CISCO SYSTEMS

# Cisco Carrier Sensitive Routing User Guide

Version 1.1

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED ORIMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

*Cisco Carrier Sensitive Routing User Guide*

**C O N T E N T S**

# Preface

This preface describes the objective, audience, organization, and conventions of the *Cisco Carrier Sensitive Routing User Guide*.

It contains the following information:

## Overview

The *Cisco Carrier Sensitive Routing User Guide* provides information on operations and functionalities of Carrier Sensitive Routing (CSR) application, its graphical user interface (GUI) and the databases associated with it.

## Who Should Use This Guide

Users of this guide should have knowledge of the following:

- UNIX operating system and commands
- SQL commands
- TCP/IP network that the CSR is connected to
- Sun/Solaris™ computer system

# Document Organization

This guide is organized as follows:

| | |
|---|---|
| Chapter1, "Overview of Cisco Carrier Sensitive Routing" | Provides an overview of the postgreSQL database, CSR application, and CSR graphical user interface (GUI). |
| Chapter2, "PostgreSQL Database Operations and Functionalities" | Describes the operations and functionalities of the postgreSQL database. |
| Chapter3, "Cisco Carrier Sensitive Routing Application Operations" | Describes the operations and functionalities of the CSR application. |
| Chapter4, "CSR GUI Functionalities" | Describes the CSR graphical user interface functionalities. |
| Appendix A, "Logging" | Describes the configuration of syslog and viewing logs. |
| Appendix B, "Data Schema" | Describes the configuration data schema and dataset schema. |
| Appendix C, "Error Conditions and Recoveries" | Describes database and gatekeeper connectivity loss. |

# Document Conventions

Following are conventions that might be used in this document.

**Table1**

| Convention | Description |
|---|---|
| **boldface** | Commands and keywords are in **bold face**. |
| *italic* | Arguments for which you supply values are in *italic*. |
| [    ] | Keywords or arguments that appear within square brackets are optional. |
| { *x* \| *y* \| *z* } | Alternative, mutually exclusive, keywords are grouped in braces and separated by vertical bars. |
| ^ orCtrl | Represent the key labeled *Control* . For example, when you read *^D* or *Ctrl-D*, you should hold down the Control key while you press the D key. |
| `screen font` | Terminal sessions and information the system displays are in `screen font`. |
| **`boldface screen font`** | Information you must enter is in **`boldface screen`** `font`. |
| <    > | Nonprinting characters, such as passwords, appear in angled brackets. |
| [    ] | Default responses to system prompts are in square brackets. |

# Related Documentation and References

- Carrier Sensitive Routing Installation Guide

For Sun/Solaris™, Java, and postgreSQL database information, go to the following URLs:

- http://www.sun.com
- http://java.sun.com
- http://www.postgresql.org

# Obtaining Documentation

These sections explain how to obtain documentation from Cisco Systems.

## World Wide Web

You can access the most current Cisco documentation on the World Wide Web at this URL:

http://www.cisco.com

Translated documentation is available at this URL:

http://www.cisco.com/public/countries_languages.shtml

## Documentation CD-ROM

Cisco documentation and additional literature are available in a Cisco Documentation CD-ROM package, which is shipped with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

## Ordering Documentation

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, U.S.A.) at 408526-7208 or, elsewhere in North America, by calling 800553-NETS (6387).

## Documentation Feedback

You can submit comments electronically on Cisco.com. In the Cisco Documentation home page, click the **Fax** or **Email** option in the "Leave Feedback" section at the bottom of the page.

You can e-mail your comments to bug-doc@cisco.com.

You can submit your comments by mail by using the response card behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Document Resource Connection
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

# Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain online documentation, troubleshooting tips, and sample configurations from online tools by using the Cisco Technical Assistance Center (TAC) Web Site. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC Web Site.

## Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com is a highly integrated Internet application and a powerful, easy-to-use tool that provides a broad range of features and services to help you with these tasks:

 •  Streamline business processes and improve productivity

 •  Resolve technical issues with online support

 •  Download and test software packages

 •  Order Cisco learning materials and merchandise

 •  Register for online skill assessment, training, and certification programs

If you want to obtain customized information and service, you can self-register on Cisco.com. To access Cisco.com, go to this URL:

http://www.cisco.com

## Technical Assistance Center

The Cisco Technical Assistance Center (TAC) is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two levels of support are available: the Cisco TAC WebSite and the Cisco TAC Escalation Center.

Cisco TAC inquiries are categorized according to the urgency of the issue:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.

- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.

- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.

- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

The Cisco TAC resource that you choose is based on the priority of the problem and the conditions of service contracts, when applicable.

## Cisco TAC Web Site

You can use the Cisco TAC Web Site to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC Web Site, go to this URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco service contract have complete access to the technical support resources on the Cisco TAC Web Site. The Cisco TAC Web Site requires a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to this URL to register:

http://www.cisco.com/register/

If you are a Cisco.com registered user, and you cannot resolve your technical issues by using the Cisco TAC Web Site, you can open a case online by using the TAC Case Open tool at this URL:

http://www.cisco.com/tac/caseopen

If you have Internet access, we recommend that you open P3 and P4 cases through the Cisco TAC WebSite.

## Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses priority level 1 or priority level2 issues. These classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer automatically opens a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to this URL:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

Before calling, please check with your network operations center to determine the level of Cisco support services to which your company is entitled: for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). When you call the center, please have available your service agreement number and your product serial number.

# Overview of Cisco Carrier Sensitive Routing

The Cisco Carrier Sensitive Routing (CSR) application provides end users with the capability to manipulate the routing of calls from the gatekeeper based on the ingress carrier and the DNIS. The routing can be based on QoS along with many other attributes pertaining to a carrier. With CSR, you can provision data specific to carriers that pertain to your network to maximize cost, QoS, and carrier relations. CSR can run on a Sun/Solaris system that has a network connection to gatekeepers.

This chapter contains the following sections:

## Prerequisites

- Root access to a UNIX machine for the following tasks:
  - Create users
  - Modify Syslog.conf
  - Setting database security
- Configured Cisco gatekeeper and gateway
- Network that provides DNS capabilities
- Knowledge of the following:
  - UNIX operating system and commands
  - SQL commands
  - TCP/IP network that the CSR is connected to
  - Sun computer system

# CSR Components

The three major components of CSR are as follows.

- postgreSQL Database
- CSR application
- CSR GUI

The CSR GUI and the CSR application are not directly connected. They are both connected to the database. Figure1-1 illustrates these components.

***Figure1-1    CSR Components***
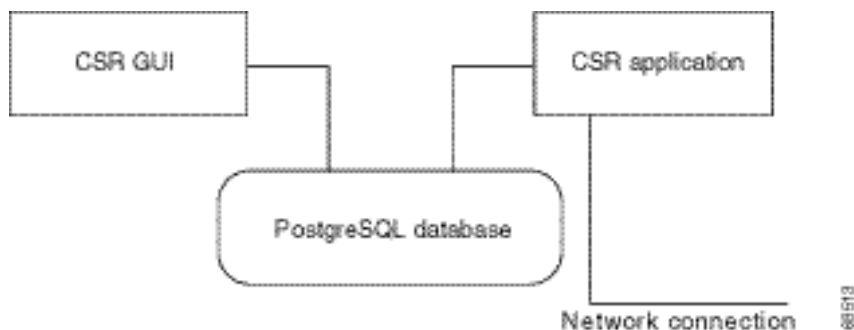


# CSR Basics

This section describes the basic operation of CSR.

After CSR is started, it registers with the gatekeeper connected to a network. Any one or a combination of the following registration messages are used:

- ARQ
- LRQ
- DRQ
- BRQ
- URQ
- RAI
- LCF
- LRJ
- IRR

The gatekeeper responds with any of the following messages:

- ANI
- DNIS
- Source carrier
- Destination carrier

CSR evaluates the predetermined selection and rejection criteria as part of the source carrier and DNIS information to determine what routing information to return to the gatekeeper.

**Note**    CSR can operate with multiple gatekeepers. CSR passes through the ANI, but its work is not based on the ANI. If the CSR receives the destination carrier, it passes the call through with the same information that it receives.

# PostgreSQL Database

The postgreSQL database stores all the provisioned data associated with the CSR. The data can be entered into the database by using the GUI or by importing. The database supports active dataset, inactive dataset, and configuration data.

Information contained in the active and inactive datasets includes:

- Carrier Table—Contains a list of carriers and associated attributes.
- Contact List Table—Contains a relationship of contacts to a key.
- Contacts Table—Contains a list of RAS addresses (zones) and associated attributes.
- Egress Route Attributes Table—Contains a list of egress attributes and associated data.
- Egress Routes Table—Contains a list of egress attributes and associated routes.
- Ingress Route Attributes Table—Contains a list ingress attributes and associated data.
- Ingress Routes Table—Contains a list of ingress attributes and associated routes.
- Match Patterns Table—Contains a list of match patterns and associated routes.
- QoS Table—Contains a list of relative QoS values associated to route, ingress, and egress.
- Routes Table—Contains a list of routes and associated rules.
- Rules Table—Contains a list of rules and associated attributes (selection/rejection).

Configuration data contained in the database includes:

- Gatekeepers Table—Contains a list of gatekeepers and associated registration messages.
- Filters Table—Contains a list of gatekeepers and associated filters.
- Configuration Settings Table—Contains logging level, application priority, incoming connection port and test routes accounting flag.
- TestRoutes—A record is added to this when a **test route** command is issued.
- TestOutputs—A set of records is added to this table after the **test route** command is complete.
- LogEntries—A set of records is added to this table after the **test route** command is complete.
- StringResources—Logging strings are located here.

# CSR Application

The CSR application contains the logic (selection and rejection) to determine the proper routing for calls based on inputs from the gatekeeper. CSR operates by the rules described in the following section.

## Ingress Rejection Rules

- Reject Max Origination Percentage
- Reject Max Ingress Units
- Reject Max Ingress Units per Route

## Egress Rejection Rules

- Reject Egress Cost Greater than Ingress
- Reject Insufficient QoS
- Reject Max Egress Units Exceeded
- Reject Max Egress Units per Route Exceeded
- Prevent ITSP to ITSP

## Selection Rules

- Select Min Cost per Egress Route
- Select Best QoS
- Select Same Carrier
- Select Percentage Egress
- Select Preferred Carrier

CSR operates on an internal copy of the active dataset, which is loaded from the active dataset of the postgreSQL database. CSR can also operate without the postgreSQL database running as long as it has an active dataset. For more information on the active dataset, see the "Loading the Active Dataset" section on page4-44.

# CSR GUI

The CSR GUI provides an interface that can be used to enter, change, and modify provisioned data in the postgreSQL database. You can import data into the database with the import capabilities provided by postgreSQL. The CSR GUI can run on Sun/Solaris and Microsoft NT platforms.

# CSR Limitations

- When two rules are available (a rule associated with the carrier and a rule associated with a route), the rule associated with the carrier is chosen first. If one of the rules (carrier's rule) does not exist, the rule associated with a route is used.

- When two contact lists are available (the contact list associated with the carrier and the contact list associated with the EgressRouteAttribute), the carrier's contact list is used first. If one of the contact lists (carrier's contact list) does not exist, then the contact list associated with the EgressRouteAttribute is used.

- If a contact is provisioned with a DNS name and a DNS server is not found—that is, cannot get an IP address—the dataset verification fails and the CSR cannot be started correctly.

- Static triggers are not supported by the CSR.

- Performance may deteriorate if more than five gatekeepers are connected to the CSR. This is also related to hardware capabilities and call volume.

**CSR Limitations**

**2**

# PostgreSQL Database Operations and Functionalities

This chapter describes the operations and functionalities of the postgreSQL database and contains the following sections:

The postgreSQL database is a relational database management system. A database in this management system stores entities such as carriers, rules, contacts, routes, and the relationships among the entities. The management system can manage many different databases simultaneously. The CSR creates a database in this system and performs call routing based on the data in the database.

## Database Operations

This section describes the following tasks:

- Activating the Database
- Exiting the Database
- Reactivating the Database
- Exporting a Database
- Reintializing a Database

## Activating the Database

To activate the postgreSQL database, perform the following steps:

**Step 1**  Log in to the Solaris system by entering **postgres** as the user.

A "postgres" password prompt may appear.

**Step 2**  Enter the following command:

```
/etc/init.d/postgres_init start
```

This command calls postgreSQL command **postmaster -S -i -B 40 -N 20 -D /usr/local/pgsql/data**. The parameters are described as follows:

- -S—Tells the postmaster to run in silent mode. This is not recorded to a log file.
- -i—Tells postmaster to allow network connections. If this is not set, the postmaster accepts connections only from the local host.

> **Note** This is needed for the GUI to connect to the database even when it is running on the same machine.

- -B and -N—Set up shared memory segments.
- -D—Tells the postmaster the location of the database files.

> **Note** For servers with console login, disable Xwindow in the file **.profile**, so the database can be restarted from a reboot.

# Exiting the Database

To exit the postgreSQL database, perform the following steps:

**Step 1**   Log in to the Solaris system by entering **postgres** as the user.

A "postgres" password prompt may appear.

**Step 2**   Enter the following command:

`/etc/init.d/postgres_init stop`

This command calls the postgreSQL command **pg_ctl stop**.

# Reactivating the Database

To reactivate the postgreSQL database, perform the following steps.

**Step 1**   Log in into the Solaris system by entering **postgres** as the user.

A "postgres" password prompt may appear.

**Step 2**   Enter the following command:

`/etc/init.d/postgres_init restart`

This command calls the postgreSQL command **pg_ctl restart**. The parameters for this command are described in "Activating the Database" section on page2-1.

# Exporting a Database

To export a database to a text file or to back up the entire database system, perform the following steps:

## Exporting to a Text File

**Step 1**    Export a database into a text file by entering the following command:

```
pg_dump –x databasename > outfile
```

**Note**    "-x" is optional. When it is used, the "dumped" database loses the ownership information.

**Step 2**    Reload the dataset by entering the following command:

```
psql –e database < outfile
```

**Note**    "-e" is optional. When it is used, every log message "echoes" on the screen.

## Backing Up the Database System

You can back up the database system file directory /usr/local/pgsql/data/base into a tar file.

# Reintializing a Database

To reinitialize a database, the existing database must be destroyed, and a new one must be created.

To reinitialize a database, performing the following steps:

**Step 1**    Destroy the database by entering the following command with "csrdb" as the name of the database:

```
dropdb csrdb
```

**Note**    The database to be destroyed must not be connected to the CSR application, CSR GUI, and pgsql.

**Step 2**    Create a new database by entering the following command:

```
createcsrdb csrdb
```

After the database is re-created, the CSR and CSR GUI can be restarted and connected to the new database. If "csrdb" is not used as the name of the database, note the following:

- To have CSR started upon reboot of the Solaris system, modify the following files:
    - etc/init.d/csr_init
    - /etc/rc2.d/S99csr_init
- To start CSR from the command prompt, specify the database name in command **csr newdbname &**

# Database Functionalities

This section describes the following:

- Database Tables, page 2-4
- PostgreSQL Database Command Summary, page 2-17

## Database Tables

This section describes database tables, constraints placed on each table, and relationships between tables.

> **Note** Advanced knowledge of SQL is required to access the database through SQL commands. Use the psql interface to access an SQL prompt. For more detail on psql interface, see the postgreSQL online manual.

### Configuration Tables

See AppendixB, "Data Schema," for the database script to create database tables, constraints and relationships.

> **Note** The configuration data must contain one record that is created when the user executes the **createcsrdb** command.

***Table 2-1    Configuration Data Table***

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| ActiveDataset | text | NO | '_1' | Valid values:<br>• _1<br>• _2 | There are two datasets in the CSR (active and inactive). There are two of each table for entities that are created (for example, Carriers_1, Carriers_2). The active dataset marks which of the above is active. |
| LoggingLevel | text | NO | 'Error' | Valid values:<br>• error<br>• info<br>• debug | When the application, is active, logs are generated. This marks the level of logging that is being done. During realtime, this runs only at error level to avoid performance deterioration. |
| CSR_ID | varchar(64) | NO | | | ID used to register with the gatekeeper which application is connecting. (Also known as the application ID). |
| CSR_Version | text | | | | Version of CSR. This is entered when the GUI requests it from the CSR. |
| GKTMP_Version | integer | | | | Version of GKTMP. This is entered when the GUI requests it from the CSR. |
| Priority | integer | NO | 1 | >=1 | Priority to register this application to the gatekeeper. |
| ListenPort | integer | | | >0 and < 32767 | TCP port to listen for incoming connection. A gatekeeper or simulator may connect to the CSR, and this is the port to connect to. |

*Table 2-1      Configuration Data Table (continued)*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|------------|---------|-------------|----------|
| ThreadCount | integer | NO | 10 | | Number of threads in the working pool that handle routes in the rules engine. This number must be adjusted for different systems. |
| MaxSizeOfQueue | integer | NO | 5000 | | Maximum size of the queue to handle calls. Typically, the queue is either empty or has an entry. During busy hours, the queue may become larger and cause long turn-around times for the calls to be routed. |
| TestRoutesAccountingEnabled | boolean | NO | FALSE | | When issuing test commands through the GUI, the call units are updated if this is set to True. |
| RejectSrcCarrierInactive | boolean | NO | FALSE | | If this rejection rule applies, the CSR returns a rejection instead of a null body. |
| RejectNullSrcCarrier | boolean | NO | FALSE | | If this rejection rule applies, the CSR returns a rejection instead of a null body. |
| RejectSrcCarrierUnknown | boolean | NO | FALSE | | If this rejection rule applies, the CSR returns a rejection instead of a null body. |
| RejectDestinationUnknown | boolean | NO | FALSE | | If this rejection rule applies, the CSR returns a rejection instead of a null body. |
| RejectSrcPercentage | boolean | NO | FALSE | | If this rejection rule applies, the CSR returns a rejection instead of a null body. |

*Table2-1    Configuration Data Table (continued)*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| RejectSrcUnits | boolean | NO | FALSE | | If this rejection rule applies, the CSR returns a rejection instead of a null body. |
| RejectNoDstCarrier | boolean | NO | FALSE | | If this rejection rule applies, the CSR returns a rejection instead of a null body. |

*Table2-2    Gatekeeper Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| GKKey | integer | NO | | | Primary key. Key of the gatekeeper. |
| GKID | varchar(64) | NO | | | ID of the gatekeeper. Must be unique. |
| GK_IP | inet | NO | | | IP address of the gatekeeper. |
| GK_Port | integer | NO | | >0 and < 32767 | Port that the gatekeeper is listening on. |
| ListenARQ | boolean | NO | FALSE | | Set to TRUE if registering for this type of message. |
| ListenLRQ | boolean | NO | FALSE | | Set to TRUE if registering for this type of message. |
| ListenIRR | boolean | NO | FALSE | | Set to TRUE if registering for this type of message. |
| ListenDRQ | boolean | NO | FALSE | | Set to TRUE if registering for this type of message. |
| ListenRRQ | boolean | NO | FALSE | | Set to TRUE if registering for this type of message. |
| ListenURQ | boolean | NO | FALSE | | Set to TRUE if registering for this type of message. |
| ListenLCF | boolean | NO | FALSE | | Set to TRUE if registering for this type of message. |

*Table2-2    Gatekeeper Table (continued)*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| ListenLRJ | boolean | NO | FALSE | | Set to TRUE if registering for this type of message. |
| ListenRAI | boolean | NO | FALSE | | Set to TRUE if registering for this type of message. |
| ListenBRQ | boolean | NO | FALSE | | Set to TRUE if registering for this type of message. |
| ConnectionState | text | NO | | Valid values:<br>• connect<br>• disconnect | If set to **connect**, the CSR attempts to connect to this gatekeeper. |

*Table2-3    TestRoutes Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| TestKey | integer | NO | | | Primary key. Key of this test route. |
| TestTime | time | | | | Time of the test. |
| TestDate | date | | | | Date of the test. |
| DatasetID | text | | '_1' | Valid values:<br>• _1<br>• _2 | This marks which dataset to issue the test on. |
| ANI | varchar(64) | | | | ANI message. |
| DNIS | varchar(64) | | | | DNIS message. |
| SrcCarrierID | varchar(64) | | | | ID of source carrier. |
| DstCarrierID | varchar(64) | | | | ID of destination carrier. |
| RunMe | Boolean | NO | FALSE | | Set to TRUE when this test is to be run by the CSR. |
| HomeCarrierID | varchar(64) | | | | ID of home carrier. |
| ResultCode | text | | | | Result code for this test. |

*Table2-4    TestOutputs Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|---|---|---|---|---|---|
| TestKey | integer | NO | | | Foreign key. Key of this test (see TestRoutes table). |
| SelectOrder | integer | NO | | | For a given TestKey, this shows the order in which the carriers will be returned. The database does not guarantee the original ordering. |
| DstCarrierID | varchar(64) | NO | | | ID of Destination Carrier. |
| ListKey | integer | NO | | | Specifies key for the contact list associated with the particular carrier. |
| IsSelected | boolean | NO | FALSE | | Specifies whether this is a selected carrier. |

*Table2-5    LogEntries Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|---|---|---|---|---|---|
| TestKey | integer | NO | | | Foreign key. Test key that this log file was generated for. |
| EntryOrder | integer | NO | | | For a given Test key, this field is used to show the order in which the log events were generated. The database does not guarantee the original ordering. |
| LogString | varchar(1024) | NO | | | The log entries associated with the test run. |

⚠
**Caution**    The following table should not be modified or deleted. It consists of error messages that are logged. The CSR application does not work without this table.

*Table2-6    StringResources Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|---|---|---|---|---|---|
| ResourceKey | integer | NO | | | Primary key. Key to the resource. |
| ResourceString | text | NO | | | String (error value). |

*Table 2-7    Filters Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| FilterKey | integer | NO | | | Primary key. Key of the filter. |
| GKID | varchar(64) | NO | | | Foreign key. ID of the gatekeeper associated with this filter. |
| MsgType | text | NO | | Valid values:<br>• ARQ<br>• LRQ<br>• RRQ<br>• URQ<br>• LCF<br>• LRJ<br>• DRQ<br>• BRQ<br>• IRR<br>• RAI | The message type for this filter. |
| FilterType | text | NO | | Valid values:<br>• endpoint<br>• supportedprefix<br>• destinationinformation<br>• remoteextesionaddress'<br>• redirectreason<br>• answercall<br>• notificationonly | Type of filter. |
| Value | text | NO | | | Value of this message. There are certain values for each filter type enforced by the GUI. |

## Dataset Tables

The dataset tables use functionality that is specific to postgreSQL, that is, table inheritance. This allows a table to be created and other tables to be inherited from it. For example, you can create a Carriers Table and have tables Carriers_1 and Carriers_2 inherit from it. This allows two datasets to be created.

*Table2-8    Carriers Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| CarrierKey | integer | NO | | <>0 | Primary key. Key of the carrier. |
| CarrierName | varchar(64) | NO | | | Name of carrier. |
| CarrierType | text | NO | | Valid values:<br>• home<br>• itsp<br>• tdm | Type of carrier. |
| CarrierID | varchar(64) | NO | | | ID of carrier. Must be unique. |
| MaxIngressUnits | integer | | | >=0 | Maximum ingress units. |
| MaxEgressUnits | integer | | | >=0 | Maximum egress units. |
| RuleKey | integer | | | | Foreign key. Rule associated with this specific carrier. |
| ListKey | integer | NO | | | Foreign key. Key of the list. |
| isActive | boolean | NO | TRUE | | Indicates whether carrier is active. |

**Note**    Tables Carriers_1 and Carriers_2 can inherit from the Carriers table.

*Table2-9    MatchPatterns Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| PatternKey | integer | NO | | <> 0 | Primary key. Key of the pattern. |
| Pattern | varchar(64) | NO | | | Unique. Valid values are 0 to 9, a to f, A to F, *, and "." |
| RouteKey | integer | | | | Foreign key. Route that this pattern is associated with. |

> **Note**   Tables MatchPatterns_1 and MatchPatterns_2 can inherit from this table.

*Table2-10   Routes Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| RouteKey | integer | NO | | <> 0 | Primary key. Key of the route. |
| RouteName | text | | | | Name of the route. |
| RuleKey | integer | | | | Foreign key. Rule associated with this route. |

> **Note**   Tables Routes_1 and Routes_2 can inherit from the Routes table.

*Table2-11   Rules Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| RuleKey | integer | NO | | <> 0 | Primary key. Key of the rule. |
| RuleName | text | | | | Optional name to describe this rule. |
| IsDefaultRule | boolean | NO | FALSE | | Indicates if this is the default rule. One default rule must exist. |
| MinQoS | integer | | | >= 1 and <=9 | Minimum quality of service. This is used with RejectInsufficientQoS rule. |
| PreferredCarrierKey | integer | | | | Foreign key. Key to a carrier. Used only when **SelectPreferredCarrier** rule is used. |

*Table2-11  Rules Table (continued)*

| Name | Type | Allow Null | Default | Constraints | Comments |
|---|---|---|---|---|---|
| RejectMaxOriginationPercentage | boolean | NO | FALSE | | Indicates whether to reject this rule. |
| RejectMaxIngressUnits | boolean | NO | FALSE | | Indicates whether to reject this rule. |
| RejectMaxIngressUnitsPerRoute | boolean | NO | FALSE | | Indicates whether to reject this rule. |
| RejectEgressCostMoreThanIngress | boolean | NO | FALSE | | Indicates whether to reject this rule. |
| RejectInsufficientQoS | boolean | NO | FALSE | | Indicates whether to reject this rule. |
| RejectMaxEgressUnits | boolean | NO | FALSE | | Indicates whether to reject this rule. |
| RejectMaxEgressUnitsPerRoute | boolean | NO | FALSE | | Indicates whether to reject this rule. |
| PreventITSPtoITSP | boolean | NO | FALSE | | Indicates whether to reject this rule. |
| SelectMinCostPerEgressRoute | integer | NO | 0 | >= 0 and <= 5 | Specifies the order of selection rule from which the CSR runs. |
| SelectBestQoS | integer | NO | 0 | >= 0 and <= 5 | Qos level. |
| SelectSameCarrier | integer | NO | 0 | >= 0 and <= 5 | Specifies carrier. |

*Table 2-11    Rules Table (continued)*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|------------|---------|-------------|----------|
| SelectPreferredCarrier | integer | NO | 0 | >= 0 and <= 5 | Specifies preferred carrier. |
| SelectPercentageEgress | integer | NO | 0 | >= 0 and <= 5 | Specifies egress percentage. |

**Note**    Tables Rules_1 and Rules_2 can inherit from the Rules table.

*Table 2-12    Contacts Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|------------|---------|-------------|----------|
| ContactKey | integer | NO | | <> 0 | Primary key. Key of the contact. |
| RASAddress | varchar(64) | NO | | | RAS address. |
| AddressType | text | NO | | Valid values:<br>• ipv4<br>• gkid<br>• dnsname | Type of address (IP address, gatekeeper ID, or DNS name). The CSR converts the DNS name to IP address. When you use the DNS name, a DNS server must be available. Otherwise, this value is not used. |
| Cost | integer | | | >= 0 | Specifies desired cost. |
| Priority | integer | | | >= 0 | Specifies desired priority. |

**Note**    Tables Contacts_1 and Contacts_2 can inherit from the Contacts table.

**Lists Table**

*Table2-13   Lists Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| ListKey | integer | NO | | <> 0 | Primary key. Key of the list. |
| ListName | text | | | | Name of the list. |

**Note**    Tables Lists_1 and Lists_2 can inherit from the Lists table.

*Table2-14   ContactLists Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| ListKey | NO | | | | Foreign key. Key of the list. |
| ContactKey | NO | | | | Foreign key. Key of the contact. |

**Note**    Tables ContactLists_1 and ContactLists_2 can inherit from this table.

*Table2-15   RouteAttributes Table*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| AttributeKey | integer | NO | | <>0 | Primary key. Key of the route attribute. |
| AttributeName | text | | | | Name of attribute associated with the route. |
| CarrierKey | integer | NO | | | Foreign key. Key of the carrier associated with this route attribute. |

*Table2-15   RouteAttributes Table (continued)*

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|-----------|---------|-------------|----------|
| DaypartType | text | NO | | Valid values:<br>• absolute<br>• weekly<br>• daily<br>• always | The CSR searches for the correct value in the following order:<br>• absolute = specific date/time range. Start date, end date, start time, and end time must be entered.<br>• weekly=a specific day of the week during a specific time. Start time, end time, and weekday must be entered.<br>• daily=each day for a specific time. Start time and end time must be entered.<br>• always=all the time. |
| WeekDay | integer | | | >= 0 and <= 6. As 0=Sunday, 6=Saturday. | This is for the weekly setting. |
| StartTime | time | | | Valid format: hour(0-23):min (0-60):sec (0-60) | This is for the absolute, weekly, and daily settings. |
| EndTime | time | | | Valid format: hour(0-23):min (0-60):sec (0-60) | This is for absolute, weekly, and daily settings. |
| Cost | integer | | | >= 1 | Cost associated with this attribute. |
| MaxUnits | integer | | | >= 0 | Maximum units. |
| MaxPercentage | integer | | | >= 1 and <=100 | Maximum percentage. |

**Note**    Tables IngressRouteAttributes_1 and IngressRouteAttributes _2 can inherit from the RouteAttributes table.

***Table2-16  EgressRouteAttributes Table***

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|------------|---------|-------------|----------|
| QoS | integer | | | >= 0 and <= 9 | Quality of service. |
| ListKey | integer | | | | Foreign key. Key of list associated with this egress route attribute. |

**Note**    EgressRouteAttributes_1 and EgressRouteAttributes_2 can inherit from the EgressRouteAttributes and RouteAttriubtes tables.

***Table2-17  QoS Table***

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|------------|---------|-------------|----------|
| IngressCarrierKey | integer | NO | | | Foreign key. Key of the ingress carrier. |
| EgressCarrierKey | integer | NO | | | Foreign key. Key of the egress carrier |
| RouteKey | integer | NO | | | Foreign key. Key of the route associated with the QoS. |
| RelativeQoS | integer | NO | | >= 1 and <= 9 | Relative quality of service. |

**Note**    Tables QoS_1 and QoS_2 can inherit from this QoS table.

***Table2-18  QoS Table***

| Name | Type | Allow Null | Default | Constraints | Comments |
|------|------|------------|---------|-------------|----------|
| RouteKey | integer | NO | | | Foreign key. Key of the route associated with this QoS. |
| AttributeKey | integer | NO | | | Foreign key. Key of the attribute associated with this QoS. |

**Note**   Tables IngressRoutes_1, IngressRoutes_2, EgressRoutes_1, and EgressRoutes_2 can inherit from this QoS table.

# PostgreSQL Database Command Summary

Following is a summary of frequently used postgreSQL commands. Refer to the online manual pages and help section for more information.

- **createdb**—Creates a new database in postgreSQL. A database name is required when using this command.

- **createuser**—Creates a postgreSQL user. A user name is required when using this command.

- **dropdb**—Removes a database from postgreSQL. A database name is required when using this command.

- **dropuser**—Removes a postgreSQL user. A user name is required when using this command.

- **initdb**—Creates a new postgreSQL database installation. This command is used only once when the database is installed. It creates the base directory. This command cannot be used again.

- **psql**—SQL interface to the postgreSQL databases. A database name is required when using this command.

- **vacuumdb**—Cleans and analyzes a postgreSQL database.

C H A P T E R **3**

# Cisco Carrier Sensitive Routing Application Operations

This chapter describes the following Cisco Carrier Sensitive Routing application operations:

## Activating CSR

This section provides instructions on running single and multiple instances of the CSR application. The CSR application typically is installed to run at one instance on a dedicated machine, although it can be run at multiple instances on the same machine.

### Single Instance

Before activating the CSR application, the postgreSQL database must be installed and set up for use. A database named "csrdb" must be loaded into the application. You can use either of the following methods to activate the CSR application.

### CSR as a Daemon

The daemon file is located in directory /etc/init.d/csr_init, and it accepts only the **start** parameter.

Step 1    To activate CSR, enter the following command:

```
csr_init start
```

You may be prompted for the postgres (UNIX user) password.

Step 2    To verify that CSR is running, enter the following command:

```
ps -ef command
```

You also can check the syslog files for the first message that the CSR prints to the logs. The message should be a startup message.

## CSR as a User Process

While the CSR is running as a daemon, other instances may run as user processes. If all instances are connected to the same database at startup, startup errors can occur because the following processes are attempted by all instances:

- Listening on the same TCP port for incoming connections
- Attempting to connect to the same gatekeepers as the same application and priority
- Listening to the database for notification events from the GUI such as shutdown, update dataset, update logging level, and update gatekeeper connections

You can create other databases with the command **createcsrdb**; for example, **createcsrdb csrdb2**. You can connect to each of the databases separately by using the GUI and provisioning the database accordingly.

---

**Step 1** To activate CSR, enter the following command:

```
csr csrdb1 &
```

This command activates CSR by connecting it to a different database and typing the process to a console.

**Step 2** Optionally, to run CSR continuously after logout, enter the following command:

```
nohup
```

---

## Multiple Instances

To run multiple CSR instances, all but one must run as a user process. If all instances are connected to the same database at startup, startup errors occur because the following processes are attempted by all instances:

- Listening on the same TCP port for incoming connections
- Attempting to connect to the same gatekeepers as the same application and priority
- Listening to the database for notification events from the GUI such as shutdown, update dataset, update logging level, and update gatekeeper connections

You can create other databases with the command **createcsrdb**; for example, **createcsrdb csrdb1**, **createcsrdb csrdb2**, **createcsrdb csrdb3**. You can connect to each of the databases separately by using the CSR GUI and provisioning the databases accordingly.

---

**Step 1** To start multiple instances of CSR that are connected to different databases such as csrdb1, csrdb2, csrdb3, enter the following commands:

```
csr csrdb1 &
csr csrdb2 &
csr csrdb3 &
```

This method ties the process to a console.

**Step 2** Optionally, to run CSR continuously after logout, enter the following command:

```
nohup
```
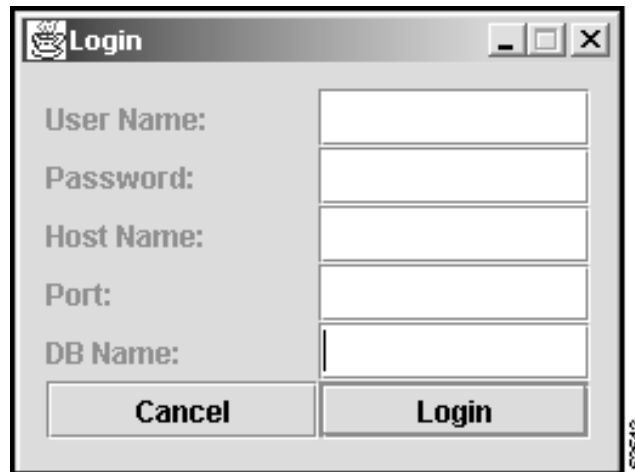
---

# Deactivating CSR

To stop CSR, log in to the CSR GUI by performing the following steps:

**Step 1**    At the system prompt, enter the following command:

`/usr/local/csr/gui/csrgui &`

The login dialog box in Figure 3-1 "Login Dialog Box" appears.

***Figure 3-1    Login Dialog Box***

**Step 2**    Enter the information on the Login screen as follows:

  **a.**   Enter the user name. The range is 1 to 32 characters.

  **b.**   Enter the password. The range is 1 to 32 characters.

  **c.**   Enter the host name that is the location of the database. This can be a remote or local host. The range is 1 to 32 characters.

  **d.**   Enter port number. This is the port on the system for the database connection. The range is 1 to 32767.

  **e.**   Enter DB Name. This is the name of the database to be connected to. The range is 1 to 32 characters.

  **f.**   Click **Login**.

    If login is successful, the CSR window in Figure 3-2 "CSR Window" appears. Otherwise, a login failure message appears and the Login dialog box reappears. The user is limited to three login attempts, then the GUI shuts down. See Chapter 4, "CSR GUI Functionalities".

*Figure 3-2    CSR Window*
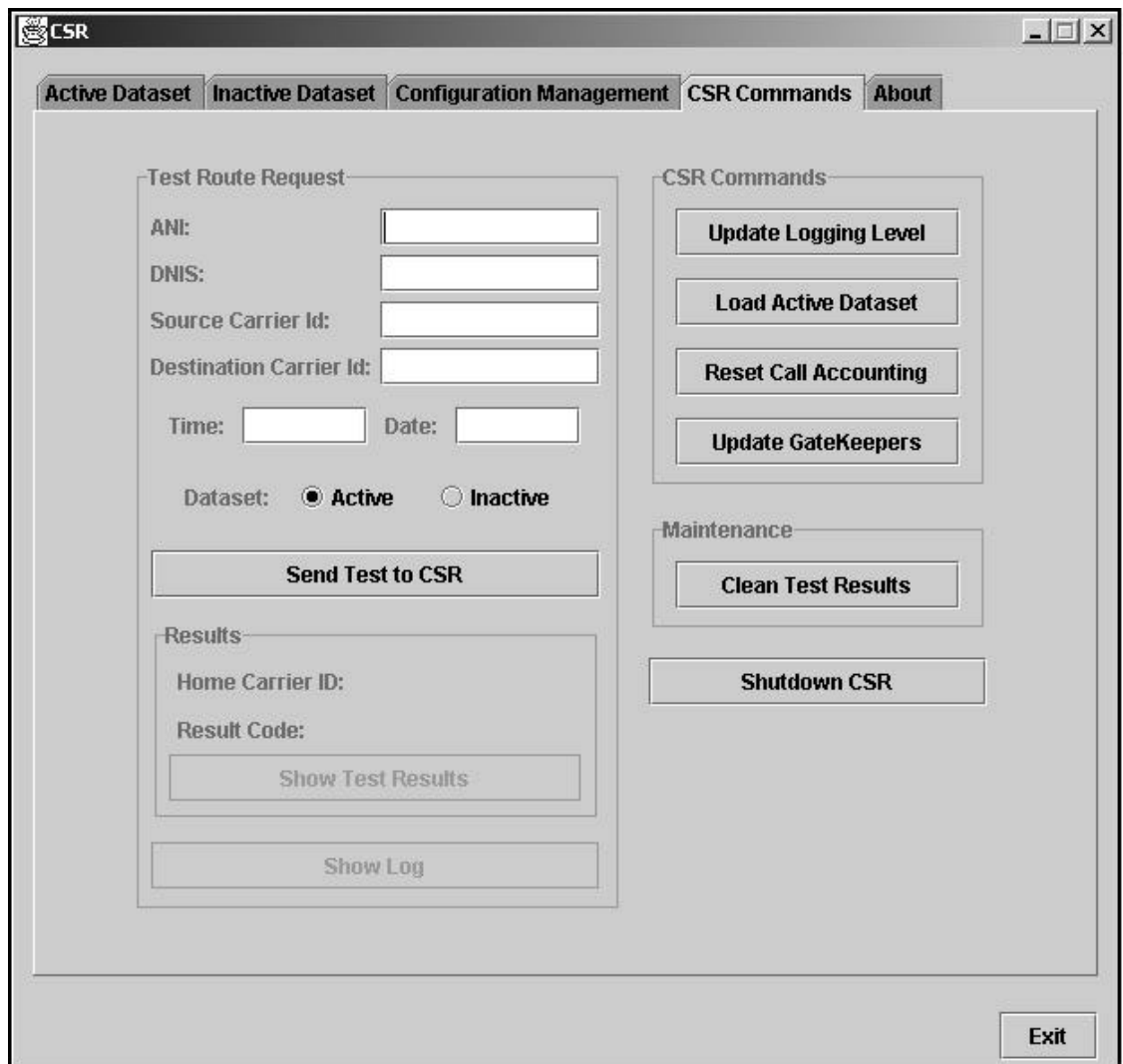


**Step 3**    Click the CSR Commands tab.

**Step 4**    Click **Shutdown CSR**.

This shuts down all CSRs connected to this database. All current calls are dropped, and connections to gatekeepers are terminated.

C H A P T E R  **4**

# CSR GUI Functionalities

This chapter describes CSR functionalities and commands, and contains the following sections:

# Activating the GUI

**Step 1**  To start the GUI, enter the following command at the system prompt:

`/usr/local/csr/gui/csrgui &`

The Login dialog box appears in Figure4-1 "Login Dialog Box", as shown below:

***Figure4-1    Login Dialog Box***
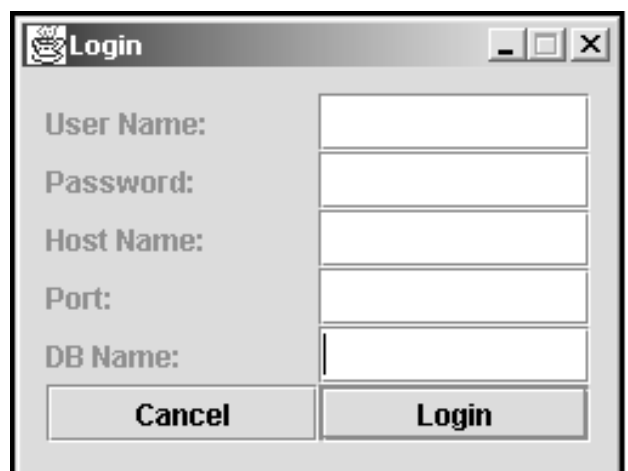
**Step 2**     Enter the information in the Login dialog box as follows:

    **a.**  Enter the user name. The range is 1 to 32 characters.

    **b.**  Enter the password. The range is 1 to 32 characters.

    **c.**  Enter the host name where the database is located. This can be a remote or local machine. The range is 1 to 32 characters.

    **d.**  Enter the port number. This is the port on the system for the database connection. The range is 1 to 32767.

    **e.**  Enter the DB Name. This is the name of the database to be connected to. The range is 1 to 32 characters.

    **f.**  Click **Login**.

If the login is successful, the CSR window appears. Otherwise, a login failure message appears and the Login dialog box reappears. After three unsuccessful login attempts, the GUI shuts down.

Alternatively, click **Cancel** to exit from the GUI.

> **Note**     Information from the last successful login is saved to the CSRLoginProperties file in the current working directory. When the Login dialog box appears next time, the login information, excluding the password, is displayed.

# Configuring CSR with the GUI

This section describes how to use the Configuration Management tab in the CSR window for the following purposes:

- Changing Configuration Settings, page 4-2
- Adding, Modifying, and Removing Gatekeepers, page 4-5
- Swapping Datasets, page 4-12

## Changing Configuration Settings

To change configuration settings, log in to CSR and click the **Configuration Management** tab in the CSR window. When the following screen (Figure 4-2 "Configuration Management Tab in CSR Window") appears, use the Configuration Settings area on the screen to make changes.
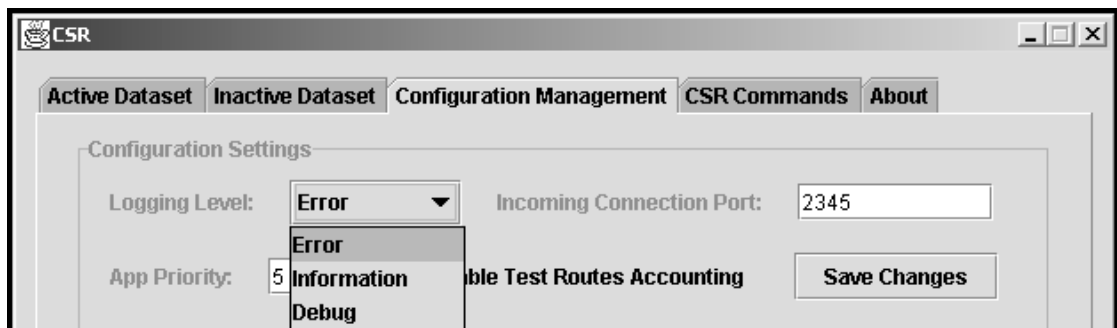
*Figure 4-2    Configuration Management Tab in CSR Window*
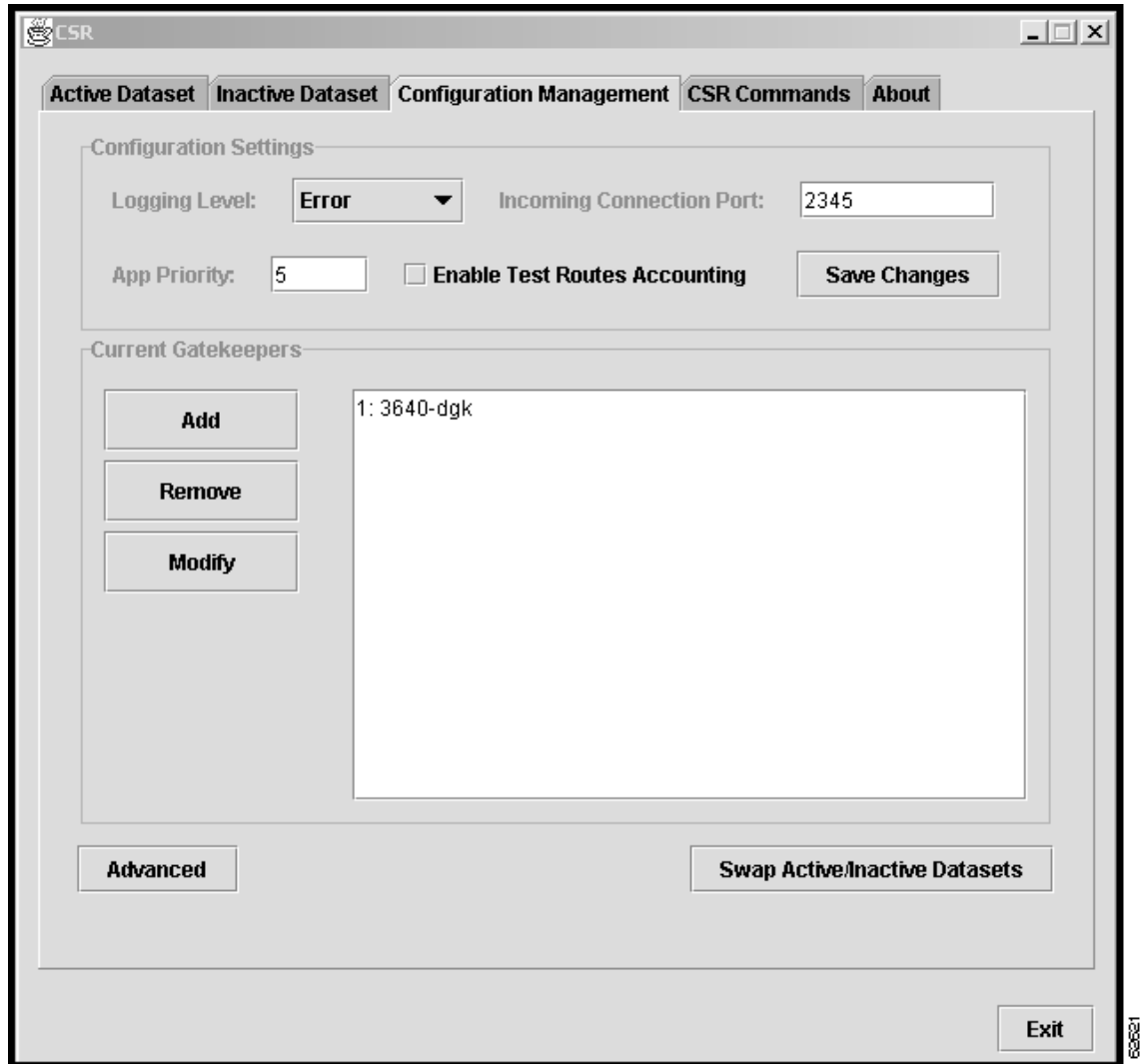


## Changing Log Levels

To change the logging level for the messages to be logged to syslog, select from the Logging Level pull-down menu. The following menu appears (Figure 4-3 "Logging Level Pull-down Menu"):

*Figure 4-3    Logging Level Pull-down Menu*



The available levels are as follows:

- Error—Only messages containing severe errors are logged. Optimally, this level may not generate any error messages.

- Information—Informational messages are generated and sent to syslog. A message at this level is likely to be generated for each route request. This logging level can slow down performance.

- Debug—Numerous messages are logged for each route request. This logging level can severely slow down performance.

See Appendix A, "Logging," for more detail on syslog.

## Changing App Priority

App Priority is the priority of the CSR with respect to the connection to the gatekeeper. If two CSRs are connected to the gatekeeper, the gatekeeper uses this priority number to distinguish which one should be used first.

To change App Priority, enter a number from 1 to 9 in the App Priority box.

## Changing Connection Port

Incoming Connection Port is the port where CSR listens for gatekeeper connection and where the simulator that acts as a client is connected to the CSR. It is normally used for testing because gatekeepers do not act as clients.

To change ports, enter a number from 1 to 32767. Entering zero disables the server connectivity.

## Enabling Test Routes Accounting

This option enables accounting associated with the test routes. When this option is enabled and a test route is issued, the call accounting units are updated. If you are testing new functionalities relating to call counts through the GUI, this must be enabled.

To enable this option, click the checkbox to the left of the option.

## Saving Configuration Settings

The Save Changes option saves the current configuration settings to the system. To save the settings, click **Save Changes**.

# Adding, Modifying, and Removing Gatekeepers

To add, modify, and remove gatekeepers, log in to CSR and select the Configuration Management tab in the CSR window. Use the Current Gatekeepers area on the screen to add, modify, or remove gatekeepers.

## Adding Gatekeepers

This section describes the steps to add a gatekeeper and its associated messages and filters.

**Step 1**    To add a gatekeeper, click **Add** in the Current Gatekeepers area. The following dialog box appears (Figure 4-4 "Registered Messages and Filters Dialog Box for a New Gatekeeper"):

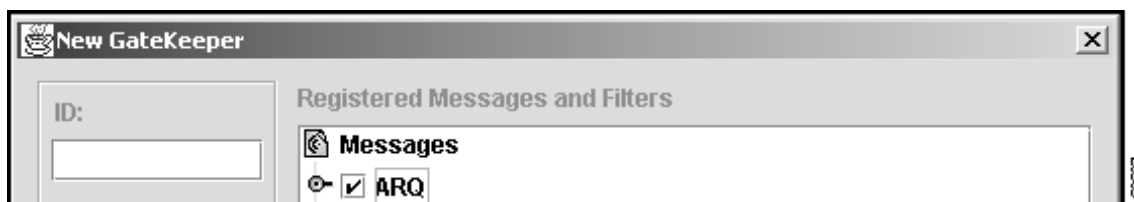*Figure 4-4    Registered Messages and Filters Dialog Box for a New Gatekeeper*

**Step 2**    Enter the ID of the gatekeeper to be added. The range is 1 to 64 characters.

**Step 3**    Enter remote address (IP address) of the gatekeeper to be connected to. The format is *x.x.x.x* where *x* is a number from 0 to 255.

**Step 4**    Enter port on the gatekeeper used for connection. The range is 1 to 32767.

**Step 5**    Click **Connected** or **Disconnected** to connect the gatekeeper to or disconnect the gatekeeper from the CSR. The default is Connected.

**Step 6**    To select a message to be associated with the gatekeeper, check the checkbox next to the message shown, as follows (Figure 4-5 "Registered Messages and Filters Checkbox"):

*Figure 4-5    Registered Messages and Filters Checkbox*



**Step 7**    To display the existing filters for the message, click the round button to the far left of the message. The following screen appears (Figure 4-6 "Filter display in the Registered Messages and Filters Dialog Box"):

*Figure 4-6    Filter display in the Registered Messages and Filters Dialog Box*



**Step 8**    To add a filter to a message, right click the icon to the left of the message. A pop-up menu containing the available filters appears (Figure 4-7 "Filters Pop-up Menu"):

*Figure4-7*    *Filters Pop-up Menu*
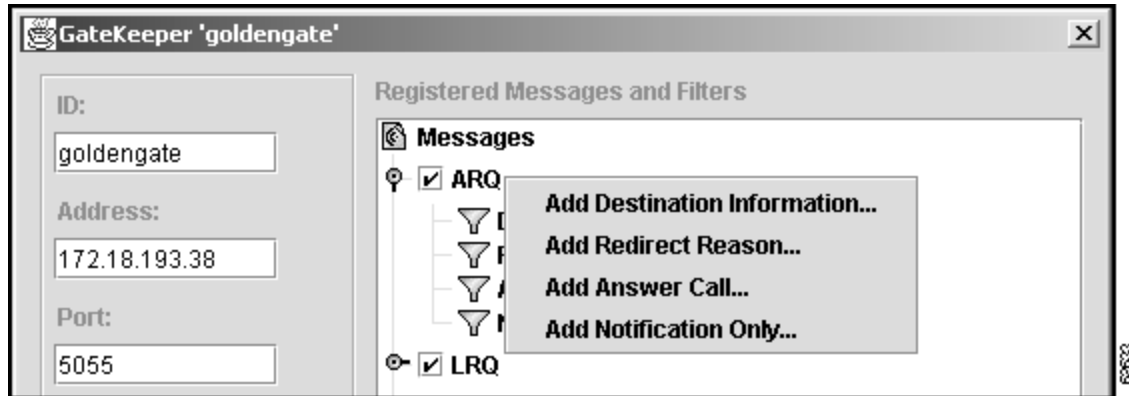


**Step 9**    Click a filter on the pop-up menu.

**Step 10**    Enter or select a value for the filter, as shown in the following sample displays (Figure4-8 "Sample Filter Display", Figure4-9 "Sample Filter Display"):
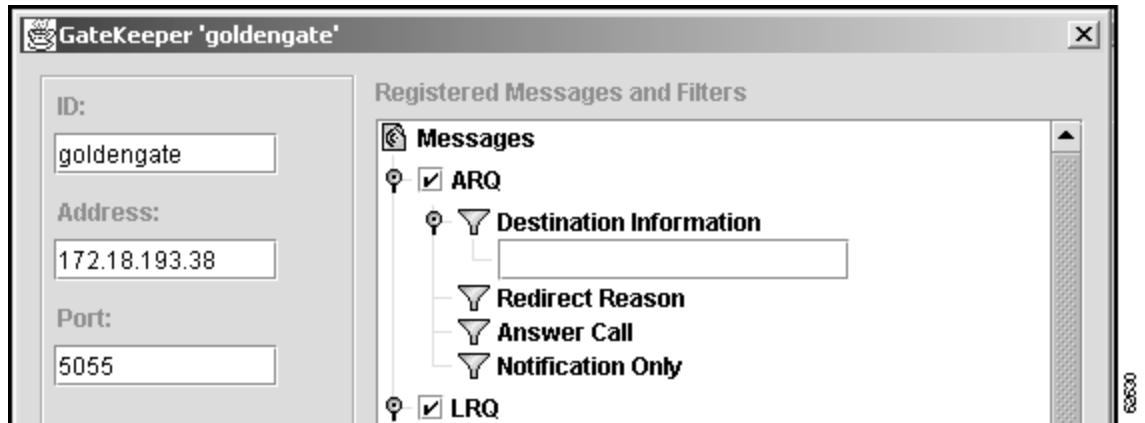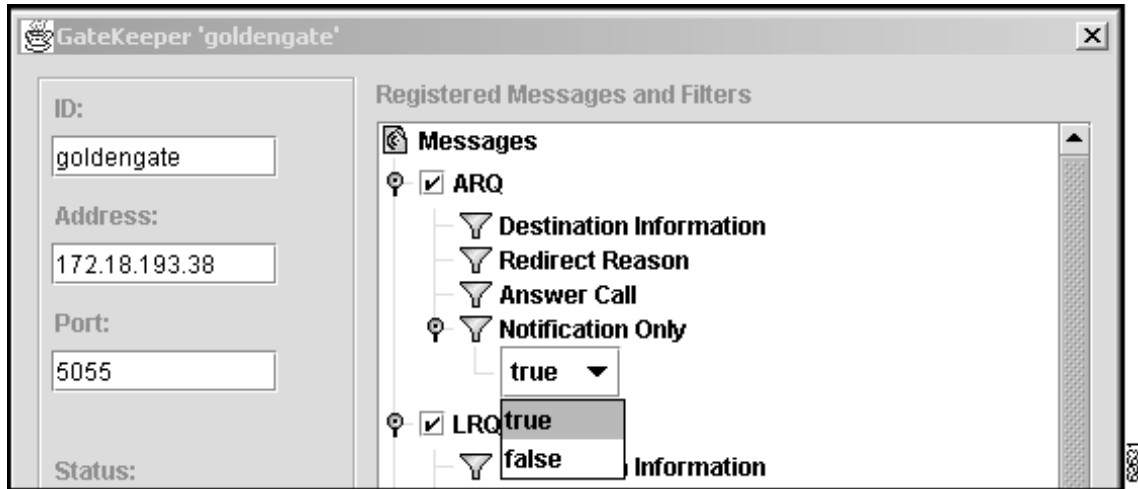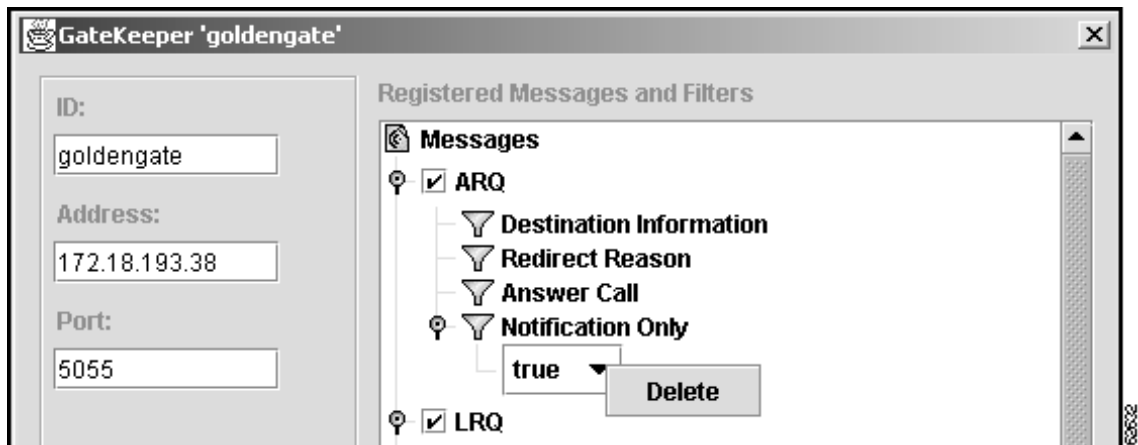
*Figure4-8*    *Sample Filter Display*

*Figure 4-9     Sample Filter Display*



Step 11    To delete a filter to the message, select from the pull-down menu for the filter. The following screen appears (Figure 4-10 "Delete Filter Display"):

*Figure 4-10   Delete Filter Display*



Step 12    Click **Delete**.

Step 13    Click **OK** to save the changes, or click **Cancel** to cancel the changes.

> **Note**    If the maximum number of filters of a particular type has already been added to a message, an error message appears.

# Modifying Gatekeepers

This section describes how to modify an existing gatekeeper.
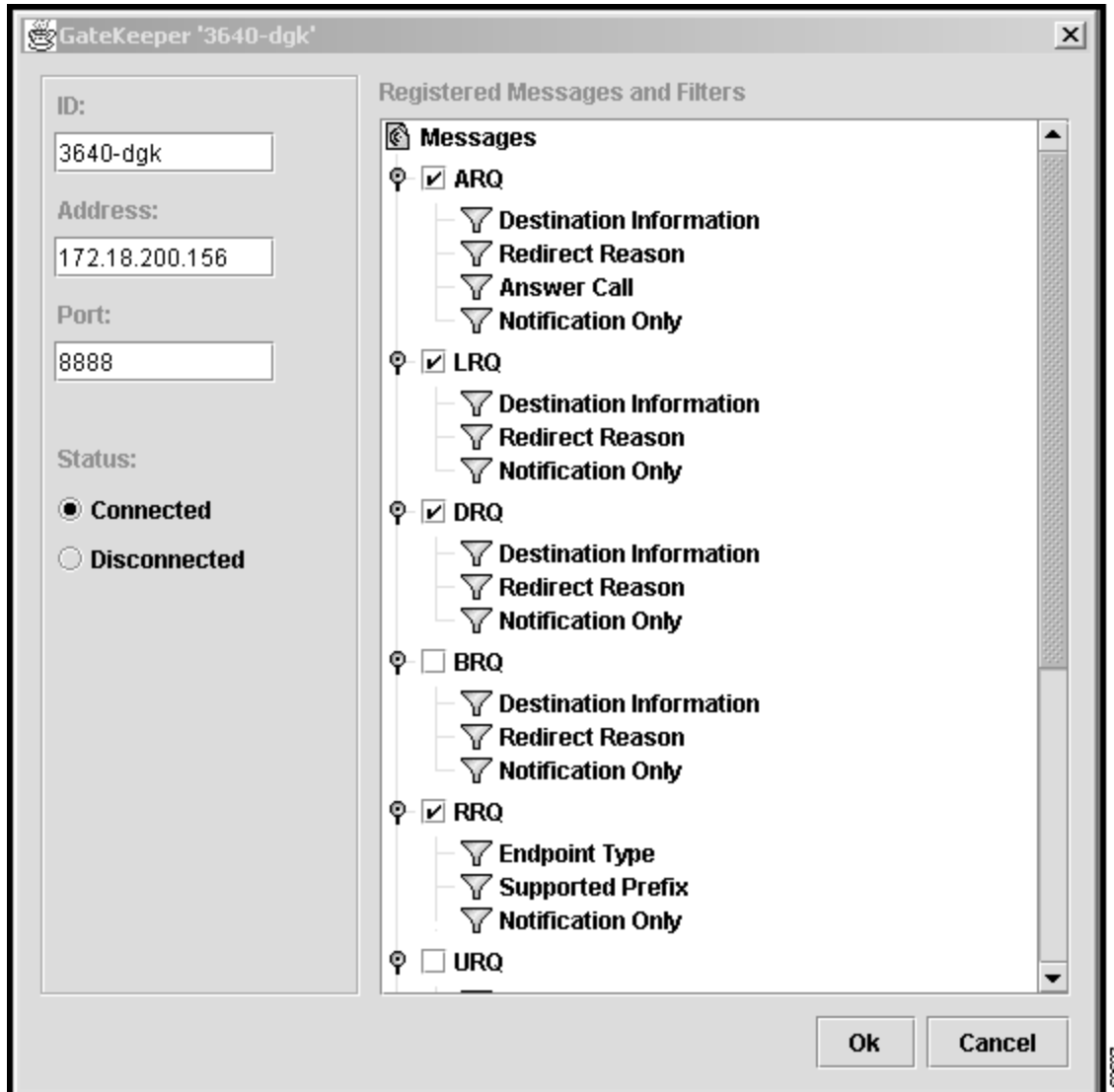
Step 1    Select a gatekeeper from the gatekeeper list in the Current Gatekeepers area, as follows (Figure 4-11 "Current Gatekeepers Area in CSR Window").

*Figure 4-11   Current Gatekeepers Area in CSR Window*
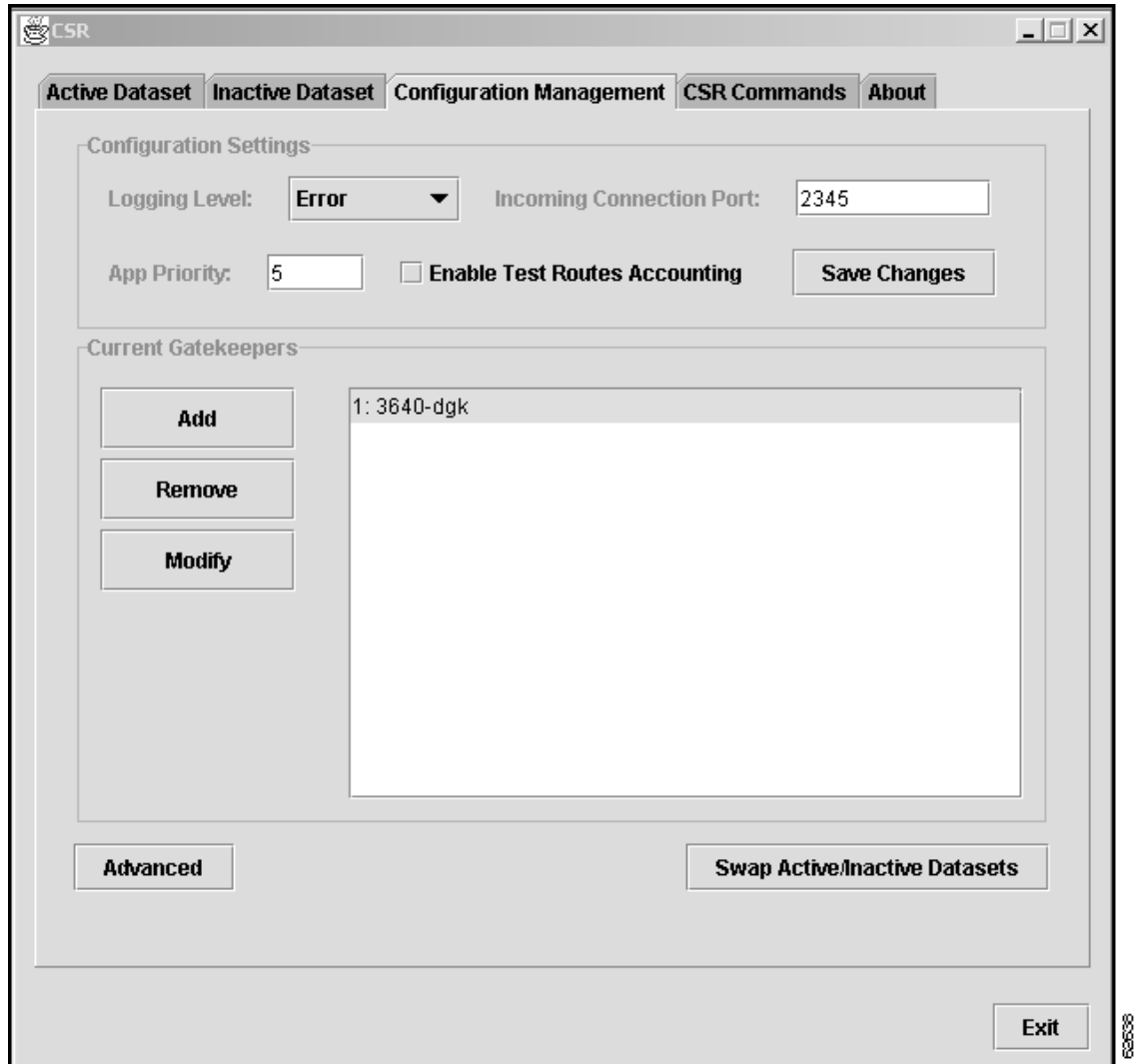


Step 2    Click **Modify.** The following dialog box appears (Figure 4-12 "Registered Messages and Filters Dialog Box for an Existing Gatekeeper").

*Figure 4-12   Registered Messages and Filters Dialog Box for an Existing Gatekeeper*



**Step 3**    To modify a message and its filters associated with the gatekeeper, click the round button to the left of the message.

**Step 4**    Use the steps in adding messages and filters in the "Adding Gatekeepers" section on page 5

**Step 5**    Click **OK** to save the changes, or click **Cancel** to cancel the changes.

## Removing Gatekeepers

This section describes how to remove an existing gatekeeper.

**Step 1**    To remove an existing gatekeeper, select a gatekeeper from the gatekeeper list in the Current Gatekeepers area.

**Step 2**    Click **Remove**. The following message appears (Figure4-13 "Delete Gatekeeper Message").

*Figure4-13    Delete Gatekeeper Message*
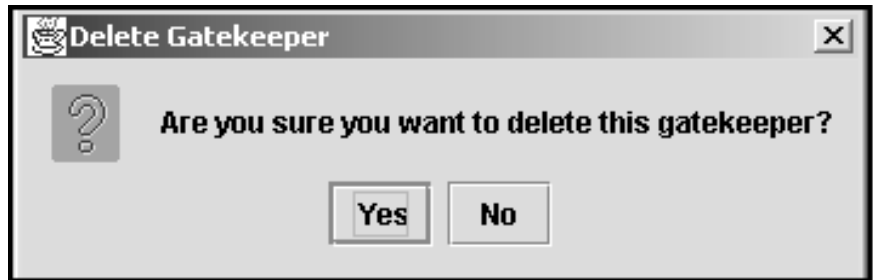


**Step 3**    Click **Yes** to remove the gatekeeper, or click **No** to retain the gatekeeper.

## Setting Threads for Processing Calls and Call Queue Size

This section describes how to set the number of working threads for processing calls in the queue and the maximum number of calls allowed to be queued.

**Step 1**    In the Configuration Management screen in the CSR window, click **Advanced**. The Advanced Configuration dialog box appears (Figure4-14 "Advanced Configuration Dialog Box"):

*Figure4-14    Advanced Configuration Dialog Box*
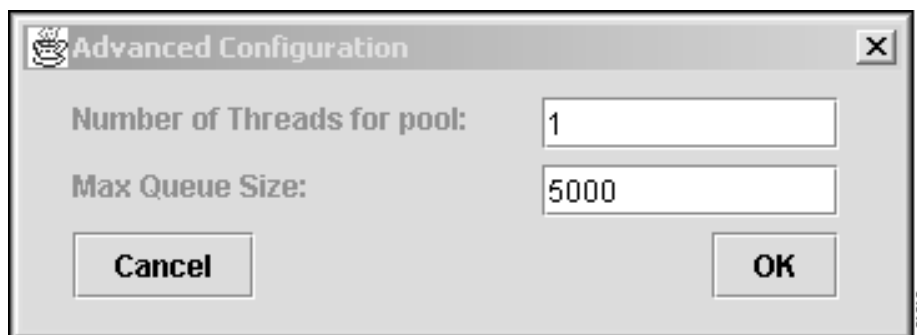


**Step 2**    To set the number of working threads for processing calls in the queue, enter the number of threads for pool.

This number should equal to the number of processors of the host machine, plus one. It can be modified to enhance the performance of the CSR in different environments.

**Step 3**  To enter the maximum number of calls allowed to be queued, enter a number for "Maximum Queue Size."

The default is 5000. If this number is too low, the gatekeeper connection waits for the CSR to catch up. If this number is too high, the turnaround time per message may be too long during busy hours and the gatekeeper routes the call on its own.

**Step 4**  Click **OK** to save the changes, or click **Cancel** to cancel the changes.

## Swapping Datasets

This section describes how to swap an active and inactive dataset.

**Step 1**  In the Configuration Management screen in the CSR window, click **Swap Active/Inactive Datasets**. The following message appears (Figure4-15 "Swap Dataset Message"):

*Figure4-15   Swap Dataset Message*
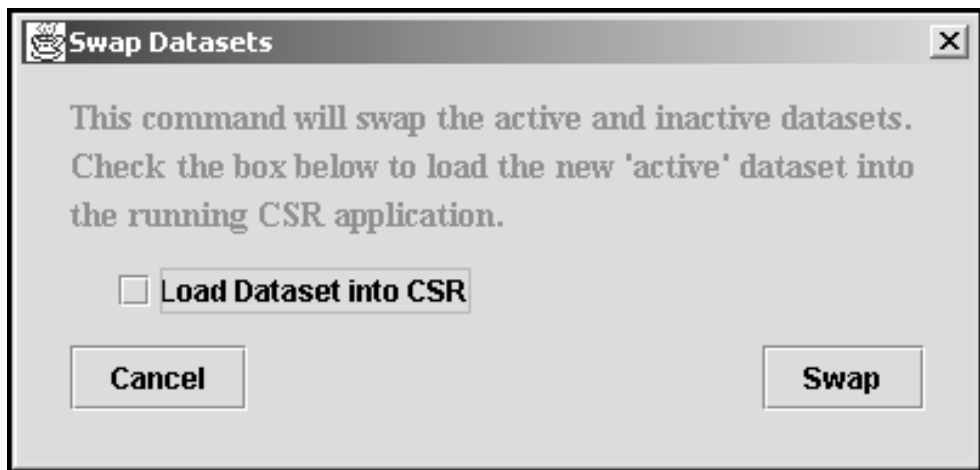


**Step 2**  Check the checkbox to the left of "Load Dataset into CSR.".

**Step 3**  Click **Swap** to swap the datasets, or click **Cancel** to cancel swapping.

# Updating Active Datasets

This section describes how to update active datasets by using the Active Dataset tab in the CSR window to create, modify, and delete the following items.

- Carriers
- Contacts
- Contact lists

- Egress and ingress route attributes

- Match patterns

- Routes

- Rules

After you log in to CSR and select the Active Dataset tab in the CSR window, the following screen appears (Figure 4-16 "Active Dataset Tab in CSR Window"):

*Figure 4-16    Active Dataset Tab in CSR Window*
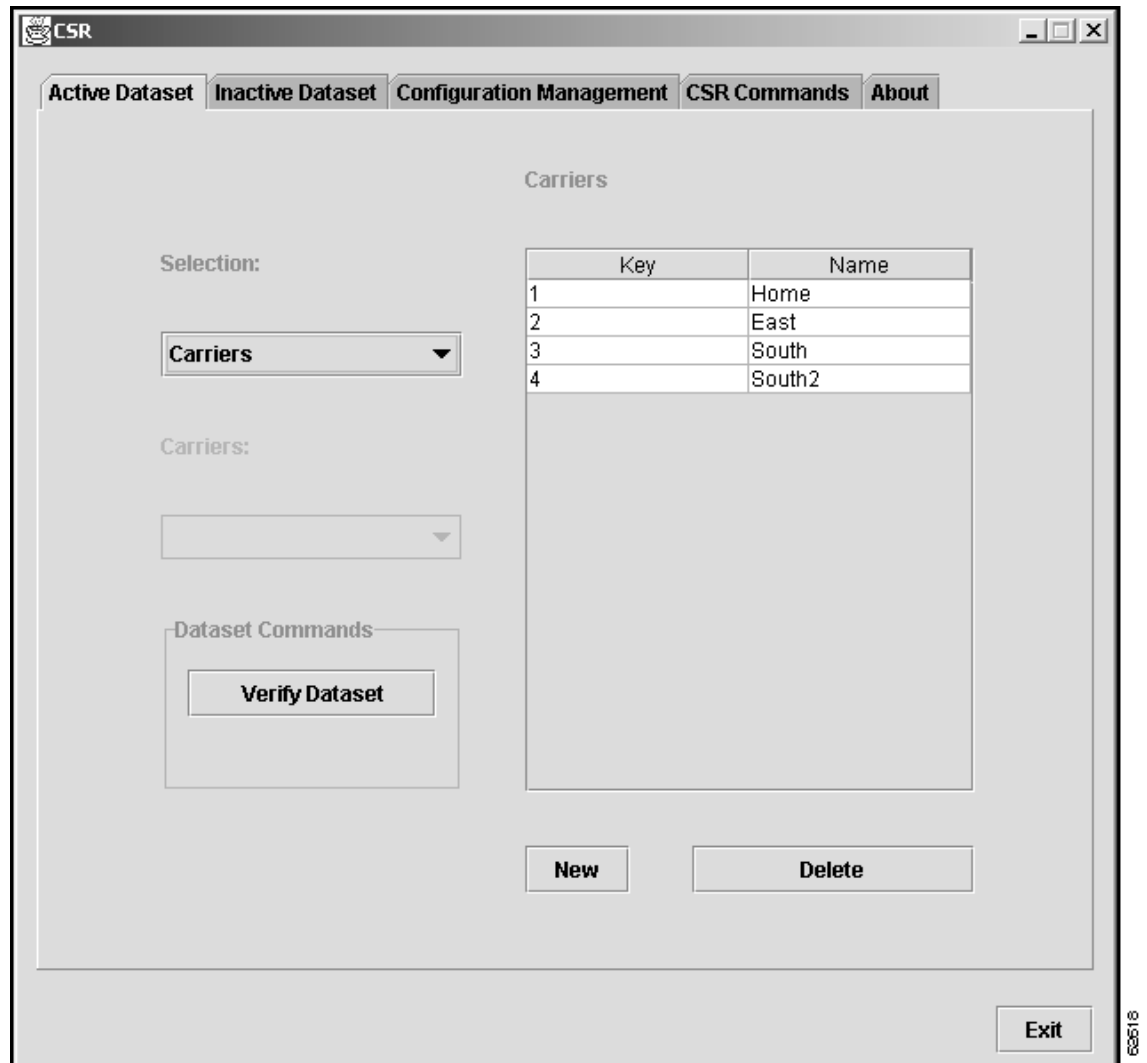


The following sections describe functionalities and operations of the Active Dataset tab.

# Creating, Modifying, and Deleting Carriers

This section describes the steps to create, modify, and delete carriers.

## Creating Carriers

Note    At least one contact list must be created before a carrier can be created. There must be a single carrier of "Home" type created for the provisioned data. See the "Deleting Contact Lists" section on page 4-21.

Step 1    In the Active Dataset screen in the CSR window, if **Carriers** is not selected in the Selection box, click the **down arrow** in that box. A pull-down menu appears (Figure 4-17 "Carrier Selection Box in Active Dataset"):

*Figure 4-17    Carrier Selection Box in Active Dataset*
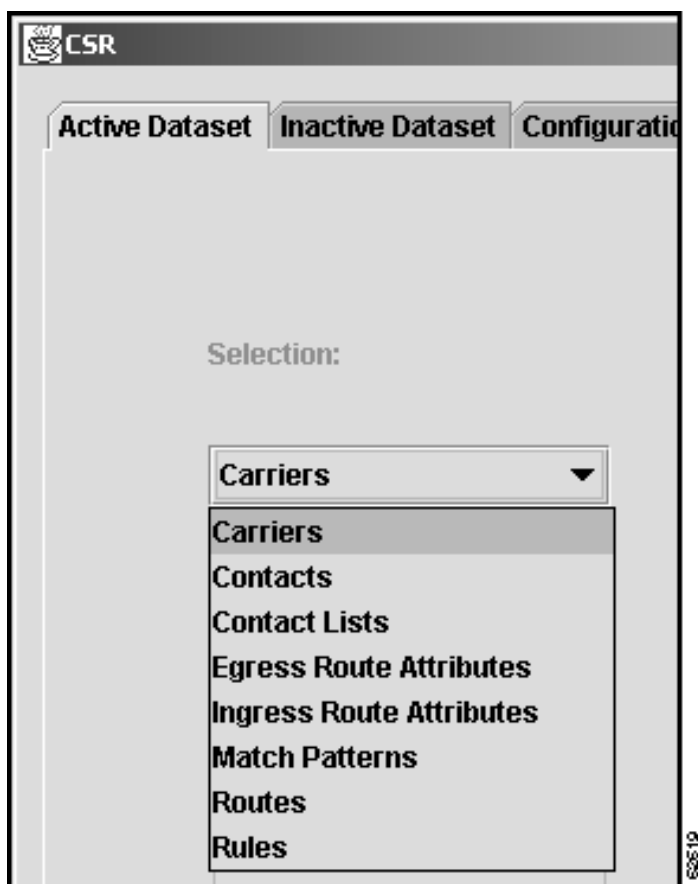


Step 2    Select **Carriers** from the pull-down menu, then click **New**. The New Carrier dialog box appears (Figure 4-18 "Dialog Box for New Carrier in Active Dataset"):

*Figure 4-18    Dialog Box for New Carrier in Active Dataset*



The carrier key is the primary key of this carrier in the Carriers table. When a new carrier is being created, "%%%%" is displayed.

Step 3    In the Carrier ID box, enter the ID of the carrier. The range is 1 to 64 characters.

Step 4    In the Carrier name box, enter name of the carrier. The range is 1 to 64 characters.

Step 5    In the Carrier Type box, click the **down arrow**, then select the type of carrier from the pull-down menu.

Step 6    In the Max Ingress Units box, enter the maximum number of ingress units this carrier may originate across all routes serviced by this carrier. The range is 0 to 2147483647. This is optional.

Step 7    In the Rule box, click the **down arrow**, then select the rule this carrier applies to from the pull-down menu.

Step 8    In the Max Egress Units box, enter the maximum number of egress units that this carrier may terminate across all routes serviced by this carrier. The range is 0 to 2147483647. This is optional.

Step 9    In the Contact List box, click the **down arrow**, then select the contact list that this carrier applies to from the pull-down menu.

Step 10    To make this carrier active, click the checkbox to the left of "Active."

Step 11    Click **Add** to add this carrier to the provisioned data, or click **Cancel** to exit.

## Modifying Carriers

Step 1    In the Active Dataset tab in the CSR window, if "Carrier" is not selected in the Selection box, click the **down arrow** in that box, then select **Carrier** from the pull-down menu.

Step 2    In the Carriers area, select a carrier by double-clicking that carrier on the list. Carrier information similar to the following appears (Figure 4-19 "Dialog Box for an Existing Carrier in Active Dataset"):

*Figure4-19   Dialog Box for an Existing Carrier in Active Dataset*



**Step 3**    Change the information accordingly.

**Step 4**    Click **Modify** to save the changes, or click **Cancel** to cancel the changes.

## Deleting Carriers

**Step 1**    In the Active Dataset tab in the CSR window, if **Carriers** is not selected in the Selection box, click the **down arrow** in that box, then select **Carrier** from the pull-down menu.

**Step 2**    In the Carriers area, click a carrier. You can select more than one carrier by holding down the Ctrl key while clicking each carrier.

**Step 3**    Click **Delete**. The following message appears (Figure4-20 "Delete Carrier Message in Active Dataset"):

*Figure4-20   Delete Carrier Message in Active Dataset*
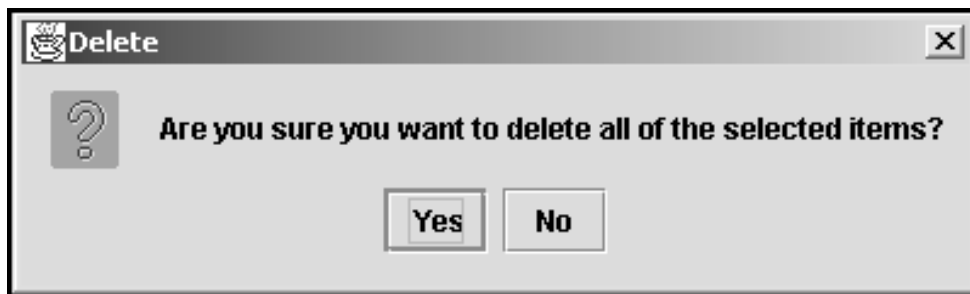


**Step 4**    Click **Yes** to delete the carrier, or click **No** to stop deletion.

> **Note**   Alternatively, you can delete a carrier by clicking **Delete** when you are in the dialog box for modifying the carrier.

# Creating, Modifying, and Deleting Contacts

This section describes the steps to create, modify, and delete contacts.

## Creating Contacts

**Step 1**   In the Active Dataset tab screen in the CSR window, click the **down arrow** in the Selection box, then select **Contacts** from the pull-down menu.

**Step 2**   Click **New**. The following dialog box appears (Figure 4-21 "Dialog Box for New Contact in Active Dataset"):

*Figure 4-21    Dialog Box for New Contact in Active Dataset*



The contact key is the primary key of this contact in the Contacts table. When a new contact is being created, "%%%%" is displayed.

**Step 3**   In the Address Type box, click the **down arrow**, then select an address type from the pull-down menu.

**Step 4**   In the RAS Address box, enter address of the contact (IP Address or Gatekeeper ID).

Step 5    In the Cost box, enter cost associated with this contact. The entry must be greater than 0. This is optional.

Step 6    In the Priority box, enter priority of this contact. The entry must be greater than 0. This is optional.

Step 7    Click **New** to add this contact, or click **Cancel** to exit.

## Modifying Contacts

Step 1    In the Active Dataset tab screen in the CSR window, click the **down arrow** in the Selection box, then select **Contacts** from the pull-down menu.

Step 2    In the Contacts area, select a contact by double-clicking on that contact on the list. Contact information similar to the following appears (Figure4-22 "Dialog Box for an Existing Contact in Active Dataset"):

*Figure4-22   Dialog Box for an Existing Contact in Active Dataset*



Step 3    Change the information accordingly.

Step 4    Click **Modify** to save the changes, or click **Change** to cancel the changes.

## Deleting Contacts

Step 1    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Contacts** from the pull-down menu.

**Step 2**   In the Contacts area, click a contact. You can select more than one contact by holding down the Ctrl key while clicking each contact.

**Step 3**   Click **Delete**. A Delete message appears:

**Step 4**   Click **Yes** to delete the contact, or click **No** to stop the deletion.

Note   Alternatively, you can delete a contact by clicking **Delete** when you are in the screen for modifying the contact.

# Creating, Modifying, and Deleting Contact Lists

This section describes the steps to create, modify, and delete Contacts Lists.

## Creating Contact Lists

**Step 1**   In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Contact Lists** from the pull-down menu.

**Step 2**   Click **New.** The following dialog box appears (Figure4-23 "Dialog Box for New Contact List in Active Dataset"):

*Figure4-23   Dialog Box for New Contact List in Active Dataset*

The key shown on the screen is the primary key of this contact list in the Contact List table. When a new contact list is being created, "%%%%" is displayed.

**Step 3**    In the Contact List Name box, enter the name of the contact list. The range is 1 to 64 characters.

**Step 4**    In the Available Contacts box, you can select a contact and click **Add** to add that contact to the Associated Contacts box.

**Step 5**    In the Associated Contacts box, you can select a contact and click **Remove** to remove that contact from the box and return it to the Available Contacts box.

**Step 6**    Click **Add** to add this contact list, or click **Cancel** to exit.

## Modifying Contact Lists

**Step 1**    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Contact Lists** from the pull-down menu.

**Step 2**    In the Contact Lists area, select a contact list by double-clicking on that contact list. Contact list information similar to the following appears (Figure 4-24 "Dialog Box for an Existing Contact List in Active Dataset"):

*Figure 4-24   Dialog Box for an Existing Contact List in Active Dataset*
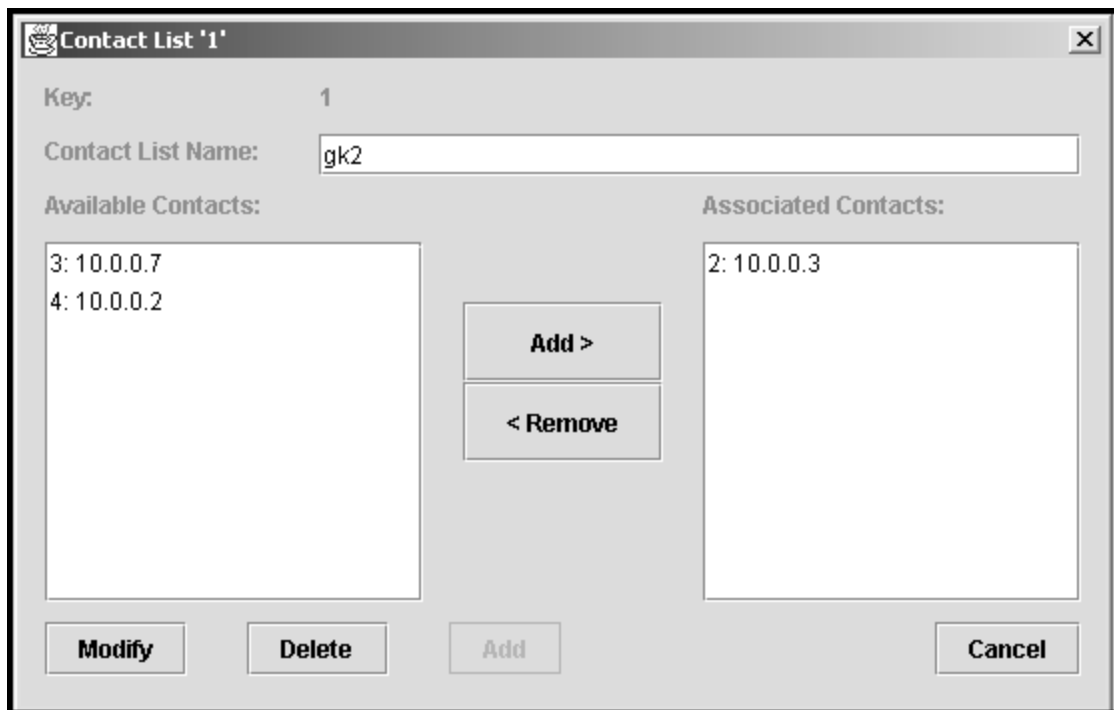


**Step 3**    Change the information accordingly.

Step 4    Click **Modify** to save the changes, or click **Cancel** to cancel the changes.

## Deleting Contact Lists

Step 1    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Contact Lists** from the pull-down menu.

Step 2    In the Contact Lists area, click a contact list. You can select more than one list by holding down the Ctrl key while clicking each contact list.

Step 3    Click **Delete.** A Delete message appears.

Step 4    Click **Yes** to delete the contact list, or click **No** to stop the deletion.

> **Note**    Alternatively, you can delete a contact list by clicking **Delete** when you are in the dialog box for modifying the list.

# Creating, Modifying, and Deleting Egress Route Attributes

This section describes the steps to create, modify, and delete Egress Route Attributes.

## Creating Egress Route Attributes

Step 1    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Egress Route Attributes** from the pull-down menu.

Step 2    Click **New**. The following screen appears (Figure 4-25 "Dialog Box for New Egress Route Attribute in Active Dataset"):

*Figure4-25   Dialog Box for New Egress Route Attribute in Active Dataset*



The key shown on the screen is the primary key of this egress route attribute in the Egress Route Attributes table. When a new egress route attribute is being created, "%%%%" is displayed.

**Step 3**    In the Name box, enter the name of this egress route attribute. The range is 1 to 64 characters. This entry is optional.

**Step 4**    In the Cost box, enter the cost associated with this egress route attribute. The entry must be greater than 0. This entry is optional.

**Step 5**    In the Max % box, enter the maximum percentage of calls that the associated carrier may terminate on a route associated with this route attribute. The range is 0 to 100. This entry is optional.

**Step 6**    In the Max Units box, enter the maximum number of egress units this carrier may terminate on a route associated with this route attribute. This entry is optional.

**Step 7**    In the Carrier box, click the **down arrow**, then select a carrier from the pull-down menu. This is the carrier this egress route applies to.

**Step 8**    In the Qos box, enter the quality of service associated with this route attribute. The range is 1 to 9. This entry is optional.

**Step 9**    In the Contact List box, click the **down arrow**, then select a contact list from the pull-down menu. This is the contact list this egress route applies to.

Step 10    In the Date/Time area, click **Daily**, **Weekly**, **Absolute** or **Constant**. The egress route attribute is being applied according to these time segments.

    **a.**    If you select Daily, enter Start Time and End Time. Use the format shown in the example at the bottom of the Date/Time area:

```
hour(0-23):min (0-60):sec (0-60)
```

    **b.**    If you select Weekly, select day of the week from the pull-down menu and enter the Start Time and the End Time.

    **c.**    If you select Absolute, enter the Start Date, Start Time, End Date, and End Time.

    The Start Date is the date when the egress route attribute becomes active. The End Date is the date when it becomes inactive. Use the formats shown in the example at the bottom of the Date/Time area:

```
4 digit year-2 digit month-2 digit day
```
for date, as in 2002-12-03
```
hour(0-23):min (0-60):sec (0-60)
```
for time, as in 12:59:10

    **d.**    If you select Constant, the egress route attribute is applied at all times of all days.

> **Note**    If two RouteAttributes exist in the same timeframe with the same date and time constraint, the RouteAttribute is "nondeterminstic." That is, the order of selection among time intervals of the same type that overlap cannot be determined.

Step 11    In the Available Routes box, click **Add** or **Remove** to add or remove a route from the Associated Routes box respectively.

Step 12    Click **Add** to add this egress route attribute, or click **Cancel** to exit.

## Modifying Egress Route Attributes

Step 1    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Egress Route Attributes** from the pull-down menu.

Step 2    Click the **down arrow** in the Carriers box, then select the appropriate carrier from the pull-down menu. A list of Egress Route Attribute names of the selected carrier appears (Figure 4-26 "List of Egress Route Attribute Names in Active Dataset"):

*Figure4-26   List of Egress Route Attribute Names in Active Dataset*
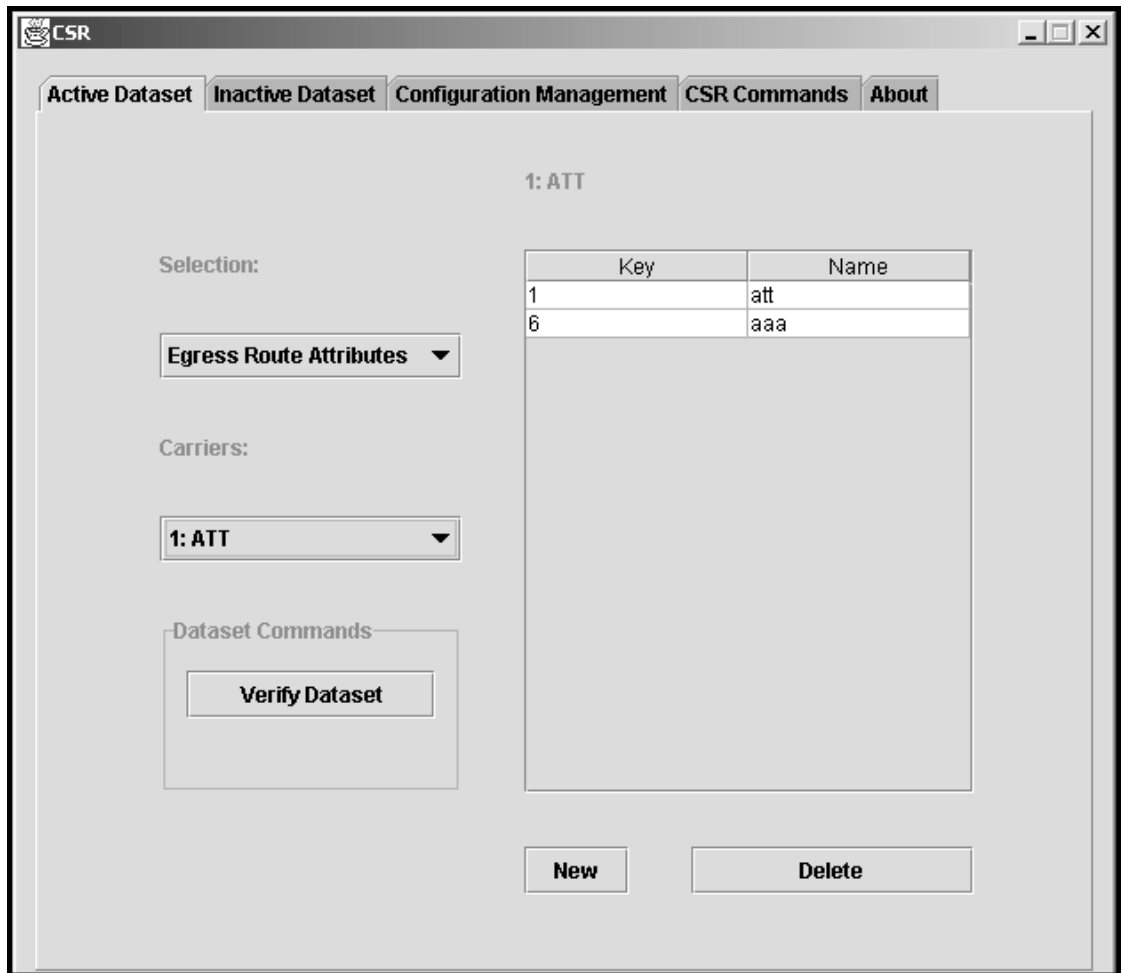
Step 3     From the list, select an Egress Route Attribute by double-clicking on that attribute. Egress Route Attributes information similar to the following appears (Figure 4-27 "Dialog Box for an Existing Egress Route Attribute in Active Dataset"):

*Figure 4-27    Dialog Box for an Existing Egress Route Attribute in Active Dataset*



Step 4     Change the information accordingly.

Step 5     Click **Modify** to save the changes, or click **Cancel** to cancel the changes.

## Deleting Egress Route Attributes

Step 1     In the Active Dataset tab screen in the CSR window, click the **down arrow** in the Selection box, then select **Egress Route Attributes** from the pull-down menu.

Step 2     Click the **down arrow** in the Carriers box, then select the appropriate carrier from the pull-down menu.

Step 3     In the Egress Route Attribute list, select an attribute. You can select more than one attribute by holding down the Ctrl key while clicking each attribute.

**Step 4**    Click **Delete**.

A Delete message appears:

**Step 5**    Click **Yes** to delete the attribute, or click **No** to stop the deletion.

> **Note**    Alternatively, you can delete an attribute by clicking **Delete** when you are in the dialog box for modifying the attribute.

.

# Creating, Modifying, and Deleting Ingress Route Attributes

This section describes the steps to create, modify, and delete Ingress Route Attributes.

## Creating Ingress Route Attributes

**Step 1**    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Ingress Route Attributes** from the pull-down menu.

**Step 2**    Click **New**. The New Ingress Route Attribute dialog box appears (Figure4-28 "Dialog Box for New Ingress Route Attribute in Active Dataset"):

*Figure4-28   Dialog Box for New Ingress Route Attribute in Active Dataset*



The key shown on the screen is the primary key of this Ingress Route Attribute in the Ingress Route Attributes table. When a new ingress route attribute is being created, "%%%%" is displayed.

**Step 3**   In the Name box, enter the name of this ingress route attribute. The range is 1 to 64 characters. This entry is optional.

**Step 4**   In the Cost box, enter the cost associated with this ingress route attribute. The entry must be greater than 0. This entry is optional.

**Step 5**   In the Max % box, enter the maximum percentage of calls that the associated carrier may originate on a route associated with this route attribute. The range is 0 to 100. This entry is optional.

**Step 6**   In the Max Units box, enter the maximum number of ingress units that this carrier may originate on a route associated with this route attribute. This entry is optional.

**Step 7**   In the Carrier box, click the **down arrow**, then select a carrier from the carrier pull-down menu. This is the carrier that this ingress route applies to.

**Step 8**   In the Date/Time area, select **Daily**, **Weekly**, **Absolute**, or **Constant**.

The Ingress Route Attribute is being applied according to these time segments:

  **a.**  If you select Daily, enter Start Time and End Time.

Use the format shown in the example at the bottom of the Date/Time area:

```
hour(0-23):min (0-60):sec (0-60)
```

**b.** If you select Weekly, select day of the week from the pull-down menu and enter the Start Time and the End Time.

**c.** If you select **Absolute**, enter the Start Date, Start Time, End Date, and End Time.

The Start Date is the date when the ingress route attribute becomes active. The End Date is the date when it becomes inactive.

Use the formats shown in the example at the bottom of the Date/Time area:

`4 digit year-2 digit month-2 digit day` for date, as in 2002-12-03
`hour(0-23):min (0-60):sec (0-60)` for time, as in 12:59:10

**d.** If you select Constant, this indicates that the ingress route attribute applies at all times of all days.

> **Note** If two RouteAttributes exist in the same time frame with the same date and time constraint, the RouteAttribute is "nondeterminstic." That is, the order of selection among time intervals of the same type that overlap cannot be determined.

**Step 9** In the Available Routes box, click **Add** or **Remove** to add or remove a route from the Associated Routes box respectively.

**Step 10** Click **Add** to add this ingress route attribute, or click **Cancel** to exit.

## Modifying Ingress Route Attributes

**Step 1** In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Ingress Route Attributes** from the pull-down menu.

**Step 2** Click the down arrow in the Carriers box, then select the appropriate carrier from the pull-down menu. A list of Ingress Route attribute names of the selected carrier appears.

**Step 3** In the list, select an ingress route attribute by double-clicking that attribute. Ingress Route Attributes information similar to the following appears (Figure 4-29 "Dialog Box for an Existing Ingress Route Attribute in Active Dataset"):

*Figure 4-29    Dialog Box for an Existing Ingress Route Attribute in Active Dataset*



**Step 4**    Change the information accordingly.

**Step 5**    Click **Modify** to save the changes, or click **Cancel** to leave the changes.

## Deleting Ingress Route Attributes

**Step 1**    In the Active Dataset tab in the CSR window, click the down arrow in the Selection box, then select **Ingress Route Attributes** from the pull-down menu.

**Step 2**    Click the **down arrow** in the Carriers box, then select the appropriate carrier from the pull-down menu.

**Step 3**    In the Ingress Route Attribute list, select an attribute. You can select more than one attribute by holding down the Ctrl key while doing the mouse-click.

**Step 4**    Click **Delete**.

A Delete message appears.

**Step 5**    Click **Yes** to delete the attribute, or click **No** to stop the deletion.

> **Note**  Alternatively, you can delete an attribute by clicking **Delete** when you are in the dialog box for modifying the attribute.

# Creating, Modifying, and Deleting Match Patterns

This section describes the steps to create, modify, and delete Match Patterns.

## Creating Match Patterns

**Step 1**  In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Match Patterns** from the pull-down menu.

**Step 2**  Click **New.** The following New Pattern dialog box appears (Figure 4-30 "Dialog Box for New Pattern in Active Dataset"):

*Figure 4-30   Dialog Box for New Pattern in Active Dataset*



The Pattern Key on this screen is the primary key of this pattern in the Match Patterns Table. When a new pattern is being created, "%%%%" is displayed.

**Step 3**  In the Pattern box, enter the match pattern that a DNIS is to mapped to.

This entry can include 0-9, A-F, *, and period (.).

**Step 4**  In the Route box, enter the route that applies to this pattern.

**Step 5**  Click **Add** to add this pattern, or click **Cancel** to exit.

## Modifying Match Patterns

**Step 1**    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select Match Patterns from the pull-down menu.

**Step 2**    In the Match Pattern area, select a pattern by double-clicking on that pattern. Match Pattern information similar to the following appears (Figure 4-31 "Dialog Box for New Match Pattern in Active Dataset"):

*Figure 4-31    Dialog Box for New Match Pattern in Active Dataset*
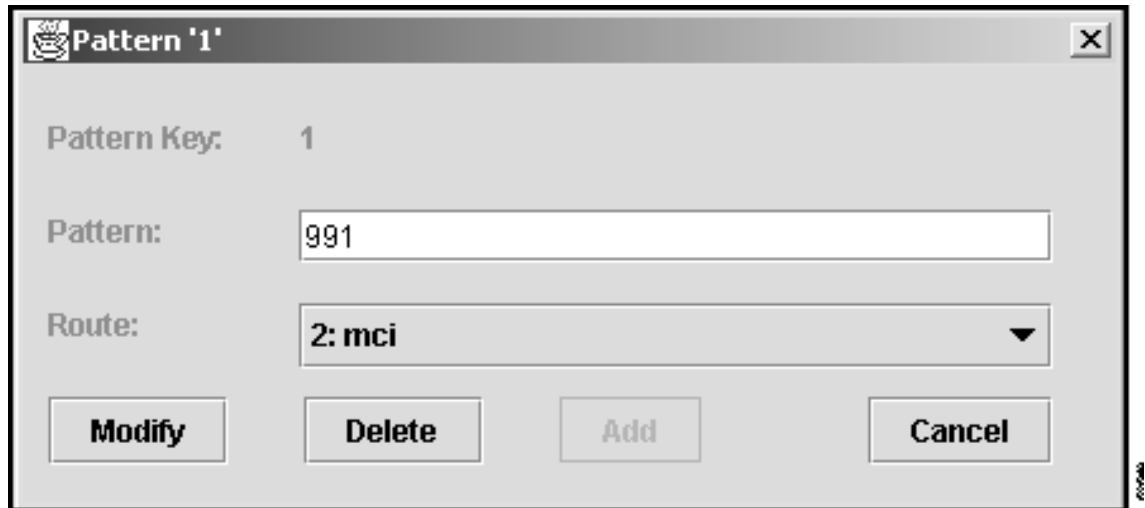


**Step 3**    Change the information accordingly.

**Step 4**    Click **Modify** to save the changes, or click **Cancel** to leave the changes.

## Deleting Match Patterns

**Step 1**    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Match Patterns** from the pull-down menu.

**Step 2**    In the Match Pattern area, select a pattern. You can select more than one pattern by holding down the Ctrl key while clicking each pattern.

**Step 3**    Click **Delete**. A Delete message appears.

**Step 4**    Click **Yes** to delete the pattern, or click **No** to stop the deletion.

> **Note**    Alternatively, you can delete a pattern by clicking **Delete** when you are in the dialog box for modifying the pattern.

# Creating, Modifying, and Deleting Routes

This section describes the steps to create, modify, and delete routes.

## Creating Routes

**Step 1**   In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Match Patterns** from the pull-down menu.

**Step 2**   Click **New.** The New Route dialog box appears (Figure4-32 "Dialog Box for New Route in Active Dataset"):

*Figure4-32   Dialog Box for New Route in Active Dataset*



The key shown on the screen is the primary key of this route in the Route table. When a new route is being created, "%%%%" is displayed.

**Step 3**   In the Name box, enter the name of the route. The range is 1 to 64 characters. This entry is optional.

**Step 4**   In the Rule box, click the **down arrow**, then select from the pull-down menu a rule that applies to the patterns to be used.

**Step 5**   Click QoS Relationships. The QoS Relationships dialog box appears (Figure4-33 "QoS Relationships Dialog Box in Active Dataset").

*Figure 4-33   QoS Relationships Dialog Box in Active Dataset*
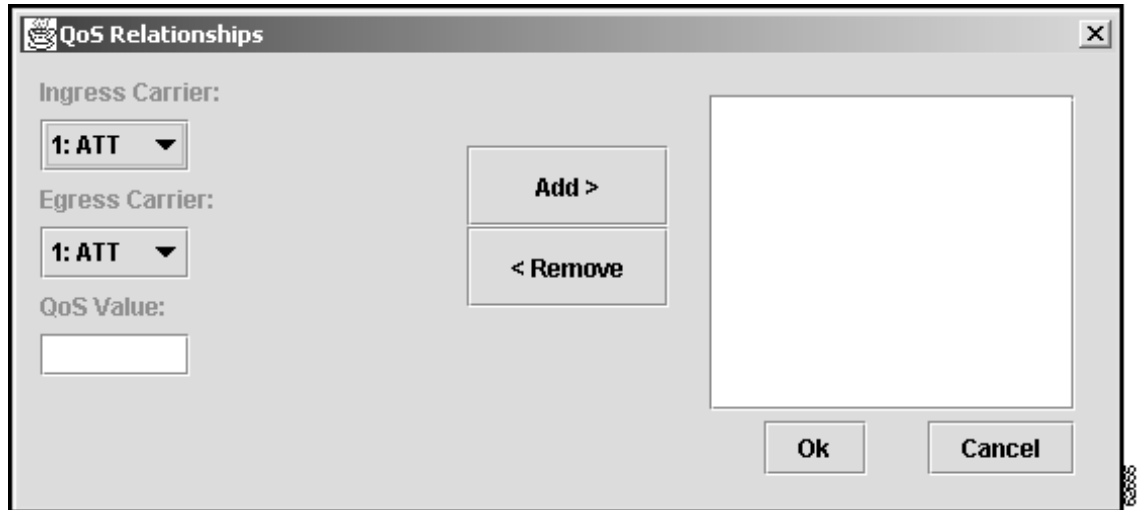


a.  In the Ingress Carrier box, click the **down arrow**, then select the ingress carrier from the pull-down menu.

b.  In the Egress Carrier box, click the **down arrow**, then select the egress carrier from the pull-down menu.

c.  In the QoS Value box, enter a number larger than zero.

d.  Click **Add** or **Remove** to add or remove the QoS relationship.

e.  Click **OK** to complete adding the relationship or click **Cancel** to exit without adding the relationship.

**Step 6**    In the Available Patterns box, select a pattern and click **Add** to add a pattern to the route. The patterns in this box are not bound to this route until they are added to the Patterns in Route box. Click **Remove** to remove a pattern bound to the route.

**Note**    A pattern can be bounded to only one route at a time. If you add a pattern to the Patterns in Route list, that pattern is removed from all other routes.

**Step 7**    Click **Add** to add this route, or click **Cancel** to exit.

## Modifying Routes

**Step 1**    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Match Patterns** from the pull-down menu.

**Step 2**    In the Routes area, select a route by double-clicking the route. Routes information similar to the following appears (Figure 4-34 "Dialog Box for an Existing Route in Active Dataset"):

*Figure 4-34   Dialog Box for an Existing Route in Active Dataset*
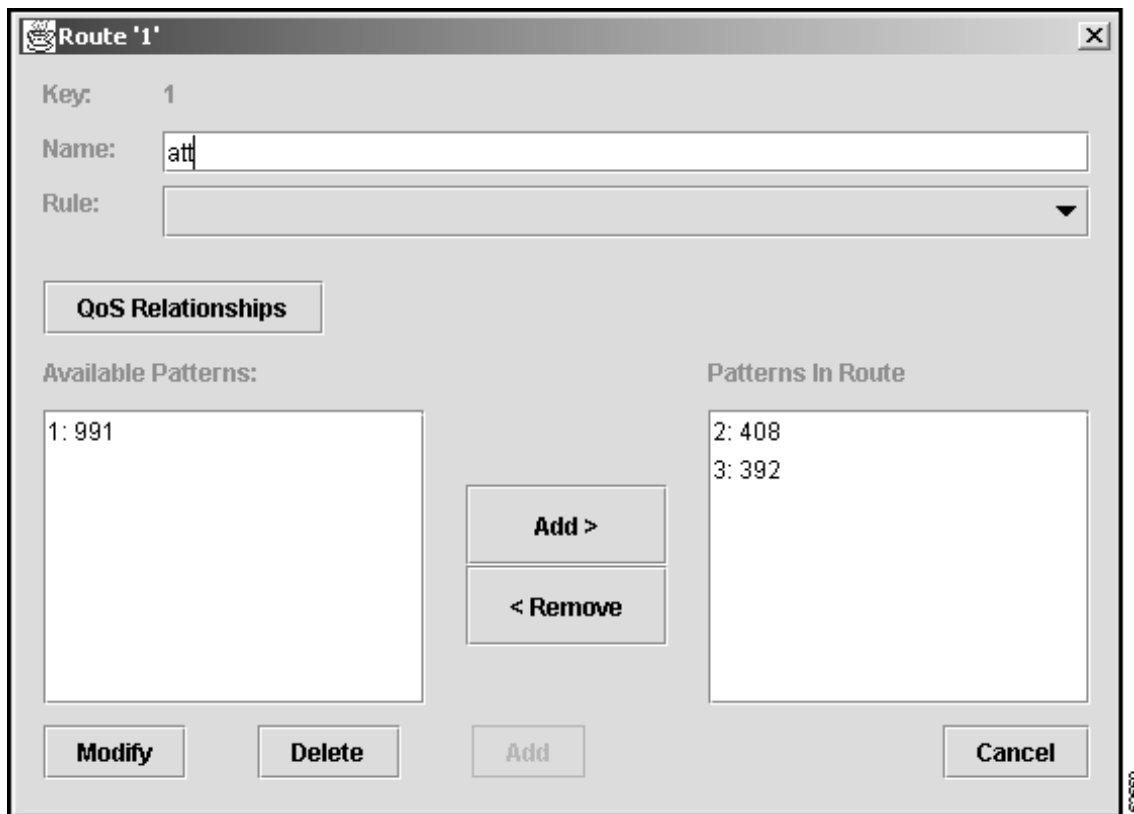


**Step 3**    Change the information accordingly.

**Step 4**    Click **Modify** to save the changes, or click **Cancel** to leave the changes.

## Deleting Routes

**Step 1**    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Route**s from the pull-down menu.

**Step 2**    In the Routes area, select a route. You can select more than one route by holding down the Ctrl key while clicking each route.

**Step 3**    Click **Delete**. A Delete message appears.

**Step 4**    Click **Yes** to delete the route, or click **No** to stop the deletion.

> **Note**    Alternatively, you can delete a route by clicking **Delete** when you are in the dialog box for modifying the route.

# Creating, Modifying, and Deleting Rules

This section describes the steps to create, modify and delete rules.

## Creating Rules

**Step 1**    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Rules** from the pull-down menu.

**Step 2**    Click **New**.

The New Rule dialog box appears (Figure 4-35 "Dialog Box for New Rule in Active Dataset"):

*Figure 4-35    Dialog Box for New Rule in Active Dataset*



The key shown on the screen is the primary key of this rule in the Rules table. When a new rule is being created, "%%%%" is displayed.

**Step 3**    In the Name box, enter the name of the rule. The range is 1 to 64 characters.

**Step 4**    In the Min QoS box, enter the lowest QoS level associated with this rule. The range is 1 to 9.

**Step 5**    Click the checkbox to the left of Is Default Rule to select this rule as the default rule.

> **Note**    One rule must be made the default rule for the provisioned data to be valid.

**Step 6**    In the Rejection Rules (Ingress) area, check the checkbox next to the rule that you want to activate as a rejection rule as follows:

- Reject Max Origination Percentage—If the ingress carrier's current percentage of calls is greater than its provisioned maximum, the call is rejected.

> **Note** This rule rejects a call if the percentage of calls originated by this ingress carrier to a specific route or match pattern exceeds the maximum percentage defined in the ingress route attributes for this ingress carrier. For example, if the maximum percentage is set to 40 percent, the first call made by carrier A to route R succeeds and 100 percent of the calls to route R are now made from ingress carrier A. The second call from the same carrier to the same DNIS fails.

- Reject Max Ingress Units—If the ingress carrier's ingress units are greater than its provisioned maximum, the call is rejected.
- Reject Max Ingress Units Per Route—If the ingress carrier's ingress units for this route are greater than its provisioned maximum for this route, the call is rejected.

> **Note** The core CSR has not been customized for Call Minute Accounting. "Units" means "call attempts". Once a specific carrier has attempted more calls than configured against "Max Ingress Units" in the Carrier menu, its calls are rejected.

**Step 7** In the Rejection Rules (Egress) area, check the checkbox to the left of the rule that you want to activate as a rejection rule as follows:

- Reject Egress Cost Greater Than Ingress—If the ingress carrier and egress carrier both have a provisioned cost and the egress carrier's cost is greater, the egress carrier is eliminated.
- Reject Insufficient QoS—If the rule has the following elements, the egress carrier is eliminated:
  - A provisioned minimum QoS
  - The egress carrier has a QoS value available
  - The egress carrier's QoS is worse (higher)
- Reject Max Egress Units Exceeded—If the egress carrier's egress units are greater than its provisioned maximum, the carrier is eliminated.
- Reject Max Egress Units Per Route Exceeded—If the egress carrier's egress units for this route are greater than its provisioned maximum for this route, the carrier is eliminated.

> **Note** The core CSR has not been customized for Call Minute Accounting. "Units" means "call attempts". Once a specific carrier has attempted more calls than configured against "Max Ingress Units" in the Carrier menu, its calls are rejected.

- Prevent ITSP to ITSP—If the ingress carrier is an ITSP type, all egress carriers that belong to the ITSP type are eliminated.

**Step 8** In the Selection Rules area, check the checkbox next to the rule you want to activate.

Each selection rule must have an associated priority if it is selected. Valid priority levels are from 1 to 5. The following rules can be applied for selecting an egress carrier:

- Select Min Cost Per Egress Route—Egress carriers without a cost value and sharing the lowest provisioned cost are selected. All others are deselected.
- Select Best QoS—Egress carriers without a QoS value and sharing the best provisioned QoS are selected. All others are deselected.

- Select Same Carrier—If the ingress carrier is also a egress carrier candidate, it is selected and all others are deselected.

- Select Percentage Egress—Carriers with a zero percentage egress and sharing the greatest difference between their percentage egress and their maximum percentage egress are selected. The purpose is to select the best egress carrier candidates that are furthest from meeting their provisioned maxPercentage on this route (based on their Egress Routing Attribute's maxPercentage value). The other carriers are moved to the deselect list.

> **Note**    Carriers with a null maxPercentage value are not moved to the deselect list.

- Select Preferred Carrier—If the preferred carrier is an egress carrier candidate, it is selected and all others are deselected. When this is selected, click the **down arrow** in the Preferred Carrier box. A preferred carrier pull-down menu appears.

- Percentage Termination—Find the best candidate egress carriers farthest from meeting their provisioned maximum percentage on this route. This is based on their Egress Routing Attribute's maxPercentage value. The other carriers, except those with a null maxPercentage value, are then moved to the "deselect" list.

**Step 9**    Click **Add** to add this rule, or click **Cancel** to exit.

## Modifying Rules

**Step 1**    In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Rules** from the pull-down menu.

**Step 2**    In the Rule area, select a rule by double-clicking the rule. Rule information similar to the following appears (Figure 4-36 "Dialog Box for Existing Rule in Active Dataset"):

*Figure 4-36   Dialog Box for Existing Rule in Active Dataset*



**Step 3**   Change the information accordingly.

**Step 4**   Click **Modify** to save the changes, or click **Cancel** to leave the changes.


## Deleting Rules

**Step 1**   In the Active Dataset tab in the CSR window, click the **down arrow** in the Selection box, then select **Rules** from the pull-down menu.

**Step 2**   In the Rule area, select a rule. You can select more than one rule by holding down the Ctrl key by clicking each rule.

**Step 3**   Click **Delete** . A Delete message appears.

**Step 4**   Click **Yes** to delete the rule, or click **No** to quit deleting.

**Note**   Alternatively, you can delete a rule by clicking **Delete** when you are in the dialog box for modifying the rule.

# Verifying Datasets

To verify whether a dataset is valid, click **Verify Dataset** in the Dataset Commands area.

This command ensures that all the provisioned data for the active dataset is valid. If the dataset is not valid, it cannot be loaded into the active CSR memory. After verification, a message with the verification results appears.

**Note**    You can use a name and an ID for a carrier, contact, contact list, route attribute, match pattern, route, and rule. The ID must be unique.

# Updating Inactive Datasets

This section describes how to update inactive datasets by using the Inactive Dataset tab in the CSR window to create, modify, and delete the following items.

- Carriers
- Contacts
- Contact lists
- Egress and ingress route attributes
- Match patterns
- Routes
- Rules

After you log in to CSR and click the Inactive Dataset tab in the CSR window, the following screen appears (Figure 4-37 "Inactive Dataset Tab in CSR Window"):

*Figure 4-37   Inactive Dataset Tab in CSR Window*
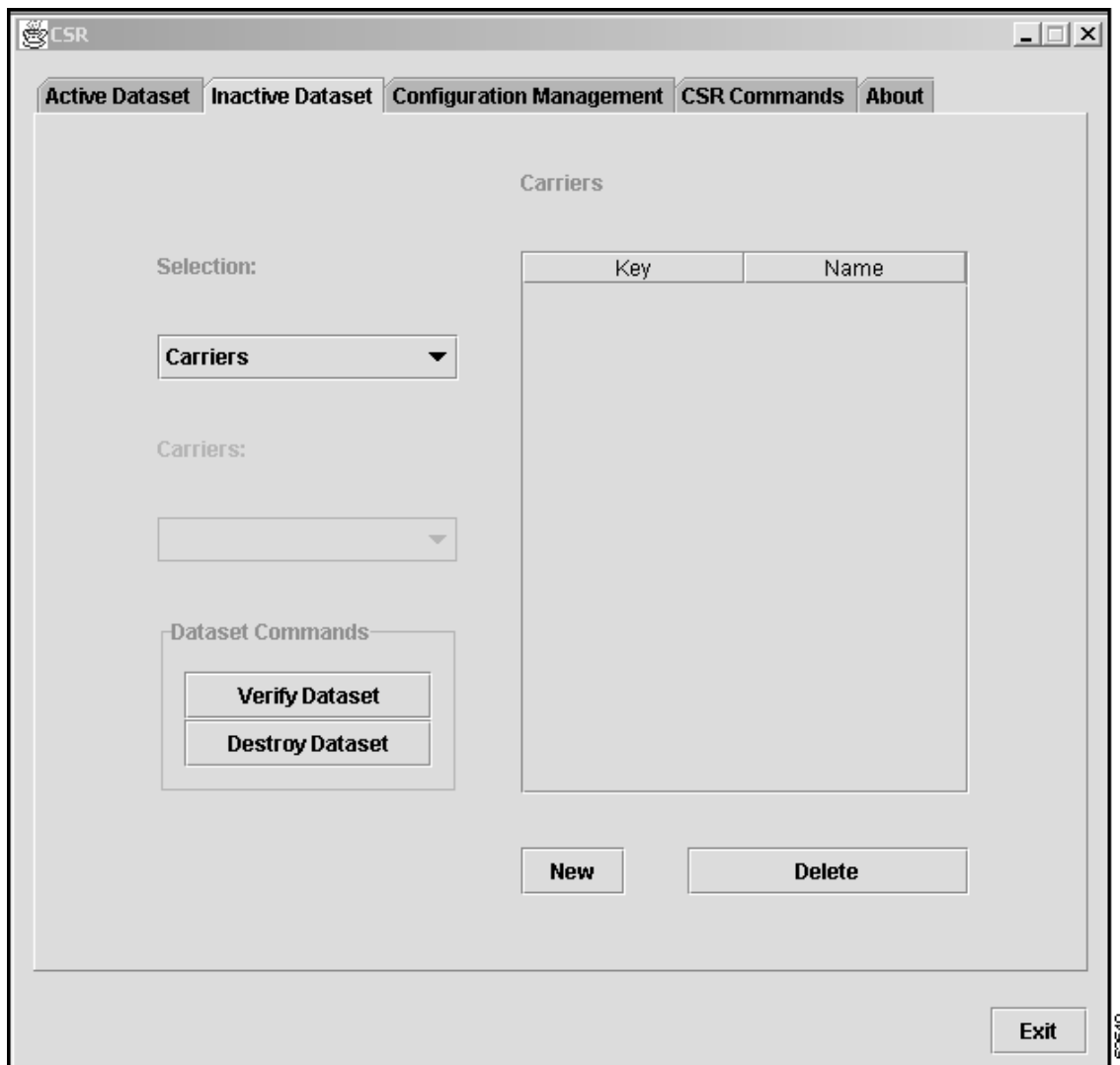


Only one functionality in the Inactive Dataset tab is different from those in the Active Dataset tab. The additional functionality, Destroy Dataset, is described in the following section. See "Updating Active Datasets" section on page 4-12 for details on all other functionalities.

## Destroying Inactive Datasets

**Step 1**  In the Inactive Dataset tab in the CSR window, click **Destroy Dataset**. A Destroy message appears (Figure 4-38 "Destroy Message in Inactive Dataset"):

*Figure4-38   Destroy Message in Inactive Dataset*
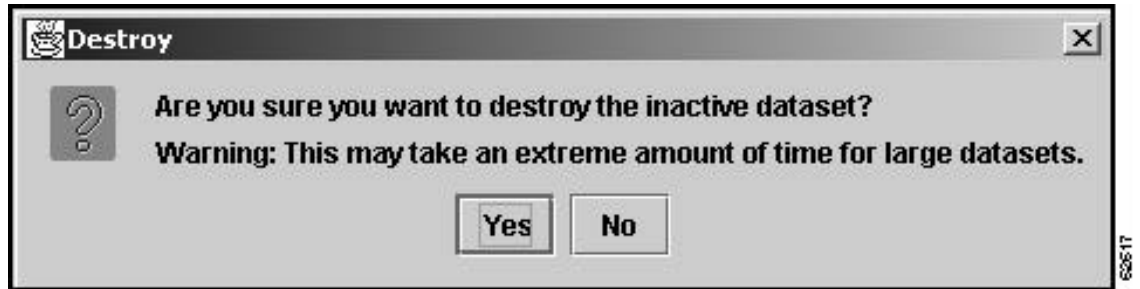


**Step 2**    Click **Yes** to destroy the inactive dataset, or **No** to quit destroying.

## Activating Inactive Datasets

An inactive dataset is not loaded into the CSR memory, with the exception of test commands. To place the dataset in the CSR memory for routing, use the following steps:

**Step 1**    In the Inactive Dataset tab in the CSR window, click **Verify Dataset** in the Dataset Commands area to verify the inactive dataset.

**Step 2**    Click the Configuration Management tab, and click **Swap Active/Inactive Datasets** to swap the inactive dataset to active dataset.

**Step 3**    In the Select Dataset box, click **Load Dataset into CSR**.

**Step 4**    Click **Swap** to swap the datasets, or click **Cancel** to cancel swapping.

# CSR Commands

This section describes how to use the CSR Commands tab in the CSR window for the following functionalities:

-
-
-
-
-
-

# Sending Test Route Request Commands to an Active CSR Application

**Step 1**    Click the **CSR Commands** tab in the CSR window. The following screen appears (CSR Commands Tab in CSR Window, page 4-42):

*Figure4-39   CSR Commands Tab in CSR Window*



**Step 2**    In the ANI box, enter the origination pattern.

**Step 3**    In the DNIS box, enter the pattern dialed.

**Step 4**    In the Source Carrier Id box, enter the ingress carrier to be used for testing.

**Step 5**    In the Destination Carrier Id box, enter the test egress carrier.

**Note**    If an egress carrier is specified, the call is ignored. It is assumed that the egress carrier has been found.

**Step 6**    In the Time box, enter the simulated time to test time-sensitive calls. If this is not specified, the current time is used. The format and an example for time entry are shown below:

`hour(0-23):min (0-60):sec (0-60)` for time, as in 12:59:10

This entry is optional.

**Step 7**    In the Date box, enter the simulated date to test date-sensitive and time-sensitive calls.

If this is not specified, the current date is used. Format and example for date entry are:

`4 digit year-2 digit month-2 digit day` for date, as in 2002-12-03

**Step 8**    Click the button to the left of Active or Inactive for the provisioned dataset for running the test. This dataset is stored in the CSR database. The current dataset in CSR memory is not used.

**Step 9**    Click **Send Test to CSR** to send the request to the CSR for processing. Test results are automatically displayed at end of test. They are stored in the database for later retrieval.

**Step 10**    To display test results as needed, click **Show Test Results**. The test results appears (Figure4-40 "Test Results Generated by the CSR Commands Tab"). The list usually contains selected carriers followed by unselected carriers.

*Figure4-40   Test Results Generated by the CSR Commands Tab*
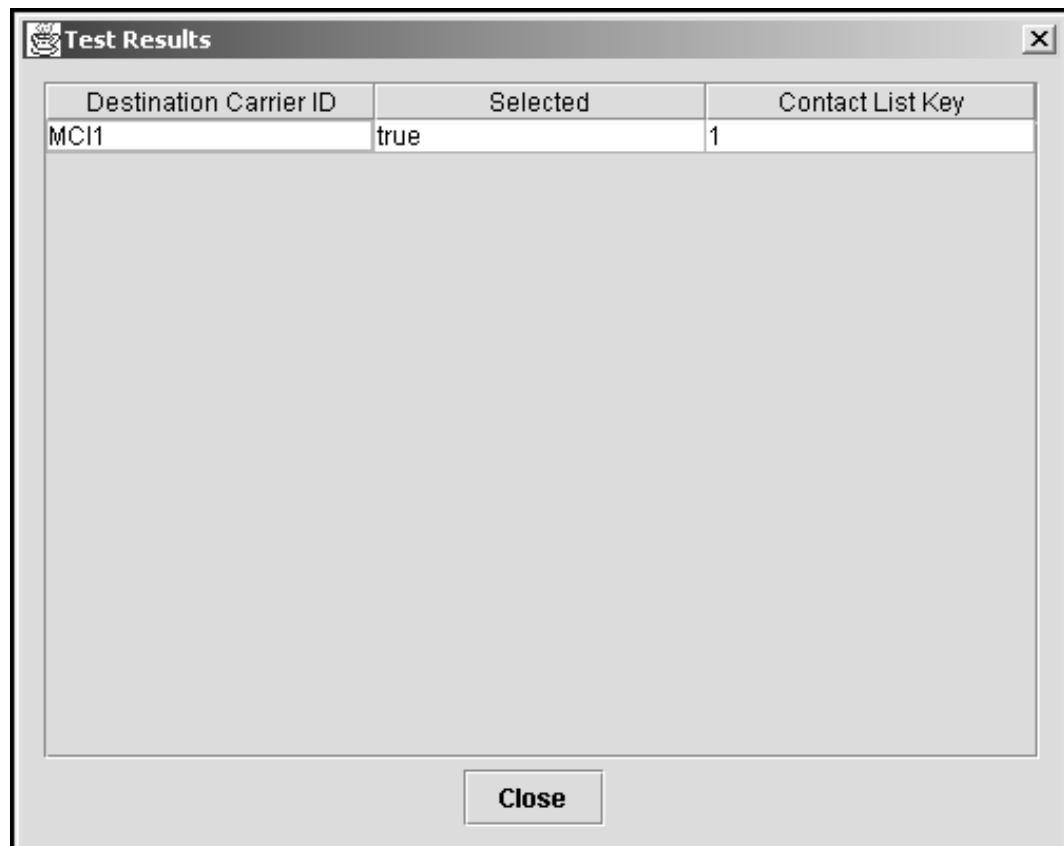
**Step 11**    To display a log of actions used to process the test results, click **Show Log**. The following screen appears (Figure4-41 "Log Entries Generated by the CSR Commands Tab"):

*Figure4-41   Log Entries Generated by the CSR Commands Tab*
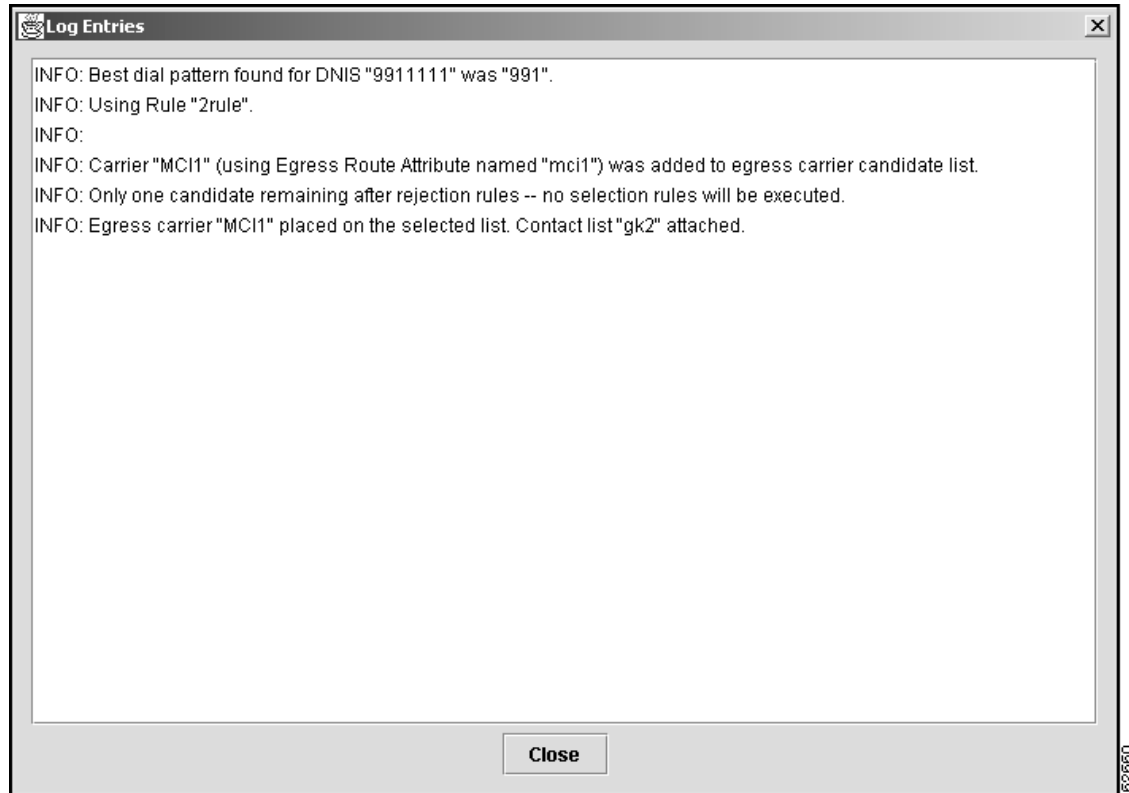


The log messages, normally routed to syslog, are logged to the database. The logging level is set to Informational for all test requests. See Appendix A, "Logging" for more detail on syslog.

The following sections describes other CSR functionalities in the CSR Commands tab in the CSR window.

# Updating Logging Level

To specify the logging level, click **Update Logging Level**.

# Loading the Active Dataset

To load the active dataset into memory for use, click **Load Active Dataset**. Because calls are always in progress, both datasets can be in use simultaneously. Calls currently in progress use the old dataset and new calls use the new dataset.

# Resetting Call Accounting

To reset all call units to zero, click **Reset Call Accounting**. This sets all accounting per carrier and per route/carrier for the CSR. This is the same data used by the Test Route Request functionality.

# Updating Gatekeepers

To update the CSR gatekeepers table, click **Update GateKeepers**. The CSR unregisters each message type from each gatekeeper and reconnects the gatekeepers on the current list.

# Cleaning Test Results

To purge all the test requests (inputs and outputs) from the database, click **Clean Test Results**.

# Checking CSR Version

To check the CSR version number, select the About tab in the CSR window.

# Multiple GUI Operation

Multiple GUIs can run against a single database and multiple databases. If all users are working in a read-only mode, the following results occur:

- If two or more users are modifying the same data, the last saved data is the current data.
- A deleted item is visible to a user if another user removes it.
- Added data may not be seen immediately by all users.

# Logging

This appendix describes the following activities:

- Configuring Syslog
- Viewing Logs

## Configuring Syslog

This section provides an overview of the UNIX syslog. For more detail, refer to the online manual page for syslog.conf.

Syslog is a general location or mechanism for logging informational messages or error messages. The messages are categorized by level, and each level can be directed to a different location specified by files, consoles, and e-mail addresses. Each line in the syslog.conf file describes the type of message, the location and how that message should be handled.

Available message levels in syslog are as follows:

- emerg—Emergency panic
- alert—Alert
- crit—Critical errors
- err—Other errors
- warning—Warning
- notice—Notice
- info—Informational messages
- debug—Debugging messages
- none—Do not send message to file

**Note** The Carrier-Sensitive Routing application sends messages at the err, info, and debug levels only. The debug level includes various GateKeeper Transaction Message Protocol interface attempts and additional logging of the dataset load process. This can be useful in determining where a valid dataset in the database may be fully or partially rejected during the loading process into the CSR.

Messages can originate from the following locations or facilities:

- user—User processes from the CSR application.
- kern—Kernel
- mail—Mail system
- daemon—System daemons
- auth—Authorization system, such as login or su.
- lpr—Print spooler
- news—Network new systems

# Examples

***Example A-1    Syslog sends all error messages from all locations to the console for everyone***

```
*.err   /dev/console
```

***Example A-2    Syslog sends all error messages generated from the kernel to the
/var/adm/kern_messages file***

```
kern.err        /var/adm/kern_messages
```

***Example A-3    Syslog sends all error messages, notice messages and error message from the
authorization system to /var/adm/some_messages file***

```
*.err,*.notice,auth.err        /var/adm/some_messages
```

***Example A-4    Syslog sends error messages to console and /var/adm/messages file separately***

```
*.err   /dev/console
*.err   /var/adm/messages
```

**Note**    Log messages can be captured in more than one location.

***Example A-5    Syslog sends message to the adminuser email address on the system***

```
*.emerg adminuser
```

**Note**    Instead of adminuser, an asterisk (*) may be used. This sends the message to all e-mail accounts.

# Viewing Logs

To view logs generated by the CSR application, check your syslog configuration file. Error logs are located in /var/adm/messages. Each log contain the following information:

- Date the log entry was generated
- Time the log entry was generated

- Name of the machine that generated the log

- Application/PID

- Log level

- Message content

## Sample logs

- Jan  2 13:01:14 ciscoe250 CSR01[8333]: ERROR: CSR Started.

**Note**    This is logged at err level to note when the CSR was started, even if debug and informational logging is off.

- Jan  2 13:01:15 ciscoe250 CSR01[8333]: DEBUG: Current size of call queue: 0 (2 known GateKeeper(s)).

- Jan  2 13:01:17 ciscoe250 CSR01[8333]: INFO: Report Version notification received -- CSR Version reported to the database.

- Jan  2 13:01:18 ciscoe250 CSR01[8333]: DEBUG: Current size of call queue: 0 (2 known GateKeeper(s)).

# Data Schema

This appendix describes the following:

- Configuration Data Schema
- Dataset Schema

## Configuration Data Schema

```
BEGIN WORK;
SET CONSTRAINTS ALL DEFERRED;


CREATE TABLE ConfigurationData (
    ActiveDataset   text        NOT NULL DEFAULT '_1' CONSTRAINT
ActiveDatasetSuffixInvalid
        CHECK (
            lower(ActiveDataset) = '_1' OR lower(ActiveDataset) = '_2'
        ),
    LoggingLevel text        NOT NULL DEFAULT 'Error' CONSTRAINT
LoggingLevelInvalid
        CHECK (
            lower(LoggingLevel) = 'error' OR
            lower(LoggingLevel) = 'info'  OR
            lower(LoggingLevel) = 'debug'
        ),
    CSR_ID         varchar(64) NOT NULL,
    CSR_Version  text,
    GKTMP_Versioninteger,
    Priority      integer     NOT NULL DEFAULT 1CONSTRAINT
                              CSRPriorityInvalid
                              CHECK (Priority >= 1),
```

```
                ListenPort      integer     CONSTRAINT CSRListenPortInvalid
                                            CHECK (ListenPort > 0 AND ListenPort <=
          32767),
                ThreadCount     integer     NOT NULL DEFAULT 10,
                MaxSizeOfQueue integer      NOT NULL DEFAULT 5000,


                TestRoutesAccountingEnabledboolean NOT NULL DEFAULT FALSE,
                RejectSrcCarrierInactive  boolean NOT NULL DEFAULT FALSE,
                RejectNullSrcCarrier      boolean NOT NULL DEFAULT FALSE,
                RejectSrcCarrierUnknown   boolean NOT NULL DEFAULT FALSE,
                RejectDestinationUnknown  boolean NOT NULL DEFAULT FALSE,
                RejectSrcPercentage       boolean NOT NULL DEFAULT FALSE,
                RejectSrcUnits            boolean NOT NULL DEFAULT FALSE,
                RejectNoDstCarrier        boolean NOT NULL DEFAULT FALSE
          );


          CREATE TABLE GateKeepers (
                GKKey           integer     NOT NULL PRIMARY KEY,
                GKID            varchar(64) NOT NULL UNIQUE,
                GK_IP           inet        NOT NULL,
                GK_Port         integer     NOT NULL CONSTRAINT GK_PortInvalid
                                            CHECK (GK_Port > 0 AND GK_Port <= 32767),
                ListenARQ       boolean     NOT NULL DEFAULT FALSE,
                ListenLRQ       boolean     NOT NULL DEFAULT FALSE,
                ListenIRR       boolean     NOT NULL DEFAULT FALSE,
                ListenDRQ       boolean     NOT NULL DEFAULT FALSE,
                ListenRRQ       boolean     NOT NULL DEFAULT FALSE,
                ListenURQ       boolean     NOT NULL DEFAULT FALSE,
                ListenLCF       boolean     NOT NULL DEFAULT FALSE,
                ListenLRJ       boolean     NOT NULL DEFAULT FALSE,
                ListenRAI       boolean     NOT NULL DEFAULT FALSE,
                ListenBRQ       boolean     NOT NULL DEFAULT FALSE,


                ConnectionStatetext         NOT NULL CONSTRAINT
                ConnectionStateInvalid
                                            CHECK (
                                             lower(ConnectionState) = 'connect' OR
                                             lower(ConnectionState) = 'disconnect'
```

```
                                  )
        );


        CREATE TABLE TestRoutes (
           TestKey        integer     NOT NULL PRIMARY KEY,
           TestTime       time,
           TestDate       date,
           DatasetID      text        DEFAULT '_1' CONSTRAINT DatasetIDInvalid
                                      CHECK (
                                        lower(DatasetID) = '_1' OR
           lower(DatasetID) = '_2'
                                      ),
           ANI            varchar(64),
           DNIS           varchar(64),
           SrcCarrierID   varchar(64),
           DstCarrierID   varchar(64),
           RunMe          boolean     NOT NULL DEFAULT FALSE,
           HomeCarrierID varchar(64),
           ResultCode     text
        );


        CREATE TABLE TestOutputs (
           TestKey        integer     NOT NULL,
           SelectOrder    integer     NOT NULL,
           DstCarrierID   varchar(64) NOT NULL,
           ListKey        integer     NOT NULL,
           IsSelected     boolean     NOT NULL DEFAULT FALSE
        );
        CREATE TABLE LogEntries (
            TestKey        integer     NOT NULL,
            EntryOrder    integer     NOT NULL,
            LogString     varchar(1024)NOT NULL
        );


        CREATE TABLE StringResources (
            ResourceKey  integer     NOT NULL PRIMARY KEY,
            ResourceStringtext        NOT NULL
        );
```

```
CREATE TABLE Filters (
     FilterKey    integer     NOT NULL PRIMARY KEY,
   GKID          varchar(64) NOT NULL,
   MsgType       text        NOT NULL CONSTRAINT MsgTypeInvalid
                 CHECK (
                  lower(MsgType) = 'arq' OR
                  lower(MsgType) = 'lrq' OR
                  lower(MsgType) = 'rrq' OR
                  lower(MsgType) = 'urq' OR
                  lower(MsgType) = 'lcf' OR
                  lower(MsgType) = 'lrj' OR
                  lower(MsgType) = 'drq' OR
                  lower(MsgType) = 'brq' OR
                  lower(MsgType) = 'irr' OR
                  lower(MsgType) = 'rai'
                 ),
    FilterType    text          NOT NULL CONSTRAINT FilterTypeInvalid
                                CHECK (
                                 lower(FilterType) = 'endpoint'OR
                                 lower(FilterType) = 'supportedprefix'OR
                                 lower(FilterType) =
                 'destinationinformation'  OR
                                 lower(FilterType) =
                 'remoteextensionaddress'  OR
                                 lower(FilterType) = 'redirectreason'OR
                                 lower(FilterType) = 'answercall'OR
                                 lower(FilterType) = 'notificationonly'
                                ),
    Value         text          NOT NULL
);

ALTER TABLE TestOutputs ADD FOREIGN KEY (TestKey) REFERENCES TestRoutes
(TestKey) DEFERRABLE;
ALTER TABLE LogEntries  ADD FOREIGN KEY (TestKey) REFERENCES TestRoutes
(TestKey) DEFERRABLE;
ALTER TABLE Filters     ADD FOREIGN KEY (GKID)    REFERENCES GateKeepers
(GKID)    DEFERRABLE;
COMMIT;
```

# Dataset Schema

```
BEGIN WORK;


SET CONSTRAINTS ALL DEFERRED;


CREATE TABLE Carriers (
   CarrierKey     integer     NOT NULL PRIMARY KEY CONSTRAINT
                               CarrierKeyNotZero
                               CHECK (CarrierKey <> 0),
   CarrierName    varchar(64) NOT NULL,
   CarrierType    text        NOT NULL
                  CONSTRAINT CarrierTypeInvalid CHECK (
                   lower(CarrierType) = 'home' OR
                   lower(CarrierType) = 'itsp' OR
                   lower(CarrierType) = 'tdm'
                  ),
   CarrierID      varchar(64) NOT NULL UNIQUE,
     MaxIngressUnitsinteger   CONSTRAINT MaxIngressUnitsInvalid
                               CHECK (MaxIngressUnits >= 0),
     MaxEgressUnitsinteger    CONSTRAINT MaxEgressUnitsInvalid
                               CHECK (MaxEgressUnits >= 0),
   RuleKey        integer,
   ListKey        integer     NOT NULL,
   isActive       boolean     NOT NULL DEFAULT TRUE
);
CREATE TABLE Carriers_1 () INHERITS (Carriers);
CREATE TABLE Carriers_2 () INHERITS (Carriers);


CREATE TABLE MatchPatterns (
   PatternKey     integer     NOT NULL PRIMARY KEY CONSTRAINT
                               PatternKeyNotZero
                               CHECK (PatternKey <> 0),
     Pattern      varchar(64) NOT NULL UNIQUE,
     RouteKey     integer
);
CREATE TABLE MatchPatterns_1 () INHERITS (MatchPatterns);
CREATE TABLE MatchPatterns_2 () INHERITS (MatchPatterns);
```

```
CREATE TABLE Routes (
   RouteKey        integer     NOT NULL PRIMARY KEY CONSTRAINT
                                RouteKeyNotZero
                                CHECK (RouteKey <> 0),
   RouteName       text,
   RuleKey         integer
);
CREATE TABLE Routes_1 () INHERITS (Routes);
CREATE TABLE Routes_2 () INHERITS (Routes);


CREATE TABLE Rules (
   RuleKey         integer     NOT NULL PRIMARY KEY CONSTRAINT
                                RuleKeyNotZero
                                CHECK (RuleKey <> 0),
   RuleName        text,
   IsDefaultRule   boolean     NOT NULL DEFAULT FALSE,
   MinQoS          integer     CONSTRAINT MinQoSInvalid
                                CHECK (MinQoS >= 1 AND MinQoS <= 9),
   PreferredCarrierKey integer,

   RejectMaxOriginationPercentageboolean NOT NULL DEFAULT FALSE,
   RejectMaxIngressUnits      boolean    NOT NULL DEFAULT FALSE,
   RejectMaxIngressUnitsPerRouteboolean  NOT NULL DEFAULT FALSE,
   RejectEgressCostMoreThanIngressbooleanNOT NULL DEFAULT FALSE,
   RejectInsufficientQoS      boolean    NOT NULL DEFAULT FALSE,
   RejectMaxEgressUnits       boolean    NOT NULL DEFAULT FALSE,
   RejectMaxEgressUnitsPerRouteboolean   NOT NULL DEFAULT FALSE,
   PreventITSPtoITSP          boolean    NOT NULL DEFAULT FALSE,

   SelectMinCostPerEgressRouteinteger    NOT NULL DEFAULT 0 CONSTRAINT
                                         SelMinCostInvalid
                                CHECK (
                                SelectMinCostPerEgressRoute >= 0 AND
                                SelectMinCostPerEgressRoute <= 5
                                ),
   SelectBestQoS                integer NOT NULL DEFAULT 0 CONSTRAINT
                                SelBestQoSInvalid
```

```
                                    CHECK (
                                     SelectBestQoS >= 0 AND
                                     SelectBestQoS <= 5
                                    ),
        SelectSameCarrier           integer    NOT NULL DEFAULT 0 CONSTRAINT
                                               SelSameCarrierInvalid
                                    CHECK (
                                     SelectSameCarrier >= 0 AND
                                     SelectSameCarrier <= 5
                                    ),
        SelectPreferredCarrier      integer    NOT NULL DEFAULT 0 CONSTRAINT
                                               SelPreferredCarrierInvalid
                                    CHECK (
                                     SelectPreferredCarrier >= 0 AND
                                     SelectPreferredCarrier <= 5
                                    ),
        SelectPercentageEgress      integer    NOT NULL DEFAULT 0 CONSTRAINT
                                               SelPercentageEgressInvalid
                                    CHECK (
                                     SelectPercentageEgress >= 0 AND
                                     SelectPercentageEgress <= 5
                                    )
);
CREATE TABLE Rules_1 () INHERITS (Rules);
CREATE TABLE Rules_2 () INHERITS (Rules);

CREATE TABLE Contacts (
   ContactKey     integer    NOT NULL PRIMARY KEY CONSTRAINT
                              ContactKeyNotZero
                              CHECK (ContactKey <> 0),
   RASAddress     varchar(64) NOT NULL,
   AddressType    text       NOT NULL CONSTRAINT
   ContactAddressTypeInvalid
                  CHECK (
                   lower(AddressType) = 'ipv4' OR
                   lower(AddressType) = 'gkid' OR
                   lower(AddressType) = 'dnsname'
                  ),
```

```
       Cost             integer     CONSTRAINT CostInvalid
                        CHECK (Cost >= 0),
    Priority            integer     CONSTRAINT PriorityInvalid
                        CHECK (Priority >= 0)
);
CREATE TABLE Contacts_1 () INHERITS (Contacts);
CREATE TABLE Contacts_2 () INHERITS (Contacts);


CREATE TABLE Lists (
    ListKey          integer     NOT NULL PRIMARY KEY CONSTRAINT
                                  ListKeyNotZero
                                  CHECK (ListKey <> 0),
    ListName text
);
CREATE TABLE Lists_1 () INHERITS (Lists);
CREATE TABLE Lists_2 () INHERITS (Lists);


CREATE TABLE ContactLists (
    ListKey          integer     NOT NULL,
    ContactKey       integer     NOT NULL
);
CREATE TABLE ContactLists_1 () INHERITS (ContactLists);
CREATE TABLE ContactLists_2 () INHERITS (ContactLists);


CREATE TABLE RouteAttributes (
    AttributeKey     integer     NOT NULL PRIMARY KEY CONSTRAINT
                                  AttributeKeyNotZero
                                  CHECK (AttributeKey <> 0),
    AttributeName text,
    CarrierKey       integer     NOT NULL,
    DaypartType      text        NOT NULL CONSTRAINT DaypartInvalid
                        CHECK (
                         lower(DaypartType) = 'absolute'OR
                         lower(DaypartType) = 'weekly'OR
                         lower(DaypartType) = 'daily'OR
                         lower(DaypartType) = 'always'
                        ),
    WeekDay          integer     CONSTRAINT WeekdayInvalid
```

```
                                              CHECK (Weekday >=0 AND Weekday <= 6),
        StartTime       time,
        EndTime         time        CONSTRAINT TimeRangeInvalid
                                    CHECK (
                                     (StartTime ISNULL AND EndTime ISNULL AND
                                     lower(DaypartType) = 'always') OR
                                    (StartTime NOTNULL AND EndTime NOTNULL
        AND
                                    (StartTime < EndTime) OR
                                     (
                                     lower(DaypartType) = 'absolute' AND
        (StartDate <
                                     EndDate) AND
                                     (lower(DaypartType) != 'weekly' OR
        WeekDay NOTNULL)
                                     )
                                    )),
        StartDate       date,
        EndDate         date,
        Cost            integer     CONSTRAINT CostInvalid
                                    CHECK (Cost >= 1),
        MaxUnits        integer     CONSTRAINT MaxUnitsInvalid
                                    CHECK (MaxUnits >= 0),
        MaxPercentage   integer     CONSTRAINT MaxPercentageInvalid
                                    CHECK (MaxPercentage >= 1 AND
        MaxPercentage <=
                                     100)
);
CREATE TABLE IngressRouteAttributes_1 () INHERITS (RouteAttributes);
CREATE TABLE IngressRouteAttributes_2 () INHERITS (RouteAttributes);


CREATE TABLE EgressRouteAttributes (
    QoS             integer     CONSTRAINT QoSInvalid
                                CHECK (QoS >= 1 and QoS <= 9),
    ListKey integer
) INHERITS (RouteAttributes);
CREATE TABLE EgressRouteAttributes_1 () INHERITS
(EgressRouteAttributes);
CREATE TABLE EgressRouteAttributes_2 () INHERITS
(EgressRouteAttributes);
```

```
CREATE TABLE QoS (
   IngressCarrierKeyinteger  NOT NULL,
   EgressCarrierKeyinteger   NOT NULL,
   RouteKey         integer   NOT NULL,
   RelativeQoS      integer   NOT NULL CONSTRAINT RelativeQoSInvalid
                              CHECK (RelativeQoS >= 1 AND RelativeQoS <=
                               9)
);
CREATE TABLE QoS_1 () INHERITS (QoS);
CREATE TABLE QoS_2 () INHERITS (QoS);


CREATE TABLE RouteParent (
   RouteKey         integer   NOT NULL,
   AttributeKey     integer   NOT NULL
);
CREATE TABLE IngressRoutes_1 () INHERITS (RouteParent);
CREATE TABLE IngressRoutes_2 () INHERITS (RouteParent);
CREATE TABLE EgressRoutes_1  () INHERITS (RouteParent);
CREATE TABLE EgressRoutes_2  () INHERITS (RouteParent);
COMMIT;


BEGIN WORK;
CREATE UNIQUE INDEX Carriers_1_PKey ON Carriers_1 (CarrierKey);
CREATE UNIQUE INDEX Carriers_2_PKey ON Carriers_2 (CarrierKey);


CREATE UNIQUE INDEX Carriers_1_ID ON Carriers_1 (CarrierID);
CREATE UNIQUE INDEX Carriers_2_ID ON Carriers_2 (CarrierID);


CREATE UNIQUE INDEX Patterns_1_PKey ON MatchPatterns_1 (PatternKey);
CREATE UNIQUE INDEX Patterns_2_PKey ON MatchPatterns_2 (PatternKey);


CREATE UNIQUE INDEX Routes_1_PKey ON Routes_1 (RouteKey);
CREATE UNIQUE INDEX Routes_2_PKey ON Routes_2 (RouteKey);


CREATE UNIQUE INDEX Rules_1_PKey ON Rules_1 (RuleKey);
CREATE UNIQUE INDEX Rules_2_PKey ON Rules_2 (RuleKey);
```

```
CREATE UNIQUE INDEX Contacts_1_PKey ON Contacts_1 (ContactKey);

CREATE UNIQUE INDEX Contacts_2_PKey ON Contacts_2 (ContactKey);


CREATE UNIQUE INDEX Lists_1_PKey ON Lists_1 (ListKey);

CREATE UNIQUE INDEX Lists_2_PKey ON Lists_2 (ListKey);


CREATE UNIQUE INDEX Match_1_Pattern ON MatchPatterns_1(Pattern);

CREATE UNIQUE INDEX Match_2_Pattern ON MatchPatterns_2(Pattern);


CREATE UNIQUE INDEX IRA_1_PKey ON IngressRouteAttributes_1
(AttributeKey);

CREATE UNIQUE INDEX IRA_2_PKey ON IngressRouteAttributes_2
(AttributeKey);


CREATE UNIQUE INDEX ECA_1_PKey ON EgressRouteAttributes_1
(AttributeKey);

CREATE UNIQUE INDEX ECA_2_PKey ON EgressRouteAttributes_2
(AttributeKey);


CREATE UNIQUE INDEX ContactLists_1_Key ON ContactLists_1 (ListKey,
ContactKey);

CREATE UNIQUE INDEX ContactLists_2_Key ON ContactLists_2 (ListKey,
ContactKey);


CREATE UNIQUE INDEX QoS_1_Key ON QoS_1 (IngressCarrierKey,
EgressCarrierKey, RouteKey);

CREATE UNIQUE INDEX QoS_2_Key ON QoS_2 (IngressCarrierKey,
EgressCarrierKey, RouteKey);


CREATE UNIQUE INDEX IngressRoutes_1_Key ON IngressRoutes_1 (RouteKey,
AttributeKey);

CREATE UNIQUE INDEX IngressRoutes_2_Key ON IngressRoutes_2 (RouteKey,
AttributeKey);


CREATE UNIQUE INDEX EgressRoutes_1_Key ON EgressRoutes_1 (RouteKey,
AttributeKey);

CREATE UNIQUE INDEX EgressRoutes_2_Key ON EgressRoutes_2 (RouteKey,
AttributeKey);

COMMIT;


BEGIN WORK;
```

```
ALTER TABLE Carriers_1 ADD FOREIGN KEY (RuleKey) REFERENCES Rules_1
(RuleKey) DEFERRABLE;

ALTER TABLE Carriers_1 ADD FOREIGN KEY (ListKey)REFERENCES Lists_1
(ListKey) DEFERRABLE;

ALTER TABLE Carriers_2 ADD FOREIGN KEY (RuleKey) REFERENCES Rules_2
(RuleKey) DEFERRABLE;

ALTER TABLE Carriers_2 ADD FOREIGN KEY (ListKey)REFERENCES Lists_2
(ListKey) DEFERRABLE;


ALTER TABLE ContactLists_1 ADD FOREIGN KEY (ListKey)REFERENCES
Lists_1    (ListKey)    DEFERRABLE;

ALTER TABLE ContactLists_1 ADD FOREIGN KEY (ContactKey)REFERENCES
Contacts_1 (ContactKey) DEFERRABLE;

ALTER TABLE ContactLists_2 ADD FOREIGN KEY (ListKey)REFERENCES
Lists_2    (ListKey)    DEFERRABLE;

ALTER TABLE ContactLists_2 ADD FOREIGN KEY (ContactKey)REFERENCES
Contacts_2 (ContactKey) DEFERRABLE;


ALTER TABLE Routes_1 ADD FOREIGN KEY (RuleKey) REFERENCES Rules_1
(RuleKey) DEFERRABLE;

ALTER TABLE Routes_2 ADD FOREIGN KEY (RuleKey) REFERENCES Rules_2
(RuleKey) DEFERRABLE;


ALTER TABLE MatchPatterns_1 ADD FOREIGN KEY (RouteKey) REFERENCES
Routes_1 (RouteKey) DEFERRABLE;

ALTER TABLE MatchPatterns_2 ADD FOREIGN KEY (RouteKey) REFERENCES
Routes_2 (RouteKey) DEFERRABLE;


ALTER TABLE Rules_1 ADD FOREIGN KEY (PreferredCarrierKey) REFERENCES
Carriers_1 (CarrierKey) DEFERRABLE;

ALTER TABLE Rules_2 ADD FOREIGN KEY (PreferredCarrierKey) REFERENCES
Carriers_2 (CarrierKey) DEFERRABLE;


ALTER TABLE IngressRouteAttributes_1 ADD FOREIGN KEY (CarrierKey)
REFERENCES Carriers_1 (CarrierKey) DEFERRABLE;

ALTER TABLE IngressRouteAttributes_2 ADD FOREIGN KEY (CarrierKey)
REFERENCES Carriers_2 (CarrierKey) DEFERRABLE;


ALTER TABLE EgressRouteAttributes_1 ADD FOREIGN KEY (CarrierKey)
REFERENCES Carriers_1 (CarrierKey) DEFERRABLE;

ALTER TABLE EgressRouteAttributes_1 ADD FOREIGN KEY (ListKey)
REFERENCES Lists_1    (ListKey)    DEFERRABLE;
```

```
ALTER TABLE EgressRouteAttributes_2 ADD FOREIGN KEY (CarrierKey)
REFERENCES Carriers_2 (CarrierKey) DEFERRABLE;

ALTER TABLE EgressRouteAttributes_2 ADD FOREIGN KEY (ListKey)
REFERENCES Lists_2    (ListKey)    DEFERRABLE;


ALTER TABLE QoS_1 ADD FOREIGN KEY (IngressCarrierKey)REFERENCES
Carriers_1 (CarrierKey) DEFERRABLE;

ALTER TABLE QoS_1 ADD FOREIGN KEY (EgressCarrierKey)REFERENCES
Carriers_1 (CarrierKey) DEFERRABLE;

ALTER TABLE QoS_1 ADD FOREIGN KEY (RouteKey)   REFERENCES Routes_1
(RouteKey)   DEFERRABLE;

ALTER TABLE QoS_2 ADD FOREIGN KEY (IngressCarrierKey)REFERENCES
Carriers_2 (CarrierKey) DEFERRABLE;

ALTER TABLE QoS_2 ADD FOREIGN KEY (EgressCarrierKey)REFERENCES
Carriers_2 (CarrierKey) DEFERRABLE;

ALTER TABLE QoS_2 ADD FOREIGN KEY (RouteKey)   REFERENCES Routes_2
(RouteKey)   DEFERRABLE;


ALTER TABLE IngressRoutes_1 ADD FOREIGN KEY (RouteKey)REFERENCES
Routes_1                (RouteKey)   DEFERRABLE;

ALTER TABLE IngressRoutes_1 ADD FOREIGN KEY (AttributeKey)REFERENCES
IngressRouteAttributes_1 (AttributeKey) DEFERRABLE;

ALTER TABLE IngressRoutes_2 ADD FOREIGN KEY (RouteKey)REFERENCES
Routes_2                (RouteKey)   DEFERRABLE;

ALTER TABLE IngressRoutes_2 ADD FOREIGN KEY (AttributeKey)REFERENCES
IngressRouteAttributes_2 (AttributeKey) DEFERRABLE;


ALTER TABLE EgressRoutes_1 ADD FOREIGN KEY (RouteKey)REFERENCES
Routes_1                (RouteKey)   DEFERRABLE;

ALTER TABLE EgressRoutes_1 ADD FOREIGN KEY (AttributeKey)REFERENCES
EgressRouteAttributes_1 (AttributeKey) DEFERRABLE;

ALTER TABLE EgressRoutes_2 ADD FOREIGN KEY (RouteKey)REFERENCES
Routes_2                (RouteKey)   DEFERRABLE;

ALTER TABLE EgressRoutes_2 ADD FOREIGN KEY (AttributeKey)REFERENCES
EgressRouteAttributes_2 (AttributeKey) DEFERRABLE;

\echo Foreign Keys done...


COMMIT;
```

■ **Dataset Schema**

# Error Conditions and Recoveries

This appendix describes the following error conditions and recoveries:

- Database Connectivity Loss
- Gatekeeper Connectivity Loss

## Database Connectivity Loss

During connectivity loss, CSR attempts to reconnect to the database every minute, and it continues to handle gatekeeper requests with the current dataset in memory. However, the following are not available:

- Tools associated with the database as follows.
  - psql
  - pg_dump
  - createdb
  - dropdb
- GUI
- Additional active dataset not in memory

# Gatekeeper Connectivity Loss

Gatekeeper Connectivity Loss can occur for any of the following reasons:

- Power failure on gatekeeper

- General network problem

- Network problem related to the Sun/Solaris™ system

During connectivity loss, CSR attempts to reconnect to the gatekeepers every minute. You can also use the GUI to attempt a reconnection by clicking **Update GateKeepers** in the CSR Commands tab in the CSR window.

**GLOSSARY**

## A

| | |
|---|---|
| **ANI** | automatic number identification. |
| **ARQ** | admission request message. |

## B

| | |
|---|---|
| **BRQ** | bandwidth request message. |

## D

| | |
|---|---|
| **DNIS** | dialed number identification service. |
| **DNS** | domain naming system. |
| **DRQ** | disengage request message. |

## G

| | |
|---|---|
| **GUI** | graphical user interface. |

## I

| | |
|---|---|
| **irr** | information request response message. |

## L

| | |
|---|---|
| **LCF** | location confirm message. |
| **LRJ** | location reject message. |
| **LRQ** | location request message. |

## R

| | |
|---|---|
| **RAI** | resource availability indication message. |
| **RRQ** | registration request message. |

## U

| | |
|---|---|
| **URQ** | unregistration request message. |

# S

Source carrier **1-2**

# T

Tables

Carrier **1-3**

Configuration Settings **1-3**

Contact List **1-3**

Contacts **1-3**

Egress Route Attributes **1-3**

Egress Routes **1-3**

Filters **1-3**

Gatekeepers **1-3**

Ingress Route Attributes **1-3**

Ingress Routes **1-3**

Match Patterns **1-3**

QoS **1-3**

Routes **1-3**

Rules **1-3**

see also Database Tables **2-4**

TestOutputs **1-3**

TestRoutes **1-3**

# U

URQ **1-2**