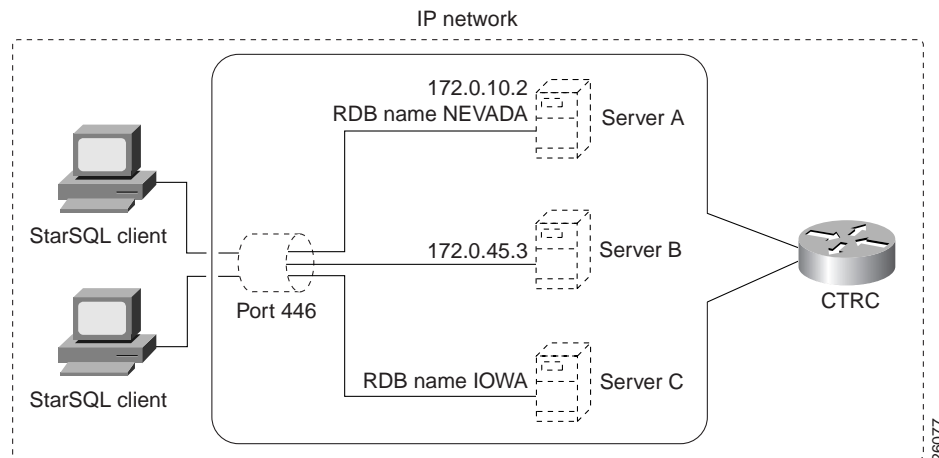
- CTRC Servers with IP Addresses Configuration Example (DB2)
- CTRC Servers with IP Addresses, RDB Names, and Ports Configuration Example 1 (DB2)
- CTRC Servers with IP Addresses, RDB Names, and Ports Configuration Example 2 (DB2)
- Server Selection by IP Addresses, RDB Names, and Ports Configuration Example (DB2)
- CTRC with CIP and DB2 on VTAM Configuration Example (DB2)
- CTRC Servers Using Token Ring to a LEN Configuration Example (CICS and DB2)
- CTRC Servers with IP Addresses, Routes, and Multi-Valued Destinations Configuration Example (CICS)

### CTRC Servers with IP Addresses Configuration Example (DB2)

Figure 4 shows a CTRC configuration where the CTRC servers are configured to listen on port 446 (by default) for IP addresses specified for these servers in the router's configuration for CTRC. When an ODBC client attempts to make a connection to DB2, a CTRC server accepts the connection if the IP address specified in its configuration matches the IP address to which the client wants to connect.

In this illustration, Servers A and B are configured with IP addresses 172.0.10.2 and 172.0.45.3. Servers A and B accept any connection that targets their IP addresses. Server C accepts any connection that targets any IP address of router on the target port of 446 and an RDB name of IOWA.

**Figure 4 CTRC Servers' Configuration with IP Addresses (for DB2 Communications)**



The following are the commands that configure servers Server A, Server B, and Server C in the Cisco router:

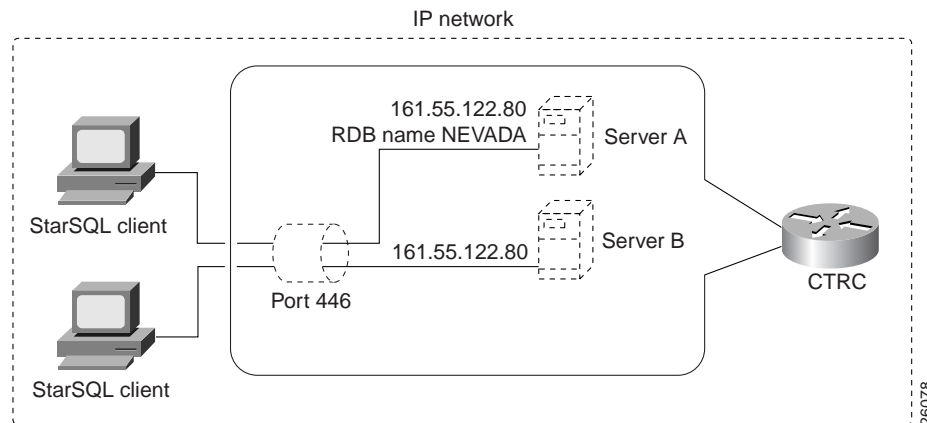
```
dbconn server SERVERA ip-address 172.0.10.2 rdbname nevada
dbconn server SERVERB ip-address 172.0.45.3
dbconn server SERVERC rdbname iowa
```

## CTRC Servers with IP Addresses, RDB Names, and Ports Configuration Example 1 (DB2)

When a client request comes in for a server, and multiple servers are configured in the router, the three configured attributes of IP address, RDB name, and port determine which server is chosen for the connection. When a server is selected for a connection, the client remains associated with that server for the duration of that connection. The APPC attributes configured for that server are used to connect to the IBM system. If a server is unconfigured while active connections exist, the active connections with that server will break.

Only one CTRC server can be configured with a unique combination of IP address, port, and RDB name. If a situation arises where multiple servers in a router meet the criteria for accepting a client connection, the CTRC server that meets the most specific criteria accepts the connection. For example, in Figure 5 Servers A and B are listening on port 446 for client connections that match their IP address of 161.55.122.80. Server A is configured to accept RDB name NEVADA and Server B is configured to accept any RDB name. A client connecting to port 446 for RDB name NEVADA matches the criteria for both servers. In this situation, Server A is selected to accept the connection because its configuration includes a specific RDB name NEVADA as compared to Server B whose configuration accepts any RDB name.

**Figure 5 CTRC Server Configuration with IP Address and RDB Name Defined**



## CTRC Servers with IP Addresses, RDB Names, and Ports Configuration Example 2 (DB2)

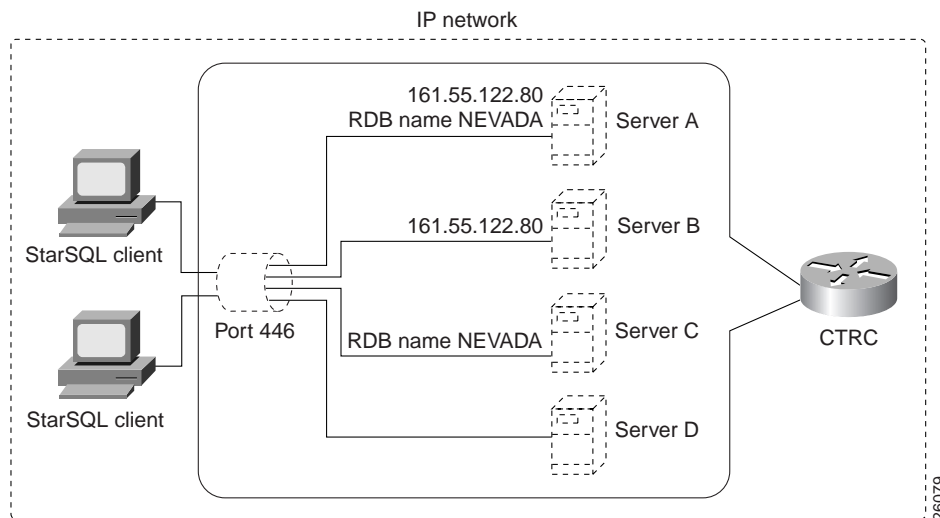
The IP address and port specified for a server in a router’s configuration also determines which server accepts a connection. For example, Server C is configured to listen on any local IP address on port 446 and RDB name IOWA. Server D is configured to listen for IP address 145.56.180.34 on port 446 and RDB name IOWA. When a client attempts to connect to IP address 145.56.180.34 on port 446 for RDB name IOWA, both servers meet the criteria in accepting the connection. In this case, CTRC selects a connection based on the IP address first, then the port, and finally, the RDB name.

## Server Selection by IP Addresses, RDB Names, and Ports Configuration Example (DB2)

If multiple servers in a router meet the criteria for accepting a client connection, the CTRC server that meets the most specific criteria accepts the connection. In Figure 6, the Cisco router contains four server configurations. All four servers listen for client connections on port 446 by default. Both Servers A and B are configured with the same IP address, 161.55.122.80. Servers A and C are configured to accept RDB name NEVADA. Servers B and D are configured to accept any RDB name.

If a client connects to IP address 161.55.122.80 on port 446 and sends RDB name NEVADA in the DRDA data stream, all four servers match the criteria for accepting the client connection. However, Server A will be selected to accept the connection because it meets the most specific criteria for IP address, RDB name, and port. If Server A was not configured, Server B would be the second choice because it meets the criteria for the IP address and port. The IP address specified in a server always has precedence when matching a connection to a server.

**Figure 6 CTRC Server Configurations with IP Addresses, RDB Names, and Default Port**



The following is the configuration for Servers A, B, C, and D in the Cisco router:

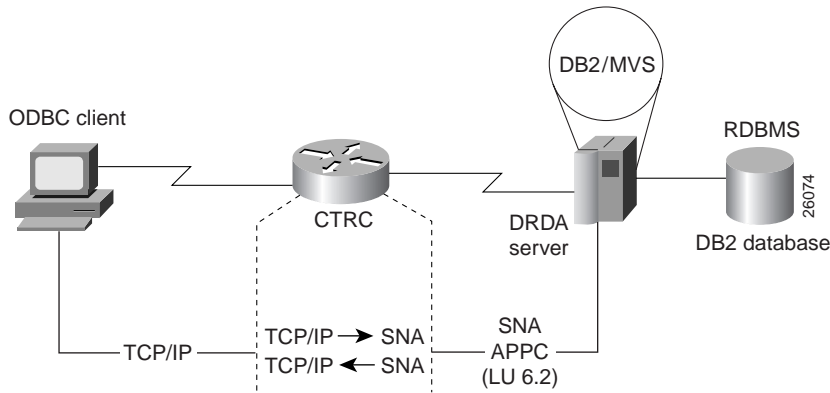
```
hostname routera
!
enable password allie

dbconn server SERVERA ip-address 161.55.122.80 rdbname NEVADA
dbconn server SERVERB ip-address 161.55.122.80
dbconn server SERVERC rdbname NEVADA
dbconn server SERVERD
```

## CTRC with CIP and DB2 on VTAM Configuration Example (DB2)

Figure 7 illustrates a Cisco router with a Channel Interface Processor (CIP) that is configured with CTRC. The CIP is networked and connected to VTAM on the mainframe. DB2 is configured on VTAM.

**Figure 7 Cisco Router with CIP and Connection to DB2 on VTAM**



The configuration in Figure 7 uses router commands to configure SNA Switching Services over CIP and CSNA via SRB. The following three pages of this document show the configuration in more detail. Note that the source-bridge ring-group of 100 matches the source bridge of 10 2 100 for interface Channel 13/2 to enable SNA Switching Services to run over SRB. In addition, the destination LAN address used by the snasw link station BUDDY corresponds to the virtual MAC address used by the adapter for Channel 13/2.

In the VTAM host definitions, the variable CONNTYPE=APPN is optional but recommended if you use APPN in your SNA environment. If CP to CP is set to YES and CONNTYPE is set to APPN, this configuration enables the Cisco router to establish CP-to-CP sessions with VTAM. By allowing CP-to-CP sessions, you gain the benefit of APPN's dynamic features such as the availability of directory and topology for locating resources and calculating optimal routes.

In this example, the router smoke is attached to the host BUDDY using a CIP processor.

---

**Note** Do not use a **snasw location** statement if you are using an APPN connection between the host and SNA Switching Services.

---

VTAM Partner PU and LU Definition

```

SMOKEPB  PU  ADDR=01,                X
          IDBLK=05D,                  X
          IDNUM=00501,                 X
          CPNAME=SMOKE,                X
          ANS=CONT,                    X
          DISCNT=NO,                   X
          IRETRY=NO,                   X
          ISTATUS=ACTIVE,              X
          PUTYPE=2,                     X
          SECNET=NO,                   X
          MAXDATA=521,                 X
          MAXOUT=7,                    X
          MAXPATH=1,                   X
          USSTAB=USSS,                 X
          MODETAB=ISTINCLM,            X
          DLOGMOD=IBMRDB,              X
          CONNTYPE=APPN
**
*
SMOKE  LU  LOCADDR=00,                INDEPENDENT LU      X
          DLOGMOD=IBMRDB
    
```

VTAM APPLID for DB2

```

DSNV510 APPL  APPC=YES,                X00006012
          AUTH=ACQ,                     X00007012
          AUTOSES=1,                     X00008012
          DMINWNL=1024,                  X00009012
          DMINWNR=1024,                  X00009112
          DSESLIM=2048,                  X00009212
          EAS=65535,                     X00009312
          MODETAB=ISTINCLM,              X00009412
          SECACPT=CONV,                  X00009512
          SRBEXIT=YES,                   X00009612
          VERIFY=NONE,                   X00009712
          VPACING=1,                      X00009812
          SYNCLVL=SYNCPT,                X00009912
          ATNLOSS=ALL                     00010012
    
```

### Cisco IOS Configuration

```
!  
source-bridge ring-group 100  
!  
interface Ethernet2/1  
  mac-address 4200.0000.0501  
  ip address 198.147.235.11 255.255.255.128  
  no ip directed-broadcast  
  no ip mroute-cache  
!  
interface Channel3/0  
  ip address 192.168.1.1 255.255.255.0  
  no ip directed-broadcast  
  no keepalive  
  channel-protocol S4  
  claw 0100 22 192.168.1.2 BUDDY CIPTCP TCPIP TCPIP  
  csna 0100 20  
!  
interface Channel3/2  
  no ip address  
  no ip directed-broadcast  
  no keepalive  
  lan TokenRing 1  
  source-bridge 10 2 100  
  adapter 1 4000.0123.9999  
!  
interface Virtual-TokenRing0  
  mac-address 4000.2222.3333  
  source-bridge 50 1 100  
  source-bridge spanning  
!  
snasw cpname STARW.SMOKE  
snasw port SRB Virtual-TokenRing0  
snasw link BUDDY port SRB rmac 4000.0123.9999  
!  
dbconn server DB2BUDD rdbname DB2510 rlu STARW.DSNV510 mode IBMRDB  
!  
ip default-gateway 198.147.235.12  
ip classless
```

## CTRC Servers Using Token Ring to a LEN Configuration Example (CICS and DB2)

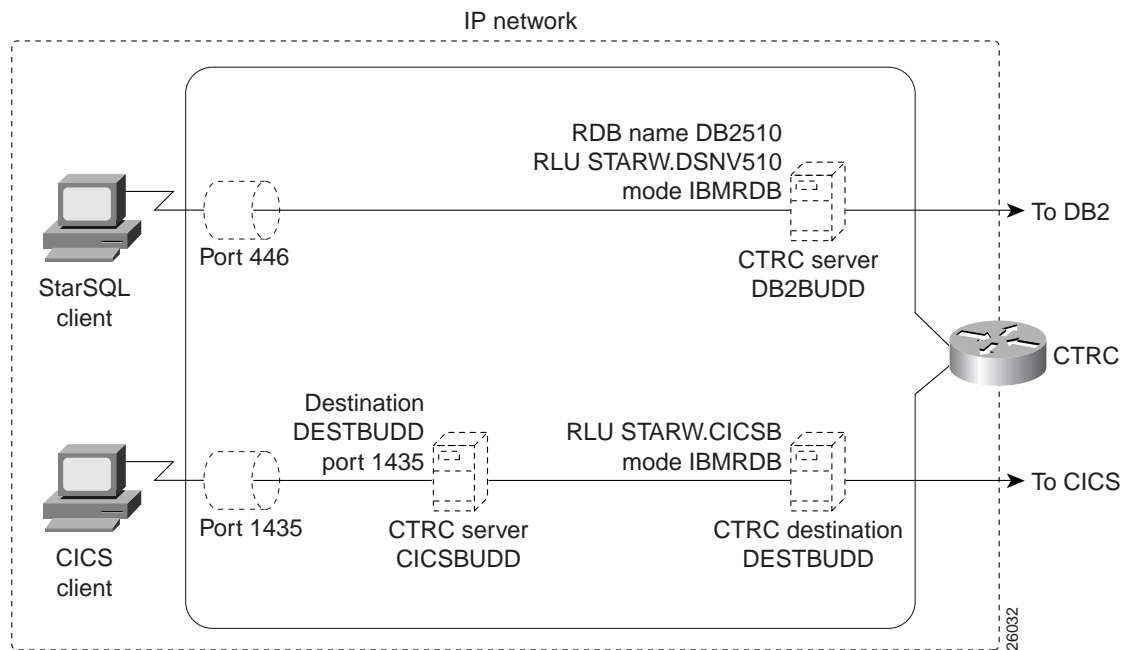
This section provides a configuration example for a router named smoke, beginning with the VTAM definition for the router, which is the same as for the previous example.

The router is connected to the host via Token Ring. The control point name of the host is BUDDY; its Token Ring MAC address is 4000.0200.0448.

The host is configured as a Subarea Node (APPN LEN); if a host is configured as an APPN Network Node, the snasw location statements are unnecessary.

Figure 8 shows a CTRC configuration for communication with DB2 and CICS.

Figure 8 CTRC Configuration for Communication with DB2 and CICS



VTAM Partner PU and LU Definition

```

RouterPB  PU  ADDR=01,                                X
           IDBLK=05D,                                X
           IDNUM=00501,                              X
           CPNAME=SMOKE,                             X
           ANS=CONT,                                  X
           DISCNT=NO,                                 X
           IRETRY=NO,                                 X
           ISTATUS=ACTIVE,                           X
           PUTYPE=2,                                  X
           SECNET=NO,                                 X
           MAXDATA=521,                              X
           MAXOUT=7,                                  X
           MAXPATH=1,                                 X
           USSTAB=USSS,                              X
           MODETAB=ISTINCLM,                         X
           DLOGMOD=IBMRDB,                           X
           CONNNTYPE=APPN
**
*
SMOKE  LU  LOCADDR=00,      INDEPENDENT LU          X
           DLOGMOD=IBMRDB
    
```

### VTAM APPLID for DB2

```

DSNV510 APPL  APPC=YES,                X00006012
                AUTH=ACQ,              X00007012
                AUTOSSES=1,            X00008012
                DMINWNL=1024,          X00009012
                DMINWNR=1024,          X00009112
                DSESLIM=2048,          X00009212
                EAS=65535,             X00009312
                MODETAB=ISTINCLM,      X00009412
                SECACPT=CONV,          X00009512
                SRBEXIT=YES,           X00009612
                VERIFY=NONE,           X00009712
                VPACING=1,              X00009812
                SYNCLVL=SYNCPT,        X00009912
                ATNLOSS=ALL             00010012
    
```

### VTAM APPLID for CICS

```

CICSAPPL VBUILD TYPE=APPL                00010001
***** 00010000
* CICS APPL DEFINITION FOR LU62 CLIENT/SERVER SUPPORT 00020000
***** 00030000
CICSB  APPL  AUTH=(ACQ,SPO,PASS,VPACE) ,    X
        MODETAB=ISTINCLM,                  X
        VPACING=0,EAS=100,PARSESS=YES,      X
        APPC=NO,                            X
        SONSCIP=YES,                        X
        ACBNAME=STARCICS
    
```

### Cisco IOS Configuration

```

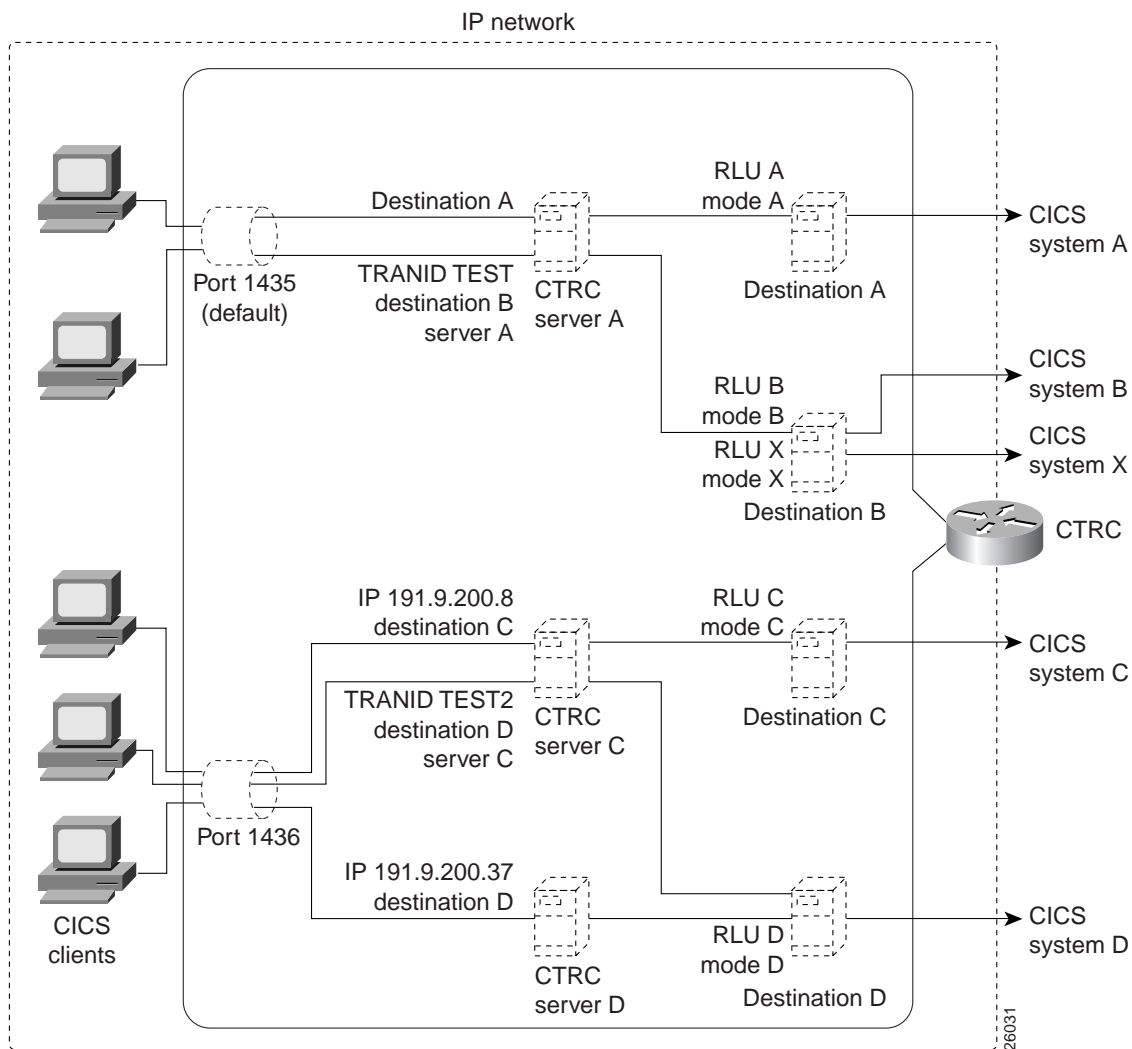
!
source-bridge ring-group 100
!
!
interface TokenRing0/1
 mac-address 4000.1111.0501
 ip address 198.147.235.196 255.255.255.224
 no ip directed-broadcast
 no ip mroute-cache
 early-token-release
 ring-speed 16
 multiring all
!
interface Ethernet2/1
 mac-address 4200.0000.0501
 ip address 198.147.235.11 255.255.255.128
 no ip directed-broadcast
 no ip mroute-cache
!
!
snasw cpname STARW.SMOKE
snasw port TR0 TokenRing0/1
snasw link BUDDY port TR0 rmac 4000.0200.0448
snasw location STARW.DSNV510 owning-cp STARW.BUDDY
snasw location STARW.CICSB owning-cp STARW.BUDDY
!
dbconn server DB2BUDD rdbname DB2510 rlu STARW.DSNV510 mode IBMRDB
!
txconn destination DESTBUDD rlu STARW.CICSB mode IBMRDB
txconn server CICSBUDD destination DESTBUDD port 1435
    
```

```
ip default-gateway 198.147.235.12
ip classless
```

## CTRC Servers with IP Addresses, Routes, and Multi-Valued Destinations Configuration Example (CICS)

Figure 9 shows a CTRC configuration that includes multiple CTRC servers, routes, default and non-default ports, and one multi-valued CTRC destination.

**Figure 9** CTRC Configuration with IP Addresses, Routes, and Multiple CICS Destinations



In Figure 9, a single router is configured to run three CTRC servers for communication with CICS. These **txconn servers** are shown as CTRC server A, CTRC server C, and CTRC server D. Server A listens on the default port, 1435, for all of the router’s IP addresses. Server C listens on port 1436 for IP address 191.9.200.8. Server D listens on port 1436 for IP address 191.9.200.37.

Server A is configured to communicate with two logical destinations. If a client communication has the value of TEST for its transaction ID (TRANID), server A sends it to logical Destination B. This is a multi-valued destination that allows communication with two CICS systems, system B (with RLU B and mode B) and system X (with RLU X and mode X). Transactions are allocated to these two destination systems on a round-robin basis.

If a client communication for server A does not have a value of TEST for TRANID, server A sends it to Destination A, which corresponds to CICS system A (with RLU A and mode A).

Server C is also configured to communicate with two logical destinations. If server C receives a client communication that has the value of TEST2 for its transaction ID, server C sends it to logical Destination D, which corresponds to CICS system D (with RLU D and mode D). Server C sends client communications with other transaction IDs to logical Destination C (CICS system C, with RLU C and mode C).

Server D is configured to send client communications to logical Destination D. This example illustrates the following CTRC configuration principles:

- One router can run multiple CTRC **txconn** servers.
- One **txconn** server can communicate with multiple logical destinations.
- One CTRC logical destination can correspond to multiple CICS destination systems.
- More than one **txconn** server can use a single port number, provided that each server listens on a different IP address.
- More than one **txconn** server can direct traffic to a single logical destination.

## Command Reference

This section documents all non-debug commands used with the CTRC feature. Some of these commands were previously used with the CDBC feature, which CTRC replaces. Other commands are new with the introduction of CTRC.

- **clear dbconn connection**
- **clear txconn connection**
- **clear txconn statistics**
- **clear dbconn statistics**
- **dbconn license**
- **dbconn pem**
- **dbconn ping**
- **dbconn server**
- **dbconn tcpserver**
- **show dbconn connection**
- **show dbconn license**
- **show dbconn ports**
- **show dbconn server**
- **show dbconn statistics**
- **show dbconn wlm**
- **show txconn destination**
- **show txconn license**
- **show txconn route**
- **show txconn server**
- **show txconn statistics**
- **show txconn transaction**
- **txconn destination**
- **txconn license**
- **txconn ping**
- **txconn route**
- **txconn server**

In Cisco IOS Release 12.0(1)T or later, you can search and filter the output for **show** and **more** commands. This functionality is useful when you need to sort through large amounts of output, or if you want to exclude output that you do not need to see.

To use this functionality, enter a **show** or **more** command followed by the “pipe” character (`|`), one of the keywords **begin**, **include**, or **exclude**, and an expression that you want to search or filter on:

```
command / {begin | include | exclude} regular-expression
```

Following is an example of the **show atm vc** command in which you want the command output to begin with the first line where the expression “PeakRate” appears:

**show atm vc / begin PeakRate**

For more information on the search and filter functionality, refer to the Cisco IOS Release 12.0(1)T feature module titled *CLI String Search*.

## clear dbconn connection

To break a client connection to DB2, use the **clear dbconn connection** privileged EXEC command.

**clear dbconn connection** *connection-id*

### Syntax Description

*connection-id* Identification number for client connection to DB2.

### Defaults

None.

### Command Modes

Privileged EXEC

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Examples

The following example shows the client connection 786A7C being cleared:

```
clear dbconn connection 786A7C
```

```
Connection 786A7C cleared
```

### Related Commands

Command	Description
<b>show dbconn connection</b>	Displays information about CTRC connections to DB2.

## clear txconn connection

To clear a CTRC connection to a CICS client and all associated transactions, use the **clear txconn connection** privileged EXEC command.

**clear txconn connection** *connection-id*

### Syntax Description

*connection-id* CICS connection identification number.

### Defaults

None.

### Command Modes

Privileged EXEC

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example clears the specified CICS client connection number 62146088 and all its transactions:

```
clear txconn connection 62146088
% Connection 62146088 cleared
```

### Related Commands

Command	Description
<b>clear txconn transaction</b>	Kills a specific CICS transaction.
<b>show txconn connection</b>	Displays information about CTRC connections to CICS.

## clear txconn statistics

To clear a specified CTRC statistic concerning communications with CICS, or to clear all such statistics, use the **clear txconn statistics** privileged EXEC command.

```
clear txconn statistics { allocatetime | clientreceived | clientsent | clientturnaround | every |
hostreceived | hostresponse | hostsent | maxconnections | maxtransactions |
totalconnections | totaltransactions }
```

### Syntax Description

<b>allocatetime</b>	Clears statistics for average time spent waiting for APPC allocate operation to complete.
<b>clientreceived</b>	Clears statistics for number of bytes received from CICS clients.
<b>clientsent</b>	Clears statistics for number of bytes sent to CICS clients.
<b>clientturnaround</b>	Clears statistics for average time from receiving a CICS client communication to sending that client a response.
<b>every</b>	Clears every statistic concerning the current router's CTRC communications with CICS.
<b>hostreceived</b>	Clears statistics for number of bytes received from CICS hosts.
<b>hostresponse</b>	Clears statistics for average CICS host response time.
<b>hostsent</b>	Clears statistics for number of bytes sent to CICS hosts.
<b>maxconnections</b>	Clears statistics for maximum number of concurrent connections to CICS clients.
<b>maxtransactions</b>	Clears statistics for maximum number of concurrent transactions with CICS hosts.
<b>totalconnections</b>	Clears statistics for total number of connections to CICS clients.
<b>totaltransactions</b>	Clears statistics for total number of CICS transactions processed.

### Defaults

None.

### Command Modes

Privileged EXEC

## clear txconn statistics

---

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example clears the stored statistics for the number of bytes the current router has received from CICS clients:

```
clear txconn statistics clientreceived
```

### Related Commands

Command	Description
<b>show txconn statistics</b>	Displays statistics related to CTRC communications with CICS.

## clear dbconn statistics

To clear all CTRC statistics concerning communications with DB2, use the **clear dbconn statistics** privileged EXEC command.

```
clear dbconn statistics { chains | clientturnaround | connectionsdown | connectionsup | dump |
  hostreceived | hostresponse | hostsent | latency | maxconnections }
```

### Syntax Description

<b>chains</b>	Clears the number of command chains created between CTRC and DB2.
<b>clientturnaround</b>	Clears statistics for average time from receiving a DB2 client communication to sending that client a response.
<b>connectionsdown</b>	Clears statistics for number of connections down between CTRC and DB2
<b>connectionsup</b>	Clears statistics for number of connections up between CTRC and DB2
<b>dump</b>	Clears the complete statistics dump between CTRC and DB2.
<b>hostreceived</b>	Clears statistics for number of bytes received from DB2 hosts.
<b>hostresponse</b>	Clears statistics for average DB2 host response time.
<b>hostsent</b>	Clears statistics for number of bytes sent to DB2 hosts.
<b>latency</b>	Clears the statistic for average server latency.
<b>maxconnections</b>	Clears statistics for maximum number of concurrent connections to CICS clients.

### Defaults

None.

### Command Modes

Privileged EXEC

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example clears the stored statistics for the number of bytes the current router has received from DB2 clients:

```
clear dbconn statistics clientreceived
```

**clear dbconn statistics**

---

Related Commands

<b>Command</b>	<b>Description</b>
<b>show dbconn statistics</b>	Displays statistics related to CTRC communications with DB2.

## clear txconn transaction

To terminate a specified CICS transaction, use the **clear txconn transaction** privileged EXEC command. This command terminates the conversation with the host and returns DEALLOC\_ABEND\_PROG to the client.

**clear txconn transaction** *transaction-id*

### Syntax Description

*transaction-id* ID of the CICS transaction to be cleared.

### Defaults

None.

### Command Modes

Privileged EXEC

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example clears the CICS transaction number 621FC8E0

```
clear txconn transaction 621FC8E0
% Transaction 621FC8E0 cleared
```

### Related Commands

Command	Description
<b>show txconn transaction</b>	Displays information about CICS transactions.

## dbconn license

To configure client licenses for CTRC connections to DB2 or CICS, use the **dbconn license** global configuration command. To remove the licenses, use the **no** form of this command.

```
dbconn license license-key [connections licensed-connections] [expiration-date yyyymmdd]  
no dbconn license
```

### Syntax Description

<i>license-key</i>	License key obtained from your Cisco representative. The license determines the maximum number of CICS conversations or connections to DB2 via a Cisco router configured with CTRC. The license key is a 32-character hexadecimal string. It is not case sensitive, but it is linked to a specific router. When you request your license key, you must supply your Cisco representative with the SNA Switching Services cpname for the router. Use the <b>show config   include cpname</b> command to view the cpname.
<b>connections</b> <i>licensed-connections</i>	(Optional) Number of licensed connections. If omitted, the number of licensed connections must be unlimited.
<b>expiration-date</b> <i>yyyymmdd</i>	(Optional) Date when a temporary license key expires, in <i>yyyymmdd</i> format, where <i>yyyy</i> is the year expressed in four digits, <i>mm</i> is the month expressed in two digits, and <i>dd</i> is the date expressed in two digits. If omitted, the license key must be permanent.

### Defaults

If the number of licensed connections is not specified, the license key must be generated for an unlimited number of licensed connections. If the expiration date is not specified, the license key must be permanent.

### Command Modes

Global configuration

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

## Usage Guidelines

Licenses are required for all CTRC installations. For testing and evaluation purposes, unlicensed CTRC installations allow you to establish two connections to DB2 or CICS. Licenses are shared between CTRC connections to DB2 and CTRC connections to CICS. If you have already configured licenses for CTRC connections to CICS, you do not need to configure licenses for DB2, and vice versa.

## Examples

The following example shows a permanent CTRC license for unlimited connections being configured:

```
dbconn license 8C09A496420BAF020BFF74B8A2FF24D2
```

## Related Commands

Command	Description
<b>show dbconn license</b>	Displays status information about licensed CTRC connections. Functionally identical to <b>show txconn license</b> command.
<b>show snasw node</b>	Displays SNA Switching Services information including the value of the cpname, which is required for generation of a CTRC license key.
<b>show txconn license</b>	Displays status information about licensed CTRC connections. Functionally identical to <b>show dbconn license</b> command.
<b>txconn license</b>	Configures CTRC license for connections to CICS or DB2. Functionally identical to <b>dbconn license</b> command.

## dbconn pem

To configure password expiration management (PEM) support for connections to DB2, use the **dbconn pem** global configuration command. To remove PEM support, use the **no** form of this command.

```
dbconn pem server server-name rlu rlu-name mode mode-name [tpname tp-name]  
no dbconn pem server server-name
```

### Syntax Description

<b>server</b> <i>server-name</i>	Name of the CTRC server whose password management you are configuring.
<b>rlu</b> <i>rlu-name</i>	Host remote LU name the server connects to when performing password management. This RLU ordinarily differs from the RLU values used in <b>dbconn server</b> or <b>txconn destination</b> commands. It may or may not be fully qualified.
<b>mode</b> <i>mode-name</i>	APPC stack mode the server uses when performing password management.
<b>tpname</b> <i>tp-name</i>	(Optional) Name of the PEM transaction program on the host (the APPC Signon transaction program, an architected APPC TP). The default value is \x06301 (0x06F3F0F1 in EBCDIC).

### Defaults

If *tp-name* is not specified, the default value is \x06301 (0x06F3F0F1 in EBCDIC).

### Command Modes

Global configuration

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example shows the configuration of PEM support:

```
dbconn pem GAZELLE rlu STARW.GAZELLE mode #INTER
```

### Related Commands

Command	Description
<b>show dbconn server</b>	Displays information about CTRC servers for DB2 communications

## dbconn ping

To determine whether or not CTRC servers are successfully connecting to DB2 host databases, use the **dbconn ping** EXEC command.

```
dbconn ping server-name [user user-id] [password password] [rdbname rdbname]
```

### Syntax Description

<i>server-name</i>	Name of the CTRC server for DB2 communications.
<b>user</b> <i>user-id</i>	(Optional) User ID used to connect to the DB2 system.
<b>password</b> <i>password</i>	(Optional) Password used to connect to the DB2 system.
<b>rdbname</b> <i>rdbname</i>	(Optional) Name of the relational database to be contacted.

### Defaults

If no user ID is specified, the ping connection is made without APPC security. The DB2 system's security settings determine whether or not the ping can succeed.

If no *rdbname* is specified, the relational database name configured for the server is used.

### Command Modes

EXEC

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Usage Guidelines

This command can be used for CTRC servers that communicate with DB2 over either SNA or TCP/IP (dbconn servers and dbconn tcpservers).

### Examples

The following example shows an attempt to verify the connection to the default database for the CTRC server canopus:

```
dbconn ping canopus

RDB named NEVADA on database server CANOPUS successfully contacted!
Database server product id is QSQ03020, DB2 for OS/400 V3R2
Elapsed time was 00:00:00
```

Related Commands

<b>Command</b>	<b>Description</b>
<b>show dbconn server</b>	Displays information about a CTRC server that communicates with DB2.
<b>ping sna</b>	Tests connectivity for SNA Switching Services by initiating an APPC session with a specified destination.

## dbconn server

To configure a CTRC server for communications to DB2, use the **dbconn server** global configuration command. To disable the server and remove its configuration, use the **no** form of this command.

```
dbconn server server-name [ipaddress ip-address] [port port-number] [rdbname rdbname]
[rlu remote-lu] [mode mode] [tpname tp-name] [idle-timeout minutes]
[window-size bytes] [wlm {off | on}]
```

```
no dbconn server server-name
```

### Syntax Description

<i>server-name</i>	Name of the CTRC server. Server names are user-defined strings up to 16 characters.
<b>ipaddress</b> <i>ip-address</i>	(Optional) IP address used by the CTRC server to receive a connection requesting DB2 communications. When a connection arrives, this IP address is used for matching and selecting the server from multiple configured servers. If you do not specify an IP address, the current server can handle DB2 connectivity requests sent to any IP address on the local router.
<b>port</b> <i>port-number</i>	(Optional) Port used to listen for connections requesting DB2 communications. If no port is specified, the default is 446.
<b>rdbname</b> <i>rdbname</i>	(Optional) DB2 remote database name on the host. When a connection arrives, this name is used to identify and select the appropriate server from multiple configured servers. The string is used to match the RDB name sent by the client in the DRDA data stream at connect time. The default RDB name is an asterisk (*), which indicates that this CTRC server serves any remote database.
<b>rlu</b> <i>remote-lu</i>	(Optional) APPC remote LU used to allocate the connection to the DB2 system. An example is NETA.S103B345. If no remote LU is specified, the default is the configured server name which is set to uppercase and truncated to eight characters. An RLU need not be qualified with a NETID. If you omit the NET ID, the NET ID of the router's SNA Switching Services control point is used to fully qualify the LU name.
<b>mode</b> <i>mode</i>	(Optional) APPC mode used to allocate the conversation to the DB2 system. If no mode is specified, the default is #INTER. Performance might improve if you choose a mode such as IBMRDB. If you specify a mode that does not already exist, CTRC will create it.
<b>tpname</b> <i>tp-name</i>	(Optional) APPC remote transaction program name used to allocate the conversation to the DB2 system. If no TP name is specified the default is the architected DRDA TP name \x076DB.

<b>idle-timeout</b> <i>idle-timeout</i>	(Optional) Time, in minutes, to wait for an idle client. If there is no activity from the client for this amount of time, the connection is forcibly broken. The time spent in waiting for a response from the DB2 system is not counted, only idle time in between client requests is counted. The maximum is 1440 minutes (24 hours). If no idle timeout is specified, the default is 0 (zero) for no timeout.
<b>window-size</b> <i>bytes</i>	(Optional) TCP/IP receive window size. If no window size is specified, the default is 4 K (4096 bytes).
<b>wlm {off   on}</b>	(Optional) Enables disables Workload Manager load balancing. The default is “inactive-enabled,” otherwise on.

## Defaults

If you do not specify an IP address, the current server can handle DB2 connectivity requests sent to any IP address on the local router.

If you do not specify a port number, the current server uses the default value of 446.

If you do not specify an rdbname, the server is configured to serve any remote database.

If you do not specify a remote LU, the default is the configured server name which is set to uppercase and truncated to eight characters. An RLU need not be qualified with a NETID. If you omit the NET ID, the NET ID of the router’s SNA Switching Services control point is used to fully qualify the LU name.

If you do not specify an APPC mode, the default value is #INTER. If you specify a mode that does not already exist, CTRC will create it.

If you do not specify an APPC transaction program name, the default value is the architected DRDA TP name \x076DB.

If you do not specify an idle timeout, client connections can continue regardless of how long they have been idle.

If you do not specify a TCP receive window size, the default value is 4 K (4096 bytes).

## Command Modes

Global configuration

## Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

## Usage Guidelines

You can configure more than one CTRC server per router for communications with DB2 or CICS. There is no limit on the number of CTRC servers. For each DB2 database system in your network, you can configure a CTRC server in the router configuration. CTRC dbconn servers that are configured on the same router can share a port. CTRC txconn servers should use a different port.

## Examples

The following example shows the configuration of a CTRC server that will communicate with DB2:

```
dbconn server coffee rdbname COFFEE rlu PARIS.COFFEE idle-timeout 20 window-size 16
```

## Related Commands

Command	Description
<b>clear dbconn connection</b>	Breaks a CTRC connection to DB2.
<b>dbconn ping</b>	Tests CTRC connectivity to DB2.
<b>show dbconn server</b>	Displays information about a CTRC server for DB2 communications.
<b>show dbconn ports</b>	Displays information about ports used for CTRC server communications to DB2.
<b>show snasw mode</b>	Displays information about SNA Switching Services modes.

## dbconn tcpserver

To configure a CTRC server to communicate with IP-enabled DB2 databases, use the **dbconn tcpserver** global configuration command. To disable such a server and remove its configuration, use the **no** form of this command.

```
dbconn tcpserver server-name [ip ip-address] [port port-num] [rdbname rdbname]
remote-ip remote-ip-address [remote-port remote-port] [idle-timeout time-in-minutes]
[window-size size-in-bytes]

no dbconn tcpserver server-name
```

### Syntax Description

<i>server-name</i>	Name of the CTRC server being configured for TCP passthrough communications with DB2.
<b>ip</b> <i>ip-address</i>	(Optional) IP address for the CTRC tcpserver process being configured. If not specified, the tcpserver receives client requests on all IP addresses configured for the router.
<b>port</b> <i>port-num</i>	(Optional) Port the tcpserver listens on for client requests. The default value is 446. A dbconn server and a dbconn tcpserver can share the same port.
<b>rdbname</b> <i>rdbname</i>	(Optional) DB2 remote database name on the host. When a connection arrives, this name is used to identify and select the appropriate tcpserver from multiple configured tcpserver. The string is used to match the RDB name sent by the client in the DRDA data stream at connect time. The default RDB name is an asterisk (*) which indicates that this CTRC tcpserver serves any remote database.
<b>remote-ip</b> <i>remote-ip-address</i>	IP address for the host where DB2 resides.
<b>remote-port</b> <i>remote-port</i>	(Optional) Host port that listens for tcpserver communications from the router. The default value is 446.
<b>idle-timeout</b> <i>time-in-minutes</i>	(Optional) Time in minutes to wait for an idle client. If there is no activity from the client for this amount of time, the connection is forcibly broken. The time spent in waiting for a response from the DB2 system is not counted, only idle time in between client requests is counted. The maximum time is 1440 minutes (24 hours). If no idle timeout is specified, the default is 0 (zero) for no timeout.
<b>window-size</b> <i>size-in-bytes</i>	(Optional) This value is used for the TCP/IP receive window size. If no window size is specified, the default is 4096 bytes (4 K).

## Defaults

If you do not specify an IP address for the tcpserver, it can receive requests on any IP address configured for the router.

If you do not specify a port for the tcpserver, the default port is 446.

If you do not specify a remote database name for the DB2 system, the tcpserver can communicate with any rdbname.

If you do not specify a port for the remote DB2 system, the tcpserver uses the default value of 446.

If you do not specify an idle timeout period, the default value is zero for no timeout.

If you do not specify a TCP/IP receive window size, the default value is 4096 bytes (4 K).

## Command Modes

Global configuration

## Command History

Release	Modification
12.0(5)XN	This command was introduced.

## Usage Guidelines

Configure a separate tcpserver for each DB2 system IP address. A dbconn server and a dbconn tcpserver can share the same port.

## Examples

The following example shows the configuration of a tcpserver:

```
Router(config)#dbconn tcpserver TCPSERVER ip 0.0.0.0 port 446 rdbname GAZELLE
remote-ip 123.456.1.2 remote-port 446
```

## Related Commands

Command	Description
<b>dbconn ping</b>	Tests connectivity between a CTRC server (or tcpserver) and a specified DB2 system.
<b>show dbconn server</b>	Displays information about CTRC servers for DB2 communications, whether those servers use TCP/IP or SNA to communicate with DB2 databases.

## show dbconn connection

To display the status of CTRC connections to DB2, use the **show dbconn connection EXEC** command.

**show dbconn connection** [*connection-id* | **server** *server-name* | **user** *user-id* | **rdbname** *rdb-name*]

### Syntax Description

<i>connection-id</i>	(Optional) Displays the status of a specified connection.
<b>server</b> <i>server-name</i>	(Optional) Displays connection information for specified server.
<b>user</b> <i>user-id</i>	(Optional) Displays connections for specified user ID.
<b>rdbname</b> <i>rdb-name</i>	(Optional) Displays connections for specified RDB name.

### Defaults

If you do not specify any arguments, this command displays information for all CTRC connections to DB2 on the current router.

### Command Modes

EXEC

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Examples

The following is sample output from the **show dbconn connection** command:

```
Router>show dbconn connection

ID          Server      Userid      ClientIPAddress  Connect  Idle
6127E428   SERVERA     ALLIE      198.999.989.36   00:01:26 00:01:12
6127D34C   SERVERB     BRIAN      198.999.989.84   00:00:48 00:00:41
```

The following is sample output from the **show dbconn connection** command for a specified connection:

```
Router>show dbconn connection 62586780
      connection id: 62586780
      connection state: active
          server: QAMELTCP
          rdbname: MELBA
          userid: QAUSER
          client name: QAMELBA
      local ip-address: 198.147.235.2
          local port: 500
      client ip-address: 198.147.235.88
          client port: 4258
          connect time: 00:53:27
          idle time: 00:00:04 (client)
      bytes received from client: 30478
          bytes received from host: 318222
          client: licensed StarSQL
```

Table 1 describes significant fields shown in the display.

**Table 1** Show Database Connection Field Descriptions

Field	Description
connection id	Identification number of the connection made by a DRDA client to the CTRC server.
connection state	Status of the connection made by a DRDA client.
server	Name of the CTRC server.
rdbname	Name of the relational database on the IBM system.
userid	Userid of the user connected through a port to the CTRC server.
client name	Name of the client system.
local ip-address	IP address of the CTRC server in the router to which the client connects.
local port	Port in the CTRC server through which the client connects.
client ip-address	IP address of the client connected to the CTRC server.
client port	Port used by the client to connect to the CTRC server.
connect time	Time when connection was made by the client to the CTRC server.
idle time	Amount of time that the active client connection has been idle.
bytes received from client	Number of bytes the router has received from the client via the specified connection.
bytes received from host	Number of bytes the router has received from the host via the specified connection
client	Indicates whether the client connection uses a licensed StarSQL ODBC-DRDA driver or another DRDA driver.

Related Commands

<b>Command</b>	<b>Description</b>
<b>show dbconn license</b>	Displays information about CTRC licensing. Functionally identical to <b>show txconn license</b> command.
<b>show dbconn ports</b>	Displays information about CTRC ports used for DB2 communications.
<b>show dbconn server</b>	Displays information about CTRC servers used for DB2 communications.

## show dbconn license

To display the status of CTRC licenses for DB2 communications, use the **show dbconn license** EXEC command.

**show dbconn license**

### Syntax Description

This command has no arguments or keywords

### Defaults

None.

### Command Modes

EXEC

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Usage Guidelines

This command produces the same results as **show txconn license** because licenses are shared.

### Examples

The following is sample output from the **show dbconn license** command for a router that is using a temporary license key:

```
Router>show dbconn license
CTRRC is licensed for 1000 connections, no licensed connections in use
Expires on 1-1-2001.
```

### Related Commands

Command	Description
<b>dbconn license</b>	Configures CTRC licenses for DB2 or CICS communications. Functionally identical to the <b>txconn license</b> command.
<b>show txconn license</b>	Displays license information for CTRC. Functionally identical to the <b>show dbconn license</b> command.
<b>txconn license</b>	Configures CTRC licenses for CICS or DB2 communications. Functionally identical to the <b>dbconn license</b> command.

## show dbconn ports

To display information about ports that CTRC is using for communications to DB2, use the **show dbconn ports EXEC** command.

**show dbconn ports**

### Syntax Description

This command has no arguments or keywords.

### Defaults

None.

### Command Modes

EXEC

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Examples

The following is sample output from the **show dbconn ports** command:

```
Router>show dbconn ports

Port  State
446   listening
447   listening
```

Table 2 describes significant fields shown in the display.

**Table 2 Show Database Connection Ports Field Descriptions**

Field	Description
Port	Port number.
State	Listening or disabled status.

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>show dbconn connection</b>	Displays connection information about CTRC communications with DB2.
<b>show dbconn license</b>	Displays license information for CTRC communications with DB2.
<b>show dbconn server</b>	Displays information about CTRC servers used for DB2 communications.

## show dbconn server

To display information about CTRC servers configured for DB2 communications, use the **show dbconn server EXEC** command.

**show dbconn server** [*server-name*]

### Syntax Description

*server-name* (Optional) Specific server for which information should be displayed. When omitted, this command displays information for all CTRC servers configured for DB2 communications on the current router.

### Defaults

If no server name is specified, this command displays information for all CTRC servers configured for DB2 communications on the current router.

### Command Modes

EXEC

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Examples

The following is sample output from the **show dbconn server** command:

```
Router>show dbconn server
Server      Port    IPAddress      RDBName      State      NumConn
SERVERA    446    0.0.0.0       MATTY        enabled    0
SERVERB    446    0.0.0.0       SCU_DSNM     enabled    4
SERVERC    446    0.0.0.0       DSN4         enabled    0
SERVERD    446    0.0.0.0       MKTG         enabled    0
SERVERE    446    0.0.0.0       ABBY         enabled    6
SERVERF    446    0.0.0.0       DB2510       enabled    0
SERVERG    446    0.0.0.0       ELLE         enabled    3
SERVERH    446    0.0.0.0       SUNSET       enabled    0
SERVERI    446    0.0.0.0       NELL         enabled    1
SERVERJ    446    198.989.999.32 SAMPLE       enabled    0
SERVERK    446    0.0.0.0       DB2410       enabled    0
SERVERL    446    0.0.0.0       SQLDS        enabled    0
SERVERM    446    0.0.0.0       STELLA       enabled    1
SERVERN    446    10.10.19.4    OAK          enabled    2
SERVERO    447    0.0.0.0       DB2510       enabled    0
```

The following is sample output from the **show dbconn server** command where the database server BUDDY is specified:

```
Router>show dbconn server BUDDY
      server: BUDDY
      server state: enabled (accepting connections)
      ip-address: 0.0.0.0
      port: 446
      rdbname: DB2510
      connection type: SNA
        rlu: STARW.DSNV50
        mode: IBMRDB
        tpname: \x076DB
      idle-timeout: 0 (none)
      window-size: 4096 bytes
      database server name: (unknown)
      database product id: (unknown)
      PEM: not configured
      number of connections: 0
      RDB server: active
      WLM: inactive-enabled
```

Table 3 describes significant fields shown in the display.

**Table 3 Show Database Connection Server Field Descriptions**

Field	Description
server	CTRC server name.
server state	Current state of the server (enabled or disabled).
ip-address	IP address of the CTRC server in the router to which the client connects.
port	Port number through which the CTRC server accepts a client connection.
rdbname	Name of the remote database accessed by the CTRC server.
connection type	Type of connection established between the CTRC router and the DB2 host. Legal values are <b>SNA</b> and <b>TCP/IP</b> .
rlu	Remote SNA LU used when connecting to the database server.
mode	SNA mode used when connecting to the database.
tpname	SNA transaction program name used for DRDA server on the database system.
idle-timeout	Maximum length of time allowed for inactive connections to the CTRC server.
window-size	TCP receive window size.
database server name	System name returned by the database server. Field shows <i>none</i> until first contact.
database product id	Database product ID. Field shows <i>none</i> until first contact.
PEM rlu	The host remote LU name the server will connect to when performing password management.
PEM mode	The APPC mode the server will use when performing password management.
PEM tpname	The name of the PEM transaction program on the host (the APPC Signon transaction program, an architected APPC TP).
number of connections	Number of all ODBC clients currently connected to the CTRC server.
RDB server	Status of the host database. Legal values are <b>active</b> and <b>unreachable</b> .
wlm	Status of Workload Manager: not enabled, inactive - enabled, active - enabled

Related Commands

<b>Command</b>	<b>Description</b>
<b>show dbconn connection</b>	Displays information about CTRC connections to DB2.
<b>show dbconn ports</b>	Displays information about CTRC ports used for DB2 communications.

## show dbconn statistics

To show all CTRC statistics concerning communications with DB2, use the **show dbconn statistics** privileged EXEC command.

```
show dbconn statistics { chains | clientturnaround | connectionsdown | connectionsup | dump |
  hostreceived | hostresponse | hostsent | latency | maxconnections }
```

### Syntax Description

<b>chains</b>	Shows statistics for number of chains created.
<b>clientturnaround</b>	Shows statistics for average time from receiving a DB2 client communication to sending that client a response.
<b>connectionsdown</b>	Shows statistics for number of connections finished between CTRC and DB2
<b>connectionsup</b>	Shows statistics for number of connections created between CTRC and DB2
<b>dump</b>	Shows a complete statistics dump
<b>hostreceived</b>	Shows statistics for number of bytes received from DB2 hosts.
<b>hostresponse</b>	Shows statistics for average DB2 host response time.
<b>hostsent</b>	Shows statistics for number of bytes sent to DB2 hosts.
<b>latency</b>	Shows average server latency time
<b>maxconnections</b>	Shows statistics for maximum number of concurrent connections to CICS clients.

### Defaults

None.

### Command Modes

Privileged EXEC

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example clears the stored statistics for the number of bytes the current router has received from DB2 clients:

```
clear dbconn statistics clientreceived
```

**show dbconn statistics**

---

Related Commands

<b>Command</b>	<b>Description</b>
<b>show dbconn statistics</b>	Displays statistics related to CTRC communications with DB2.

## show dbconn wlm

To display information about CTRC servers configured for DB2 communications using Workload Manager, use the **show dbconn wlm** EXEC command.

**show dbconn wlm** [*server-name*]

### Syntax Description

*server-name* (Optional) Specific server for which information should be displayed. When omitted, this command displays information for all CTRC servers configured for DB2 communications on the current router.

### Defaults

If no server name is specified, this command displays information for all CTRC servers configured for DB2 communications with Workload Manager on the current router.

### Command Modes

EXEC

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Examples

The following is sample output from the **show dbconn wlm** command:

IP	Port	Weight	Hits
198.147.235.2	500	1	0
198.147.235.2	501	2	0
198.147.235.2	502	3	0

### Related Commands

Command	Description
<b>show dbconn connection</b>	Displays information about CTRC connections to DB2.
<b>show dbconn ports</b>	Displays information about CTRC ports used for DB2 communications.

## show txconn connection

To display a list of all of the router's CTRC connections to CICS clients, a list of a specified CTRC server's connections to CICS clients, or detailed status information for a specific CTRC connection to a CICS client, use the **show txconn connection** EXEC command.

**show txconn connection** [*connection-id* | **server** *server-name*]

### Syntax Description

*connection-id* (Optional) Specifies a CTRC connection to a CICS client for which to display detailed status information.

**server** *server-name* (Optional) Specifies a CTRC server for which to list connections to CICS clients.

### Defaults

If neither *connection-id* nor *server-name* are specified, a list of all of the current router's CTRC connections to CICS clients is displayed.

### Command Modes

EXEC

### Command History

Release	Modification
12.0(5)XN	This command was introduced

### Examples

The following example displays information about a CICS client connections for the current router:

```
Router>show txconn connection
```

Server	ConnID	State	#Transact.	IP Address	Port	Bytes
CICSB	6241464C	receiving	2	198.147.235.88	1365	2.89K
AMELIA	625443BC	receiving	0	198.147.235.88	1371	15.60K

The following example displays information about a specified CICS client connection:

```
Router>show txconn connection 6241464C
connection: 6241464C
server: CICSB
state: receiving
transactions: 2
ip address: 198.147.235.88
port: 1365
total transactions: 20
connect timestamp: 06:04:26
idle time: 06:17:34
total bytes received: 2963
total bytes sent: 28121 (2.89K)
idle timeout: 0 (none)
```

Table 4 describes significant fields shown in the display.

**Table 4 Show Txconn Connection Field Descriptions**

Field	Description
connection	Unique identifier for the CICS client connection.
server	CTRC txconn server process that is handling the connection.
state	Status of the connection. Possible values are: <ul style="list-style-type: none"> <li>• <b>closing</b> indicates that the connection is in the process of closing.</li> <li>• <b>halt</b> indicates that the connection has been manually cleared and is in the process of releasing resources.</li> <li>• <b>receiving</b> indicates that the connection is receiving data from the client.</li> <li>• <b>reset</b> indicates that the connection has just opened or just closed.</li> </ul>
transaction	Number of CICS transactions currently in progress for the connection.
ip address	IP address of the CICS client that is using the connection.
port	Port of the CICS client that is using the connection.
total transactions	Total number of CICS transactions performed using the connection.
connect timestamp	Amount of time elapsed since the connection was first established. Values of less than 24 hours are displayed in hours, minutes, and seconds. Longer periods are displayed in days and hours.
idle time	Amount of time that the connection has been idle.
total bytes received	Number of bytes received from the CICS client via this connection.
total bytes sent	Number of bytes sent to the CICS client via this connection.
idle timeout	Number of minutes after which the connection will be automatically closed if there is no activity. A value of zero (0) indicates that the connection will not be closed for lack of activity.

## Related Commands

Command	Description
<b>clear txconn connection</b>	Clears a specific CTRC connection to a CICS client, along with all that connection's transactions.
<b>show txconn transaction</b>	Displays information about CICS transactions.

## show txconn destination

To display a list of all of the current router's CICS destinations for CTRC, or to display detailed status information for a specified CTRC CICS destination, use the **show txconn destination EXEC** command.

**show txconn destination** [*destination-name*]

### Syntax Description

*destination-name* (Optional) CTRC destination for which to display detailed status information. A destination is defined by a unique remote LU and mode pair.

### Defaults

If *destination-name* is omitted, displays a list of all CTRC destinations for the current router.

### Command Modes

EXEC

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following examples show the CTRC destinations for CICS communications that are available on the current router and information specifically about the destination GEN:

```
Router>show txconn destination
Name          Remote LU      Mode           Hits
-----
CICSB         CICSB          IBMRDB         31
GEN           CICSB          IBMRDB         50
              CICSC          IBMRDB         51
GUAVA         GUAVA          IBMRDB         0
CICSC         CICSC          IBMRDB         7

Router>show txconn destination GEN
Name          Remote LU      Mode           Hits
-----
GEN           CICSB          IBMRDB         50
              CICSC          IBMRDB         51
```

The HITS column displays the number of times the router has routed transactions or pings to each destination since the last time the router was started up.

## Related Commands

<b>Command</b>	<b>Description</b>
<b>txconn destination</b>	Configures CICS destinations for CTRC.

## show txconn license

To show the status of licenses used for CTRC, use the **show txconn license EXEC** command.

**show txconn license**

### Syntax Description

This command has no arguments or keywords.

### Defaults

None.

### Command Modes

EXEC

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Usage Guidelines

This command produces the same result as the **show dbconn license** command because CTRC licenses are shared between DB2 connections and CICS conversations.

### Examples

The following example shows license information for CICS communications being displayed on the current router:

```
Router>show txconn license
CTRRC is licensed for 4990 connections, 20 licensed connections in use
This is a permanent license
```

### Related Commands

Command	Description
<b>dbconn license</b>	Configures licenses for CTRC communications to DB2 or CICS. Functionally identical to the <b>txconn license</b> command
<b>show dbconn license</b>	Displays license information for CTRC. Functionally identical to the <b>show txconn license</b> command.
<b>txconn license</b>	Configures licenses for CTRC communications to CICS or DB2. Functionally identical to the <b>dbconn license</b> command

## show txconn route

To display a list of all CTRC routes defined for specified CICS transaction IDs, or to display a particular CTRC server's routes to CICS, use the **show txconn route EXEC** command.

```
show txconn route [server server-name]
```

### Syntax Description

**server** *server-name* (Optional) Server for which you wish to display routing information. If not specified, a list of all CICS communications routes for CTRC servers on the current router is displayed.

### Defaults

If *server-name* is not specified, a list of all CICS communications routes for CTRC servers on the current router is displayed.

### Command Modes

EXEC

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following examples show information about all the CTRC routes to CICS that are available on the current router and information about the route for CTRC server CICSB&C. A <default> in the SERVER column indicates a global route that is used by all txconn servers on the router. A <default> in the TRANID column indicates the default route for the listed txconn server.

```
Router>show txconn route
Server          TranID          Destination
-----
CICSC           <default>       CICSC
CICSB           <default>       CICSB
CICSB&C         <default>       GEN
GUAVA           <default>       GUAVA
<default>       CPMI            CICSC
CICSB           CPMI            CICSB

Router>show txconn route CICSB&C
Server          TranID          Destination
-----
CICSB&C         <default>       GEN
```

### Related Commands

Command	Description
<b>txconn route</b>	Configures CTRC routes to CICS for specified transaction IDs.

## show txconn server

To display information about the current router's CTRC servers for CICS communications, or to display detailed status information for a single CTRC server, use the **show txconn server** EXEC command.

**show txconn server** [*server-name*]

### Syntax Description

*server-name* (Optional) CTRC server for which to display detailed status information. When omitted, a list of CTRC servers is displayed.

### Defaults

When *server-name* is not specified, a list of the current router's CTRC servers that communicate with CICS is displayed.

### Command Modes

EXEC

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following examples show summary information about the CTRC servers for CICS communications that are available on the current router and more detailed information about the CTRC server CICSB:

```
Router>show txconn server
Server      Port  IP Address  Dest      State    NumConn
-----
AMELIA     1436  0.0.0.0    AMELIA   enabled  0
CICSB     1444  0.0.0.0    CICSB   enabled  0
CICSC     1434  0.0.0.0    CICSC   enabled  0
TEST      1446  0.0.0.0    CICSC   enabled  0
```

```
Router>show txconn server TEST
      server: TEST
      destination: CICSB
      server state: enabled (accepting connections)
      ip address: 0.0.0.0
      port: 1437
      client timeout: 0 (none)
      host timeout: 0 (none)
      window size: 4096 bytes
      fold program name: on
      ccsid: 273
      number of connections: 1
      number of transactions: 2
```

Table 5 describes significant fields shown in the display.

**Table 5 Show Txconn Server Field Descriptions**

Field	Description
server	Name of the txconn server.
destination	Default destination for the server.
server state	Status of the server process. Possible values are: <ul style="list-style-type: none"> <li>• <b>disabled (unable to accept connections)</b> indicates that CICS client connections will be rejected.</li> <li>• <b>enabled (accepting connections)</b> indicates that CICS clients can connect to the server.</li> </ul>
ip address	TCP/IP address for which the server accepts connections. A value of 0.0.0.0 indicates that the server accepts connections for any IP address that is configured on the router.
port	TCP/IP port number on which the server listens.
client timeout	Number of minutes a CICS client can remain idle before it is automatically disconnected. A value of zero (0) indicates that the server does not disconnect clients for inactivity.
host timeout	Number of minutes a CICS host may remain idle before it is automatically disconnected. A value of zero (0) indicates that the server does not disconnect hosts for inactivity.
window size	TCP/IP receive window size.
fold	CTRC folds the CICS program name to upper case. Options are off on. Default is on.
number of connections	Number of currently active CICS client connections to the server.
number of transactions	Number of currently active CICS transactions being handled by the server.

## Related Commands

Command	Description
<b>txconn server</b>	Configures CTRC servers for CICS communications.

## show txconn statistics

To display information about the current router's CTRC communications with CICS, use the **show txconn statistics** EXEC command.

```
show txconn statistics [kind histogram | kind summary] name { activeconnections |  
  activetransactions | allocatetime | clientreceived | clientsent | clientturnaround | dump |  
  hostreceived | hostresponse | hostsent | latency | maxconnections | maxtransactions |  
  totalconnections | totaltransactions }
```

### Syntax Description

**kind histogram | kind summary**

(Optional) Desired format for the statistics to be displayed. Legal values are as follows:

- **histogram** displays the named statistic in a graphical format.
- **summary** displays the named statistic in a tabular format.

If the kind of statistics display to show is not specified, **summary** is used. Statistics are shown for the clock hour in progress and the preceding 24 full hours.

**name** { **activeconnections** |  
**activetransactions** |  
**allocatetime** | **clientreceived** |  
**clientsent** | **clientturnaround** |  
**dump** | **hostreceived** |  
**hostresponse** | **hostsent** |  
**latency** | **maxconnections** |  
**maxtransactions** |  
**totalconnections** |  
**totaltransactions** }

Specific statistic to display. Legal values are as follows:

- **activeconnections** displays the number of connections to CICS clients currently active.
- **activetransactions** displays the number of CICS transactions currently being processed.
- **allocatetime** displays the average time in seconds spent waiting for APPC allocate operation to complete.
- **clientreceived** displays the total number of bytes received from CICS clients during the indicated time period.
- **clientsent** displays the total number of bytes sent to CICS clients during the indicated time period.
- **clientturnaround** displays the average time in seconds from receiving a request from a CICS client to sending that client a response during the indicated time period.
- **dump** displays a compact statistics summary for the last 24 hours, including all named txconn statistics except activeconnections and activetransactions.
- **hostreceived** displays the total number of bytes received from hosts for CICS connections during the indicated time period.
- **hostresponse** displays the average host response time in seconds for CICS connections during the indicated time period.
- **hostsent** displays the total number of bytes sent to hosts for CICS connections during the indicated time period.
- **latency** displays the average amount of time in seconds used by the txconn server per CICS client request (clientturnaround minus hostresponse).
- **maxconnections** displays the maximum number of concurrent connections to CICS clients during the indicated time period.
- **maxtransactions** displays the maximum number of concurrent CICS transactions during the indicated time period.
- **totalconnections** displays the total number of connections to CICS clients used during the indicated time period.
- **totaltransactions** displays the total number of CICS transactions processed during the indicated time period.

## Defaults

If the kind of statistics display is not specified, **summary** is used.

## Command Modes

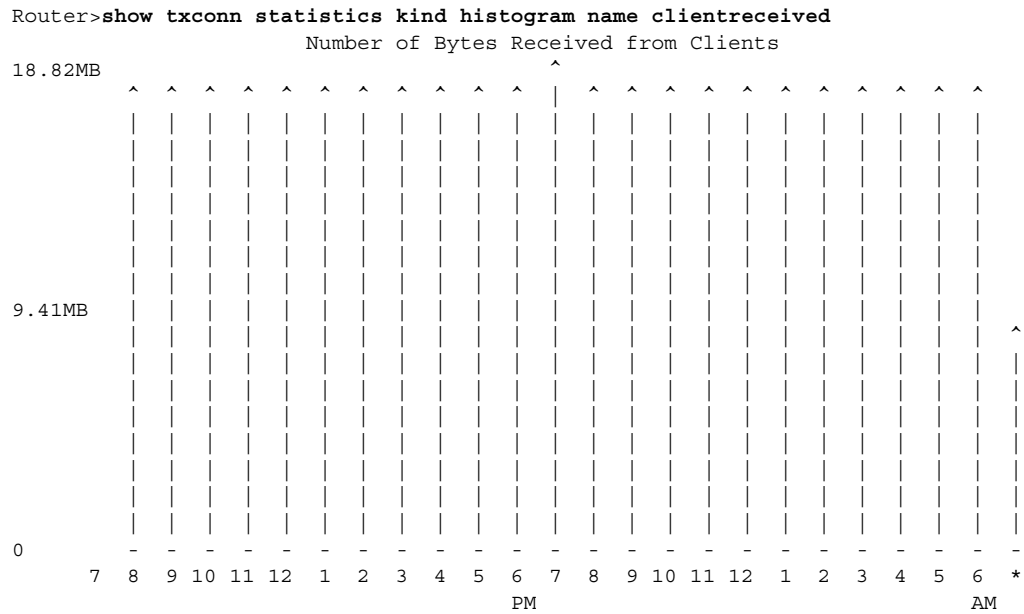
EXEC

## Command History

Release	Modification
12.0(5)XN	This command was introduced.

Examples

The following examples show histogram and summary displays of the **clientreceived** statistic.



```
Router>show txconn statistics name clientreceived
Number of Bytes Received from Clients
```

hour	yesterday	yesterday	today
----	-----AM-----	-----PM-----	-----AM-----
12:00-12:59		19728481	19727299
01:00-01:59		19732711	19727299
02:00-02:59		19722903	19727382
03:00-03:59		19728398	19731695
04:00-04:59		19729497	19726200
05:00-05:59		19730596	19733893
06:00-06:59		19722986	19708616
07:00-07:59	0	19734992	8736034 *
08:00-08:59	19726283	19725101	
09:00-09:59	19725101	19728398	
10:00-10:59	19726283	19727382	
11:00-11:59	19729497	19730596	

24-hour total: 453731589 (excludes hour in progress \*)

## Related Commands

<b>Command</b>	<b>Description</b>
<b>show txconn connection</b>	Displays information about CTRC connections to CICS clients.
<b>show txconn destination</b>	Displays information about CICS destinations for CTRC.
<b>show txconn license</b>	Displays license information for CTRC. Functionally identical to the <b>show dbconn license</b> command.
<b>show txconn route</b>	Displays information about CTRC routes to CICS.
<b>show txconn server</b>	Displays information about CTRC servers that communicate with CICS.
<b>show txconn transaction</b>	Displays information about CICS transactions.

## show txconn transaction

To display a list of all the current router's CTRC transactions with CICS, a specified CTRC server's transactions, or a specified CICS client connection's transactions, use the **show txconn transaction EXEC** command.

**show txconn transaction** [**server** *server-name* | **connection** *connection-id* | *transaction-id*]

### Syntax Description

<b>server</b> <i>server-name</i>	(Optional) Specifies a CTRC server for which to display a list of transactions.
<b>connection</b> <i>connection-id</i>	(Optional) Specifies a CICS client connection to CTRC for which to display a list of transactions.
<i>transaction-id</i>	(Optional) Specifies an individual transaction for which to display detailed status information.

### Defaults

If no arguments are specified, all the CICS transactions for the current router are listed.

### Command Modes

EXEC

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following two examples display general information about CICS transactions being handled by the current router and detailed information about a specified CICS transaction, respectively:

```
Router>show txconn transaction
```

Transaction ID	Server	Conn ID	State	TP Name	User ID
6246ECD8	QATIM	62494598	receiving	CPMI	QAUSER
62476188	QATIM	62494598	receiving	CPMI	QAUSER
623130D4	QATIM	62494598	receiving	CPMI	QAUSER
6229E88C	QATIM	62494598	receiving	CPMI	QAUSER

```

Router>show txconn transaction 6246ECD8
transaction: 6246ECD8
server: QATIM
connection id: 62494598
state: receiving
tp name: CPMI
user id: QAUSER
session RU address (OAF+DAF): 6
idle time: 1788
--- Transaction Totals ---
number of transactions executed: 1
number of bytes received from client: 1099
number of bytes received from host: 0

```

Table 6 describes significant fields shown in the display.

**Table 6 Show Txconn Transaction Field Descriptions**

Field	Description
transaction	Unique identifier for the transaction.
server	CTRC txconn server process that is handling the transaction.
connection id	Unique identifier for the CICS client connection associated with the transaction.
state	Status of the transaction. Possible values are: <ul style="list-style-type: none"> <li>• <b>closing</b> indicates that the SNA session is in the process of closing.</li> <li>• <b>exception</b> indicates that an error has occurred. An error indication will be sent to the client and the host session will be terminated.</li> <li>• <b>exc. resp.</b> indicates that the router has sent an error indication to the client.</li> <li>• <b>opening</b> indicates that the SNA session is about to open.</li> <li>• <b>parsing FMH</b> indicates that the SNA session has received the first portion of an FM header.</li> <li>• <b>parsing FMH5</b> indicates that CTRC is about to establish a SNA session with the host.</li> <li>• <b>parsing FMH7</b> indicates that the router just received an error from the CICS client.</li> <li>• <b>parsing DFC</b> indicates that the SNA session is about to close.</li> <li>• <b>receiving</b> indicates that the SNA session is receiving data from the host.</li> <li>• <b>reset</b> indicates that the SNA session is idle, waiting for a new transaction request from the CICS client.</li> <li>• <b>sending</b> indicates that the SNA session is sending data to the host.</li> <li>• <b>waiting</b> indicates that the SNA session is waiting for data from the client.</li> </ul>
tp name	CICS transaction program name.
user id	CICS user ID associated with the transaction.
session RU address	SNA architected address that allows multiple sessions to share one connection.
idle time	Time in milliseconds that the SNA session has been idle.
number of transactions executed	Number of CICS transactions executed by the current SNA session.
number of bytes received from client	Number of bytes received from the CICS client during the current SNA session.
number of bytes received from host	Number of bytes received from the CICS host during the current SNA session.

Related Commands

<b>Command</b>	<b>Description</b>
<b>show txconn connection</b>	Displays information about CTRC connections to CICS clients.
<b>show txconn destination</b>	Displays information about CICS destinations for CTRC.
<b>show txconn server</b>	Displays information about CTRC servers that communicate with CICS.

## txconn destination

To configure a CTRC destination, use the **txconn destination** global configuration command. To remove configuration for a txconn destination, use the **no** form of this command.

```
txconn destination destination-name rlu rlu-name mode mode-name
no txconn destination destination-name [rlu rlu-name mode mode-name]
```

### Syntax Description

<i>destination-name</i>	Name of the destination being defined or added to. This name is used in the route configuration command to identify the destination for the route.  If the destination does not exist, it is created; if it exists, the <b>rlu</b> and <b>mode</b> parameters are added as an additional routing target for this destination. When a destination contains multiple routing targets, it is like configuring a cluster where the various targets are chosen on a round-robin basis for load balancing.
<b>rlu</b> <i>rlu-name</i>	Remote LU name on the host. This parameter defines to which remote LU the server will connect when using this destination. A remote LU corresponds directly to a CICS region. The value you enter here should match your VTAM APPLID.
<b>mode</b> <i>mode-name</i>	Name of the APPC mode. This parameter defines which mode the server will use for its APPC connections when using this destination. If the mode you specify does not already exist, CTRC will create it.

### Defaults

No defaults exist for the **txconn destination** command. However, the remote LU name of the host and the APPC mode name are optional for the **no** form of the command. If you omit them, CTRC removes the configuration for all routing targets defined for the destination. If you use **rlu** *rlu-name* **mode** *mode-name* to specify a particular routing target within a destination that has multiple targets, the configuration is removed only for the specified target.

### Command Modes

Global configuration

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example shows the CTRC destination newdest being defined on the current router:

```
txconn destination newdest rlu CICS mode IBMRDB
```

## txconn destination

---

A second pair of rlu and mode values could be assigned to this same logical destination, to allow load sharing between the two destination CICS systems:

```
txconn destination newdest rlu CICS mode IBMRDB
```

### Related Commands

Command	Description
<b>show snasw mode</b>	Displays information about predefined SNA Switching Services modes.
<b>show txconn destination</b>	Displays information about CICS destinations for CTRC.

## txconn license

To license a Cisco router for CTRC communications with CICS or DB2, use the **txconn license** global configuration command. To remove the license, use the **no** form of this command.

```
txconn license license-key [connections licensed-connections] [expiration-date yyyymmdd]  
no txconn license
```

### Syntax Description

<i>license-key</i>	License key obtained from your Cisco representative. The license determines the maximum number of DB2 connections or conversations with CICS possible via a Cisco router configured with CTRC. It is not case sensitive, but it is linked to a specific router. When you request your license key, you must supply your Cisco representative with the SNA Switching Services cpname for the router. Use the <b>show config   include cpname</b> command to view the cpname.
<b>connections</b> <i>licensed-connections</i>	(Optional) Number of licensed connections. If omitted, the number of licensed connections must be unlimited.
<b>expiration-date</b> <i>yyyymmdd</i>	(Optional) Date when a temporary license key expires, in <i>yyyymmdd</i> format, where <i>yyyy</i> is the year expressed in four digits, <i>mm</i> is the month expressed in two digits, and <i>dd</i> is the date expressed in two digits. If omitted, the license key must be permanent.

### Defaults

If the number of licensed connections is not specified, the license key must be generated for an unlimited number of licensed connections. If the expiration date is not specified, the license key must be permanent.

### Command Modes

Global configuration

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Usage Guidelines

Licenses are required for all CTRC installations. Within CTRC, a common pool of node-linked licenses is used for communications with both DB2 and CICS. Therefore, if you have already configured licenses for CTRC connections to DB2, you do not need to configure licenses for CICS. The **txconn license** command provides exactly the same functionality as **dbconn license**.

For testing and evaluation purposes, unlicensed CTRC installations allow you to establish two connections to DB2 or two conversations with CICS.

Examples

The following example shows a CTRC license being configured for communications with CICS:

```
txconn license 8C09A496420BAF020BFF74B8A2FF24D2
```

Related Commands

<b>Command</b>	<b>Description</b>
<b>dbconn license</b>	Configures CTRC license for connections to DB2 or CICS. Functionally identical to the <b>txconn license</b> command.
<b>show dbconn license</b>	Shows the status of CTRC licenses. Functionally identical to the <b>show txconn license</b> command.
<b>show txconn license</b>	Shows the status of CTRC licenses. Functionally identical to the <b>show dbconn license</b> command.

## txconn ping

To test communications between the CTRC router and a CTRC destination (a host defined by a pair of RLU and mode values), use the **txconn ping** EXEC command.

**txconn ping** *destination-name*

### Syntax Description

*destination-name* Specifies the CICS system for which to test communications.

### Defaults

None.

### Command Modes

EXEC

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example shows communications being tested between the current router and the CTRC destination GEN. Note that GEN is a destination that has two RLU-Mode pairs defined, and that the **txconn ping** command automatically tests connections to both:

```
txconn ping GEN
Trying GEN CICSC:IBMRDB
Destination GEN successfully contacted!
Elapsed time was 00:00:01.001
Trying GEN CICSB:IBMRDB
Destination GEN successfully contacted!
Elapsed time was 00:00:01.001
```

Elapsed time is noted in hours, minutes, seconds, and milliseconds.

### Related Commands

Command	Description
<b>dbconn ping</b>	Tests CTRC communications with DB2 databases.

## txconn route

To configure a CTRC route that will transmit specified transactions to a particular CICS destination, use the **txconn route** global configuration command. To remove the configuration of a CTRC route, use the **no** form of this command.

**txconn route** [**server** *server-name*] **tranid** *transaction-id* **destination** *destination-name*

**no txconn route** [**server** *server-name*] **tranid** *transaction-id*

### Syntax Description

<b>server</b> <i>server-name</i>	(Optional) Name of the CTRC server to which this route applies. If omitted, this route is applied to all CTRC servers on the current router that are configured for communication with CICS.
<b>tranid</b> <i>transaction-id</i>	CICS transaction ID (a TP name). When the server processes a transaction that uses this transaction ID, the server routes the transaction using this route entry.
<b>destination</b> <i>destination-name</i>	Name of the destination to which the transaction is routed.

### Defaults

None.

### Command Modes

Global configuration

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example shows a CTRC route to destination GEN being defined on the current router for transaction ID PNG1:

```
txconn route server newsvr tranid PNG1 destination GEN
```

### Related Commands

Command	Description
<b>show txconn route</b>	Displays information about CTRC routes to CICS.

## txconn server

To configure a CTRC server for communications with CICS, use the **txconn server** global configuration command. To disable a CTRC server, use the **no** form of this command.

```
txconn server server-name destination destination-name [ccsid number]
[ipaddress ip-address] [port port-number] [client-timeout minutes] [host-timeout minutes]
[window-size bytes] [fold on/off]
```

```
no txconn server server-name
```

### Syntax Description

<i>server-name</i>	Name of the server being defined. This name is used in other commands to identify the server being administered.
<b>ccsid</b> <i>number</i>	The Coded Character Set Identifier. This is used for Tx Series Clients.
<b>destination</b> <i>destination-name</i>	Name of the server's default destination. Any transactions whose <i>tranid</i> is not associated with a particular route will be routed to this destination. The destination must already be defined when configuring the server.
<b>ipaddress</b> <i>ip-address</i>	(Optional) TCP/IP network address for which the server accepts connections. If this parameter is omitted, the server accepts connections for any IP address, like a wildcard address. If multiple servers are configured to listen on the same port, they must each specify a different IP address. If a server is configured with the IP address omitted, no other servers may listen on the same port. So, on a given port, you may configure either 1 wildcard IP address server, or <i>n</i> address-specific servers, where <i>n</i> is 1 or more.
<b>port</b> <i>port-number</i>	(Optional) TCP/IP port number on which the server listens. If no IP address is specified, only one server can listen on a port. Multiple servers can use the same port number if the combination of IP address and port number is unique to each server. If the port number is omitted, the server listens on port 1435.
<b>client-timeout</b> <i>minutes</i>	(Optional) Number of minutes of client connection inactivity after which the server decides the client has gone away. When this happens the server closes the client connection. If no client timeout is specified, the default is 0 (zero) for no timeout.
<b>host-timeout</b> <i>minutes</i>	(Optional) Number of minutes of host connection inactivity after which the server decides the host has gone away. When this happens the server closes the host connection. If no host timeout is specified, the default is 0 (zero) for no timeout.
<b>window-size</b> <i>bytes</i>	(Optional) Size, in bytes, of the TCP/IP window for incoming CICS client connections. If no window size is specified, the default is 4096 bytes (4 K).

**fold {on / off}** Enables/disables the fold program. Default is on. CTRC folds the CICS program name to upper case.

### Defaults

If the CTRC server's IP address is not configured, the server accepts connections for any IP address that is configured for the router.

If the port number is not configured, the server listens on port 1435.

If you do not specify a client timeout, CICS client connections can continue regardless of how long they have been idle.

If you do not specify a host timeout, host connections can continue regardless of how long they have been idle.

If you do not specify a TCP/IP window size, the default value is 4096 bytes (4 K).

### Command Modes

Global configuration

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Usage Guidelines

You can configure more than one CTRC server per router for communications with DB2 or CICS. There is no limit on the number of CTRC servers. However, be sure that CTRC txconn servers and CTRC dbconn servers are configured to use different ports, and that each txconn server is configured to use a unique combination of port number and IP address, or a unique port number with no IP address.

### Examples

The following example shows the CTRC server newsvr being defined on the current router:

```
txconn server newsvr destination GEN port 1438
```

### Related Commands

Command	Description
<b>show txconn destination</b>	Displays information about CICS destinations for CTRC.
<b>show txconn server</b>	Displays information about CTRC servers that communicate with CICS.

## Debug Commands

This section documents debugging commands used with the CTRC feature. Some of these commands were previously used with the Cisco Database Connection feature (CDBC), which CTRC replaces. Other commands are new with the introduction of CTRC.

- **debug dbconn all**
- **debug dbconn appc**
- **debug dbconn config**
- **debug dbconn drda**
- **debug dbconn event**
- **debug dbconn tcp**
- **debug txconn all**
- **debug txconn appc**
- **debug txconn config**
- **debug txconn data**
- **debug txconn event**
- **debug txconn tcp**
- **debug txconn timer**

## debug dbconn all

Use the **debug dbconn all** privileged EXEC command to turn on all debug flags for CTRC communications with DB2. These flags include **appc**, **config**, **drda**, **event**, and **tcp**. Use the **no** form of this command to disable all debugging output.

**[no] debug dbconn all**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the dbconn subsystem.

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Examples

See the examples provided for the **debug dbconn appc**, **debug dbconn config**, **debug dbconn drda**, **debug dbconn event**, and **debug dbconn tcp** commands.

### Related Commands

Command	Description
<b>debug dbconn appc</b>	Displays APPC-related trace or error messages for communications with DB2.
<b>debug dbconn config</b>	Displays trace or error messages for CTRC configuration and control blocks for DB2 communications.
<b>debug dbconn drda</b>	Displays error messages or stream traces for DRDA communications with DB2.
<b>debug dbconn event</b>	Displays trace or error messages for CTRC events related to DB2 communications.
<b>debug dbconn tcp</b>	Displays error messages or traces for TCP/IP communications with DB2.
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>show debugging</b>	Displays the state of each debugging option.

## debug dbconn appc

Use the **debug dbconn** privileged EXEC command to display APPC-related trace or error messages for communications with DB2. Use the **no** form of this command to disable debugging output.

**[no] debug dbconn appc**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the dbconn subsystem.

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Examples

The following example shows APPC debugging output using the **debug dbconn appc** command:

```
01:18:05: DBCONN-APPC-622ADF38: Verb block =
01:18:05: DBCONN-APPC-622ADF38: 0001 0200 0300 0000 0400 0000 0000 0000
01:18:05: DBCONN-APPC-622ADF38: 0000 00FC 0000 0000 0000 0000 0000 0000
01:18:05: DBCONN-APPC-622ADF38: 0000 0000 0840 0007 0000 0000 0000 0000
01:18:05: DBCONN-APPC-622ADF38: 7BC9 D5E3 C5D9 4040 07F6 C4C2 4040 4040
01:18:05: DBCONN-APPC-622ADF38: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: DBCONN-APPC-622ADF38: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: DBCONN-APPC-622ADF38: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: DBCONN-APPC-622ADF38: 4040 4040 4040 4040 0000 0000 0000 0000
01:18:05: DBCONN-APPC-622ADF38: 0000 0000 0000 0000 0000 0000 0000 0000
01:18:05: DBCONN-APPC-622ADF38: 0000 0000 0000 0000 0000 0000 0000 0000
01:18:05: DBCONN-APPC-622ADF38: 00E2 E3C1 D9E6 4BC7 C1E9 C5D3 D3C5 4040
01:18:05: DBCONN-APPC-622ADF38: 4040 0000 0000 0000 0000 0000
01:18:05: DBCONN-APPC-621E5730: Verb block =
01:18:05: DBCONN-APPC-621E5730: 0001 0200 0300 0000 0400 0000 0000 0000
01:18:05: DBCONN-APPC-621E5730: 0000 00FD 0000 0000 0000 0000 0000 0000
01:18:05: DBCONN-APPC-621E5730: 0000 0000 0840 0007 0000 0000 0000 0000
01:18:05: DBCONN-APPC-621E5730: C9C2 D4D9 C4C2 4040 07F6 C4C2 4040 4040
01:18:05: DBCONN-APPC-621E5730: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: DBCONN-APPC-621E5730: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: DBCONN-APPC-621E5730: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: DBCONN-APPC-621E5730: 4040 4040 4040 4040 0000 0000 0000 0000
01:18:05: DBCONN-APPC-621E5730: 0000 0000 0000 0000 0000 0000 0000 0000
01:18:05: DBCONN-APPC-621E5730: 0000 0000 0000 0000 0000 0000 0000 0000
01:18:05: DBCONN-APPC-621E5730: 00E2 E3C1 D9E6 4BE2 E3C5 D3D3 C140 4040
01:18:05: DBCONN-APPC-621E5730: 4040 0000 0000 0000 0000 0000
```

Related Commands

<b>Command</b>	<b>Description</b>
<b>debug dbconn all</b>	Displays all CTRC debugging information related to communications with DB2.
<b>debug dbconn config</b>	Displays trace or error messages for CTRC configuration and control blocks for DB2 communications.
<b>debug dbconn drda</b>	Displays error messages or stream traces for DRDA communications with DB2.
<b>debug dbconn event</b>	Displays trace or error messages for CTRC events related to DB2 communications.
<b>debug dbconn tcp</b>	Displays error messages or traces for TCP/IP communications with DB2.
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>show debugging</b>	Displays the state of each debugging option.

## debug dbconn config

Use the **debug dbconn config** privileged EXEC command to display trace or error messages for CTRC configuration and control blocks for DB2 communications. Use the **no** form of this command to disable debugging output.

**[no] debug dbconn config**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the dbconn subsystem.

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Examples

The following example shows output from the **debug dbconn config** command when SNA Switching Services is not available:

```
01:21:05: DBCONN-CONFIG: server 6231CA1C takes connection 621E12CC
01:21:05: %CTRC-3-SNASWNOTSTARTED: SNASW is not started -- unable to connect to
remote database.
01:21:05: %CTRC-3-SNASWNOTSTARTED: SNASW is not started -- unable to connect to
remote database.
01:21:05: DBCONN-CONFIG: server 6231BA6C releases connection 62138E50
01:21:05: DBCONN-CONFIG: deleting connection 62138E50
01:21:05: DBCONN-CONFIG: server 6231CA1C releases connection 621E12CC
01:21:05: DBCONN-CONFIG: deleting connection 621E12CC
01:21:05: DBCONN-CONFIG: listen 6231BB88 received ping reply - ping failed.
01:21:05: DBCONN-CONFIG: listen 62324FE8 received ping reply - ping failed.
01:21:19: %CTRC-3-SNASWSTOPPED: SNASW was stopped -- existing connections have been
closed.
01:21:20: DBCONN-CONFIG: listen 62171740 received alarm msg
```

Related Commands

<b>Command</b>	<b>Description</b>
<b>debug dbconn all</b>	Displays all CTRC debugging information related to communications with DB2.
<b>debug dbconn appc</b>	Displays APPC-related trace or error messages for communications with DB2.
<b>debug dbconn drda</b>	Displays error messages or stream traces for DRDA communications with DB2.
<b>debug dbconn event</b>	Displays trace or error messages for CTRC events related to DB2 communications.
<b>debug dbconn tcp</b>	Displays error messages or traces for TCP/IP communications with DB2.
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>show debugging</b>	Displays the state of each debugging option.

## debug dbconn drda

Use the **debug dbconn drda** privileged EXEC command to display error messages and stream traces for DRDA. Use the **no** form of this command to disable debugging output.

**[no] debug dbconn drda**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the dbconn subsystem.

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Examples

The following example displays output from the **debug dbconn drda** command:

```
Router#debug dbconn drda
*Jun 30 16:09:32.363: DBCONN-DRDA-62008300: DSS X'006CD0410001', length 108, in chain,
REQDSS, correlator 1
*Jun 30 16:09:32.363: DBCONN-DRDA-62008300: OBJECT X'00661041', length 98, code point
X'1041'
*Jun 30 16:09:32.363: DBCONN-DRDA-62008300: OBJECT X'0020115E' in COLLECTION X'1041',
length 28, code point X'115E'
*Jun 30 16:09:32.363: DBCONN-DRDA-62008300: OBJECT X'000C116D' in COLLECTION X'1041',
length 8, code point X'116D'
*Jun 30 16:09:32.363: DBCONN-DRDA-62008300: OBJECT X'0013115A' in COLLECTION X'1041',
length 15, code point X'115A' (skipping...)
```

### Related Commands

Command	Description
<b>debug dbconn all</b>	Displays all CTRC debugging information related to communications with DB2.
<b>debug dbconn appc</b>	Displays APPC-related trace or error messages for communications with DB2.
<b>debug dbconn config</b>	Displays trace or error messages for CTRC configuration and control blocks for DB2 communications.
<b>debug dbconn event</b>	Displays trace or error messages for CTRC events related to DB2 communications.
<b>debug dbconn tcp</b>	Displays error messages or traces for TCP/IP communications with DB2.
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>show debugging</b>	Displays the state of each debugging option.

## debug dbconn event

Use the **debug dbconn event** privileged EXEC command to display trace or error messages for CTRC events related to DB2 communications. Use the **no** form of this command to disable debugging output.

**[no] debug dbconn event**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the dbconn subsystem.

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Examples

The following examples display output from the **debug dbconn event** command in a variety of situations. A normal trace for the **debug dbconn event** displays as follows:

```
Router#debug dbconn event
DBCONN-EVENT: Dispatch to 60FD6C00, from 0, msg 60F754CC, msgid 6468 'dh',
buffer 0.
DBCONN-EVENT: [*] Post to 61134240(cn), from 60EC5470(tc), msg 611419E4,
msgid 0x6372 'cr', buffer 612BF68C.
DBCONN-EVENT: Flush events called for pto 61182742, pfrom 61239837.
DBCONN-EVENT: Event discarded: to 61182742 (cn), from 61239837(ap), msg
61339273, msgid 0x6372 'cr' buffer 0.
DBCONN-EVENT: == Send to 1234abcd, from 22938acd, msg 72618394, msgid
0x6372 'cr', buffer 0.
```

If the following messages are displayed, contact Cisco technical support personnel:

```
DBCONN-TCPFSM-1234abcd: Cannot occur in state 2 on input 6363 ('cc')
DBCONN-APPCFSM-1234abcd: Cannot occur in state 3 on input 6363 ('cc')
```

## Related Commands

<b>Command</b>	<b>Description</b>
<b>debug dbconn all</b>	Displays all CTRC debugging information related to communications with DB2.
<b>debug dbconn appc</b>	Displays APPC-related trace or error messages for communications with DB2.
<b>debug dbconn config</b>	Displays trace or error messages for CTRC configuration and control blocks for DB2 communications.
<b>debug dbconn drda</b>	Displays error messages or stream traces for DRDA communications with DB2.
<b>debug dbconn tcp</b>	Displays error messages or traces for TCP/IP communications with DB2.
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>show debugging</b>	Displays the state of each debugging option.

## debug dbconn tcp

Use the **debug dbconn tcp** privileged EXEC command to display error messages and traces for TCP. Use the **no** form of this command to disable debugging output.

**[no] debug dbconn tcp**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the dbconn subsystem.

### Command History

Release	Modification
11.3(2)T	This command was introduced.
12.0(5)XN	Command moved from CDBC feature to CTRC feature.

### Examples

The following example displays output from the **debug dbconn tcp** command:

```
Router#debug dbconn tcp
DBCONN-TCP-63528473: tcpdriver_passive_open returned NULL
DBCONN-TCP-63528473: (no memory) tcp_reset(63829482) returns 4
DBCONN-TCP: tcp_accept(74625348,&error) returns tcb 63829482, error 4
DBCONN-TCP: (no memory) tcp_reset(63829482) returns 4
DBCONN-TCP-63528473: (open) tcp_create returns 63829482, error = 4
DBCONN-TCP-63528473: tcb_connect(63829482,1.2.3.4,2010) returns 4
DBCONN-TCP-63528473: (open error) tcp_reset(63829482) returns 4
DBCONN-TCP-63528473: tcp_create returns 63829482, error = 4
DBCONN-TCP-63528473: tcb_bind(63829482,0.0.0.0,2001) returns 4
DBCONN-TCP-63528473: tcp_listen(63829482,,) returns 4
DBCONN-TCP-63528473: (errors) Calling tcp_close (63829482)
```

## Related Commands

<b>Command</b>	<b>Description</b>
<b>debug dbconn all</b>	Displays all CTRC debugging information related to communications with DB2.
<b>debug dbconn appc</b>	Displays APPC-related trace or error messages for communications with DB2.
<b>debug dbconn config</b>	Displays trace or error messages for CTRC configuration and control blocks for DB2 communications.
<b>debug dbconn drda</b>	Displays error messages or stream traces for DRDA communications with DB2.
<b>debug dbconn event</b>	Displays trace or error messages for CTRC events related to DB2 communications.
<b>debug ip tcp</b>	Displays debugging information related to TCP/IP.
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>show debugging</b>	Displays the state of each debugging option.

## debug txconn all

Use the **debug txconn all** privileged EXEC command to turn on all debug flags for CTRC communications with CICS. These flags include **appc**, **config**, **data**, **event**, **tcp**, and **timer**. Use the **no** form of this command to disable all debugging output.

**[no] debug txconn all**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the txconn subsystem.

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example shows the immediate output of the debug txconn all command. For examples of specific debugging messages, see the examples provided for the **debug txconn appc**, **debug txconn config**, **debug txconn data**, **debug txconn event**, **debug txconn tcp**, and **debug txconn timer** commands.

```
Router#debug txconn all
All possible TXConn debugging has been turned on
```

### Related Commands

Command	Description
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>debug txconn appc</b>	Displays APPC-related trace or error messages for communications with CICS.
<b>debug txconn config</b>	Displays trace or error messages for CTRC configuration and control blocks for CICS communications.
<b>debug txconn data</b>	Displays CICS client and host data being handled by CTRC, in hexadecimal notation.
<b>debug txconn event</b>	Displays trace or error messages for CTRC events related to CICS communications.
<b>debug txconn tcp</b>	Displays error messages or traces for TCP/IP communications with CICS.
<b>debug txconn timer</b>	Displays performance information related to CICS communications.
<b>show debugging</b>	Displays the state of each debugging option.

## debug txconn appc

Use the **debug txconn** privileged EXEC command to display APPC-related trace or error messages for communications with CICS. Use the **no** form of this command to disable debugging output.

**[no] debug txconn appc**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the txconn subsystem.

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example shows APPC debugging output from the **debug txconn appc** command:

```
01:18:05: TXCONN-APPC-622ADF38: Verb block =
01:18:05: TXCONN-APPC-622ADF38: 0001 0200 0300 0000 0400 0000 0000 0000
01:18:05: TXCONN-APPC-622ADF38: 0000 00FC 0000 0000 0000 0000 0000 0000
01:18:05: TXCONN-APPC-622ADF38: 0000 0000 0840 0007 0000 0000 0000 0000
01:18:05: TXCONN-APPC-622ADF38: 7BC9 D5E3 C5D9 4040 07F6 C4C2 4040 4040
01:18:05: TXCONN-APPC-622ADF38: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: TXCONN-APPC-622ADF38: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: TXCONN-APPC-622ADF38: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: TXCONN-APPC-622ADF38: 4040 4040 4040 4040 0000 0000 0000 0000
01:18:05: TXCONN-APPC-622ADF38: 0000 0000 0000 0000 0000 0000 0000 0000
01:18:05: TXCONN-APPC-622ADF38: 0000 0000 0000 0000 0000 0000 0000 0000
01:18:05: TXCONN-APPC-622ADF38: 00E2 E3C1 D9E6 4BC7 C1E9 C5D3 D3C5 4040
01:18:05: TXCONN-APPC-622ADF38: 4040 0000 0000 0000 0000 0000
01:18:05: TXCONN-APPC-621E5730: Verb block =
01:18:05: TXCONN-APPC-621E5730: 0001 0200 0300 0000 0400 0000 0000 0000
01:18:05: TXCONN-APPC-621E5730: 0000 00FD 0000 0000 0000 0000 0000 0000
01:18:05: TXCONN-APPC-621E5730: 0000 0000 0840 0007 0000 0000 0000 0000
01:18:05: TXCONN-APPC-621E5730: C9C2 D4D9 C4C2 4040 07F6 C4C2 4040 4040
01:18:05: TXCONN-APPC-621E5730: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: TXCONN-APPC-621E5730: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: TXCONN-APPC-621E5730: 4040 4040 4040 4040 4040 4040 4040 4040
01:18:05: TXCONN-APPC-621E5730: 4040 4040 4040 4040 0000 0000 0000 0000
01:18:05: TXCONN-APPC-621E5730: 0000 0000 0000 0000 0000 0000 0000 0000
01:18:05: TXCONN-APPC-621E5730: 0000 0000 0000 0000 0000 0000 0000 0000
01:18:05: TXCONN-APPC-621E5730: 00E2 E3C1 D9E6 4BE2 E3C5 D3D3 C140 4040
01:18:05: TXCONN-APPC-621E5730: 4040 0000 0000 0000 0000 0000
```

Related Commands

<b>Command</b>	<b>Description</b>
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>debug txconn all</b>	Displays all CTRC debugging information related to communications with CICS.
<b>debug txconn config</b>	Displays trace or error messages for CTRC configuration and control blocks for CICS communications.
<b>debug txconn data</b>	Displays CICS client and host data being handled by CTRC, in hexadecimal notation.
<b>debug txconn event</b>	Displays trace or error messages for CTRC events related to CICS communications.
<b>debug txconn tcp</b>	Displays error messages or traces for TCP/IP communications with CICS.
<b>debug txconn timer</b>	Displays performance information related to CICS communications.
<b>show debugging</b>	Displays the state of each debugging option.

## debug txconn config

Use the **debug txconn config** privileged EXEC command to display trace or error messages for CTRC configuration and control blocks for CICS communications. Use the **no** form of this command to disable debugging output.

**[no] debug txconn config**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the txconn subsystem.

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example shows output for the **debug dbconn config** command:

```
Router#debug txconn config
22:11:37: TXCONN-CONFIG: deleting transaction 61FCE414
22:11:37: TXCONN-CONFIG: deleting connection 61FB5CB0
22:11:37: TXCONN-CONFIG: server 62105D6C releases connection 61FB5CB0
22:11:44: TXCONN-CONFIG: new connection 61FB64A0
22:11:44: TXCONN-CONFIG: server 6210CEB4 takes connection 61FB64A0
22:11:44: TXCONN-CONFIG: new transaction 61E44B9C
22:11:48: TXCONN-CONFIG: deleting transaction 61E44B9C
22:11:53: TXCONN-CONFIG: new transaction 61E44B9C
22:11:54: TXCONN-CONFIG: deleting transaction 61E44B9C
```

### Related Commands

Command	Description
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>debug txconn all</b>	Displays all CTRC debugging information related to communications with CICS.
<b>debug txconn appc</b>	Displays APPC-related trace or error messages for communications with CICS.
<b>debug txconn data</b>	Displays CICS client and host data being handled by CTRC, in hexadecimal notation.
<b>debug txconn event</b>	Displays trace or error messages for CTRC events related to CICS communications.
<b>debug txconn tcp</b>	Displays error messages or traces for TCP/IP communications with CICS.
<b>debug txconn timer</b>	Displays performance information related to CICS communications.
<b>show debugging</b>	Displays the state of each debugging option.

## debug txconn data

To display a hexadecimal dump of CICS client and host data being handled by CTRC, plus information about certain CTRC internal operations, use the **debug txconn data** privileged EXEC command. Use the no form of this command to disable the debugging output.

**[no] debug txconn data**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the txconn subsystem.

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example shows selected output from the **debug txconn data** command when a connection is established, data is received from the client via TCP/IP, data is sent to the client, and then the connection is closed.

```
Router#debug txconn data
TXConn DATA debugging is on

00:04:50: TXConn(62197464) Created
00:04:50: TXConn(62197464) State(0) MsgID(0) -> nextState(1)
00:04:50: TXConn(62197464) Client->0000 003A 0000 0002 000B 90A0
00:04:50: TXConn(62197464) Received LL 58 for session(0 0 2).
00:06:27: TXConn(62197464) Client<-0000 0036 0000 0003 000B 8001 0707 0864
00:06:53: TXConn(62175024) Deleted
```

The following lines show output when data is sent to the host:

```
00:04:50: TXTrans(id:62197910 conn:62197464 addr:2) LL(58) FMH5(0) CEBI(0)
00:04:50: TXTrans(id:62197910 conn:62197464 addr:2) State(0) MsgID(7844) ->
nextState(1)
00:04:50: TXTrans(id:62197910 conn:62197464 addr:2) conversationType(mapped)
syncLevel(1) sec(0)
00:04:50: TXTrans(id:62197910 conn:62197464 addr:2) TPName CCIN
00:04:50: TXTrans(id:62197910 conn:62197464 addr:2) apDataLength(32) GDSID(12FF)

00:04:50: TXTrans(id:62197910 conn:62197464 addr:2) ->Host 0000 0008 03F4 F3F7 0000
0008 0401 0000
```

The following lines show output when data is received from the host:

```
00:05:01: TXTrans(id:62197910 conn:62197464 addr:2) <-Host 0092 12FF 0000 000C 0102
0000 0000 0002
```

The following lines show CTRC generating an FMH7 error message indicating that a CICS transaction has failed at the host or has been cleared by a router administrator:

```
00:06:27: TXTrans(id:6219853C conn:62197464 addr:3) Generating FMH7.
00:06:27: %TXCONN-3-TXEXCEPTION: Error occurred from transaction 3 of client
157.151.241.10 connected to server CICSC, exception type is 9
```

The following line shows CTRC responding to an FMH7 error message sent by the CICS client program:

```
00:07:11: TXTrans(id:62197910 conn:62197464 addr:2) Generating FMH7 +RSP.
```

## Related Commands

Command	Description
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>debug txconn all</b>	Displays all CTRC debugging information related to communications with CICS.
<b>debug txconn appc</b>	Displays APPC-related trace or error messages for communications with CICS.
<b>debug txconn config</b>	Displays trace or error messages for CTRC configuration and control blocks for CICS communications.
<b>debug txconn event</b>	Displays trace or error messages for CTRC events related to CICS communications.
<b>debug txconn tcp</b>	Displays error messages or traces for TCP/IP communications with CICS.
<b>debug txconn timer</b>	Displays performance information related to CICS communications.
<b>show debugging</b>	Displays the state of each debugging option.

## debug txconn event

Use the **debug txconn event** privileged EXEC command to display trace or error messages for CTRC events related to CICS communications. Use the **no** form of this command to disable debugging output.

**[no] debug txconn event**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the txconn subsystem.

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example shows output for the **debug txconn event** command:

```
Router#debug txconn event
TXConn event debugging is on
Router#
22:15:08: TXCONN-EVENT: [*] Post to 62146464(cn), from 6211E744(tc), msg
61FC6170, msgid 0x6372 'cr', buffer 6211289C.
22:15:08: TXCONN-EVENT: Dispatch to 62146464, from 6211E744, msg 61FC6170,
msgid 6372 'cr', buffer 6211289C.
22:15:08: TXCONN-EVENT: [*] Post to 61E44BA0(sn), from 62146464(cn), msg
621164D0, msgid 0x7844 'xD', buffer 0.
22:15:08: TXCONN-EVENT: [*] Post to 6211E744(tc), from 62146464(cn), msg
61FC6170, msgid 0x6347 'cG', buffer 0.
22:15:08: TXCONN-EVENT: Dispatch to 61E44BA0, from 62146464, msg 621164D0,
msgid 7844 'xD', buffer 0.
22:15:08: TXCONN-EVENT: Dispatch to 6211E744, from 62146464, msg 61FC6170,
msgid 6347 'cG', buffer 0.
22:15:08: TXCONN-EVENT: [*] Post to 62146464(cn), from 6211E744(tc), msg
61FC6170, msgid 0x6372 'cr', buffer 6211289C.
22:15:08: TXCONN-EVENT: Dispatch to 62146464, from 6211E744, msg 61FC6170,
msgid 6372 'cr', buffer 6211289C.
22:15:08: TXCONN-EVENT: [*] Post to 61E44BA0(sn), from 62146464(cn), msg
61FBFBF4, msgid 0x7844 'xD', buffer 0.
22:15:08: TXCONN-EVENT: [*] Post to 6211E744(tc), from 62146464(cn), msg
61FC6170, msgid 0x6347 'cG', buffer 0.
22:15:08: TXCONN-EVENT: Dispatch to 61E44BA0, from 62146464, msg 61FBFBF4,
msgid 7844 'xD', buffer 0.
22:15:08: TXCONN-EVENT: [*] Post to 61FC6394(ap), from 61E44BA0(sn), msg
621164D0, msgid 0x634F 'cO', buffer 0.
22:15:08: TXCONN-EVENT: Dispatch to 6211E744, from 62146464, msg 61FC6170,
msgid 6347 'cG', buffer 0.
```

## Related Commands

<b>Command</b>	<b>Description</b>
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>debug txconn all</b>	Displays all CTRC debugging information related to communications with CICS.
<b>debug txconn appc</b>	Displays APPC-related trace or error messages for communications with CICS.
<b>debug txconn config</b>	Displays trace or error messages for CTRC configuration and control blocks for CICS communications.
<b>debug txconn data</b>	Displays CICS client and host data being handled by CTRC, in hexadecimal notation.
<b>debug txconn tcp</b>	Displays error messages or traces for TCP/IP communications with CICS.
<b>debug txconn timer</b>	Displays performance information related to CICS communications.
<b>show debugging</b>	Displays the state of each debugging option.

## debug txconn tcp

Use the **debug txconn tcp** privileged EXEC command to display error messages and traces for TCP. Use the **no** form of this command to disable debugging output.

**[no] debug txconn tcp**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the txconn subsystem.

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example displays output from the **debug txconn tcp** command:

```
Router#debug txconn tcp
TXCONN-TCP-63528473: tcpdriver_passive_open returned NULL
TXCONN-TCP-63528473: (no memory) tcp_reset(63829482) returns 4
TXCONN-TCP: tcp_accept(74625348,&error) returns tcb 63829482, error 4
TXCONN-TCP: (no memory) tcp_reset(63829482) returns 4
TXCONN-TCP-63528473: (open) tcp_create returns 63829482, error = 4
TXCONN-TCP-63528473: tcb_connect(63829482,1.2.3.4,2010) returns 4
TXCONN-TCP-63528473: (open error) tcp_reset(63829482) returns 4
TXCONN-TCP-63528473: tcp_create returns 63829482, error = 4
TXCONN-TCP-63528473: tcb_bind(63829482,0.0.0.0,2001) returns 4
TXCONN-TCP-63528473: tcp_listen(63829482,,) returns 4
TXCONN-TCP-63528473: (errors) Calling tcp_close (63829482)
```

## Related Commands

<b>Command</b>	<b>Description</b>
<b>debug ip</b>	Displays debugging information related to TCP/IP communications.
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>debug txconn all</b>	Displays all CTRC debugging information related to communications with CICS.
<b>debug txconn appc</b>	Displays APPC-related trace or error messages for communications with CICS.
<b>debug txconn config</b>	Displays trace or error messages for CTRC configuration and control blocks for CICS communications.
<b>debug txconn data</b>	Displays CICS client and host data being handled by CTRC, in hexadecimal notation.
<b>debug txconn event</b>	Displays trace or error messages for CTRC events related to CICS communications.
<b>debug txconn timer</b>	Displays performance information related to CICS communications.
<b>show debugging</b>	Displays the state of each debugging option.

## debug txconn timer

To display performance information regarding CTRC communications with CICS, use the **debug txconn timer** privileged EXEC command. Use the **no** form of this command to disable the debugging output.

**[no] debug txconn timer**

### Syntax Description

This command has no arguments or keywords.

### Defaults

By default, debugging is not enabled for the txconn subsystem.

### Command History

Release	Modification
12.0(5)XN	This command was introduced.

### Examples

The following example shows turnaround time and host response time in milliseconds for a CICS transaction requested through CTRC. Turnaround time is measured from when CTRC receives the first request packet for the transaction until CTRC sends the last response packet of the transaction to the client. Host response time is measured from when CTRC sends the last request packet for a transaction to the host until CTRC receives the first response packet from the host for that transaction.

```
Router#debug txconn timer  
TXConn timer debugging is on  
00:04:14: TXTrans(id:622F4350 conn:62175024 addr:1) Turnaround Time = 4536(msec)  
HostResponseTime = 120(msec)
```

## Related Commands

<b>Command</b>	<b>Description</b>
<b>debug snasw</b>	Displays debugging information related to SNA Switching Services.
<b>debug txconn all</b>	Displays all CTRC debugging information related to communications with CICS.
<b>debug txconn appc</b>	Displays APPC-related trace or error messages for communications with CICS.
<b>debug txconn config</b>	Displays trace or error messages for CTRC configuration and control blocks for CICS communications.
<b>debug txconn data</b>	Displays CICS client and host data being handled by CTRC, in hexadecimal notation.
<b>debug txconn event</b>	Displays trace or error messages for CTRC events related to CICS communications.
<b>debug txconn tcp</b>	Displays error messages or traces for TCP/IP communications with CICS.
<b>show debugging</b>	Displays the state of each debugging option.

## Error Messages

### Error Message

%CTRC-3-APPCVERB: Unexpected APPC error: opcode [hex], primary code [hex], secondary code [hex]

**Explanation** An unexpected APPC error occurred. CTRC uses APPC API calls to SNASw to communicate with the remote host. Opcode identifies the APPC API call that received the error. Primary and secondary return codes are returned by the APPC API.

**Recommended Action** This message usually indicates a problem in the SNA network. Use the **dbconn ping** command to test connectivity to a remote database server. Use the **txconn ping** command to test connectivity to a remote CICS system. If the ping fails, check the status of the SNA link to the remote host with the **show snasw link** command. If the link is active then stop and start the link. If the problem persists, contact your SNA network administrator for help in troubleshooting. If the message recurs, copy the error message exactly as it appears on the console or in the system log, call the Cisco TAC and provide the representative with the gathered information. If possible, also provide the output of the **debug dbconn appc** command (for a database connection) or the **debug txconn appc** command (for a CICS connection), and an SNA line trace between the router and the remote host for the period during which this message appeared.

### Error Message

%CTRC-3-SNASWNOTSTARTED: SNASW is not started -- unable to connect to remote host.

**Explanation** A client connected to a CTRC server, but CTRC was unable to connect out to the remote host because SNASw is not started. CTRC uses SNASw to connect to the remote host. You must start SNASw before CTRC clients attempt to connect.

**Recommended Action** Start SNASw and retry the client connection.

### Error Message

%CTRC-3-SNASWSTOPPED: SNASW was stopped -- existing connections have been closed.

**Explanation** SNASw was stopped while CTRC connections were open. Stopping SNASw caused all open SNA connections to the remote host to be lost. As a result, all client connections were disconnected. You should not stop SNASw while there are open CTRC connections.

**Recommended Action** Restart SNASw. Disconnected clients must reconnect.

### Error Message

%TXCONN-3-BADLICENSEKEY: Bad license key configured for Transaction Connection.

**Explanation** The license key configured with the **txconn license** configuration command is not valid.

**Recommended Action** Check that the license key was entered correctly in the **txconn license** configuration command. Refer to the txconn feature documentation for information on how to obtain a valid license key.

**Error Message**

%TXCONN-3-BADMODE: Unknown mode '[chars]' configured for server [chars]

**Explanation** The mode configured for the txconn server was rejected by the remote host.

**Recommended Action** Contact your SNA network administrator to determine the correct name of the mode to use to connect to the remote host.

**Error Message**

%TXCONN-3-BADRLU: Bad remote LU '[chars]' configured for server [chars]

**Explanation** The remote LU configured for the txconn server is incorrect.

**Recommended Action** Contact your SNA network administrator to determine the correct name of the remote LU for the remote host. Be sure to check that the SNA network name is correct as well as the LU name.

**Error Message**

%TXCONN-3-INVALIDTRANS: Bad Transaction ID, flushing out request for client [int] connected server [chars], killing connection.

**Explanation** Bad input stream.

**Recommended Action** This is an informational message. No action is required.

**Error Message**

%TXCONN-3-NOSESSION: Out of resource, cannot create transaction for client [int] connected server [chars], killing connection.

**Explanation** The router is out of memory.

**Recommended Action** This is an informational message. No action is required.

**Error Message**

%TXCONN-5-CONNIDLETIMEOUT: Client [int] connected to server [chars] has timed out after %TE.

**Explanation** The client was idle for too long, and the idle timeout configured for the txconn server expired.

**Recommended Action** This is an informational message only. No action is required.

**Error Message**

%TXCONN-5-SECFAIL: APPC security failed, client [int] using userid '[chars]' for server [chars]

**Explanation** The userid or password passed in by the client were rejected by the remote host.

**Recommended Action** This is an informational message only. No action is required.

**Error Message**

%TXCONN-5-TARGETDOWN: CICS server [chars] has become unavailable. Recovery is in progress.

**Explanation** A CICS server for a configured destination has become unavailable. All client connections to this server will be rejected until it becomes available again.

**Recommended Action** This is an informational message only. No action is required.

**Error Message**

%TXCONN-5-TARGETUP: CICS server [chars] is now available.

**Explanation** A CICS server that was previously unavailable has become available. All client connections to this server will now be accepted.

**Recommended Action** This is an informational message only. No action is required.

**Error Message**

%TXCONN-5-TRANSIDLETIMEOUT: Transaction [hex] of Client [int] connected to server [chars] has timed out after %TE.

**Explanation** The transaction was idle for too long, and the idle timeout configured for the txconn server expired.

**Recommended Action** This is an informational message only. No action is required.

**Error Message**

%TXCONN-3-TXEXCEPTION: Error occurred from transaction [hex] of client [int] connected to server [chars], exception type is [dec]

**Explanation** An unexpected transaction error occurred.

**Recommended Action** Copy the error message exactly as it appears on the console or in the system log, call the Cisco TAC, and provide the representative with the gathered information. If possible, also provide the output of the **debug txconn data** command and an SNA line trace between the router and the remote database server for the period during which this message appeared.

**Error Message**

%TXCONN-3-UNEXPECTEDREQ: Unexpected client data, flushing out the request for client [int] connected server [chars].

**Explanation** Bad input stream.

**Recommended Action** This is an informational message. No action is required.

## Glossary

“See cross references”. For example: Advanced Program-to-Program Communications—See APPC.

**APPC**—Advanced Program-to-Program Communications. An implementation of the SNA/SDLC LU6.2 protocol that allows interconnected systems to communicate and share the processing of programs.

**APPN**—Advanced Peer-to-Peer Networking. An extension to SNA featuring greater distributed network control that avoids critical hierarchical dependencies, thereby isolating the effects of single points of failure. APPN also features dynamic exchange of network topology information to foster ease of connection, reconfiguration, and adaptive route selection, dynamic definition of network resources, and automated resource registration and directory lookup.

**CDBC**—Cisco Database Connection. Prior to the introduction of CTRC, CDBC provided ODBC client applications with access to data in DB2 databases.

**CEDA**—A CICS-supplied transaction that lets you update the CICS system definition or define resources for your CICS system while the system is running.

**CICS**—IBM’s Customer Information Control System. An online transaction monitoring system. Available versions include CICS for OS/390, CICS/ESA, CICS/400, and TXSeries.

**CTRC**—Cisco Transaction Connection. Router software that provides ODBC client applications with connectivity to DB2 databases, and that provides CICS client applications with connectivity to CICS transaction monitoring systems.

**DB2**—IBM relational database management system. Available versions include DB2 for MVS/ESA, DB2 for OS/400, DB2 Parallel Edition, and DB2 for VSE and VM.

**DDM**—Distributed Data Management Architecture. An IBM architected data management interface used for data interchange among like or unlike systems. DRDA is based on DDM, describing the content of and restrictions on DDM flows related to relational database management.

**DRDA**—Distributed Relational Database Architecture. A connection protocol for distributed relational database processing that is used by IBM's relational database products.

**ECI**—external call interface. Allows a non-CICS client program to call a CICS transaction. The client application communicates with the program in the CICS server using a COMMAREA.

**EPI**—extended presentation interface. Interface suitable for distributed applications that call CICS transactions that were originally accessed via 3270 terminals.

**ISC**—intersystem communication protocol. Used to interconnect CICS with other CICS and IMS regions. ISC protocol is also used by IBM Common Client and Microsoft COMTI client software to communicate with CICS and IMS. ISC protocol supports transaction routing, function shipping, and distributed program link. Of these three functions, CTRC implements transaction routing and distributed program link.

**ODBC**—Open Database Connectivity. A call-level interface developed by Microsoft Corporation that allows a single application to access database management systems from different vendors using a single interface.

**SNA**—Systems Network Architecture. A large, complex, feature-rich network architecture developed by IBM.

**SNA Switching Services**—Cisco software that replaces Cisco's APPN offering. SNA Switching Services is an enhancement to the original IBM SNA architecture. It handles session establishment between peer nodes, dynamic transparent route calculation, and traffic prioritization for APPC traffic.

**TCP/IP**—Transmission Control Protocol/Internet Protocol. Common name for the suite of protocols developed by the U.S. Department of Defense in the 1970s to support the construction of worldwide internetworks. TCP and IP are the two best-known protocols in the suite.

**Workload Manager** —The purpose of workload management is to balance the available system resources to meet the demands of S/390 subsystems work managers such as DB2 in response to incoming work requests.