



Configuring Cisco Transaction Connection

This chapter describes how to configure the Cisco Transaction Connection (CTRC) feature. For a complete description of the CTRC commands mentioned in this chapter, refer to the “Cisco Transaction Connection Commands” chapter of the *Cisco IOS Bridging and IBM Networking Command Reference* (Volume 2 of 2). To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

This chapter contains the following sections:

- [Technology Overview, page 1](#)
- [Configuration CTRC Task List, page 4](#)
- [Defining the CTRC Router to VTAM, page 6](#)
- [Preparing a CICS Host for Remote Access, page 7](#)
- [Preparing a DB2 Host for Remote Access, page 11](#)
- [Configuring the CTRC Router, page 15](#)
- [Verifying the CTRC Configuration, page 18](#)
- [Configuring CTRC Clients, page 21](#)
- [Monitoring and Maintaining CTRC, page 25](#)
- [CTRC Configuration Examples, page 27](#)

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

Technology Overview

CTRC provides TCP/IP end-users and servers with fast, reliable, and secure access to IBM DB2 databases and Customer Information Control System (CICS) transaction programs. The CTRC feature of the Cisco router provides a flexible, cost-effective, and scalable solution for enterprise-wide database access and transaction processing. CTRC allows Windows or UNIX client applications to call CICS transactions without changes to the client or host software. Any client running a Distributed Relational Database Architecture (DRDA) requestor, which is included in most Open Database Connectivity (ODBC) applications, can use CTRC to access data in DB2 databases.



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

With CTRC, you can continue using current CICS client/server applications on a more robust, higher-performing platform than the general-purpose operating system gateways. CTRC provides protocol independence between client workstations and the host, enabling the applications to communicate directly with CICS and DB2 without costly mainframe application upgrades or expensive middleware servers.

The CTRC software feature provides:

- Access to DB2 databases from TCP/IP clients
- Access to CICS applications from TCP/IP clients
- A keepalive timer to maintain the TCP/IP connection
- Integration with the Cisco IOS software to provide intelligent network services for application connectivity, workload management, and fault tolerance

CTRC is a standards-based solution that can be managed either from the host, using mainframe management software, or from a Simple Network Management Protocol (SNMP) workstation. The following MIBs allow monitoring the CTRC router from the management platform of choice:

- CISCO-DATABASE-CONNECTION-MIB.my - 93
- CISCO-TRANSACTION-CONNECTION-MIB.my - 144

For descriptions of supported MIBs and how to use MIBs, see the Cisco MIB website on Cisco.com.

Using CTRC for CICS Access

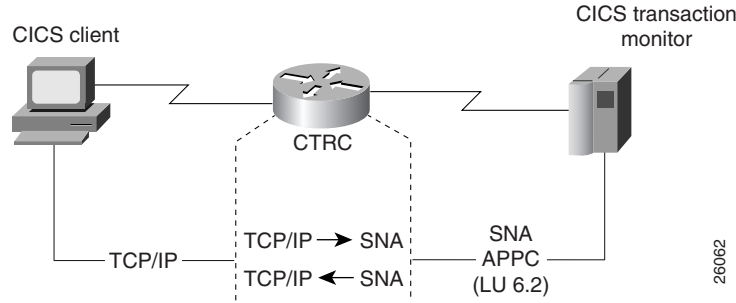
When a router is configured to use CTRC for communications with CICS systems, the router converts Inter-System Communications (ISC) packets over TCP/IP to ISC packets over Advanced Program-to-Program Communications (APPC) LU 6.2, and then routes them to the appropriate CICS region. CTRC converts CICS client messages received via TCP/IP to SNA messages and uses Cisco SNA Switching Services (SNASw) to send them to the host.

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. CTRC allows you to configure specific routes for CICS transactions, giving you control over which transaction is routed to which CICS region.

CTRC supports connectivity to CICS from the IBM Universal Client (also referred to as the Common Client), TXSeries clients, and Microsoft Common Object Module Transaction Interface (COMTI) clients. See the [Configuration CTRC Task List, page 4](#) for details on the hardware and software that CTRC supports.

[Figure 1](#) illustrates how CTRC allows CICS client applications on TCP/IP networks to interact with CICS transaction monitoring systems on IBM hosts.

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



Using CTRC for DB2 Access

In addition to its CICS-related functionality, CTRC includes the feature previously known as Cisco Database Connection (CDBC). CTRC allows Cisco routers to use IBM’s DRDA protocol to provide a gateway between client workstations on TCP/IP networks and IBM DB2 databases on SNA networks. CTRC also provides full duplex TCP passthrough to DB2 systems that support direct TCP/IP access.

Clients use a CTRC IP address and port on the router to connect to the IBM host system in either an SNA network or a TCP/IP network.

Figure 2 illustrates how the Cisco router configured with the CTRC feature enables the exchange of database information between an ODBC client application running DRDA in a TCP/IP network and a DB2 system in an SNA network. For an SNA host connection, the CTRC router converts DRDA packets over TCP/IP to DRDA packets over APPC (LU 6.2) and then routes them to DB2 databases. 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 Cisco Router Configured with the CTRC Feature for DB2 Communications (SNA Host Network)

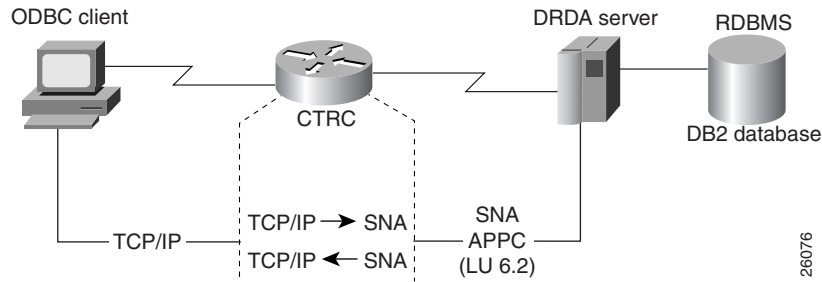


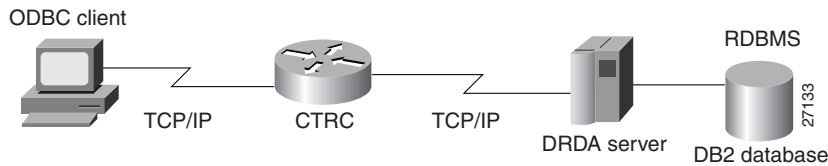
Figure 3 illustrates a configuration where CTRC supports direct TCP/IP access to DB2. For a TCP/IP host connection, CTRC routes the DRDA packets over TCP/IP without protocol changes. To use this TCP/IP passthrough feature of CTRC, the host database version must support direct TCP/IP access and the SNA Switching Services must be available.



Note

Licensing of the CTRC router is based on the cpname assigned to the router in the SNA Switching Services configuration. You must install and start SNA Switching Services with at least a minimal configuration to support the TCP/IP connections. Refer to the [“Configuring SNA Switching Services” section on page 17](#), for more information about configuring the CTRC license and the SNA Switching Services that CTRC requires.

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



Using the CTRC Keepalive Timer

In environments where there is heavy network traffic or limited processing capabilities, TCP/IP connections can time out before transactions are completed. The Keepalive Timer feature enables CTRC servers to send acknowledgment messages to clients at specific intervals to maintain the TCP/IP connection. CTRC servers that support direct TCP/IP connections to a DB2 host also can be configured to send keepalive messages to the host. The Keepalive Timer feature keeps TCP/IP connections active so they do not time out from inactivity.

Configuration CTRC Task List

CTRAC can be configured for use with CICS, with DB2, or both. Both CICS and DB2 configurations require Cisco SNA Switching Services.

General Tasks

Setting up CTRAC involves the following general tasks:

- [Defining the CTRAC Router to VTAM, page 6](#)
- [Preparing a CICS Host for Remote Access, page 7](#)
- [Preparing a DB2 Host for Remote Access, page 11](#)
- [Configuring the CTRAC Router, page 15](#)
- [Verifying the CTRAC Configuration, page 18](#)
- [Configuring CTRAC Clients, page 21](#)

To configure CTRAC for use with both CICS and DB2, complete all the configuration tasks. Otherwise, skip the sections that are related only to CICS or DB2, as appropriate for your needs. The “[CTRAC Configuration Examples](#)” section on [page 27](#) provides example configurations for using CTRAC in various network topologies.

The following sections describe the hardware and software required to use CTRAC.

Router Requirements

CTRC became available in Cisco IOS Release 12.05(XN). It is available for the following platforms:

- Cisco 7200 series routers
- Cisco 7500 series routers

CTRC consists of a system image and a microcode image, which are virtually bundled as one combined image. Within the Cisco IOS software listings, look for a software feature called Enterprise/SNASw Plus.

If you want to run CTRC on a router with a CIP card, also be sure to download the CIP hardware microcode appropriate for the Cisco IOS software level you are using.

Host Requirements

Mainframe hosts using SNA with the CTRC server must be running VTAM V3.0 or later.

CICS Host Requirements

Using CTRC for CICS access requires CICS Version 4.0 or later. CTRC supports the following CICS servers:

- CICS Transaction Server for OS/390, Version 1 or later
- CICS/400, Version 3.1
- CICS on Open Systems and NT (TXSeries)
- CICS/ESA, Version 3.3*
- CICS/ESA, Version 4.1
- CICS/MVS, Version 2.12.*
- CICS/VSE, Version 2.2*
- CICS/VSE, Version 2.3
- CICS for OS/2, Version 2.01 or later

**Note**

Versions marked with an asterisk (*) have limited server support. These versions support ECI but they *do not* support EPI or the Terminal Emulation function.

DB2 Host Requirements

When CTRC is configured for access to DB2 in an SNA network, client-based ODBC applications can connect to the following IBM DB2 relational databases:

- DB2 for OS/390 (DB2/MVS), Version 2.3 or later
- SQL/DS (DB2 for VM and VSE), Version 3.3 or later
- DB2/400 (OS/400), Version 2.2 or later
- DB2 Universal Database for UNIX, OS/2, and Windows NT, Version 5.1 or later
- DB2 Common Server, Version 2.1 or later

CTRC for DB2 access via direct TCP/IP is supported for the following versions of DB2:

- DB2 for OS/390, Version 5.1 or later (requires OS/390 Version 1.3 or later)
- DB2 for VM and VSE, Version 6.1 or later
- DB2/400 (OS/400), Version 4 Release 2 or later
- DB2 Universal Database for UNIX, OS/2, and Windows NT, Version 5.1 or later

Client Requirements

CTRC supports connectivity to DB2 from any client that supports the Level 3 DRDA. Many of the available workstation-based DRDA requestors are ODBC client applications, such as StarSQL.

CTRC supports connectivity to CICS from the following clients:

- IBM Universal Client, version 2.0 or later, using the Extended Presentation Interface (EPI) or the Extended Call Level Interface (ECI)
- IBM TXSeries for AIX or NT, version 4.2 or later, running as clients
- Microsoft COMTI

Defining the CTRC Router to VTAM

Regardless of whether you want to connect to a CICS or a DB2 host, the CTRC router must be defined to VTAM so that the host recognizes and accepts session initiation requests from it. VTAM handles network communications for MVS for direct VTAM and SNA gateway configurations. For each CTRC router, the VTAM system programmer must create a logmode table entry and major node definitions for the CTRC router link.

The following sections provide information about the logmode table entry and major node definitions required for CTRC. Consult your VTAM documentation for detailed instructions on configuring VTAM. You also may want to take advantage of VTAM's support for dynamic definition of independent LU's, which is described in the VTAM documentation.

Logmode Table Entry

The logmode table entry contains information that governs how conversations take place in VTAM. It defines pacing, RU sizes and class of service (COS) parameters. The mode entry can be placed in any mode table under VTAM—the default mode table or the one used in the APPL statement for the LU definitions. (See the [“Defining the CICS Subsystem to VTAM”](#) section on page 7 and the [“Defining the DB2 Subsystem to VTAM”](#) section on page 11 for example APPL statements).

The following example shows a logmode table entry for APPC, with a LOGMODE name of IBMRDB. Make a note of the LOGMODE name because you must use the same name for the DLOGMODE value in the major node definitions and also in the SNA configuration. The PSERVIC field identifies the LU traffic protocol—the value shown in the following example is for an independent LU using LU6.2.

```
IBMRDB    MODEENT  LOGMODE=IBMRDB,
          FMPROF=X'13',
          TSPROF=X'01',
          PRIPROT=X'B0',
          SECPROT=X'B0',
          COMPROT=X'50A1',
```

```

RUSIZES=X'8989',
TYPE=0,
PSNDPAC=X'03',
SRVCPAC=X'03',
SSNDPAC=X'02',
PSERVIC=X'060200000000000000002F00'

```

Major Node Definitions

The VTAM system programmer creates an XCA major node definition for the connection to the CTRC router. Additionally, a switched major node definition and a Cross Domain Resource definition can be created to represent the LU for the CTRC router.

In the switched major node definition, the DLOGMOD value must match the LOGMODE value in the mode table entry. The name of IBMRDB is specified for both the LOGMODE value in the previous example and in the following switched major node definition example. Make a note of the values for the LU and PU names, and the CPNAME, DLOGMOD, and CONNTYPE parameters because you must specify the same values in the SNA configuration.

```

S02CTRC      VBUILD      TYPE=SWNET
* CTRC  DOWNSTREAM  PU
CTRCPU      PU          ADDR=01,
                    CPNAME=CTRCPU,
                    ANS=CONT,
                    DISCNT=NO,
                    IRETRY=NO,
                    ISTATUS=ACTIVE,
                    PUTYPE=2,
                    SECNET=NO,
                    MAXDATA=521,
                    MAXOUT=2,
                    MAXPATH=1,
                    USSTAB=USSS,
                    MODETAB=ISTINCLM,
                    DLOGMOD=IBMRDB,
                    CONNTYPE=APPN
*
CTRCCIP  PATH  GRPNM=G02E20A, CALL=IN
*
CTRCPU  LU          LOCADDR=00,      INDEPENDENT LU
                    DLOGMOD=IBMRDB,

```

Preparing a CICS Host for Remote Access

CTRC connects to CICS using the SNA LU6.2 (APPC) communication protocol. The SNA functions are provided by a separate SNA product on the host, and CICS uses the services of that product. On a mainframe host, the SNA product is VTAM (also known as eNetwork Communications Server). You must configure both the CICS subsystem and VTAM to enable ISC.

Defining the CICS Subsystem to VTAM

The APPL statement defines the CICS subsystem to VTAM to support remote access. If your CICS subsystem is not already supporting remote access, you must create an appropriate APPL statement.

The following example shows an APPL statement that defines CICS to VTAM. Make a note of the APPL statement label, which is CICSB in this example, and the password, if one is specified, because you must specify the same values in the SNA configuration. Note that the DLOGMOD value, IBMRDB in this example, must match the LOGMODE value that is specified in the VTAM mode table entry (see the “Logmode Table Entry” section on page 6).

```
A02CICS  VBUILD  TYPE=APPL
CICSB  APPL    AUTH=(ACQ,SPO,PASS,VPACE),
          MODETAB=ISTINCLM,
          DLOGMOD=IBMRDB,
          HAVAIL=YES,
          VPACING=9,
          EAS=10000,
          PARSESS=YES,
          APPC=NO,
          SONSCIP=YES
```

Configuring CICS for ISC

To use CTRC to communicate with CICS, you must configure CICS for APPC connections. If you have configured another product, such as TXSeries for AIX, to connect to CICS, some of these steps might be completed already.

-
- Step 1** Set the ISC parameter in the CICS system initialization table (SIT) to YES. The following example overrides the CICS SIT parameters with the APPL statement label (CICSB in this example), and a value of YES for the ISC parameter.

```
APPLID=(CICSB),
GMTEXT='CICS TS V1.2',
AUXTR=OFF,
EDSALIM=80M,
FCT=NO,
ISC=YES,
MXT=100
```

- Step 2** Install the CICS-supplied resource definition group, DFHCLNT. This installation includes definitions of the CICS internal transactions, CCIN and CTIN, and of the programs they use.

- Step 3** When a CICS client sends a request, the server controller calls a routine that supports code page translations and data conversions. Regardless of whether translations and conversions are required, you need to create or modify a DFHCNV table to allow the server controller to handle incoming requests. The use of the DFHCNV macro for defining the table is described in the *CICS Family, Communicating from CICS on System/390* document. The following example shows the DFHCNV table entries:

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



Note It is not necessary to code the pages used with CICS clients on the CLINTCP and SRVERCP operands of the DFHCNV TYPE=INITIAL macro.

- Step 4** Messages relating to client support are written to the CSCC transient data queue, which you must define to CICS. There is a sample definition in the supplied resource definition group, DFHDCTG. The sample defines CSCC as an indirect extra partition destination, pointing to CSSL.
-

Defining APPC Connections to CTRC

You must install APPC connections to define the CTRC connection to CICS. This section describes the definitions and methods for installing them.

In the CONNECTION definition you specify information about the CTRC router and how it connects to CICS. The following example shows a CONNECTION definition named CTRC. Note that the NETNAME value must be the same as the CTRC router LU name, which is CTRCBOX in this example. Setting the AUTOCONNECT option to YES allows CICS to dynamically activate the router connection. See the [“Supporting CICS Security Models” section on page 10](#) for information about specifying security parameters in the CONNECTION definition.

```
DEFINE
  CONNECTION (CTRC)
  DESCRIPTION (CTRC)
  AUTOCONNECT (YES)
  NETNAME (CTRCBOX)
  ACCESSMETHOD (VTAM)
  PROTOCOL (APPC)
  SINGLESESS (NO)
  ATTACHSEC (IDENTIFY)
  BINDPASSWORD (NO)
  BINDSECURITY (NO)
  USEDFLTUSER (YES)
```

Following is an example SESSIONS definition. Note that the value for the CONNECTION parameter must be the same as the name of the CONNECTION definition, which is CTRC for this example.

```
DEFINE
  SESSIONS (CTRC)
  CONNECTION (CTRC)
  MODENAME (IBMRDB)
  PROTOCOL (APPC)
  MAXIMUM (64, 1)
  SENDSIZE (4096)
  RECEIVESIZE (4096)
```

The connections can be single- or parallel-session links. Install APPC connections to CICS either by creating static definitions for the router or using an autoinstall. The installation methods are addressed in the following sections.

Creating Static Definitions for Router Connections

You can use the CICS CEDA transaction DEFINE and INSTALL commands to create static definitions. For more information about defining APPC connections, refer to the *CICS Intercommunication Guide*.

Using Autoinstall for Router Connections

Another method of installing router connections is to use autoinstall. If you use autoinstall you must create suitable CONNECTION and SESSIONS template definitions. For information about autoinstall and defining templates, see the *CICS Resource Definition Guide*. For information about customizing your autoinstall user program to handle APPC connections, see the *CICS Customization Guide*.

Installing Client Virtual Terminals

Virtual terminals are used by the EPI and terminal emulator functions of the CICS client products. Both IBM-supplied autoinstall programs support virtual terminal autoinstall. Refer to the *CICS Customization Guide* for detailed information on autoinstall for virtual terminals.

Supporting CICS Security Models

This section addresses how to configure the the Bind, Link, and User security models that are supported in CICS.

Bind Security

Bind-time security currently cannot be configured on the Cisco router. Therefore, specify BINDSECURITY(NO) in the CONNECTION definitions that define the router to CICS.

Link Security

Link security provides the lowest level of resource security for intercommunication links. It defines the total set of resources that can be accessed across the connection.

To set link security for a CICS client connection, specify a userid for the link for the SECURITYNAME option of the CONNECTION definition. Then define a profile to your External Security Manager for the link userid. Users of the connection will be able to access only those resources that the link userid is authorized to access.

If you do not specify a userid for the SECURITYNAME option, the authority of the link is that of the CICS default user.

User Security

User (attach-time) security defines how individual users of an intercommunication link are to be checked. It also affects the resources that individual users are able to access. Unless you specify LOCAL user security (in which case all potential users share the authority of the link userid), you must define user profiles to your External Security Manager.

Preparing a DB2 Host for Remote Access

CTRC provides a gateway between DRDA client requests over TCP/IP to DB2 in SNA networks. CTRC also provides full duplex TCP passthrough to DB2 systems that support direct TCP/IP access. Perform the steps in this section if you want to use CTRC to provide access to DB2 hosts. Otherwise, skip to the [“Configuring the CTRC Router”](#) section on page 15.

Defining the DB2 Subsystem to VTAM

The APPL statement defines the DB2 subsystem to VTAM to support remote access. If your DB2 system is not already supporting remote access, you must create an appropriate APPL statement.

The following is an example of an APPL statement. Make a note of the APPL statement label, which is DSNV510 in the following example, and the password, if one is specified. You need to specify the same values when you configure or update the distributed data facility (DDF) record in the Bootstrap Data Set (BSDS) as described in the next section.

```
DB2APPL    VBUILD    TYPE=APPL
DSNV510   APPL      AUTH=(ACQ) ,
           APPC=YES,
           AUTOSES=1 ,
           DMINWNL=10 ,
           DMINWNR=10 ,
           DSESLIM=20 ,
           MODETAB=ISTINCLM,
           SECACPT=ALREADYV,
           SRBEXIT=YES,
           VERIFY=NONE,
           VPACING=2
```

Configuring DB2 for Remote Access

To use CTRC as a gateway between TCP/IP clients and the DB2 host, you need to configure and start DDF and define the CTRC router in the DB2 communications database table.

Configuring DDF

DB2 reads the BSDS during start up to obtain the system installation parameters. The DDF record in the BSDS contains information used by DB2 to connect to VTAM. If the DB2 system supports direct TCP/IP access, the DDF record specifies which port to use for TCP/IP communications.

If you are installing DB2, use the DDF installation panel DSNTIPR to provide the following parameters. If DB2 is already installed, use the change log inventory utility DSNJU003 to update this information in BSDS.

- DDF location name
- DDF LUNAME
- Password used when connecting DB2 to VTAM, if a password is required
- IP port to use for TCP/IP access

The following example updates the BSDS with a location name of DB2510, LU name of DSNV510 for SNA access, a password of STARPASS, and a port of 446 for TCP/IP communications. The RESPORT and PORT parameters are required only for TCP/IP access and can be omitted if using only SNA.

```
// *
//DSNTLOG EXEC PGM=DSNJU003,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=DSN510.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DSN5CAT.BSDS01
//SYSUT2 DD DISP=OLD,DSN=DSN5CAT.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
DDF LOCATION=DB2510,LUNAME=DSNV510,
PASSWORD=STARPASS,RESPORT=5020,PORT=446
// *
```

LOCATION is used as the Remote Database (RDB) name. If your system does not require a password to connect DB2 to VTAM, replace the PASSWORD parameter with NOPASSWD. Note the DDF LUNAME because you must specify the same value in the SNA configuration. Also make a note of the LOCATION name because you must specify the same value as the Database Server Name during data source configuration on the desktop (described in the [“Setting Up DB2 DRDA Client Connections”](#) section on page 21).



Note

You also can determine the DDF location name from the syslog. The DB2 message “DSNL004I (starting DDF)” contains the location name.

For complete information about configuring DDF, consult IBM’s DB2/MVS installation documentation.

Starting DDF

Use the following command, which requires authority of SYSOPR or higher, to start DDF:

```
-START DDF
```

When DDF starts successfully, the following messages are displayed:

```
DSNL003I - DDF IS STARTING
DSNL004I - DDF START COMPLETE LOCATION locname LU netname.luname
```

If DDF has not been properly installed, the START DDF command fails and displays the following message:

```
DSN9032I - REQUESTED FUNCTION IS NOT AVAILABLE
```

If DDF has already been started, the START DDF command fails and displays the following message:

```
DSNL001I - DDF IS ALREADY STARTED
```

Defining CTRC in the DB2 Communications Database

The DB2 host maintains a database table that defines the network attributes of remote systems. To enable communication between a CTRC client and the DB2 host, there must be an entry in this table. On DB2 for OS/390 or later versions, the name of this table is SYSIBM.LUNAMES. For DB2 on MVS v4.1, the name of this table is SYSIBM.SYSLUNAMES. Table 1 describes the table entry parameters and indicates which are applicable to one or both versions of the table.

Table 1 DB2 Communications Database Table Entry

Parameter	SYSLUNAMES	LUNAMES	Description
LUNAME	Yes	Yes	LUNAME of the remote system. An empty string means that any LU is valid for this row.
SYSMODENAME	Yes	Yes	VTAM login mode name used for DB2 for MVS/ESA intersystem conversations. A blank frame indicates that IBMDB2LM should be used. Use the mode name specified in the logmode table.
ENCRYPTPSWDS	Yes	Yes	Indicates whether passwords exchanged with this partner are encrypted. Use the default value of NO for passing passwords between a client and DB2 host using CTRC.
MODESELECT	Yes	Yes	If 'Y,' the SYSMODESELECT table is used to obtain the mode name for each outbound distributed database request. If not 'Y,' the mode name IBMDB2LM is used for system-directed access requests, and the mode name IBMRDB is used for DRDA requests.
USERNAMES	Yes	Yes	Indicates the level of come-from checking and user ID translation required. It also specifies the security parameters this DB2 for MVS/ESA subsystem uses when requesting data from the remote partner (outbound security requirements). 'I' indicates an "inbound" ID is subject to translation. 'O' indicates an "outbound" ID, sent to the corresponding LUNAME, is subject to translation. 'B' indicates that both inbound and outbound IDs are subject to translation. A blank indicates no translation for inbound or outbound IDs.
USERSECURITY	Yes	—	Network security acceptance options required of the remote system when the DB2 for MVS/ESA system acts as a server for the remote system (inbound security requirements).
SECURITY_IN	—	Yes	Defines the security options that are accepted by this host when an SNA client connects. 'V' for "verify" indicates that the incoming connection request must include a password. 'A' for "already verified" indicates the request does not require a password, although the password is checked if it is sent.
SECURITY_OUT	—	Yes	Defines the security option that is used when local DB2 SQL applications connect to any remote server associated with this LUNAME. 'A' for "already verified" indicates that outbound connection requests contain an authorization id and no password. 'P' for "password" indicates that outbound connection requests contain an authorization id and password. 'R' for "RACF PassTicket" indicates that outbound connection requests contain a userid and RACF PassTicket.

The following command inserts a row into the SYSIBM.SYSLUNAMES table that any LU can use because the value of the LUNAME column is an empty string:

```
INSERT INTO SYSIBM.SYSLUNAMES (LUNAME, SYSMODENAME, USERSECURITY, ENCRYPTPSWDS,
MODESELECT, USERNAMES) VALUES (' ', ' ', 'C', 'N', 'N', ' ');
```

The following command inserts a row into the SYSIBM.LUNAMES table that any LU can use:

```
INSERT INTO SYSIBM.LUNAMES (LUNAME, SECURITY_IN, ENCRYPTPSWDS, USERNAMES) VALUES (' ',
'V', 'N', ' ');
```

Configuring Password Expiration Management

Users of DRDA-based applications, such as StarSQL, can change their host password using CTRC's Password Expiration Management (PEM) feature. This feature is supported by CTRC using IP passthrough and APPC. PEM support for IP passthrough is provided by DB2 for OS390 V5 or later. PEM support when using APPC is provided by either APPC/MVS or CICS.

PEM Support for IP Passthrough

There is no CTRC configuration required for PEM support as it is native in DRDA over TCP/IP. However, the DB2 host must be enabled to support PEM. To enable PEM support on DB2 for OS390 V5 or later, you must configure and use extended security using either:

- The DSNTIPR (DDF) panel on the DB2 installation dialog
- A customized configuration job DSNTIJUZ, with the option EXTSEC=YES specified

Refer to the *DB2 Installation Guide* for details on setting up and using extended security.



Note

If you are using DB2 for OS390 V5, install the maintenance fix PTF UQ21052. The IBM APAR PQ15977 describes the problems fixed by this PTF. This maintenance fix is not required for later releases.

PEM Support for APPC

The CTRC PEM support over APPC is implemented using SNA architecture TPs. Therefore, CTRC requires that a surrogate subsystem such as APPC/MVS or CICS be used to change passwords. Both APPC/MVS and CICS support the SNA architecture TPs.

To allow PEM support for DB2 connections, use the **dbconn pem** command to turn on PEM support as needed for the CTRC routers handling the connections. In the **dbconn pem** command statement, specify the LU name of the APPC/MVS base configuration. APPC/MVS configuration statements are in SYS1.PARMLIB(APPCPMxx). Consult your MVS systems programmer to obtain the name of the target LU that will be used by CTRC. The PEM support does not require any explicit definitions of the SNA architecture TPs. The following example shows a LUADD statement, such as found in SYS1.PARMLIB.

```
LUADD ACBNAME(MVSLU01) BASE TPDATA(SYS1.APPCTP)
```

The following is an example VTAM APPL definition for the APPC/MVS LU:

```
MVSLU01    APPL          ACBNAME=MVSLU01,      ACBNAME FOR APPC
              APPC=YES,
              AUTOSES=0,
              DDRAINL=NALLOW,
              DLOGMOD=IBMRDB,
```

```

DMINWNL=5,
DMINWNR=5,
DRESPL=NALLOW,
DSESLIM=10,
LMDENT=19,
PARSESS=YES,
SECACPT=CONV,
SRBEXIT=YES,
VPACING=1

```

Another alternative for providing PEM support is through the CICS support for SNA architecture TPs, which is provided in resource group DFHISC. To use this method, define the connection to CTRC as described in the “[Defining APPC Connections to CTRC](#)” section on page 9, and use the CICS APPLID as the rlu value in the `dbconn pem` command.

Configuring the CTRC Router

After you define the CTRC router to VTAM and prepare the CICS and DB2 hosts for remote access, you must configure the router.

Configuring CTRC for CICS Communications

To configure CTRC to communicate with CICS, you must define a destination and specify a particular server process. You also can define specific routes to be used for particular transaction programs.

Configuring a CTRC Destination for CICS

To configure CTRC to communicate with CICS, you must configure a CTRC destination. A CTRC destination is typically a single CICS system defined in terms of its remote LU name and APPC mode. To configure a destination, use the following global configuration command:

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

If you want to assign more than one CICS system or region to a single CTRC destination name, such as to help balance the workload, repeat the `txconn destination` command with the same destination name and different remote LU and mode values. 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 for CICS

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

Command	Purpose
<pre>Router(config)# txconn server server-name destination destination-name [access {cics comti}] [client-timeout minutes] [ccsid number] [host-timeout minutes] [ipaddress ip-address] [keepalive attempts number] [keepalive interval seconds] [port port-number] [target {cics ims-tm}] >window-size bytes][fold {on off}]</pre>	Configures a CTRC server process for communicating with CICS. 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. By default, the CTRC server port for CICS client communications is 1435. You can create multiple CTRC server processes for both CICS and DB2 on one router.

Configuring a CTRC Route for 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 defined default destination. To configure a CTRC route, use the following global configuration command:

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

Configuring CTRC for DB2 Communications

To configure a CTRC server process for APPC communications with DB2, use the **dbconn server** command in global configuration mode. To configure a CTRC server to communicate with an IP-enabled DB2 database, use the **dbconn tcpserver** global configuration command.

Command	Purpose
<pre>Router(config)# dbconn server server-name [idle-timeout minutes] [ipaddress ip-address] [keepalive attempts number] [keepalive interval seconds] [mode mode] [port port-number] [rdbname rdbname] [rlu remote-lu] [tpname tp-name] [window-size bytes][wlm {off on}]</pre>	Configures a CTRC server for APPC communications with DB2.
<pre>Router(config)# dbconn tcpserver server-name remote-hostname remote-hostname remote-ip remote-ipaddress [idle-timeout minutes] [ip ip-address] [keepalive attempts number] [keepalive interval seconds] [port port-number] [rdbname rdbname] [remote-keepalive attempts number] [remote-keepalive interval seconds] [remote-port remote-port] [window-size bytes][wlm {off on}]</pre>	Configures a CTRC server to communicate with IP-enabled DB2 databases.

When a client attempts to connect to a CTRC server for DB2, the server's port, IP address, and RDB name determine whether that connection is accepted. By default, the CTRC server port for client requests for DB2 communications is 446. You can create multiple CTRC server processes for both CICS and DB2 on one router.

Configuring SNA Switching Services

CTRC uses the SNA Switching Services (SNASw) of the Cisco router. Even if you do not need to convert client messages received over TCP/IP to SNA messages (such as in a TCP/IP passthrough topology), SNASw must be present, and you must specify a CPNAME for the CTRC router. The following command illustrates the minimal SNASw configuration required to enable the CTRC license:

```
snasw cpname netid.cpname
```

To configure basic SNASw, complete the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# snasw cpname {netid.cpname netid [hostname ip-address interface-name]}	Defines an SNASw control point name. For the <i>netid.name</i> variable, specify the fully qualified CP name for the router, which consists of both network ID and cpname.
Step 2	Router(config)# snasw port portname [hpr-ip vdlc ring-group mac mac-address] interfacename [conntype nohpr len dyncplen] [nns-required] [hpr-sap hpr-sap-value] [max-links link-limit-value] [maxbtu max-btu-size] [sap sap-value] [vname virtual-node-name] [nns] [nostart]	Associates an SNASw port with an interface.
Step 3	Router(config)# snasw link linkname port portname rmac mac-address ip-dest ip-address [rsap sap-value] [nns] [tgp [high low medium secure]] [nostart]	Configures upstream links.



Note

For a LEN-level connection between SNASw and the host, you 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 SNASw and the host. See the “[Cisco IOS Software Configuration](#)” section on page 35 for an example of the SNASw configuration statements.

For additional information about configuring SNASw, consult the SNA Switching Services chapter of this document.

Configuring the CTRC License

An unlicensed installation of CTRC allows up to two DB2 connections, two CICS conversations, or one DB2 connection and one CICS conversation for evaluation purposes. To use more than two connections or conversations, you must configure the CTRC license.

The CTRC license key is locked to one node and is based on the SNASw control point name (cpname) for the router. Use the **show config | include cpname** command to determine the cpname for the router you want to license. Then contact your Cisco representative and request a CTRC license key. You will receive a license key along with information about the number of connections you are licensing and, if the license has a time limit, the expiration date.

For communications with DB2, CTRC checks the number of connections in use against the licensed number of connections. For communications with CICS, CTRC checks the number of concurrent and queued conversations. One license key is used for both CICS and DB2 communications, so you can use

either of the following global configuration commands to configure the CTRC license. If your license is not for an unlimited number of connections and period of time you must specify the number of connections and expiration date.

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

Verifying the CTRC Configuration

After preparing the host systems and configuring the CTRC router, perform the following steps to ensure CTRC can communicate with the host systems:

- Step 1** To verify that you have SNA connectivity between the router and each host system, use the **ping sna** command, specifying the mode and the fully-qualified remote LU name appropriate for your environment in place of IBMRDB and STARW.BUDDY in the following example.

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

- Step 2** If you configured CTRC for communications with CICS, perform the following steps to verify the router is properly configured. Skip to Step 3 if you are using CTRC only for DB2 communications.

- a.** Enter the **show txconn destination** command in EXEC or privileged EXEC mode. Make sure that all 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         0
GEN            CICSB              IBMRDB         0
               CICSC              IBMRDB         0
GUAVA          GUAVA              IBMRDB         0
CICSC          CICSC              IBMRDB         0
```

- b.** For each CICS destination shown in the previous step, 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
```

- c.** Enter the **show txconn server** command. Make sure that all 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    0
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
```

Use the **show txconn server** *server-name* form of the command to display detailed 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
      CCSID: 037
      number of connections: 0
      number of transactions: 0
      client type: CICS

```

- d. If you defined any routes for specific transaction IDs to take to CICS destinations, enter the **show txconn route** command. Make sure that all CTRC routes you defined are listed with the configuration values you specified. A <default> in the SERVER column indicates a global route that can be 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 3 If you configured CTRC for communications with DB2, perform the following steps to verify the router is properly configured. If you are using CTRC only for CICS communications, skip to Step 4.

- a. Enter the **show dbconn server** command. Make sure the servers you defined are listed with the configuration values you specified.

```

Router# show dbconn server
Server  Port  IPAddress      RDBName      State      NumConn
SERVERA 446  0.0.0.0        MATY         enabled    0
SERVERB 446  0.0.0.0        SCU_DSNM     enabled    0
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    0
SERVERF 446  0.0.0.0        DB2510       enabled    0
SERVERG 446  0.0.0.0        ELLE         enabled    0
SERVERH 446  0.0.0.0        SUNSET       enabled    0
SERVERI 446  0.0.0.0        NELL         enabled    0
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    0
SERVERN 446  10.10.19.4     OAK          enabled    0
SERVERO 447  0.0.0.0        DB2510       enabled    0
BUDDY   446  0.0.0.0        DB2510       enabled    0

```

Use the **show dbconn server server-name** form of the command to display more information for an individual server.

```

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.DSNV510
        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

```

- b. For each dbconn server shown in the previous step, 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 BUDDY successfully contacted!
Elapsed time was 00:00:00

```

- Step 4** Verify that the CTRC license configuration matches the number of licensed connections that you purchased. Enter either the **show dbconn license** command or the **show txconn license** command as shown below.

```

Router# show txconn license

```

```

Router# show dbconn license

```

The command displays information about the license, as shown in the following example:

```

CTRRC is licensed for 4990 connections, no licensed connections in use
This is a permanent license

```

Configuring CTRC Clients

This section provides information about setting up DRDA client connections for DB2 access, and for setting up the supported CICS clients.

Setting Up DB2 DRDA Client Connections

To configure a connection between a DRDA-based client and a DB2 database, you must define a data source to the ODBC driver. For each DB2 database that will be accessed, you need to specify the following data source information to configure the DRDA requestor to use the CTRC router:

- The RDB name of the DB2 database you want to access. This value must match the rdbname that you specify with the **dbconn server** command to configure the CTRC router for communicating with DB2 (see the [“Configuring CTRC for DB2 Communications”](#) section on page 16). The RDB name also must match the DDF location defined on the DB2 host (see the [“Configuring DDF”](#) section on page 11).
- The router’s host name or the IP address of the interface that will accept the connection requests.
- The port number on which the CTRC router is listening for connection requests. The default is 446.

The procedures for configuring a data source are specific to the client implementation. Refer to the documentation for your DRDA client for details.

Setting Up CICS Clients

CTRC supports IBM CICS Universal Client, IBM TXSeries, and Microsoft COMTI clients. These clients connect to the Cisco router via TCP/IP.

Setting Up CICS Universal Client Connections

To set up the CICS Universal Client, perform the following tasks:

-
- Step 1** Install the Universal Client for your platform.
 - Step 2** Choose TCP/IP as your network connection.
 - Step 3** To have the Universal Client connect to your CTRC server, add an entry in the Server section of the CICSCLI.INI file to define the CTRC server. The following example entry defines a server named CTRCSERV with a TCP/IP hostname (NetName) of CTRCBOX. Substitute the LU name of your router for the NetName.

```
Server = CTRCSERV  
Description = TCP/IP Server  
Protocol = TCPIP  
NetName = CTRCBOX  
Port = 1435
```

- Step 4** If necessary, stop and restart the Universal Client to have the changes take effect and connect to the CTRC server.

To connect through multiple servers, increase the `MaxServers` value in the Client section of the `CICSCLI.INI` file from the default of 1. If you have multiple servers configured in `CICSCLI.INI`, some applications may display a list of servers from which to choose. If security is turned on in CICS, a user/password dialog may appear after selecting a CICS Server.

If you have specified `UseDfltUser=NO` and `AttachSec=Verify` in your APPC CONNECTION definition on CICS (see the [“Defining APPC Connections to CTRC”](#) section on page 9), a userid and password will be required to use the CICS Terminal. If you are using ECI, pass the userid and password using a command such as:

```
cicscli /c=ctrcserv /u=p390 /p=p390
```

The CICS Terminal status line displays the virtual terminal name. When you enter a command on the terminal (such as “CEOT”), you will see the SYSID and APPLID of the CICS system to which you are connected.

Setting Up TXSeries as a CTRC Client

To connect a machine running TXSeries to another CICS host through a CTRC connection, you must create the following CICS resource definitions:

- Listener Definition
- Communications Definition
- Program Definition for each remote program you want to use

You can create these resource definitions using the `cicsadd` command, or you can use the CICS System Management Interface Tool (SMIT) to build the commands. The following sections describe both methods.



Note

The procedures in the following sections show how to create the resource definitions for TXSeries on AIX. If you are using TXSeries on Windows NT, refer to your TXSeries documentation for the commands and configuration panels provided for creating resource definitions on that platform.

Using `cicsadd` to Create the Definitions

To use the `cicsadd` command to add CICS resource definitions on TXSeries for AIX, specify the values appropriate for your definition in place of the variables shown in *italic* in the following command syntax.

```
cicsadd -c className[-r regionName] [-P | -B] [-f fileName] [-m modelId] resourceName
[attributeName=attributeValue ...]
```

To use the CTRC router, the value for the *resourceName* in the Communications Definition (CD) must be the same as the *attributeValue* specified for the RemoteSysId attribute in the Program Definition. And, the ListenerName specified in the CD must match the name of the Listener Definition. For example, issuing the following command creates a Communications Definition for the CTRC router with a *resourceName* of CTRC and a ListenerName of TCP:

```
cicsadd -c cd -r TX6000 -B CTRC ResourceDescription="Connection thru CTRC"
ConnectionType=cics_tcp ListenerName=TCP OutboundUserIds=sent RemoteCodePageTR="IBM-037"
RemoteNetworkName="CICSB" RemoteSysSecurity=trusted RemoteTCPAddress="ctrcbox"
RemoteTCPPort=1435 RemoteLUName="CTRCBOX"
```

To use a remote program named PNG1, the Program Definition for PNG1 must set the RemoteSysId attribute to CTRC, as shown in the following command.

```
cicsadd -c pd -r TX6000 -B PNG1 ResourceDescription="eciPing back end" RemoteSysId=CTRC
RemoteName=PNG1 RSLKey=public
```

You specify the protocol that the CICS client will use in the Listener Definition. For example, to allow the TXSeries client to connect to the CICS region specified in the above example commands, TX6000, add a Listener Definition for TCP/IP as shown in the following command.

```
cicsadd -c ld -r TX6000 -B TCP ResourceDescription="TCP/IP Listener" Protocol=TCP
```

Using SMIT to Create the Definitions

To use SMIT to build the commands for creating the resource definitions, start SMIT and display the Manage Resources menu, which lists the types of definitions you can create.

Following are example definitions, assuming the values below for the CTRC-related parameters:

- TX6000—Name of the CICS region on an RS/6000 running TXSeries.
- CTRCBOX—IP host name of CTRC router.
- CICSB—APPLID of CICS server running on a mainframe.
- PNG1—ECI host program running on the mainframe.

Listener Definition Example

```
* New Listener Identifier [TCP]
* Listener Identifier TCP
* Region name TX6000
  Update Permanent Database OR
    Install OR Both Both
  Group to which resource belongs []
  Activate resource at cold start? yes
  Resource description [Listener Definition]
* Number of updates 0
  Protect resource from modification? no
  Protocol type TCP
  TCP adapter address [198.147.235.8]
  TCP service name []
  local SNA Server Protocol Type TCP
  local SNA Server Identifier []
  local SNA Node Name []
  local Named Pipe name []
```

Communication Definition Example

The following definition shows a TCP/IP link to a CICS host STARW.CICSB through the CTRC router named CTRCBOX:

```

New Communication Identifier          [CTRC]
Communication Identifier              CTRC
Region name                          TX6000
Update Permanent Database OR
    Install OR Both                   Both
Group to which resource belongs      []
Activate the resource at cold start?  yes
Resource description                  [Communications Definit>
* Number of updates                   2
Protect resource from modification?   no
Connection type                       cics_tcp
Name of remote system                 [CICSB]
SNA network name for the remote system [STARW]
SNA profile describing the remote system []
Default modename for a SNA connection []
Gateway Definition (GD) entry name    []
Listener Definition (LD) entry name   [TCP]
TCP address for the remote system     [CTRCBOX]
TCP port number for the remote system [1435]
DCE cell name of remote system        [././]
Timeout on allocate (in seconds)      [60]
Code page for transaction routing     [IBM-037]
Set connection in service?            yes
Send userids on outbound requests?    sent
Security level for inbound requests   verify
UserId for inbound requests           []
Transaction Security Level (TSL) Key Mask [none]
Resource Security Level (RSL) Key Mask [none]
Transmission encryption level         none

```

Program Definition Example

The following definition describes a program named PNG1 that is running on the remote system accessed through the Communication Definition named CTRC (see the [“Communication Definition Example”](#) section on page 24):

```

New Program Identifier                [PNG1]
Program Identifier                    PNG1
Region name                          TX6000
Update Permanent Database OR Install
    OR Both                            Both
Group to which resource belongs      []
Activate resource at cold start?     yes
Resource description                  [Program Definition]
* Number of updates                   0
Protect resource from modifications?  no
Program enable status                 enabled
Remote system on which to run program [CTRC]
Name to use for program on remote system [PNG1]
Transaction name on remote system for program []
Resource Level Security Key           [public]
Program path name                     []
Program type program                  []
User Exit number                      [0]
Is a user conversion template defined? no
Is this a program that should be cached? no

```

Refer to the IBM TXSeries CICS documentation for more information about specifying CICS resource definitions on TXSeries.

Setting Up COMTI Client Connections

When a COMTI application is built using Microsoft's COMTI Component Builder, it must be defined with the following information to provide remote access to CICS.

- "CICS and IMS via TCP/IP" as the remote environment type
- "CICS" as the target environment
- "MS Link" as the server mode

For the COMTI client to access CICS using the CTRC router, you must define CTRC as a TCP Remote Environment. Use Microsoft's COMTI Manager to define the remote environment with the following values.

- Select "CICS and IMS using TCP/IP" as the remote environment type
- Specify the IP address and TCP port address as configured on the CTRC router
- Specify a name and comment for the new remote environment

Refer to the *Microsoft COM Transaction Integrator Online Guide* for details about setting up and using COMTI.

Monitoring and Maintaining CTRC

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



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.

Monitoring and Maintaining CTRC Communications with CICS

To monitor and maintain CTRC communications with CICS, use the following commands in privileged EXEC mode:

Command	Purpose
Router# clear txconn connection <i>connection-id</i>	Terminates the specified CTRC connection to a CICS client and all associated transactions.
Router# clear txconn statistic <i>name</i> { allocatetime clientreceived clientsent clientturnaround every hostreceived hostresponse hostsent maxconnections maxtransactions totalconnections totaltransactions }	Clears the named statistic or all statistics (every keyword) related to CTRC communications with CICS.
Router# clear txconn transaction <i>transaction-id</i>	Terminates the specified CTRC transaction.
Router# debug txconn { all appc config data event tcp timer }	Enables debugging of CTRC communications with CICS.
Router# show debugging	Displays current status of debugging for the router.

Command	Purpose
Router# show txconn connection [server <i>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 destination <i>destination-name</i>	Displays a list of all the current router's destinations for CICS communications, or detailed status information for the specified CTRC destination.
Router# show txconn license or show dbconn license	Shows the status of the CTRC license.
Router# show txconn route [server <i>server-name</i>]	Displays a list of CTRC routes to CICS for the current router or a particular server.
Router# show txconn server	Lists the CTRC servers that are configured for CICS communications on the current router.
Router# show txconn server <i>server-name</i>	Displays detailed status information for the specified CTRC server.
Router# show txconn transaction [server <i>server-name</i> connection <i>connection-id</i>]	Displays a list of the current router's CTRC transactions with CICS, or the transactions of a particular server or connection.
Router# show txconn transaction <i>transaction-id</i>	Displays detailed status information for the specified CTRC transaction.
Router# show txconn statistic [kind { histogram summary }] name { activeconnections activetransactions allocatetime clientreceived clientsent clientturnaround dump hostreceived hostresponse hostsent latency maxconnections maxtransactions totalconnections totaltransactions }	Displays statistics related to CTRC communications with CICS.
Router# txconn ping <i>destination-name</i>	Tests communications between the CTRC router and a CTRC destination (a host defined by a pair of RLU and mode values).

Monitoring and Maintaining CTRC Communications with DB2

To monitor and maintain CTRC communications with DB2, use the following commands in privileged EXEC mode:

Command	Purpose
Router# clear dbconn connection <i>connection-id</i>	Breaks the specified client connection to the server.
Router# clear dbconn statistic { chains clientturnaround connectionsdown connectionsup every hostreceived hostresponse hostsent maxconnections }	Clears statistics related to CTRC communications with DB2.
Router# dbconn ping <i>server-name</i> [userid <i>userid</i>] [password <i>password</i>] [rdbname <i>rdbname</i>]	Verifies connectivity to the specified DB2 database.
Router# debug dbconn { all appc config drda event tcp }	Enables debugging of CTRC communications with DB2.

Command	Purpose
Router# show dbconn connection	Displays the status of each CTRC connection to DB2.
Router# show dbconn connection <i>connection-id</i>	Displays a detailed status of the specified CTRC connection to DB2.
Router# show dbconn connection server <i>server-name</i>	Displays the status of CTRC connections to DB2 for the specified server.
Router# show dbconn connection userid <i>userid</i>	Displays the status of a user connected to CTRC for DB2 communications.
Router# show dbconn connection rdbname <i>rdb-name</i>	Displays a status of each connection to DB2 that matches the specified RDB name.
Router# show dbconn license OR Router# show txconn license	Displays the status of the CTRC license for both DB2 and CICS.
Router# show dbconn ports	Displays information on all ports through which CTRC servers are accepting connections to DB2.
Router# show dbconn server	Displays a summary of information about each CTRC server configured to communicate with DB2.
Router# show dbconn server <i>server-name</i>	Displays a detailed status of the specified CTRC server for DB2 communications.
Router# show dbconn statistic [<i>kind</i> { <i>histogram</i> <i>summary</i> }] name { <i>chains</i> <i>clientturnaround</i> <i>connectionsdown</i> <i>connectionsup</i> <i>dump</i> <i>hostreceived</i> <i>hostresponse</i> <i>hostsent</i> <i>latency</i> <i>maxconnections</i> }	Displays current statistics related to CTRC communications with DB2.
Router# show debugging	Displays current status of debugging for CTRC.

CTRC Configuration Examples

The following sections provide CTRC configuration examples:

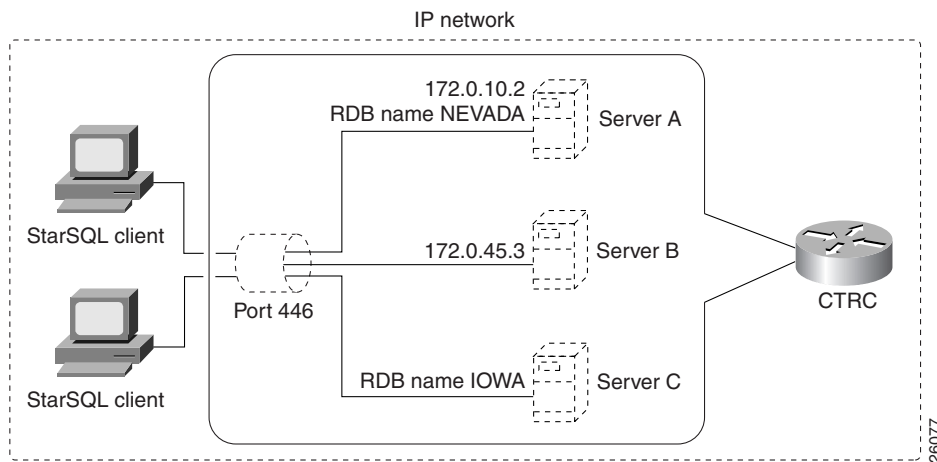
- [CTRC Servers with IP Addresses Configuration Example \(DB2\), page 28](#)
- [CTRC Servers with IP Addresses, RDB Names, and Ports Configuration Example 1 \(DB2\), page 28](#)
- [CTRC Servers with IP Addresses, RDB Names, and Ports Configuration Example 2 \(DB2\), page 29](#)
- [Server Selection by IP Addresses, RDB Names, and Ports Configuration Example \(DB2\), page 29](#)
- [CTRC with CIP and DB2 on VTAM Configuration Example \(DB2\), page 30](#)
- [CTRC Servers Using Token Ring to a LEN Configuration Example \(CICS and DB2\), page 33](#)
- [CTRC Servers with IP Addresses, Routes, and Multi-Valued Destinations Configuration Example \(CICS\), page 36](#)

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 Server A, Server B, and Server C in the Cisco router:

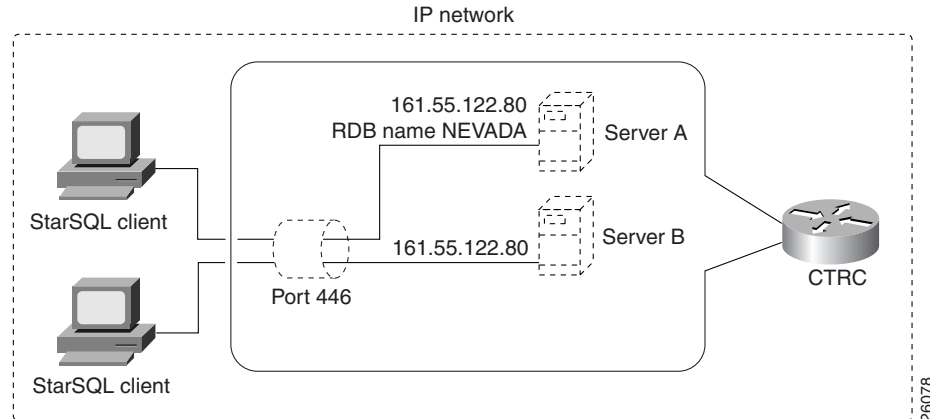
```
dbconn server SERVERA ip-address 172.0.10.2 rdbname nevada
dbconn server SERVERB ip-address 172.0.45.3
dbconn server SERVERC rdbname iowa
```

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)

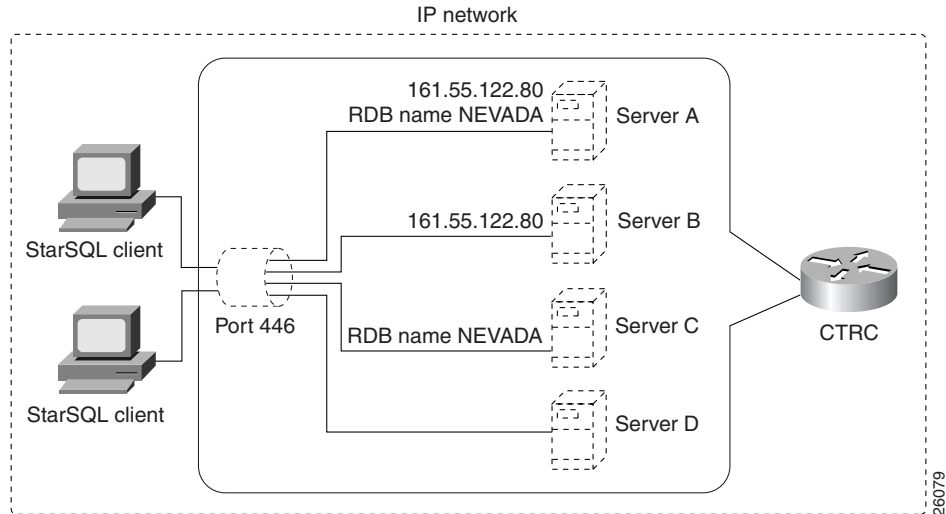
The IP address and port specified for a server in a router's configuration also determines which server accepts a connection. For example, Server C is configured to listen on any local IP address on port 446 and RDB name IOWA. Server D is configured to listen for IP address 145.56.180.34 on port 446 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:

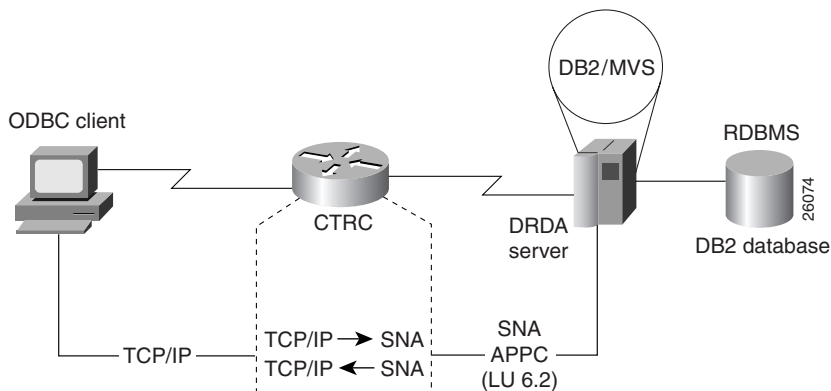
```
hostname routera
!
enable password allie

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 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 examples show the configuration in more detail.

In the VTAM host definitions, the variable CONNTYPE=APPN is optional, but is 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.

VTAM Partner PU and LU Definition

```

CTRCPU PU ADDR=01, X
        IDBLK=05D, X
        IDNUM=00501, X
        CPNAME=CTRCPBOX, 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
CTRCPBOX 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

```

XCA for a CIP-Attached Router

```

XCAE20 VBUILD TYPE=XCA
XPE20R PORT CUADDR=E20,
          ADAPNO=1,
          SAPADDR=4,
          MEDIUM=RING,
          DELAY=0,
          TIMER=60
G02E20A GROUP ANSWER=ON, CALL=INOUT, DIAL=YES, ISTATUS=ACTIVE
K02T201S LINE
P02T201S PU
K02T202S LINE
P02T202S PU

```

Cisco IOS Software Configuration

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

```

!
source-bridge ring-group 100
!
interface Ethernet2/1
  mac-address 4200.0000.0501
  ip address 198.147.235.11 255.255.255.0
  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.CTRCBOX
snasw port SRB Virtual-TokenRing0
snasw link BUDDY port SRB rmac 4000.0123.9999
snasw location DSNV510 owning-cp STARW.BUDDY (see Note below)

!
dbconn server DB2BUDD rdbname DB2510 rlu STARW.DSNV510 mode IBMRDB
!
ip default-gateway 198.147.235.12
ip classless

```



Note

Do not use an **snasw location** statement if you are using an APPN connection between the host and SNASw.

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

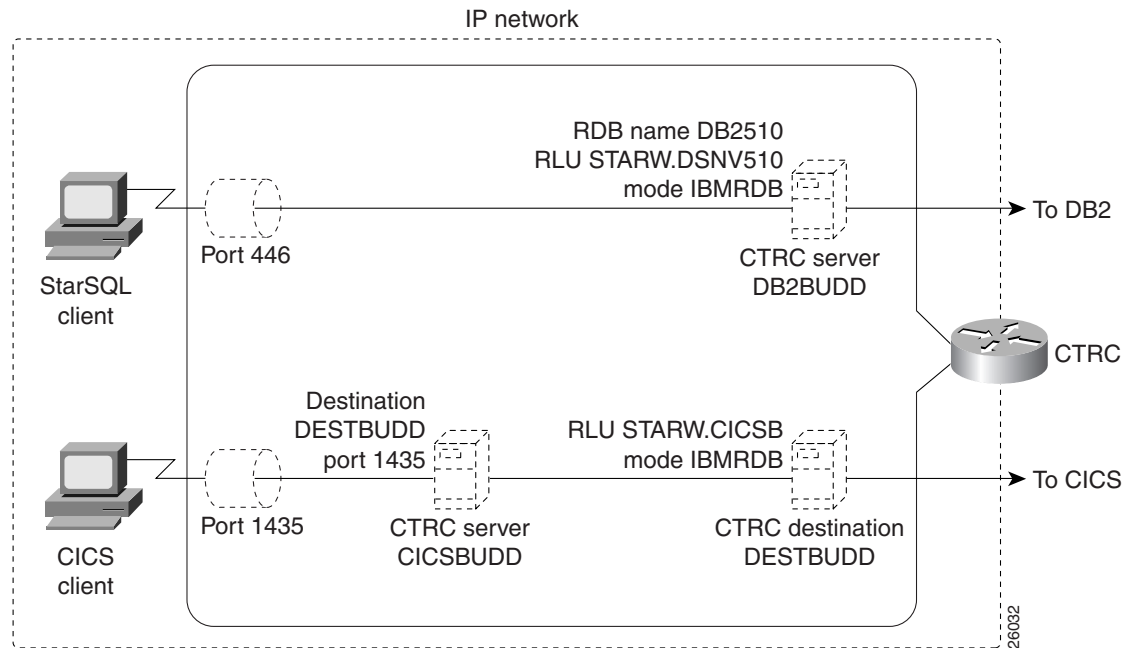
This section provides a configuration example for a router named CTRCBOX, beginning with the VTAM definition for the router, which is the same as for the previous example.

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

```

CTRCPU PU ADDR=01, X
        IDBLK=05D, X
        IDNUM=00501, X
        CPNAME=CTRCBOX, 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
        ONNTYPE=APPN
CTRCBOX LU LOCADDR=00, INDEPENDENT LU X
        DLOGMOD=IBMRDB
    
```

VTAM APPLID for CICS

```

CICSAPPL VBUILD TYPE=APPL                                00010001
*****
* CICS APPL DEFINITION FOR LU62 CLIENT/SERVER SUPPORT    00020000
*****
* CICSAPPL VBUILD TYPE=APPL                                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=CICSB

```

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

```

VTAM APPLID for PEM Support

```

MVSLU01   APPL   ACBNAME=MVSLU01,      ACBNAME FOR APPC
          APPC=YES,
          AUTOSES=0,
          DDRAINL=NALLOW,
          DLOGMOD=IBMRDB,
          DMINWNL=5,
          DMINWNR=5,
          DRESPL=NALLOW,
          DSESLIM=10,
          LMDENT=19,
          PARSESS=YES,
          SECACPT=CONV,
          SRBEXIT=YES,
          VPACING=1

```

DB2 BSDS DDF Record

The following example updates the BSDS with a location name of DB2510, LU name of DSNV510 for SNA access, a password of STARPASS, and a port of 446 for TCP/IP communications. The RESPORT and PORT parameters are required only for TCP/IP access and can be omitted if using only SNA.

```

/*
//DSNTLOG EXEC PGM=DSNJU003, COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=DSN510.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DSN5CAT.BSDS01
//SYSUT2 DD DISP=OLD,DSN=DSN5CAT.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
          DDF LOCATION=DB2510, LUNAME=DSNV510,
          PASSWORD=STARPASS, RESPORT=5020, PORT=446
/*

```

XCA for Token Ring Attached Router

```

XCAE40  VBUILD  TYPE=XCA
XPE40R  PORT    CUADDR=E40,
          ADAPNO=1,
          SAPADDR=4,
          MEDIUM=RING,
          DELAY=0,
          TIMER=30
G02E40A  GROUP  DIAL=YES, CALL=INOUT, ANSWER=ON, ISTATUS=ACTIVE
*
K02T001S  LINE
P02T001S  PU
*
K02T002S  LINE
P02T002S  PU

```

Cisco IOS Software Configuration

```

source-bridge ring-group 100
!
!
interface TokenRing0/1
  mac-address 4000.1111.0501
  ip address 198.147.236.196 255.255.255.0
  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 198.147.235.11 255.255.255.0
  no ip directed-broadcast
  no ip mroute-cache
!
!
snasw cpname STARW.CTRCBOX
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
dbconn tcpserver BUDDTCP port 446 rdbname DB2510 remote-ip-address 198.147.235.39
remote-port 446
      dbconn pem DB2BUDD rlu MVSLU01 mode #INTER
!
txconn destination DESTBUDD rlu STARW.CICSB mode IBMRDB
txconn server CICSBUDD destination DESTBUDD port 1435
ip default-gateway 198.147.235.12
ip classless

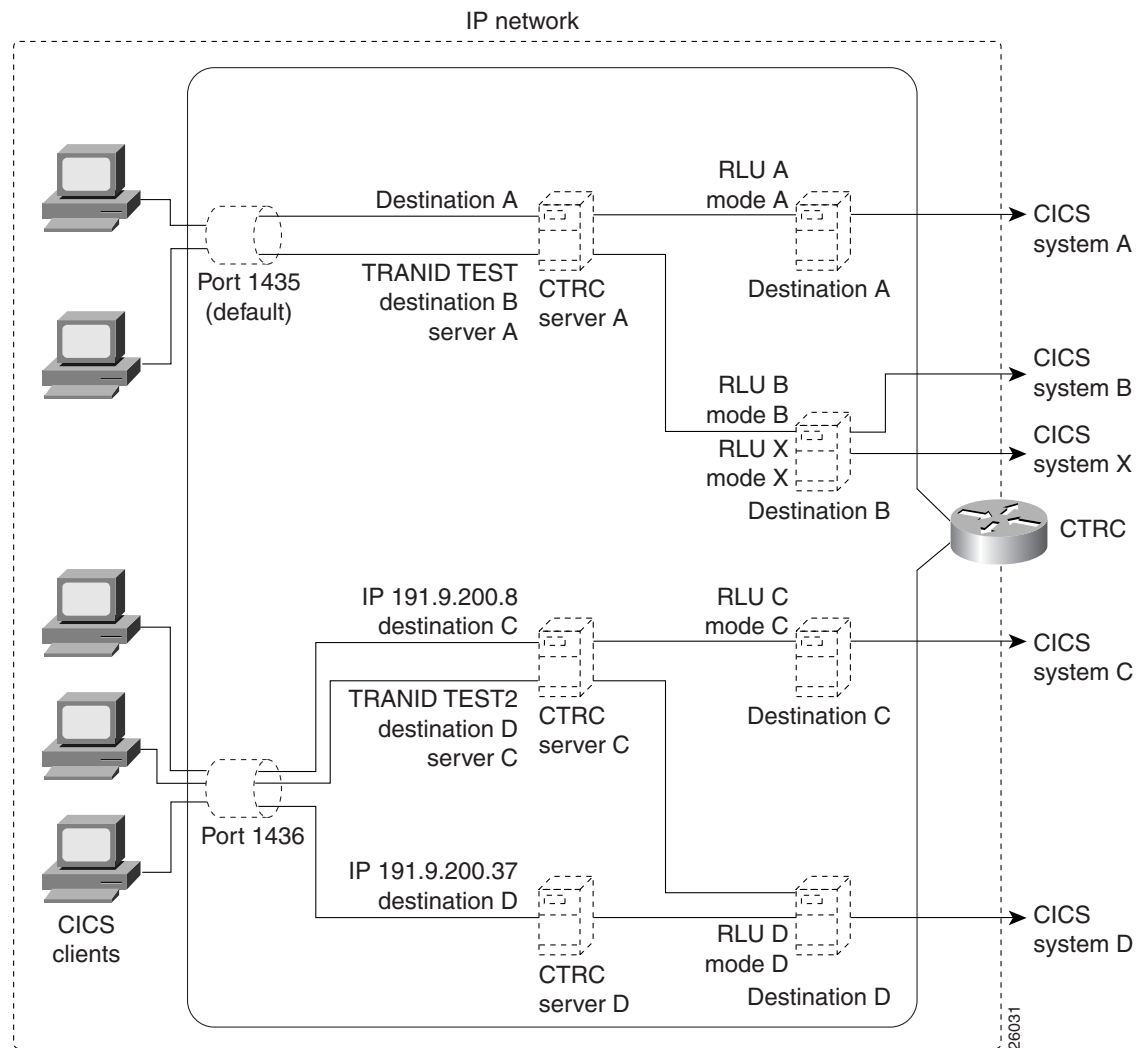
```

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. 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.

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). CTRC allocates transactions 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.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2008 Cisco Systems, Inc. All rights reserved.

