# Care Team Sync Configuration Guide

Version 2.5.1

# Notice

Vocera Communications, Inc.
www.vocera.com
tel :: +1 408 882 5100
fax :: +1 408 882 5101

**Last modified:** 2023-02-07 08:40

VCTS-251Rel-Docs build 162

# Contents

# Introduction

This section introduces Vocera Care Team Sync and lists system requirements.

## What is Vocera Care Team Sync?

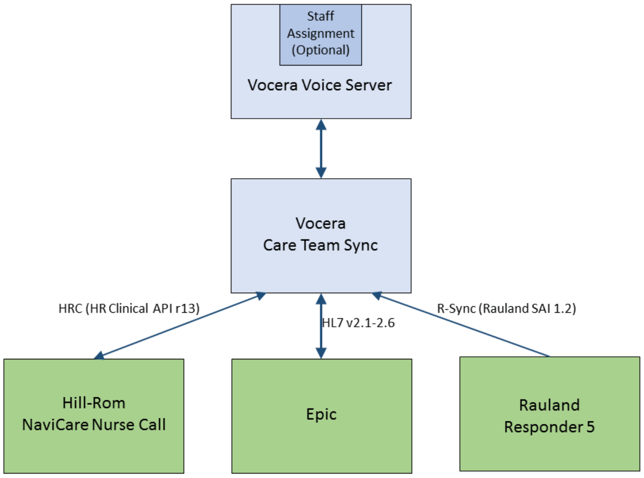
The Vocera Care Team Sync (CTS) application integrates the care team role assignments of the Vocera Voice Server (and optionally the Vocera Staff Assignment application) with a foreign clinical care team assignment system.

CTS allows you to choose a "source of truth" for your assignments—either the assignments in the Vocera Voice Server database or a foreign clinical assignment system—and optionally synchronize the care team assignments of the two systems.

Vocera Care Team Sync supports all of the following foreign clinical assignment systems:

- A system whose connection mechanism uses the Hill-Rom Clinical API r13 (HRC).
  These systems include some versions of Critical Alert and Connexall as well as the Hill-Rom Navicare Nurse Call system itself.
- A system whose connection mechanism uses HL7 v 2.1-2.6.
  These systems include Admit/Discharge/Transfer (ADT) systems such as Epic and others.
- A system whose connection mechanism uses the Rauland Responder 5 R-Sync (SAI 1.2) web service.

The following figure shows the CTS architecture.



## CTS Operation Requirements

The following information provides guidelines for implementing the Vocera Care Team Sync solution.

## System Requirements

These are the installation requirements and prerequisites for the CTS server.

### Hardware:

Following are the minimum hardware requirements for the CTS server.

| Item | Requirement |
|---|---|
| Processor | Intel Xeon 2.60 GHz Quad Core |
| RAM | 4 GB |
| Available Hard Disk Space | 180 GB |

Vocera has tested the above hardware configuration to confirm that CTS provides the following throughput:

| Connection Type | Number of Simultaneous Assignments | Approximate Time in Minutes |
|---|---|---|
| Rauland (Publisher only) | 6000 | 4.5 to 5.0 |
| Hill-Rom (Publisher or subscriber) | 6000 | 5.0 |
| HL7 (Publisher or subscriber) | 6000 | 4.5 to 5.0 |

### Operating System:

- Windows Server 2008 R2 Standard or newer

### Supported Browser Versions:

- Chrome 65.0 or greater
- Internet Explorer 10 or greater
- Mozilla Firefox 51.0 or greater

### Additional Software:

- Vocera Voice Server version 4.4.3 or newer

## VMware Support

Vocera supports running CTS in a virtualized VMware environment.

The following VMware platforms are supported:

- VMware Virtual Infrastructure 3.5 (ESX or ESXi)
- VMware vSphere 4 (ESX or ESXi)
- VMware vSphere 5 (ESXi) or later

Vocera strongly recommends that the VMware virtualization platform includes VMware vCenter Server (formerly VMware Virtual Center) with a Standard license or better.

Vocera recommends the VMware environment conform to the following guidelines:

***Table 1: VMware Recommendations***

| Component | Recommendation |
|---|---|
| Processors per VM | 2 |
| RAM per VM | 4 GB |
| Resource Reservation | • CPU: 500 MHz<br>• RAM: 512 MB |

| Component | Recommendation |
|---|---|
| Network Interface | • VMware vSphere 4: VMXNET 3<br>• VMware ESX 3.5: Enhanced VMXNET (VMXNET 2) |
| Minimum Disk Space | 180 GB |
| Guest Operating System | Windows 2008 R2 Standard or newer |

# Installing and Upgrading CTS

## Setting Up New CTS Integrations

If you are setting up Vocera Care Team Sync for the first time at your site, you need to allow time for creating normalizer and templates, configuring connections, and setting up the entire CTS environment. The actual software installation is fast, but the configuration to make CTS work with other clinical staff assignment systems can be extensive. The topics is this section provide a guide to first-time CTS deployments.

### How to Deploy CTS

This topic summarizes how to install, configure, and fully deploy a Vocera Care Team Sync system in an environment that has never used CTS. It shows you the most efficient way to fully configure the CTS product and begin syncing assignments with foreign clinical staff assignment systems.

To deploy CTS for the first time:

1. Install the software.
   a. Run the CTS software installation program.

      See *How to Install the CTS Software* on page 10.

   b. If necessary, customize the settings in the `vcts.properties` file.

      You likely need to perform this task only if you are installing CTS on a machine with an IP address outside the 10.x.x.x range.

      See *Customizing CTS Properties* on page 13.
2. Perform the initial system set up.
   a. Change the default password of the CTS Console administrator account.

      See *How to Change the Default Password* on page 23.

   b. Register the IP address of the CTS and Vocera Voice Server machines with each other.

      See *Registering the VS and CTS with each other* on page 24.
3. Configure email settings to receive alert messages when a connection fails.

   See *How to Set Up Email* on page 80.
4. Set up the units whose assignments CTS will synchronize.
   a. Create a **Vocera Group Template** for each unit.

      See *Creating Group Templates for Units* on page 96.

   b. Create a **Vocera Group Location Normalizer** for each unit.

      See *Creating Group Normalizers for Units* on page 101.

   c. Create a **Vocera Group Role Normalizer** for each unit.

      See *Creating Group Normalizers for Units* on page 101.

   **d.** Specify the unit definition.

   See *How to Create a Unit* on page 74.

5. Set up a connection to each of the third-party systems with which CTS is syncing assignments.

   **a.** Create the normalizers and maps that you need for each connection

   - For a Rauland connection:
     - Create a **Subscriber Location Normalizer**.
       See *Subscriber Location Normalizer Example* on page 109.
     - Create a **Subscriber Role Normalizer**.
       See *Subscriber Role Normalizer Example* on page 111.
   - For a Hill-Rom connection where CTS is acting as the subscriber:
     - Create a **Subscriber Location Normalizer**.
       See *Subscriber Location Normalizer Example* on page 109.
     - Create a **Subscriber Role Normalizer**.
       See *Subscriber Role Normalizer Example* on page 111.
   - For an HL7 connection:
     - Create an **Assigned Patient Location Normalizer**.
       See *Normalizing and Mapping Workflow* on page 113.
     - Create a **Patient ID Normalizer**.
       See *Normalizing and Mapping Workflow* on page 113.
     - Create a **Prior Patient Location Normalizer**.
       See *Normalizing and Mapping Workflow* on page 113.
     - Create an **ADT Field Map**.
       See *ADT Mapping Methods* on page 116.
     - Create an **ADT Event Map**.
       See *ADT Event Mapping Examples* on page 116.

   **b.** Specify the connection definition.

   See the appropriate topic in the following list:

   - *How to Add a Rauland Connection* on page 28
   - *How to Add a Hill-Rom Connection* on page 32
   - *How to Add an HL7 Connection* on page 43

6. Specify the source of truth for each role.

   See *How to Synchronize Care Team Information* on page 53.

7. Confirm that assignments are updating.

   - Look for logs
   - Look for values in the CTS database
   - Look for assignments in the end point (e.g. Staff Assignment or the VS)

8. If necessary, map any users that CTS fails to map.

   See *How to Add a User Mapping* on page 70.

9. Back up the configuration and the database.

   See *How to Back Up Configuration Data* on page 57 and *How to Back Up the CTS Database* on page 58.

## Installing CTS Software

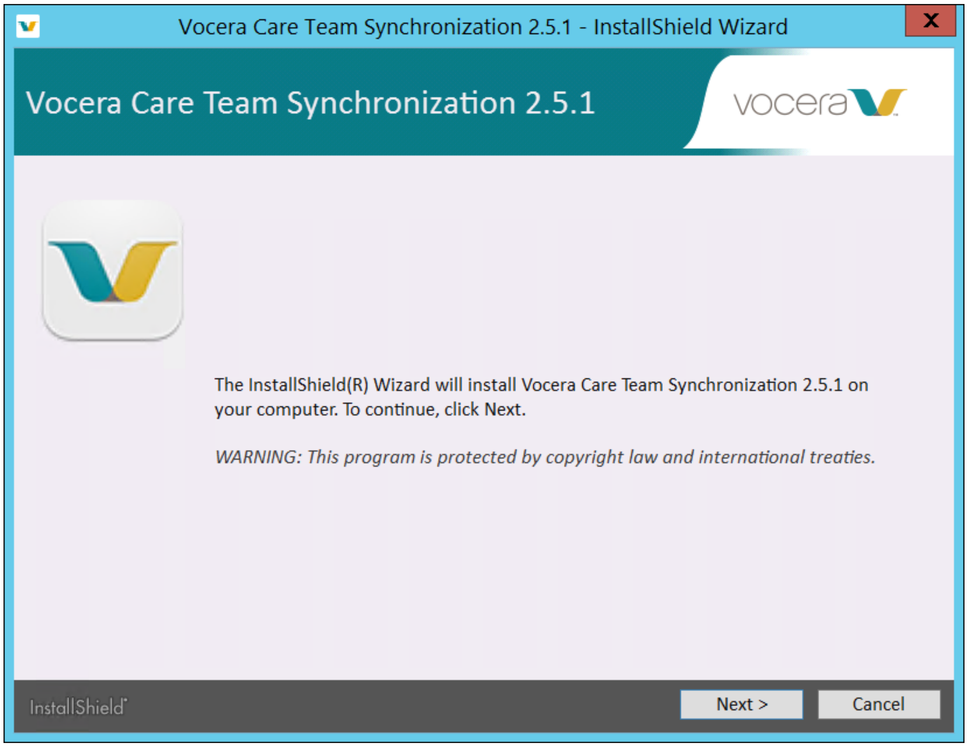
The topics in this section show you how to install the CTS software; they also describe the directories and environment variables that the installation program creates.

### How to Install the CTS Software

Download the CTS Installer and run it to set up the software.

Follow this procedure to install the CTS software.

1. Download the CTS Installer from the link in the Technical Support Knowledge Base article.
2. Log into the computer you intend to use for the CTS application with administrator privileges.
3. Copy the **VoceraCareTeamSync-Installer-<version>.exe** file to the server and double-click it to start the installation.
   The **Welcome** dialog box appears.

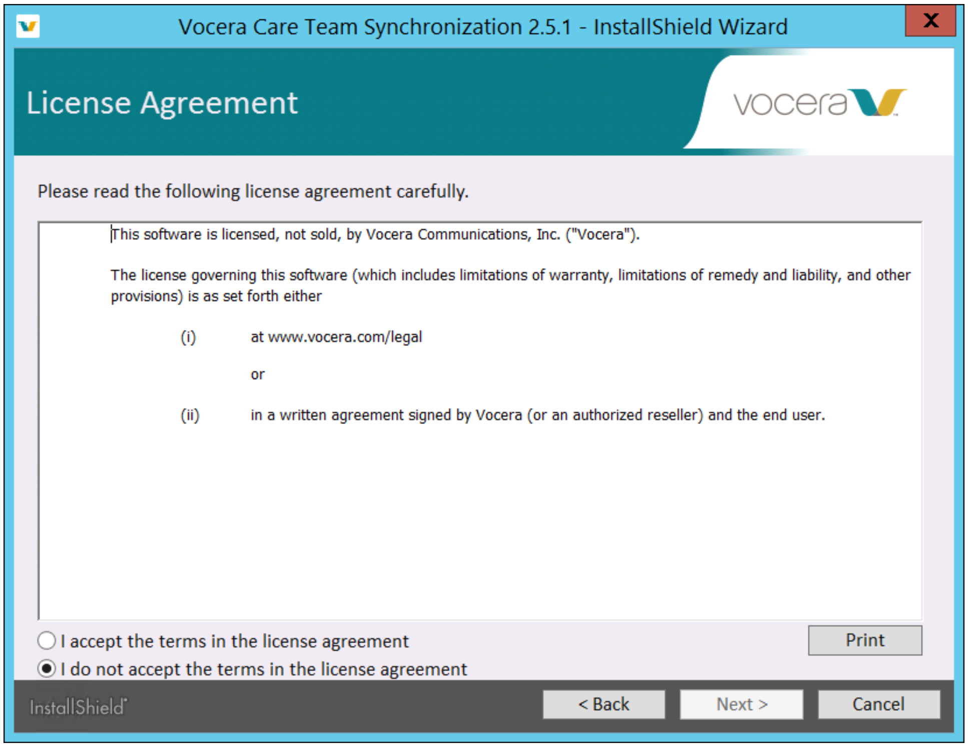


Vocera Care Team Synchronization 2.5.1 - InstallShield Wizard
Vocera Care Team Synchronization 2.5.1
vocera
InstallShield
Next >
Cancel

The InstallShield(R) Wizard will install Vocera Care Team Synchronization 2.5.1 on your computer. To continue, click Next.

*WARNING: This program is protected by copyright law and international treaties.*

4. Click **Next** in the **Welcome** dialog box.
   The **License Agreement** dialog box appears.



Vocera Care Team Synchronization 2.5.1 - InstallShield Wizard

License Agreement

Please read the following license agreement carefully.

This software is licensed, not sold, by Vocera Communications, Inc. ("Vocera").

The license governing this software (which includes limitations of warranty, limitations of remedy and liability, and other provisions) is as set forth either

(i) at www.vocera.com/legal

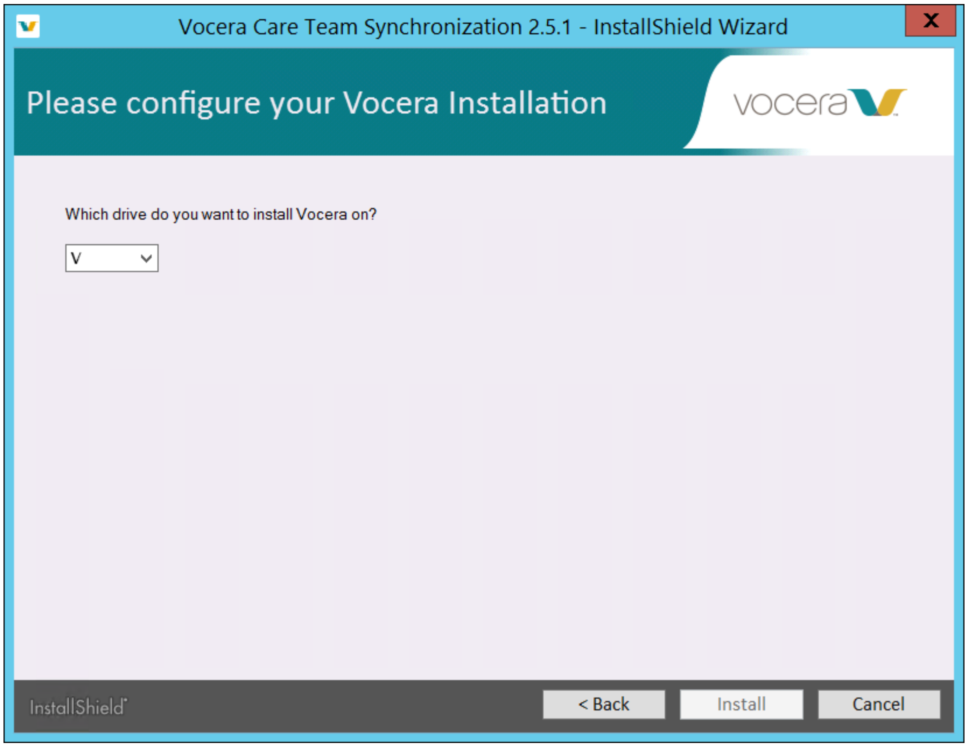
or

(ii) in a written agreement signed by Vocera (or an authorized reseller) and the end user.

○ I accept the terms in the license agreement

● I do not accept the terms in the license agreement

5. In the License Agreement dialog box, select **I accept the terms in the license agreement** and click **Next**.
   The **Configuration** dialog box appears.



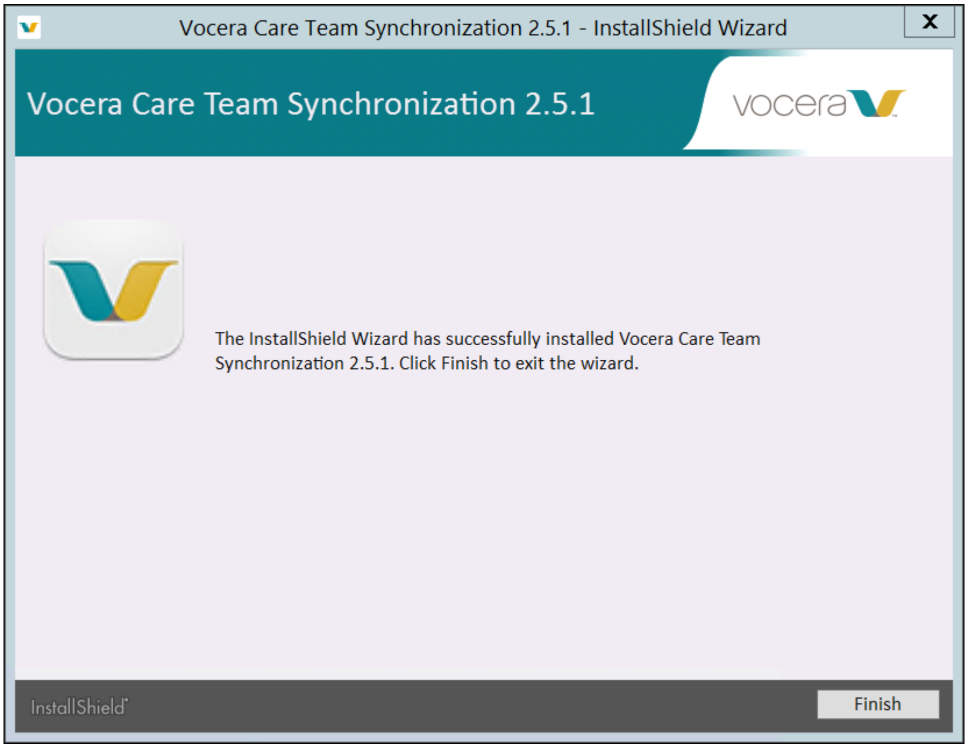
Vocera Care Team Synchronization 2.5.1 - InstallShield Wizard

Please configure your Vocera Installation

Which drive do you want to install Vocera on?

V

6. Specify the installation drive (the default is C) in the **Configuration** dialog box and click **Install**.
   The Installer runs and a progress bar indicates the status. When the installation is complete, the Installer displays the following dialog box:



Vocera Care Team Synchronization 2.5.1 - InstallShield Wizard

Vocera Care Team Synchronization 2.5.1

The InstallShield Wizard has successfully installed Vocera Care Team Synchronization 2.5.1. Click Finish to exit the wizard.

7. Click **Finish**.
   The installer exits and the CTS application shortcuts (CTS Release Notes, Wireshark Installer, and Send Vocera Logs) are visible on the desktop.

The installation is now complete.

## Installation Directories

Review the installation directories created by the CTS software installation.

When you install CTS, the following folders are created:

***Table 2: CTS Install Folders***

| Directory | Description |
|---|---|
| C:\InstallLogs-Vocera | Installation logs |
| *%VOCERA_DRIVE%*\vocera\agent | Vocera Agent (messaging broker) |
| *%VOCERA_DRIVE%*\vocera\jre18 | Java Runtime Environment 1.8.0 |
| *%VOCERA_DRIVE%*\vocera\logs | CTS logs |
| *%VOCERA_DRIVE%*\vocera\myodbc52 | MySQL ODBC Connector |
| *%VOCERA_DRIVE%*\vocera\mysql56 | MySQL database |
| *%VOCERA_DRIVE%*\vocera\mysql-workbench | MySQL Workbench |
| *%VOCERA_DRIVE%*\vocera\support | Vocera Support tools (including AutoSendLogs) |
| *%VOCERA_DRIVE%*\vocera\tools | Wireshark installer |
| *%VOCERA_DRIVE%*\vocera\vcts | CTS application |

## Environment Variables

Review the environment variables created by the CTS software installation.

When you install CTS, the following environment variables are created:

***Table 3: Environment variables***

| Environment variable | Default value |
|---|---|
| JAVA_HOME | *%VOCERA_DRIVE%*\vocera\jre18 |
| MYSQL_HOME | *%VOCERA_DRIVE%*\vocera\mysql56 |
| VOCERA_DRIVE | C: (By default) |
| VOCERA_HOME | *%VOCERA_DRIVE%*\VOCERA |
| VOCERA_RTSLOG | file |
| VOCERA_VCTS_SERVICE_PARAMETERS | -JVM-Xms256m<br>-JVM-Xmx1g<br>-JVM-Dcom.sun.management.jmxremote.port=9998<br>-JVM-Dcom.sun.management.jmxremote.ssl=false<br>-JVM-Dcom.sun.management.jmxremote.authenticate=false<br>-JVM-XX:+PrintGCDetails<br>-JVM-XX:+PrintGCTimeStamps<br>-JVM-XX:+PrintGCDateStamps<br>-JVM-verbose:gc<br>-JVM-Xloggc:V:/vocera/logs/vcts-gc.log<br>-JVM-XX:+UseGCLogFileRotation<br>-JVM-XX:NumberOfGCLogFiles=10<br>-JVM-XX:GCLogFileSize=10M<br>-JVM-XX:MaxPermSize=128m<br>-JVM-XX:+UseG1GC<br>-JVM-XX:InitiatingHeapOccupancyPercent=40<br>-JVM-XX:G1ReservePercent=15<br>-JVM-XX:-DisableExplicitGCwindir=C:\Windows |

## Customizing CTS Properties

The CTS application uses a properties file named `\vocera\vcts\resources\vcts.properties` to configure the service at start up. For most installations, you do not need to customize the default settings in this file.

## Customizing HL7 Logging Properties

The default logging configuration values are sufficient for most installations of CTS. However, you can customize them to accommodate special situations.

To customize the HL7 logging properties:

1. Open `vocera\vcts\resources\vcts.properties` in a text editor.
2. Modify any of the following properties and remove the # comment, if necessary.

***Table 4: HL7 logging properties***

| Property | Default Value | Description |
|---|---|---|
| hl7.log.encrypted | true | Specifies whether to encrypt the HL7 data. |
| hl7.log.obfuscate | false | Specifies whether to obfuscate HL7 values. If you set this to **true**, manually create the `hl7.log.obfuscate.val` property and specify which HL7 values to obfuscate. Contact Vocera Technical Support if you need to obfuscate log data. |
| hl7.log.xml | false | Whether to save the log in XML format rather than HL7. |

| Property | Default Value | Description |
|---|---|---|
| hl7.log.lines | 10000 | Maximum number of lines in each log file.<br>**Note:** If you receive a large number of HL7 messages each day, you can increase the value of hl7.log.lines or hl7.log.files to accommodate the higher volume. |
| hl7.log.files | 5 | Number of log files to save before moving them to the upload folder. |
| hl7.log.upload.enable | true | Whether to move log files to the upload folder once the hl7.log.files number has been reached. |
| hl7.log.upload.folder | %VOCERA_DRIVE% \vocera\support\upload | The folder that the AutoSendLogs utility uses to send log files to Vocera. |

3. Save the file.
4. Restart the Vocera Care Team Sync service.

## Customizing VSI Properties

Modify the VSI properties only under direction from Vocera Technical Support.

1. Open `vocera\vcts\resources\vcts.properties` in a text editor.
2. Modify any of the following properties and remove the # comment, if necessary.

***Table 5: VSI (Vocera Services Interface) Properties***

| Property | Default Value | Description |
|---|---|---|
| vsi.enable | N/A | This property is deprecated; in previous versions of CTS it controlled the logging of CTS-specific events from the Vocera Voice Server. VSI logging is now managed by the following entry in the `\vocera\vcts\resources\logback.xml` file:<br>`<logger name="com.vocera.vsi" level="ERROR"/>` |
| vsi.property.vs.server.addres | N/A | This property is deprecated; in previous versions of CTS it specified the IP address of the Vocera Voice Server machines. This property is now managed by the **Vocera Server** section in the CTS. |
| vsi.property.vs.server.port | 8080 | Change the value of this property only by direction of Vocera Technical Support. |
| vsi.property.event.service.clie | ENTITY | Change the value of this property only by direction of Vocera Technical Support. |
| vsi.property.event.service.clie | VCTS | Change the value of this property only by direction of Vocera Technical Support. |
| vsi.property.event.service.clie | 172.25.10.100 | By default, CTS assumes IP addresses are in the 10.x.x.x range. If the IP address of your CTS machine is outside this range, you must uncomment this property and specify your IP address as its value. |
| vsi.property.event.service.reg | enableUserEvents,enableSys | Change the value of this property only by direction of Vocera Technical Support. |
| vsi.property.event.service.coı | 0 | Change the value of this property only by direction of Vocera Technical Support. |

3. Save the file.
4. Restart the Vocera Care Team Sync service.

## Upgrading Existing CTS Integrations

CTS supports a direct upgrade from version 2.3.0 and 2.4.0 without requiring an uninstall. If you need to upgrade from version 2.2.0, you must uninstall, re-install, and then manually run SQL INSERT statements to migrate data from the earlier version to the new database.

### How to Upgrade from CTS 2.3.0 or Greater

Upgrade the CTS software and make sure a facility mapping exists in HL7 patient normalizers.

To upgrade from CTS Version 2.3.0 or greater:

1. In the CTS Console, navigate to the **System Backup** page and back up both the configuration data and the database.

   See *How to Back Up Configuration Data* on page 57 and *How to Back Up the CTS Database* on page 58.
2. Make a copy of the backup files in a safe location so you can use them to roll back, if necessary.
3. Install the latest version of the CTS software.

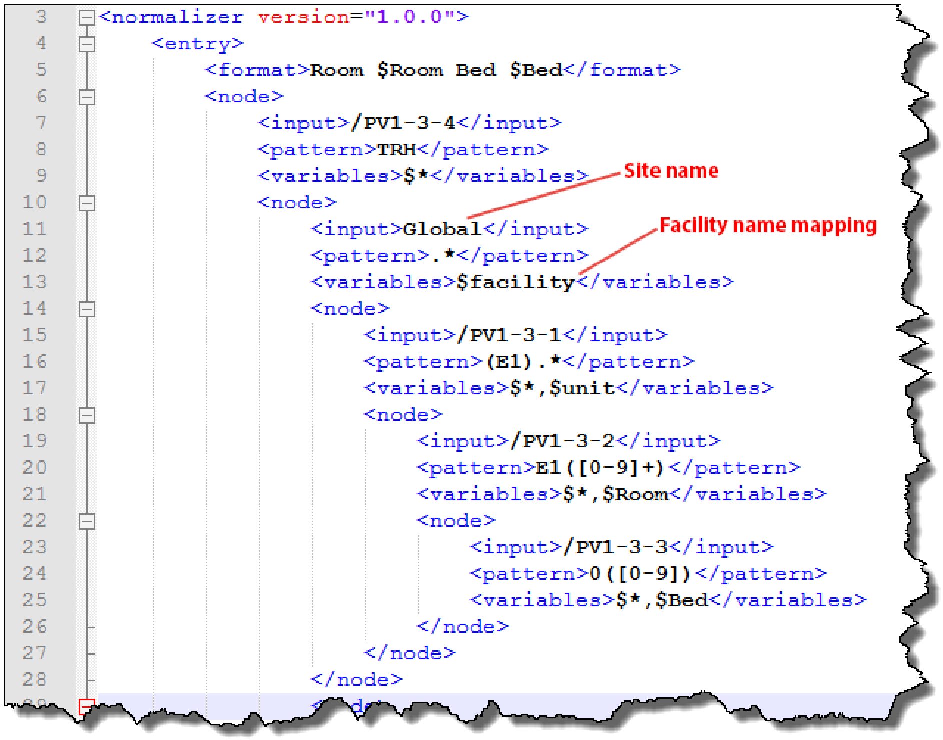
   See *How to Install the CTS Software* on page 10.

   **Important:** Do not uninstall the existing software. Install Version 2.5.1 directly over Version 2.4.0 or Version 2.3.0.
4. Enter your VAI credentials in the **Update Vocera Server Configuration** dialog box.

   In earlier versions of CTS, you provided credentials in the `vai.username` and `vai.password` properties of the `vcts.properties` file. These properties are now deprecated, and you must provide the credentials in the UI. If you do not provide credentials, CTS will not be able to connect to the Vocera Analytics.

   See *How to Register the Network Address of the VS with the CTS* on page 24.
5. If necessary, create a facility name (`$facility`) mapping in the files that you use for the **Assigned Patient Location** and **Prior Patient Location** normalizers in the HL7 connection.

   Open each normalizer in an editor and confirm that the mapping exists. If it does not exist, create it within the <node> definition for each site.



```
<normalizer version="1.0.0">
    <entry>
        <format>Room $Room Bed $Bed</format>
        <node>
            <input>/PV1-3-4</input>
            <pattern>TRH</pattern>
            <variables>$*</variables>
            <node>
                <input>Global</input>
                <pattern>.*</pattern>
                <variables>$facility</variables>
                <node>
                    <input>/PV1-3-1</input>
                    <pattern>(E1).*</pattern>
                    <variables>$*,$unit</variables>
                    <node>
                        <input>/PV1-3-2</input>
                        <pattern>E1([0-9]+)</pattern>
                        <variables>$*,$Room</variables>
                        <node>
                            <input>/PV1-3-3</input>
                            <pattern>0([0-9])</pattern>
                            <variables>$*,$Bed</variables>
                        </node>
                    </node>
                </node>
```

The most recent version of CTS requires a `$facility` mapping. Earlier versions of CTS do not require this mapping, but some deployments already include it in the normalizers.

6. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.
7. Open the CTS Console in a browser and navigate from page-to-page, confirming that your upgraded data has migrated successfully.

The Vocera Care Team Sync system is successfully upgraded.

**Note:** If you are unable to log into the CTS Console after upgrading, delete the cache in your browser and retry.

## How to Roll Back an Upgrade

You can manually roll back an upgrade if necessary for any reason.

To roll back an upgrade:

1. Uninstall the current version of the CTS software.

   See *How to Uninstall CTS* on page 17.
2. Install the version of the CTS software that you were originally using.

   See *How to Install the CTS Software* on page 10.
3. Copy your original configuration and database backup files from the archive that you created to the `\vocera\vcts\backup` directory.
4. Restore both the configuration and the database backup.

See *How to Restore Configuration Data* on page 59 and *How to Restore the CTS Database* on page 61.

5. Open the CTS Console in a browser and navigate from page-to-page, confirming that your upgraded data has migrated successfully.

The CTS system is successfully rolled back to its previous state.

## How to Uninstall CTS

It is not necessary to uninstall the CTS application to upgrade from an earlier version. If you are uninstalling for any other reason, make sure you back up the configuration data and the database before you uninstall the CTS application. The uninstallation process removes everything from your system.

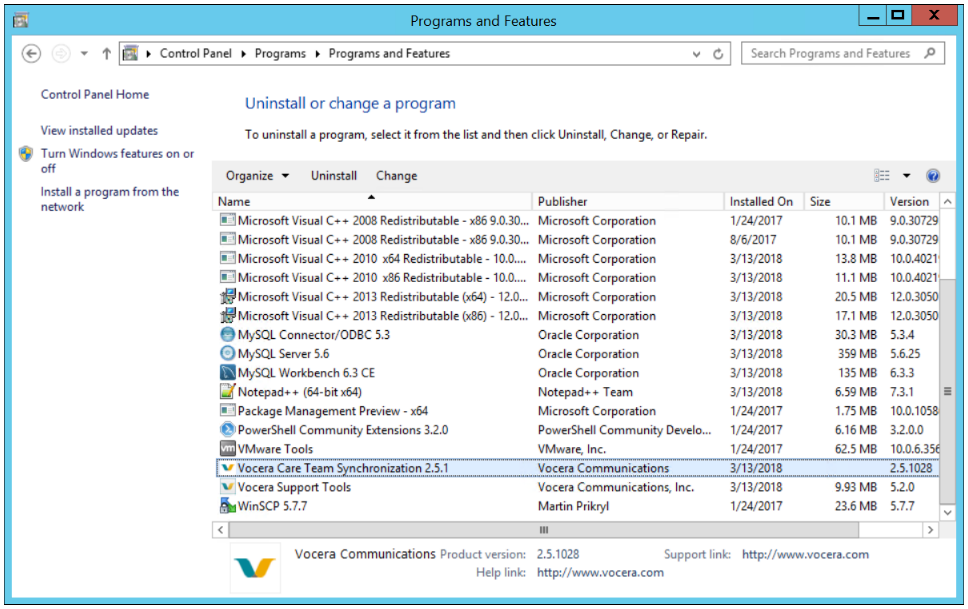
Use the following steps to uninstall the CTS application.

1. In the CTS Console, navigate to the **System Backup** page and back up both the configuration data and the database.

   See *How to Back Up Configuration Data* on page 57 and *How to Back Up the CTS Database* on page 58.

2. In the Windows Programs Control Panel, choose **Uninstall a Program**.

   The **Programs and Features** window opens.


Programs and Features
Control Panel ▸ Programs ▸ Programs and Features
Search Programs and Features
Control Panel Home
View installed updates
Turn Windows features on or off
Install a program from the network
Organize ▾
Uninstall
Change

Uninstall or change a program

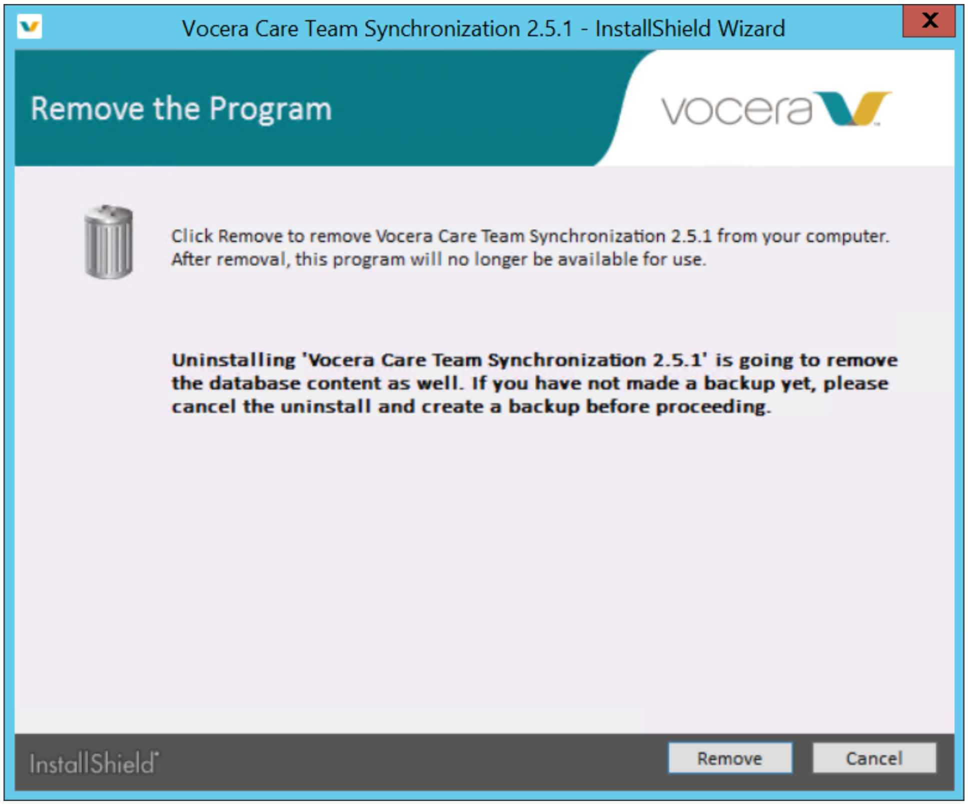
To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

| Name | Publisher | Installed On | Size | Version |
|---|---|---|---|---|
| Microsoft Visual C++ 2008 Redistributable - x86 9.0.30... | Microsoft Corporation | 1/24/2017 | 10.1 MB | 9.0.30729 |
| Microsoft Visual C++ 2008 Redistributable - x86 9.0.30... | Microsoft Corporation | 8/6/2017 | 10.1 MB | 9.0.30729 |
| Microsoft Visual C++ 2010 x64 Redistributable - 10.0.... | Microsoft Corporation | 3/13/2018 | 13.8 MB | 10.0.4021 |
| Microsoft Visual C++ 2010 x86 Redistributable - 10.0.... | Microsoft Corporation | 3/13/2018 | 11.1 MB | 10.0.4021 |
| Microsoft Visual C++ 2013 Redistributable (x64) - 12.0... | Microsoft Corporation | 3/13/2018 | 20.5 MB | 12.0.3050 |
| Microsoft Visual C++ 2013 Redistributable (x86) - 12.0... | Microsoft Corporation | 3/13/2018 | 17.1 MB | 12.0.3050 |
| MySQL Connector/ODBC 5.3 | Oracle Corporation | 3/13/2018 | 30.3 MB | 5.3.4 |
| MySQL Server 5.6 | Oracle Corporation | 3/13/2018 | 359 MB | 5.6.25 |
| MySQL Workbench 6.3 CE | Oracle Corporation | 3/13/2018 | 135 MB | 6.3.3 |
| Notepad++ (64-bit x64) | Notepad++ Team | 3/13/2018 | 6.59 MB | 7.3.1 |
| Package Management Preview - x64 | Microsoft Corporation | 1/24/2017 | 1.75 MB | 10.0.1058 |
| PowerShell Community Extensions 3.2.0 | PowerShell Community Develo... | 1/24/2017 | 6.16 MB | 3.2.0.0 |
| VMware Tools | VMware, Inc. | 1/24/2017 | 62.5 MB | 10.0.6.356 |
| Vocera Care Team Synchronization 2.5.1 | Vocera Communications | 3/13/2018 | | 2.5.1028 |
| Vocera Support Tools | Vocera Communications, Inc. | 3/13/2018 | 9.93 MB | 5.2.0 |
| WinSCP 5.7.7 | Martin Prikryl | 1/24/2017 | 23.6 MB | 5.7.7 |

Vocera Communications Product version: 2.5.1028 Support link: http://www.vocera.com
Help link: http://www.vocera.com

3. Right-click **Vocera Care Team Synchronization <version>** and select **Uninstall**.

   The **Vocera Care Team Synchronization InstallShield Wizard** appears.



Vocera Care Team Synchronization 2.5.1 - InstallShield Wizard

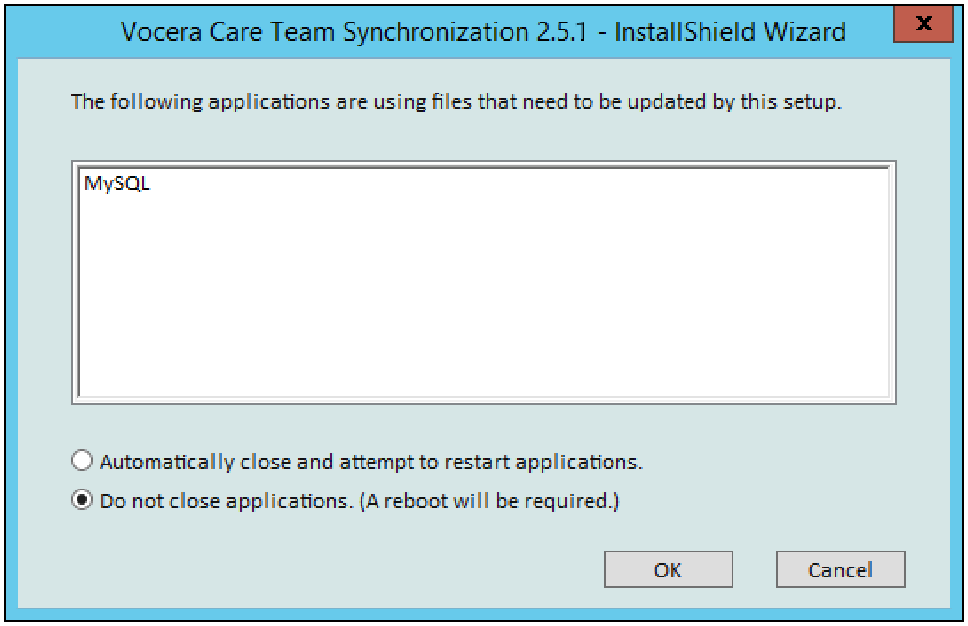
Remove the Program

Click Remove to remove Vocera Care Team Synchronization 2.5.1 from your computer. After removal, this program will no longer be available for use.

**Uninstalling 'Vocera Care Team Synchronization 2.5.1' is going to remove the database content as well. If you have not made a backup yet, please cancel the uninstall and create a backup before proceeding.**

4. Click **Remove** to uninstall the application.

   The uninstall proceeds and a progress bar indicates the status. If the MySQL database was in use, the following dialog box asks you how to proceed.



Vocera Care Team Synchronization 2.5.1 - InstallShield Wizard

The following applications are using files that need to be updated by this setup.

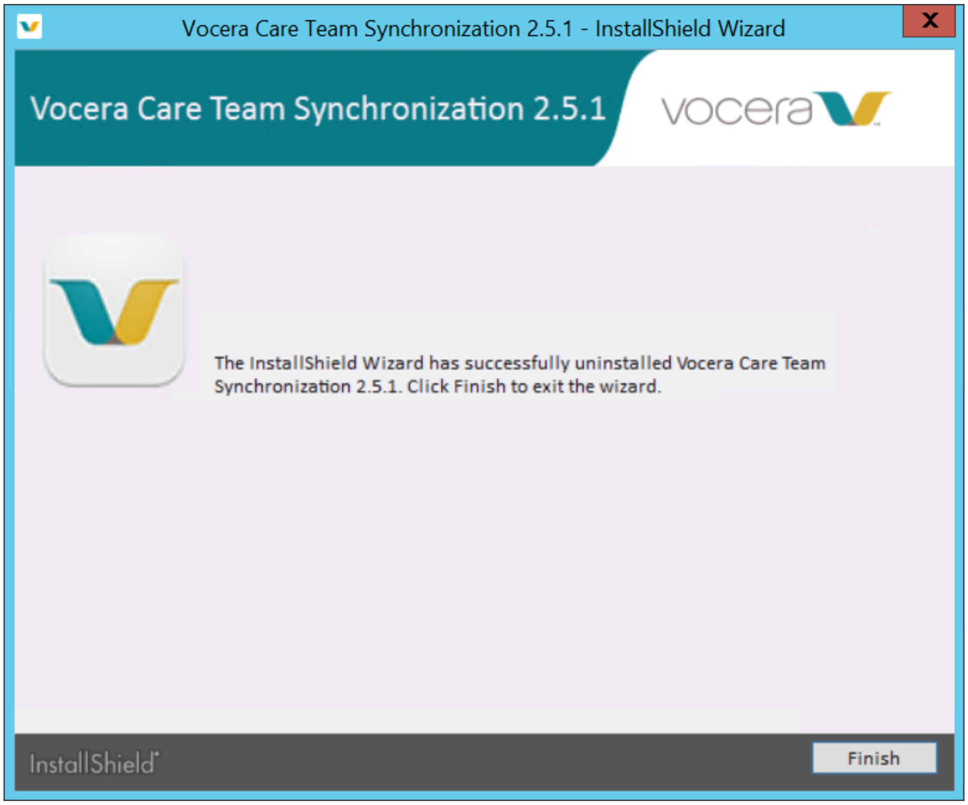
MySQL

○ Automatically close and attempt to restart applications.

● Do not close applications. (A reboot will be required.)

5. Check **Automatically close and attempt to restart applications**, and then click **OK**.

   The uninstall proceeds and a progress bar indicates the status. When the uninstallation is complete, the Installer displays the following dialog box:


Vocera Care Team Synchronization 2.5.1 - InstallShield Wizard
Vocera Care Team Synchronization 2.5.1
vocera
InstallShield
Finish

The InstallShield Wizard has successfully uninstalled Vocera Care Team Synchronization 2.5.1. Click Finish to exit the wizard.

6. Click **Finish**.

The software is removed.

# Performing Initial Setup

After you have installed the Care Team Sync software, you are ready to perform the tasks in the initial system setup. These tasks include logging into the system, changing your password, and registering the CTS and VS with each other.

## Getting Started

Open the CTS Console, log in, and change the default password for the console.

### Opening the CTS Console

Follow these steps to launch the CTS Console and log in.

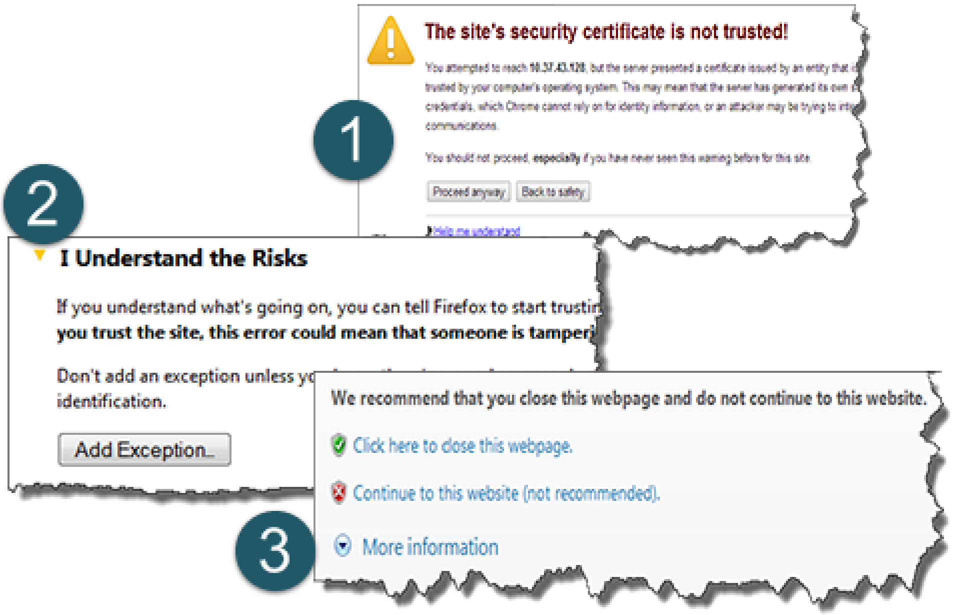
To log into the CTS Console:

1. Open an Internet Explorer browser window.
2. Enter the CTS Console URL in the address bar using the following format:

   `https://`*CTS_IP_Address*

   For example, if the IP address of the CTS machine is 10.98.2.300, enter the following URL into the address bar:

   `https://10.98.2.300`
3. For this release the browser opens a page indicating the site certificate is untrusted. Depending on your browser, choose to accept the security certificate as follows:
   - For Chrome, select **Proceed Anyway**
   - For Firefox, select **I Understand the Risks**, to reveal and select the **Add Exception** button.
   - For Internet Explorer, select **Continue to this website**



**The site's security certificate is not trusted!**

You attempted to reach **10.37.43.128**, but the server presented a certificate issued by an entity that i
trusted by your computer's operating system. This may mean that the server has generated its own s
credentials, which Chrome cannot rely on for identity information, or an attacker may be trying to int
communications.

You should not proceed, **especially** if you have never seen this warning before for this site.

Proceed anyway | Back to safety

Help me understand

**I Understand the Risks**

If you understand what's going on, you can tell Firefox to start trustin
**you trust the site, this error could mean that someone is tamperi**

Don't add an exception unless yo
identification.

Add Exception...

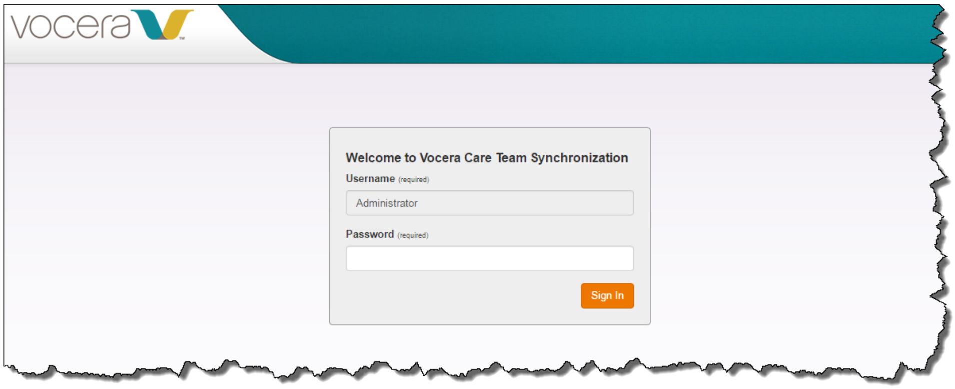
We recommend that you close this webpage and do not continue to this website.

Click here to close this webpage.

Continue to this website (not recommended).

More information

The CTS **Login** screen appears.



Welcome to Vocera Care Team Synchronization

Username (required)

Administrator

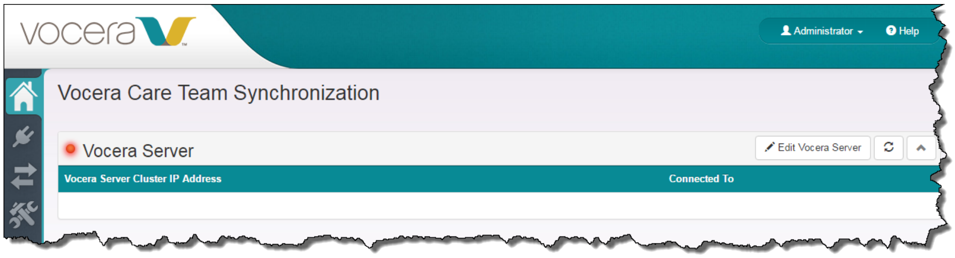
Password (required)

Sign In

**4.** Enter the following values and choose **Log In**:

| Field | Value |
|---|---|
| Username | Administrator |
| Password | admin |

The CTS Console **Home** screen appears.



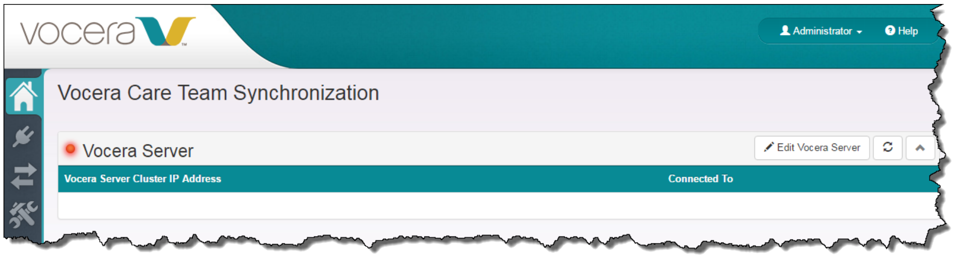
Vocera Care Team Synchronization

Vocera Server

| Vocera Server Cluster IP Address | Connected To |
|---|---|

## A First Look at the CTS Console

The **Home** screen of the CTS Console displays basic status information about your connection and a navigation bar on the left.



Click any icon on the navigation bar to display a different screen in the console:

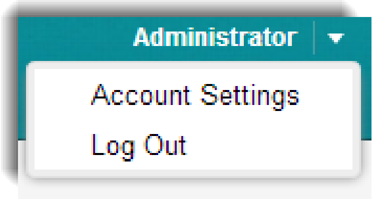
- CTS Home ( ⌂ )
  The **Home** screen displays basic status information and allows you to establish a connection to the Vocera Voice Server.
- Connections ( 🔌 )
  The **Connections** screen allows you to configure connectors between the Vocera Voice Server and each clinical system.
- Sync ( ⇄ )
  The **Sync** screen allows you sync care team assignments between your Vocera Voice Server and each clinical system.
- Tools ( 🛠 )
  The **Tools** screen lets you back up and restore both configuration data and also the CTS database. It also allows you to perform other basic administrative tasks.

## How to Log Out of the CTS Console

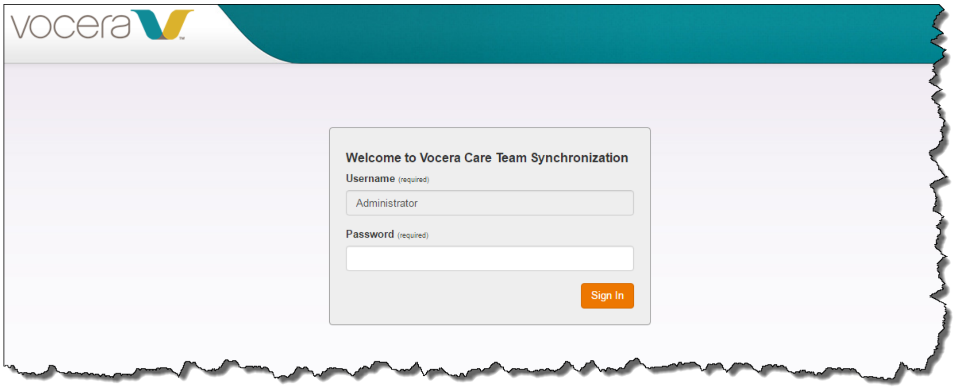
Log out of the CTS Console when you are not using it to prevent unauthorized access.

To log out of the CTS Console:

1. In the CTS Console, click the down arrow ( ▾ ) next to the name of the logged in user (by default, "Administrator") at the top right of the screen.



2. Choose **Log Out**.
   The CTS Console closes and the CTS **Login** screen appears.



Welcome to Vocera Care Team Synchronization

Username (required)

Administrator

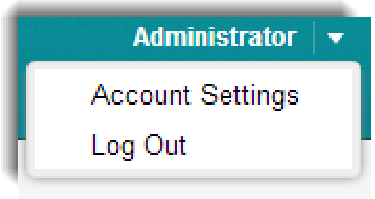
Password (required)

## How to Change the Default Password

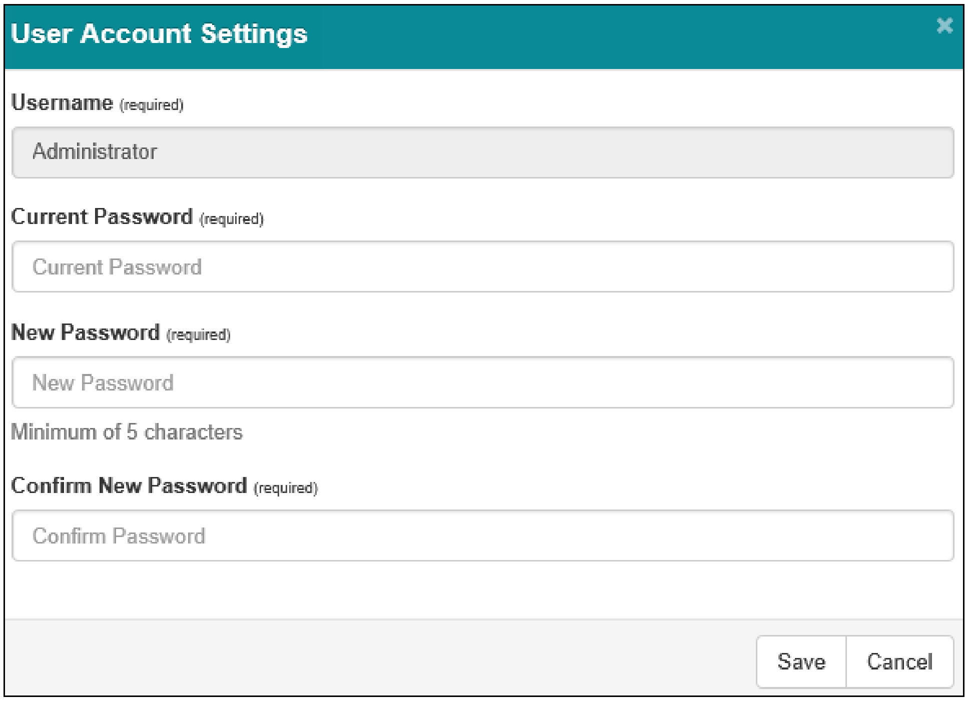
For additional security, you may want to change the default password of the Administrator account.

To change the default password that provides access to the CTS Console:

1. In the CTS Console, click the **Administrator** button at the top right of the screen.



2. Choose **Account Settings**.
   The **User Account Settings** dialog box appears.



User Account Settings

Username (required)

Administrator

Current Password (required)

Current Password

New Password (required)

New Password

Minimum of 5 characters

Confirm New Password (required)

Confirm Password

3. In the **Current Password** field, enter the password you used to access the CTS Console:
4. In the **New Password** field, enter your new password using any combination of alphabetical, numeric, and special characters.

The new password must have a minimum of five characters.

5. In the **Confirm New Password** field, type your new password again.
6. Click **Save**.

   The **User Account Settings** dialog box closes and the CTS Console displays a message stating that your account settings have been updated.

## Registering the VS and CTS with each other

You need to mutually identify the Vocera Voice Server and the Vocera Care Team Sync server by registering the network address of each system with the other system.

### About Mutual Registration

You must register the network addresses of the Vocera Care Team Sync machine and the Vocera Voice Server with each other to establish a connection properly.

- On the **System|License Info** page of the Vocera Voice Server Administration Console, enter the IP address of the CTS server in the **VAI Application IP Addresses** field.

  When you register IP addresses in this manner, the Vocera Voice Server rejects VAI connections from any server not in the **VAI Application IP Addresses** list.
- In the CTS Console, enter the IP addresses or the fully qualified domain names of the Vocera Voice Server cluster nodes (or the standalone Vocera Voice Server IP address) in the **Vocera Server Cluster IP Addresses** field in the **Update Vocera Server Configuration** dialog box.

  When you use this field to register the Vocera Voice Server, you identify the network location of the VS to the CTS application.

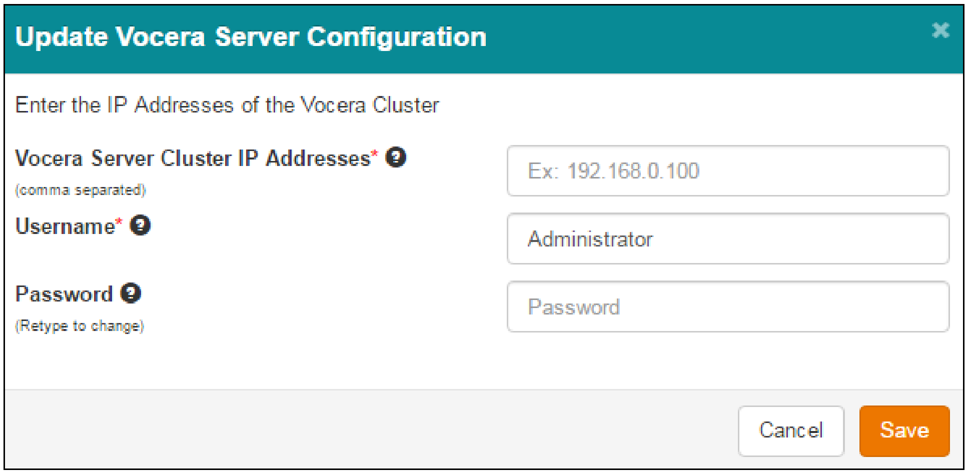
### How to Register the Network Address of the VS with the CTS

The **Vocera Server** section of the CTS home screen displays a red icon to indicate that you need to provide the network address of the Vocera Voice Server.

Use the following steps to register the network address of the Vocera Voice Server with the Vocera Care Team Sync server.

1. From the CTS home screen, click the **Edit Vocera Server** button in the **Vocera Server** section.

   The **Update Vocera Server Configuration** dialog box appears.



Update Vocera Server Configuration

Enter the IP Addresses of the Vocera Cluster

Vocera Server Cluster IP Addresses* (comma separated): Ex: 192.168.0.100

Username*: Administrator

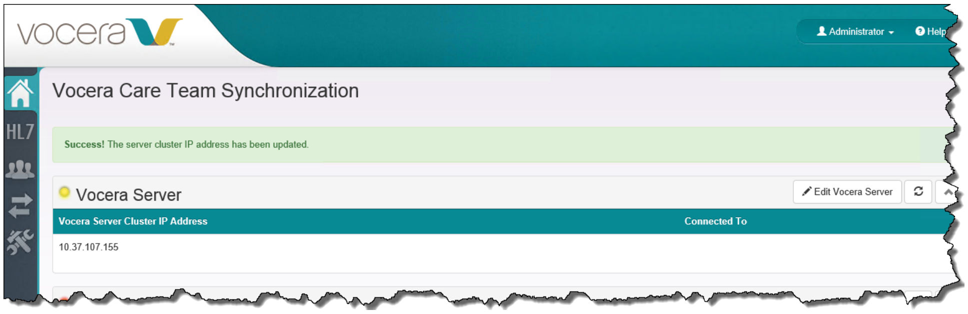
Password (Retype to change): Password

2. Enter the comma-separated list of network addresses for the Vocera Voice Server cluster nodes in the **Vocera Server Cluster IP Addresses** field. If you do not have a cluster, enter the network address of your standalone Vocera Voice Server.

Use either numeric IP addresses or the fully qualified domain names of the machines. You may also use a mix of IP addresses and fully qualified domain names for Vocera Voice Server cluster nodes.

3. In the **Username** field, enter the user name that you provide to establish a VAI connection to the Vocera Voice Server. You must provide credentials that have full administrative permissions to the Vocera Voice Server Administration Console.
4. In the **Password** field, enter the password that you provide to establish a VAI connection to the Vocera Voice Server. You must provide credentials that have full administrative permissions to the Vocera Voice Server Administration Console.
5. Click **Save**.

   The CTS displays the message, "Success! The server cluster IP address has been updated." when it is complete. In addition, the **Vocera Server** section of the CTS home screen displays the network address of the VS as well as a yellow icon to indicate that you need to register the CTS with the Vocera Voice Server.



Vocera Care Team Synchronization

**Success!** The server cluster IP address has been updated.

Vocera Server

| Vocera Server Cluster IP Address | Connected To |
|---|---|
| 10.37.107.155 | |

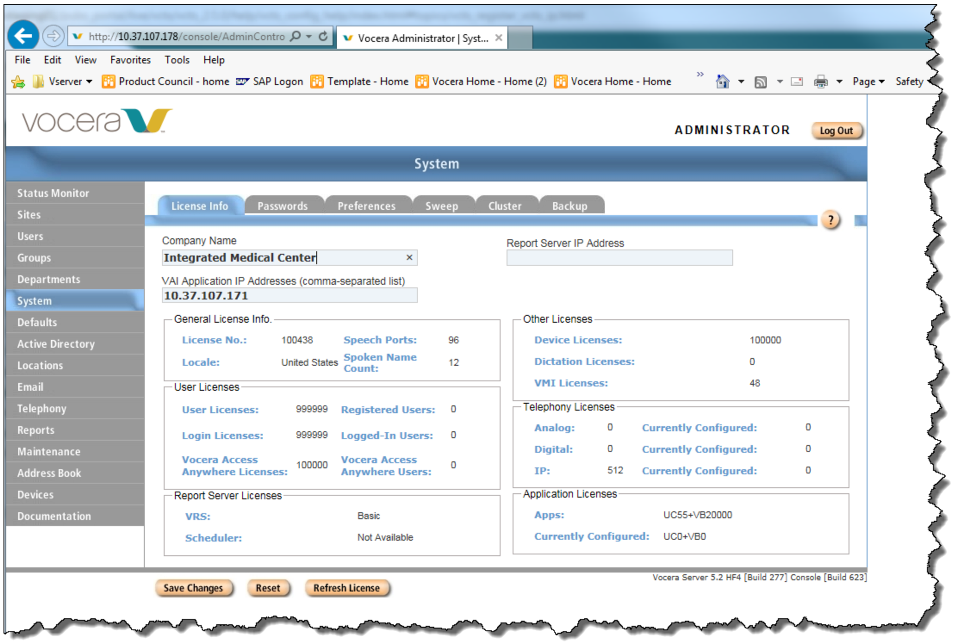
6. Click the close icon in the top right of the message banner to dismiss the message.

## How to Register the IP Address of the CTS with the VS

Log into the Vocera Voice Server Administration Console to register the IP address of the CTS server.

Use the following steps to register the IP address of the Vocera Care Team Sync server with the Vocera Voice Server.

1. From the Vocera Voice Server Administration Console, navigate to the **System|License Info** page.



Company Name: Integrated Medical Center

Report Server IP Address:

VAI Application IP Addresses (comma-separated list): 10.37.107.171

General License Info.

| | | | |
|---|---|---|---|
| License No.: | 100438 | Speech Ports: | 96 |
| Locale: | United States | Spoken Name Count: | 12 |

User Licenses

| | | | |
|---|---|---|---|
| User Licenses: | 999999 | Registered Users: | 0 |
| Login Licenses: | 999999 | Logged-In Users: | 0 |
| Vocera Access Anywhere Licenses: | 100000 | Vocera Access Anywhere Users: | 0 |

Report Server Licenses

| | |
|---|---|
| VRS: | Basic |
| Scheduler: | Not Available |

Other Licenses

| | |
|---|---|
| Device Licenses: | 100000 |
| Dictation Licenses: | 0 |
| VMI Licenses: | 48 |

Telephony Licenses

| | | | |
|---|---|---|---|
| Analog: | 0 | Currently Configured: | 0 |
| Digital: | 0 | Currently Configured: | 0 |
| IP: | 512 | Currently Configured: | 0 |

Application Licenses

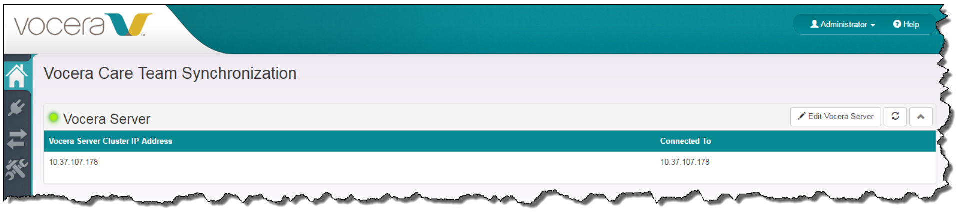
| | |
|---|---|
| Apps: | UC55+VB20000 |
| Currently Configured: | UC0+VB0 |

Vocera Server 5.2 HF4 [Build 277] Console [Build 623]

2. Enter the IP address of the CTS machine in the **VAI Application IP Addresses** field, and then click **Save Changes**.

   The Vocera Voice Server saves the IP address in its database.

When the CTS home screen refreshes, the **Vocera Server** section displays a green icon to indicate that you have completed the mutual registration of the CTS and the VS.



Vocera Care Team Synchronization

🟢 Vocera Server

| Vocera Server Cluster IP Address | Connected To |
|---|---|
| 10.37.107.178 | 10.37.107.178 |

# Adding and Updating Connections

*Connections* maintain the parameters that allow Vocera Care Team Sync and foreign end points to communicate. When you add these connections, CTS stores the communication parameters that you specify in its database. When you update the connections, you edit these parameters to accommodate any necessary changes.

The **All Connections** section displays a brief status of all the connections you have currently created. The Vocera Voice Server connection appears on this page with the **Connection Name** "Vocera" after you perform the mutual registration. You cannot edit the VS connection on this page; use the **Edit Vocera Server** button on the Vocera Care Team Sync Home page to update this connection.

| Field | Description |
|---|---|
| Connection Name | Displays the name of the connection that is specified in the **Add Connection** dialog box. A green icon indicates that the connection is enabled; a red icon indicates that it is disabled. |
| Connection Type | Displays one of the following connection types: HillromAdapter, RaulandAdapter, or HL7. |
| IP Address | Displays the URI of the connection's end point. |
| Last Processed Time | Displays the date and time that the most recent data was received from an inbound connection with the associated system, regardless of whether the sending system is the source of truth. |

Use the **Connections** page to perform any of the following actions:

- Click **Add Connection** and choose a connection type from the menu to create a new connection.
- Click the **Refresh** ( ⟳ ) icon to refresh the display of the **Connections** page.
- Click the **Disable** button inline with the connection name to update it. VS stops processing data, and making assignments, and the status icon turns red.
- Click the **Enable** button inline with the connection name to update it. VS begins processing data, and making assignments, and the status icon turns green. Connections are enabled by default after you add or update them.
- Click the **Edit** button inline with a connection name to update the connection.

**Important:** You must restart the Vocera CTS service after making any changes to connections to force the updates to take effect. CTS reads connections parameters into memory at start up only.

## The Rauland Connection

The Rauland connection allows Vocera Care Team Sync to connect to a Rauland Responder 5 system that implements the R-Sync (SAI 1.2) web service.

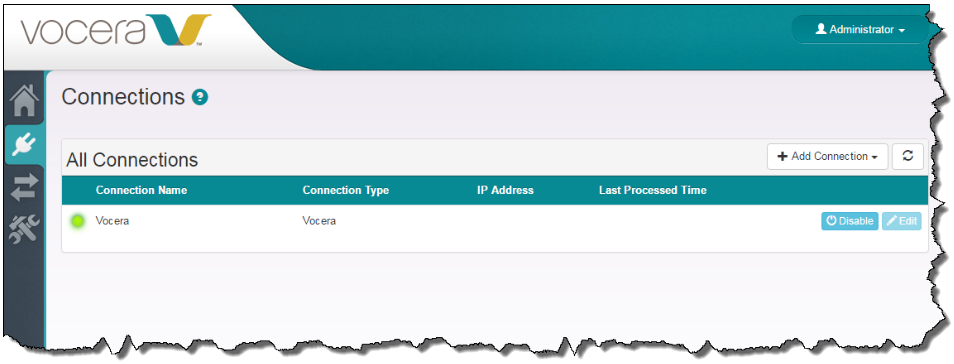
The Rauland system is always the source of truth for this connection; it publishes its assignments to the CTS system.

### How to Add a Rauland Connection

Specify the location of any normalizers required to process Rauland data correctly, along with the additional parameters that allow the end point on the far end to communicate with the Vocera Care Team Sync server.

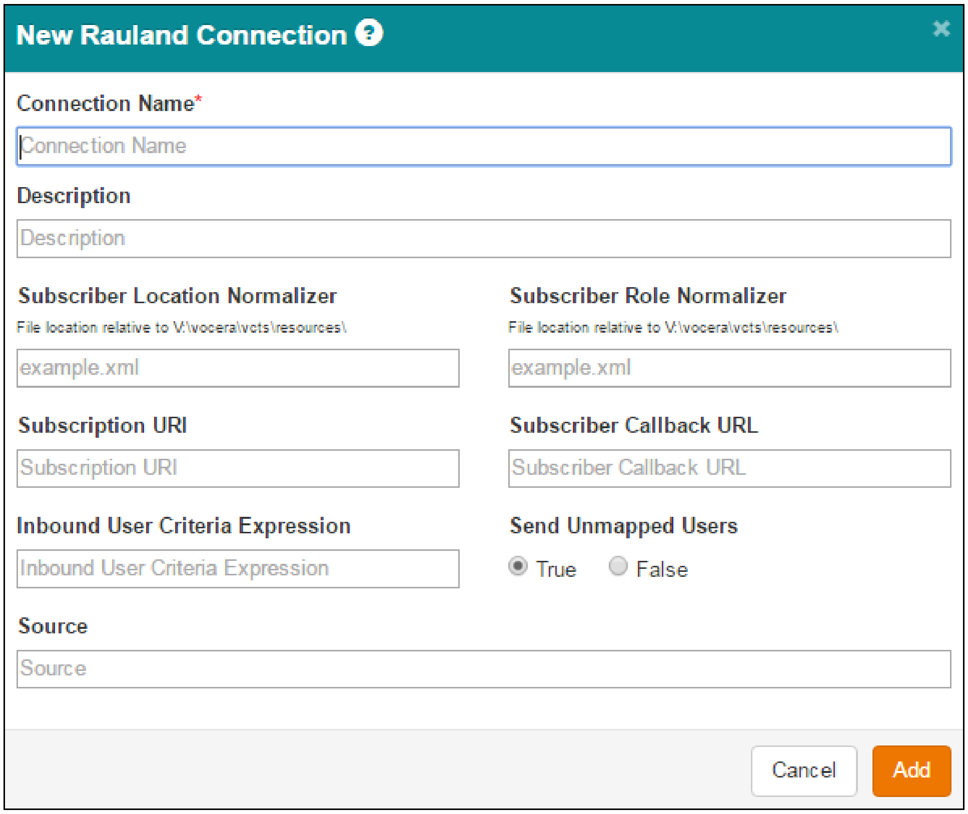
Use the following steps to connect the Vocera Care Team Sync server with your Rauland system.

1. In the CTS Console, click the **Connections** icon ( ) in the navigation bar on the left.
   The **Connections** screen appears.



| Connection Name | Connection Type | IP Address | Last Processed Time |
|---|---|---|---|
| Vocera | Vocera | | |

2. Click the **Add Connection** button and select **Rauland** from the drop-down list that appears.
   The **New Rauland Connection** dialog box appears.



New Rauland Connection

Connection Name*: Connection Name

Description: Description

Subscriber Location Normalizer — File location relative to V:\vocera\vcts\resources\ : example.xml

Subscriber Role Normalizer — File location relative to V:\vocera\vcts\resources\ : example.xml

Subscription URI: Subscription URI

Subscriber Callback URL: Subscriber Callback URL

Inbound User Criteria Expression: Inbound User Criteria Expression

Send Unmapped Users: ● True ○ False

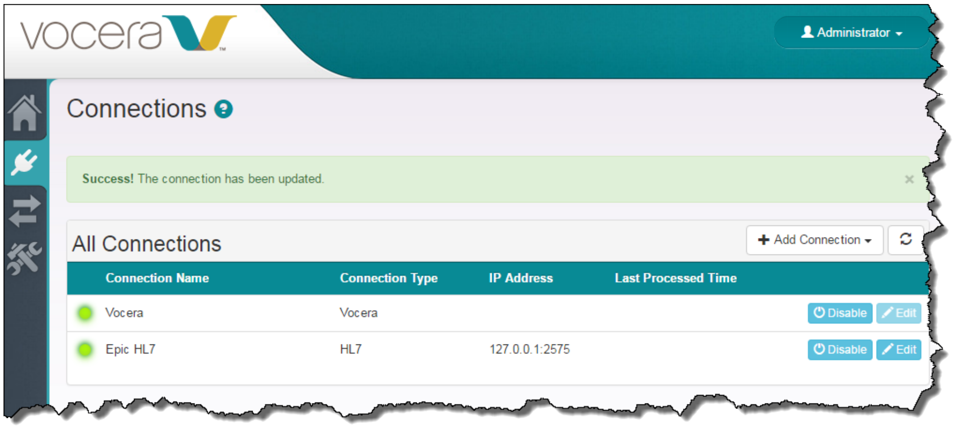
Source: Source

3. Specify the Rauland connection parameters in the **New Rauland Connection** dialog box as follows:

| Field | Description |
|---|---|
| Connection Name | Specify a name for the connection to help you identify it in the future. This name appears as an identifier in other screens of the CTS Console. |
| Description | Optionally enter a more verbose description of the connection for your own information. The description does not appear on the **Connections** page, but it is visible for your reference if you edit the connection at a later time. |
| Subscriber Location Normalizer | The **Subscriber Location Normalizer** normalizes location data such as unit, room, and bed that is used by the foreign system to match the values in CTS. This normalizer is required to resolve naming differences between the two systems.<br>Enter the location of the normalizer relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the normalizers used by different connectors. |
| Subscriber Role Normalizer | The **Subscriber Role Normalizer** normalizes role data such as RN, PCT, and LPN that is used by the foreign system to match the values in CTS. This normalizer is required to resolve naming differences between the two systems.<br>Enter the location of the normalizer relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the normalizers used by different connectors. |
| Subscription URI | Enter the URI of the foreign system in the `http://<hostname>:<port>/path` format.<br>**Note:** If you leave this field blank, it will disable the connection with the end point. In some situations, you may want to complete all the set up for end point, test it, but not have it go live immediately. You can temporarily delete the value in this field to prevent the connection from occurring. |
| Subscriber Callback URL | Enter the URL of callback service on the Vocera Care Team Sync server. The URL comprises the IP address or fully qualified domain name of the Vocera Care Team Sync machine and the hard-coded path of the service (`rauland-adapter/AssignmentService`).<br>For example, if the IP address of the Vocera Care Team Sync server is 10.37.107.171, the full **Subscriber Callback URL** is the following:<br>`http://10.37.107.171/rauland-adapter/AssignmentService` |
| Inbound User Criteria Expression | The **Inbound User Criteria Expression** optionally allows you to specify a template that maps "u-name" values (user IDs) from the external system to values in the **User ID** field in the Vocera Voice Server Administration Console.<br>Specify the full mapping expression in the **Inbound User Criteria Expression** field. You may create multi-line expressions on the single line in this field by using semicolon delimiters. |
| Send Unmapped Users | The **Send Unmapped Users** determines whether the Vocera Voice Server "optimistically" sends user IDs that are possibly unmapped to the external system. By default, this value is set to **True**, and the VS sends all users to the external system. |
| Source | Optionally specify a text string to indicate the source of truth; for example "CTS" or "Rauland".<br>**Note:** You must set the source of truth interactively in the CTS console. This field is for your own reference only. |

4. Click **Add**.

   The **New Rauland Connection** dialog box closes, your changes are saved to the database, and the connection appears in the **All Connections** section of the **Connections** screen.



Connections

**Success!** The connection has been updated.

All Connections

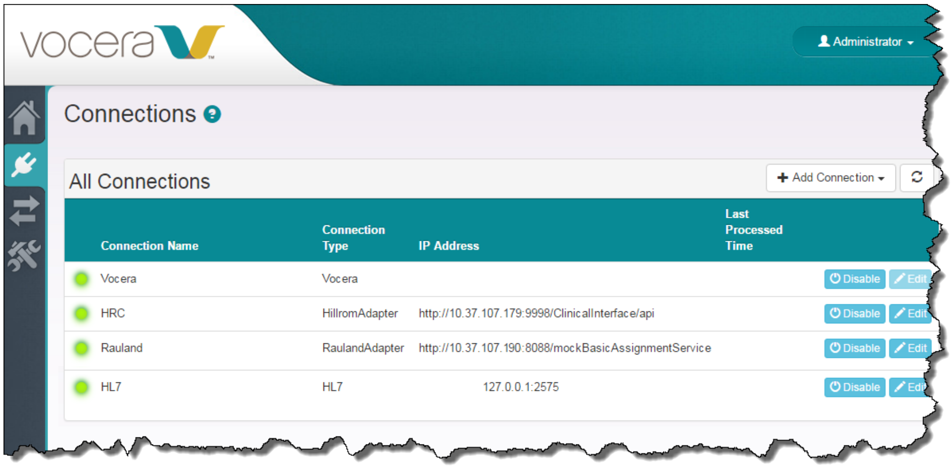
| | Connection Name | Connection Type | IP Address | Last Processed Time | |
|---|---|---|---|---|---|
| ● | Vocera | Vocera | | | Disable Edit |
| ● | Epic HL7 | HL7 | 127.0.0.1:2575 | | Disable Edit |

5. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

When the Vocera Care Team Sync-to-Rauland connection is set up correctly, a green light icon (enabled) appears next to the connection in the **Connection Name** column of the **Connections** screen.



Connections

All Connections

| | Connection Name | Connection Type | IP Address | Last Processed Time | |
|---|---|---|---|---|---|
| ● | Vocera | Vocera | | | Disable Edit |
| ● | HRC | HillromAdapter | http://10.37.107.179:9998/ClinicalInterface/api | | Disable Edit |
| ● | Rauland | RaulandAdapter | http://10.37.107.190:8088/mockBasicAssignmentService | | Disable Edit |
| ● | HL7 | HL7 | 127.0.0.1:2575 | | Disable Edit |

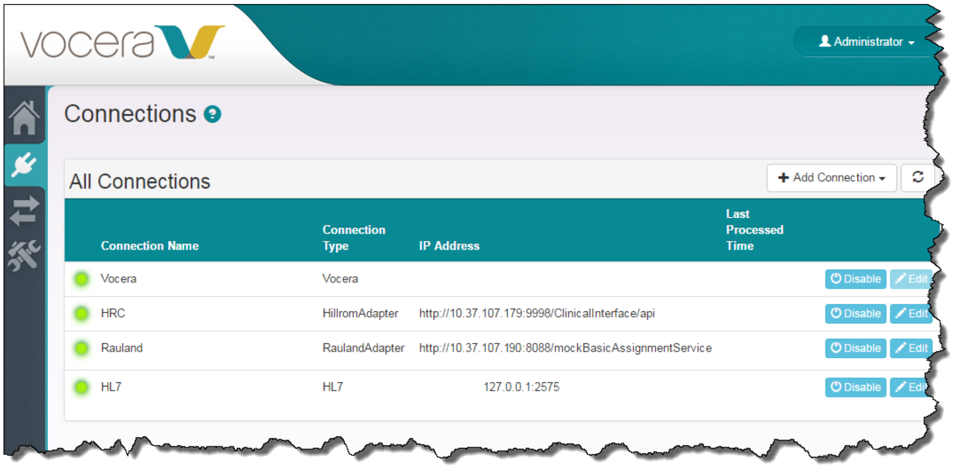
## How to Update a Rauland Connection

Edit the Rauland connection at any time to update the configuration parameters. After saving, you must restart the Vocera Vocera Care Team Sync service to allow changes to take effect.

Use the following steps to update the CTS connection to a Rauland system.

1. In the CTS Console, click the **Connections** icon ( 🔌 ) in the navigation bar on the left.

   The **Connections** screen appears.



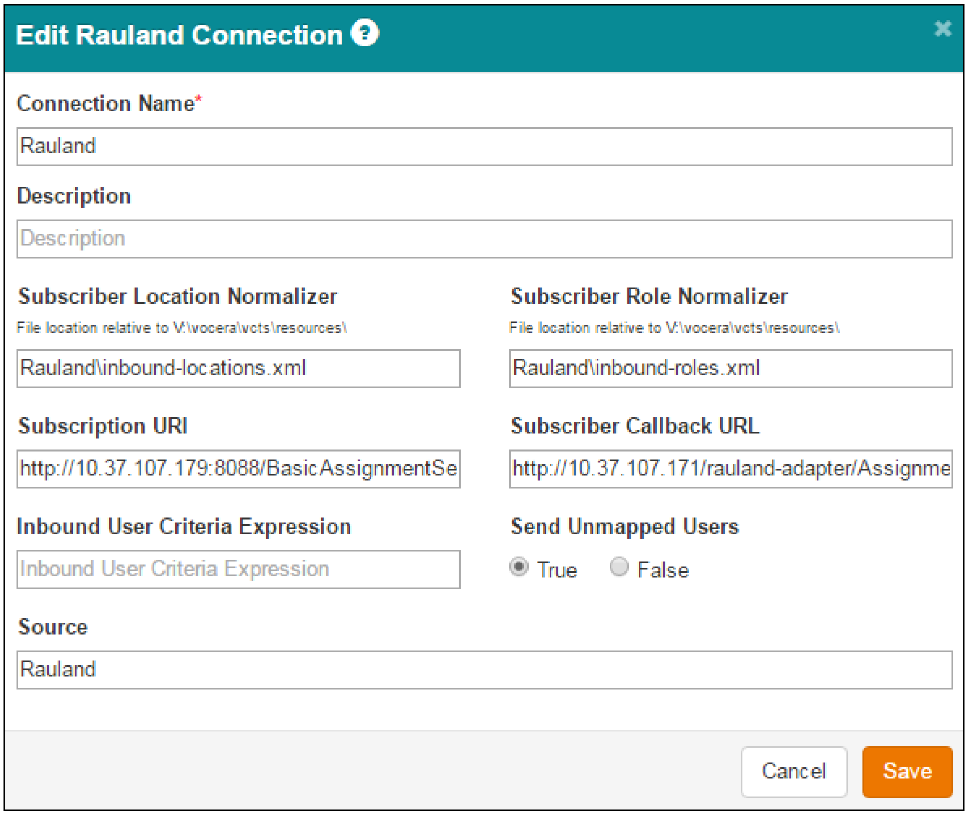
Connections

All Connections

| Connection Name | Connection Type | IP Address | Last Processed Time |
|---|---|---|---|
| Vocera | Vocera | | |
| HRC | HillromAdapter | http://10.37.107.179:9998/ClinicalInterface/api | |
| Rauland | RaulandAdapter | http://10.37.107.190:8088/mockBasicAssignmentService | |
| HL7 | HL7 | 127.0.0.1:2575 | |

2. Click the **Edit** button to the right of the connection that you want to edit.

   The **Edit Rauland Connection** dialog box appears.



Edit Rauland Connection

Connection Name*: Rauland

Description: Description

Subscriber Location Normalizer — File location relative to V:\vocera\vcts\resources\: Rauland\inbound-locations.xml

Subscriber Role Normalizer — File location relative to V:\vocera\vcts\resources\: Rauland\inbound-roles.xml

Subscription URI: http://10.37.107.179:8088/BasicAssignmentSe

Subscriber Callback URL: http://10.37.107.171/rauland-adapter/Assignme

Inbound User Criteria Expression: Inbound User Criteria Expression

Send Unmapped Users: ● True ○ False

Source: Rauland

3. Edit any fields that need to be updated.

   See the descriptions of the fields in *How to Add a Rauland Connection* on page 28.

4. Click **Save** to record your changes.

   The **Edit Rauland Connection** dialog box closes, and the connection changes are saved to the database.

5. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

## The Hill-Rom Connection

The Hill-Rom connection allows Vocera Care Team Sync to connect to a Hill-Rom NaviCare nurse call system or any other nurse call system that implements the HR Clinical API r13 interface.

You can specify either the Hill-Rom system or the CTS as the source of truth for this type of connection.

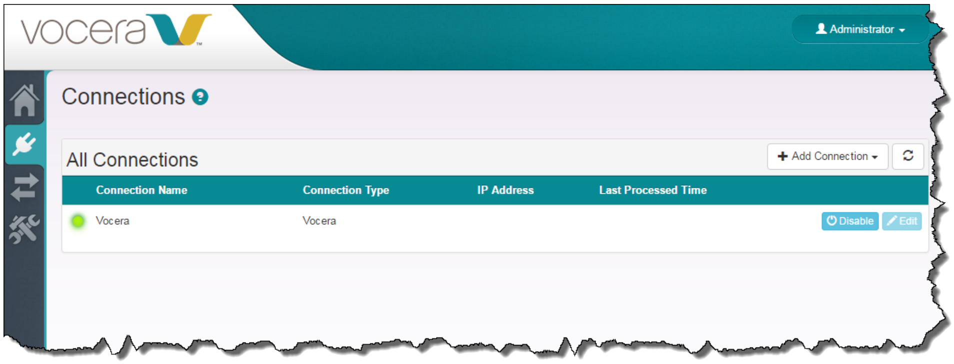
**Note:** When documentation in this section references "Hill-Rom", the content is also true for any system that has implemented the HR Clinical API r13 interface, unless otherwise noted.

### How to Add a Hill-Rom Connection

Specify the location of any normalizers required to process Hill-Rom data correctly, along with the additional parameters that allow the end point on the far end to communicate with the Vocera Care Team Sync server.

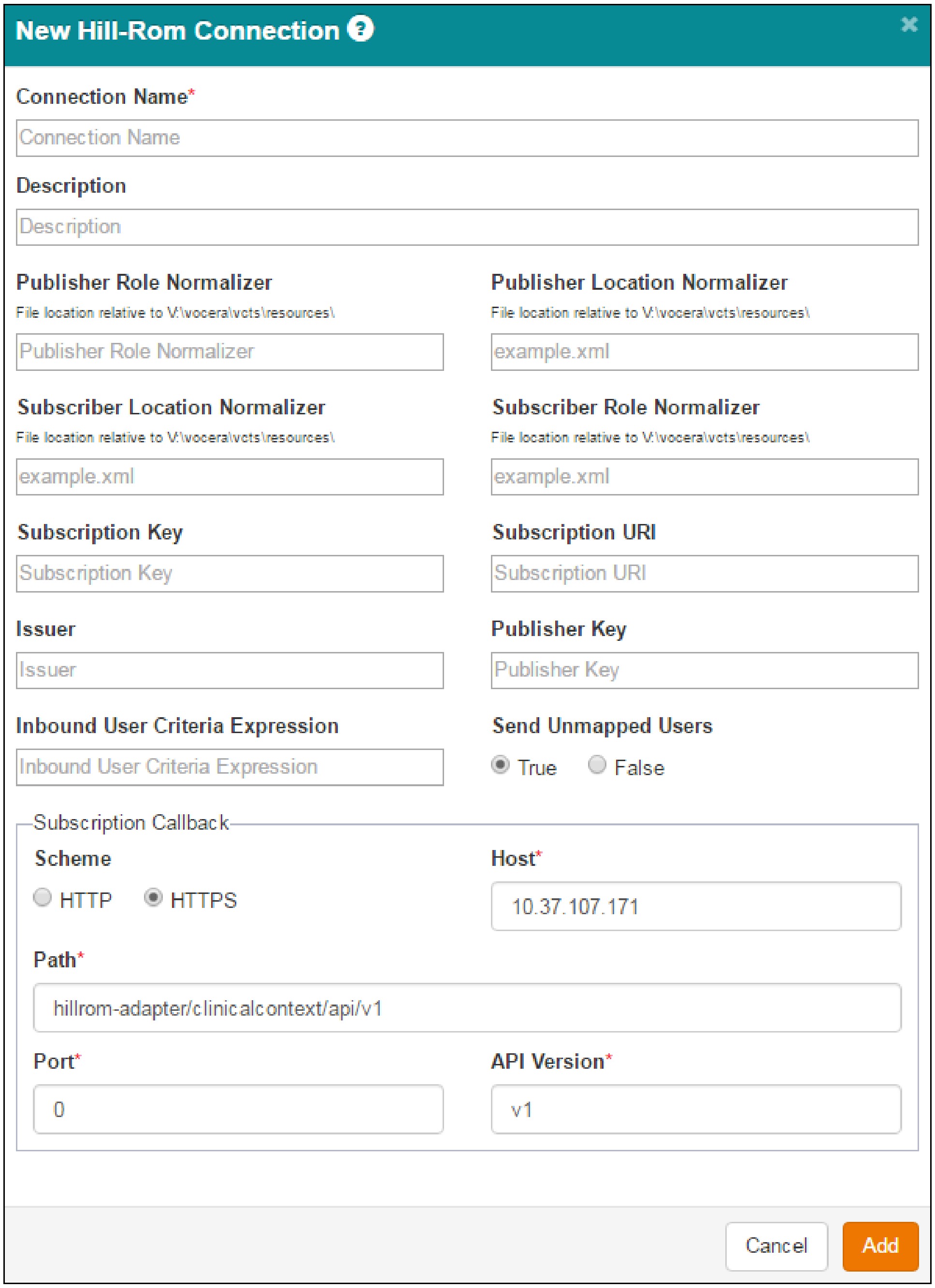
Use the following steps to connect the Vocera Care Team Sync server with your Hill-Rom system.

1. In the CTS Console, click the **Connections** icon ( 🔌 ) in the navigation bar on the left.
   The **Connections** screen appears.



| Connection Name | Connection Type | IP Address | Last Processed Time |
|---|---|---|---|
| Vocera | Vocera | | |

2. Click the **Add Connection** button and select **Hill-Rom** from the drop-down list that appears.
   The **New Hill-Rom Connection** dialog box appears.



New Hill-Rom Connection

Connection Name*: Connection Name

Description: Description

Publisher Role Normalizer (File location relative to V:\vocera\vcts\resources\): Publisher Role Normalizer

Publisher Location Normalizer (File location relative to V:\vocera\vcts\resources\): example.xml

Subscriber Location Normalizer (File location relative to V:\vocera\vcts\resources\): example.xml

Subscriber Role Normalizer (File location relative to V:\vocera\vcts\resources\): example.xml

Subscription Key: Subscription Key

Subscription URI: Subscription URI

Issuer: Issuer

Publisher Key: Publisher Key

Inbound User Criteria Expression: Inbound User Criteria Expression

Send Unmapped Users: ● True ○ False

Subscription Callback

Scheme: ○ HTTP ● HTTPS

Host*: 10.37.107.171

Path*: hillrom-adapter/clinicalcontext/api/v1

Port*: 0

API Version*: v1

3. Specify the Hill-Rom connection parameters in the **New Hill-Rom Connection** dialog box as follows:

| Field | Description |
|---|---|
| Connection Name | Specify a name for the connection to help you identify it in the future. This name appears as an identifier in other screens of the CTS Console. |
| Description | Optionally enter a more verbose description of the connection for your own information. The description does not appear on the **Connections** page, but it is visible for your reference if you edit the connection at a later time. |
| Publisher Role Normalizer | The **Publisher Role Normalizer** normalizes role data that is used by CTS to match the values in the foreign system. This normalizer is required to resolve naming differences between the two systems.<br>Enter the location of the normalizer relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the normalizers used by different connectors. |

| Field | Description |
|---|---|
| Publisher Location Normalizer | The **Publisher Location Normalizer** normalizes location data that is used by CTS to match the values in the foreign system. This normalizer is required to resolve naming differences between the two systems.<br>Enter the location of the normalizer relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the normalizers used by different connectors. |
| Subscriber Location Normalizer | The **Subscriber Location Normalizer** normalizes location data such as unit, room, and bed that is used by the foreign system to match the values in CTS. This normalizer is required to resolve naming differences between the two systems.<br>Enter the location of the normalizer relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the normalizers used by different connectors. |
| Subscriber Role Normalizer | The **Subscriber Role Normalizer** normalizes role data such as RN, PCT, and LPN that is used by the foreign system to match the values in CTS. This normalizer is required to resolve naming differences between the two systems.<br>Enter the location of the normalizer relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the normalizers used by different connectors. |
| Subscription Key | The **Subscription Key** is an encoded JSON web token that securely identifies the CTS to the Hill-Rom system in mutual authentication. Hill-Rom may issue a token on Vocera's behalf and provide it for this purpose.<br>Enter the encoded token that identifies the CTS system. |
| Subscription URI | Enter the URI of the foreign system in the `http://<hostname>:<port>/path` format.<br>**Note:** If you leave this field blank, it will disable the connection with the end point. In some situations, you may want to complete all the set up for end point, test it, but not have it go live immediately. You can temporarily delete the value in this field to prevent the connection from occurring. |
| Issuer | Optionally specify a text string to indicate the party who issued the subscription key; for example, "CTS" or "Hill-Rom". |
| Publisher Key | The **Publisher Key** is an encoded JSON web token that securely identifies the Hill-Rom system to the CTS in mutual authentication. Hill-Rom will issue a token and provide it for this purpose.<br>Enter the encoded token that identifies the Hill-Rom system. |
| Inbound User Criteria Expression | The **Inbound User Criteria Expression** optionally allows you to specify a template that maps "u-name" values (user IDs) from the external system to values in the **User ID** field in the Vocera Voice Server Administration Console.<br>Specify the full mapping expression in the **Inbound User Criteria Expression** field. You may create multi-line expressions on the single line in this field by using semicolon delimiters. |
| Send Unmapped Users | The **Send Unmapped Users** determines whether the Vocera Voice Server "optimistically" sends user IDs that are possibly unmapped to the external system. By default, this value is set to **True**, and the VS sends all users to the external system. |

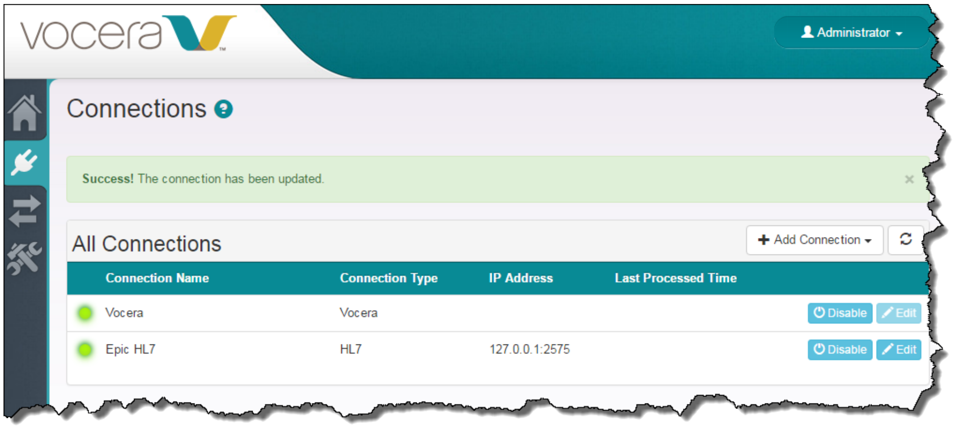
4. In the **Subscription Callback** section of the **New Hill-Rom Connection** dialog box, specify the following information:

| Field | Description |
|---|---|
| Scheme | Specify whether the protocol used by the CTS server to call back to the Hill-Rom system is **HTTP** or **HTTPS**. |
| Host | Specify the IP address or the fully qualified domain name of the CTS system. |
| Path | The path to the adapter on the Hill-Rom system. CTS supplies this value for you automatically. Do not change this value unless the Hill-Rom system mandates it. |

| Field | Description |
|---|---|
| Port | The port on which the Hill-Rom adapter listens for a connection from the CTS server. CTS supplies this value for you automatically. Do not change this value unless the Hill-Rom system mandates it. |
| API Version | The API version used by the Hill-Rom adapter. CTS supplies this value for you automatically. Do not change this value unless the Hill-Rom system mandates it. |

5. Click **Add**.

   The **New Hill-Rom Connection** dialog box closes, your changes are saved to the database, and the connection appears in the **All Connections** section of the **Connections** screen.



Connections

**Success!** The connection has been updated.

All Connections

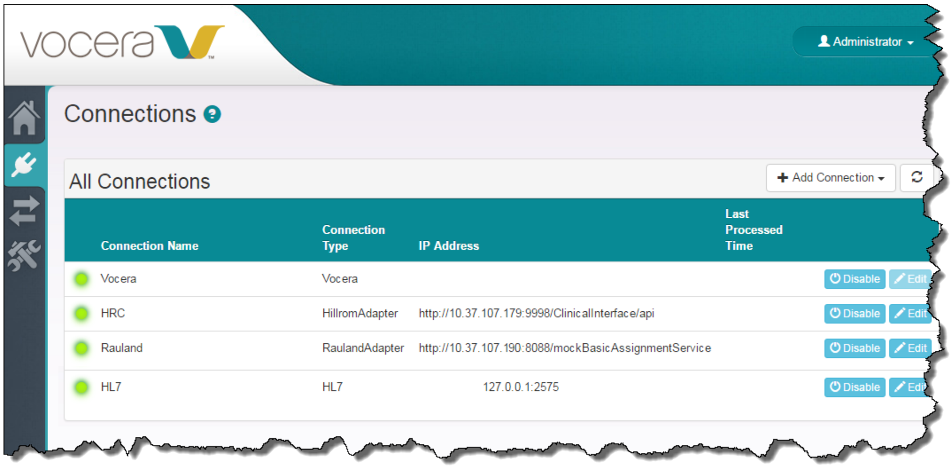
| Connection Name | Connection Type | IP Address | Last Processed Time |
|---|---|---|---|
| Vocera | Vocera | | |
| Epic HL7 | HL7 | 127.0.0.1:2575 | |

6. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

When the Vocera Care Team Sync-to-Hill-Rom connection is set up correctly, a green light icon (enabled) appears next to the connection in the **Connection Name** column of the **Connections** screen.



Connections

All Connections

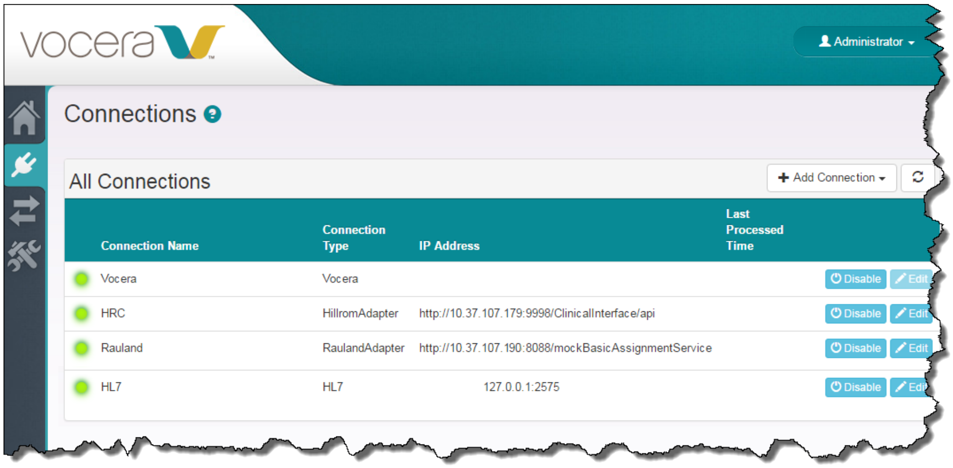
| Connection Name | Connection Type | IP Address | Last Processed Time |
|---|---|---|---|
| Vocera | Vocera | | |
| HRC | HillromAdapter | http://10.37.107.179:9998/ClinicalInterface/api | |
| Rauland | RaulandAdapter | http://10.37.107.190:8088/mockBasicAssignmentService | |
| HL7 | HL7 | 127.0.0.1:2575 | |

### How to Update a Hill-Rom Connection

Edit the Hill-Rom connection at any time to update the configuration parameters. After saving, you must restart the Vocera Vocera Care Team Sync service to allow changes to take effect.

Use the following steps to update the CTS connection to a Hill-Rom system.

1. In the CTS Console, click the **Connections** icon ( 🔌 ) in the navigation bar on the left.
   The **Connections** screen appears.

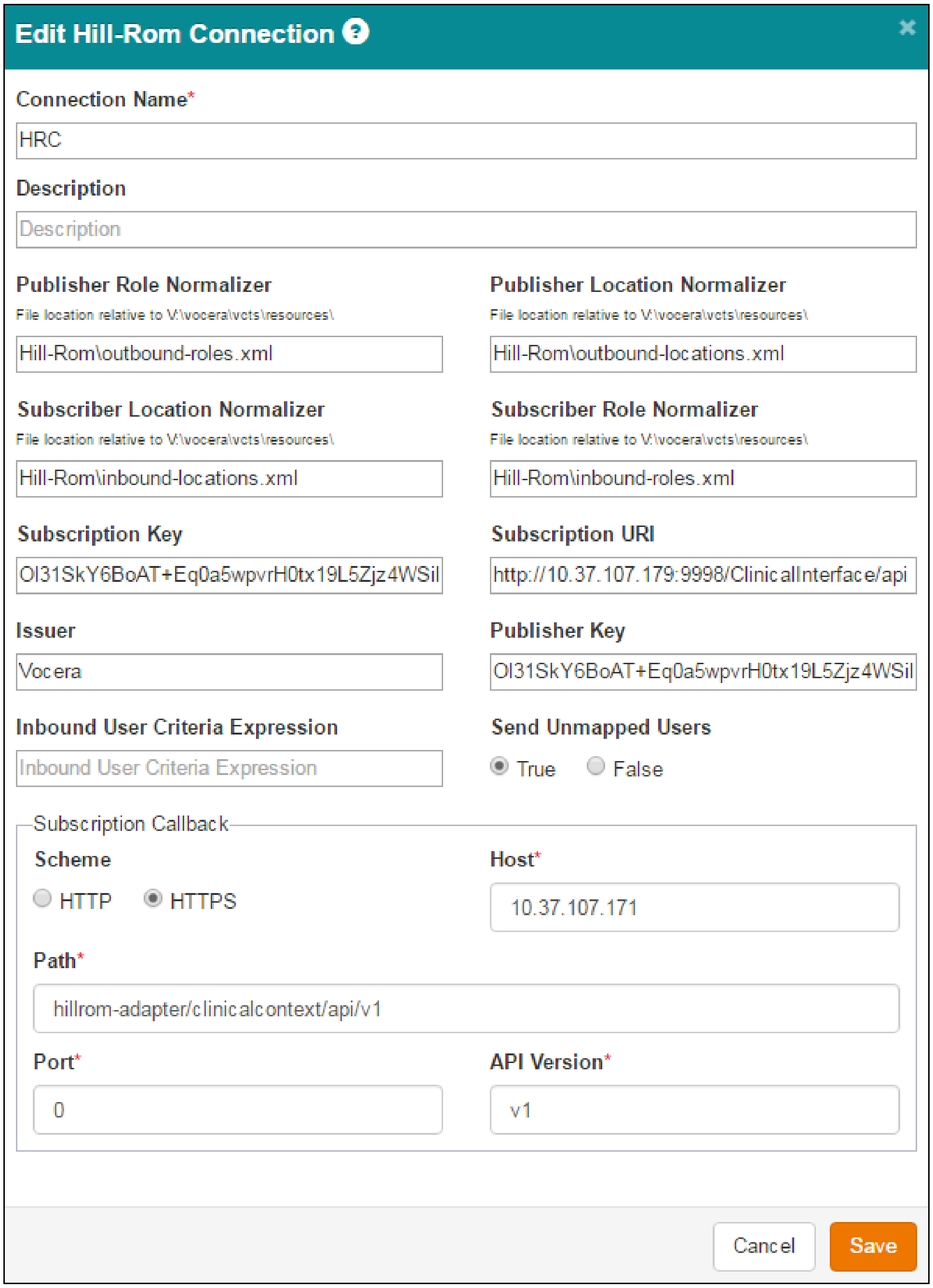


Connections

All Connections

| | Connection Name | Connection Type | IP Address | Last Processed Time | |
|---|---|---|---|---|---|
| ● | Vocera | Vocera | | | Disable Edit |
| ● | HRC | HillromAdapter | http://10.37.107.179:9998/ClinicalInterface/api | | Disable Edit |
| ● | Rauland | RaulandAdapter | http://10.37.107.190:8088/mockBasicAssignmentService | | Disable Edit |
| ● | HL7 | HL7 | 127.0.0.1:2575 | | Disable Edi |

2. Click the **Edit** button to the right of the connection that you want to edit.
   The **Edit HL7 Connection** dialog box appears.



Edit Hill-Rom Connection

Connection Name*
HRC

Description
Description

Publisher Role Normalizer
File location relative to V:\vocera\vcts\resources\
Hill-Rom\outbound-roles.xml

Publisher Location Normalizer
File location relative to V:\vocera\vcts\resources\
Hill-Rom\outbound-locations.xml

Subscriber Location Normalizer
File location relative to V:\vocera\vcts\resources\
Hill-Rom\inbound-locations.xml

Subscriber Role Normalizer
File location relative to V:\vocera\vcts\resources\
Hill-Rom\inbound-roles.xml

Subscription Key
OI31SkY6BoAT+Eq0a5wpvrH0tx19L5Zjz4WSil

Subscription URI
http://10.37.107.179:9998/ClinicalInterface/api

Issuer
Vocera

Publisher Key
OI31SkY6BoAT+Eq0a5wpvrH0tx19L5Zjz4WSil

Inbound User Criteria Expression
Inbound User Criteria Expression

Send Unmapped Users
● True ○ False

Subscription Callback

Scheme
○ HTTP ● HTTPS

Host*
10.37.107.171

Path*
hillrom-adapter/clinicalcontext/api/v1

Port*
0

API Version*
v1

3. Edit any fields that need to be updated.
   See the descriptions of the fields in *How to Add a Hill-Rom Connection* on page 32.
4. Click **Save** to record your changes.
   The **Edit Hill-Rom Connection** dialog box closes, and the connection changes are saved to the database.
5. Restart the Vocera CTS service.
   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.
   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

## The HL7 Connection

The HL7 connection allows Vocera Care Team Sync to connect to any system whose connection mechanism uses HL7 v 2.1-2.6. These systems include Admit/Discharge/Transfer (ADT) systems such as Epic and others.

You can specify either the HL7 system or the CTS as the source of truth for this type of connection.

### About the HL7 Map Files

The HL7 integration may require map files to associate the events and fields used by the HL7 end point with the events and fields that CTS expects.

CTS provides sample ADT map files in the `\vocera\vcts\samples` directory:

- The `adt-field-map.xml` file is a sample field map.
- The `adt-map.xml` file is a sample event map.

If necessary, you can copy these map files from the `\vocera\vcts\samples` directory to the proper location in the `\vocera\vcts\resources` tree: either directly within `resources` or within the subdirectory of your choice.

The field types (segment elements) that Vocera expects are documented in *Segment Element Recommendation* on page 124. If the end point is using a different field type, update the `adt-field-map.xml` file to accommodate it. See *How to Update the ADT Field Map* on page 38.

The event types that Vocera expects are documented in *Event Types* on page 121. If the end point is using a different or additional event type for admit, discharge, or transfer events, update the `adt-map.xml` file to accommodate it. See *How to Update the ADT Event Map* on page 40.

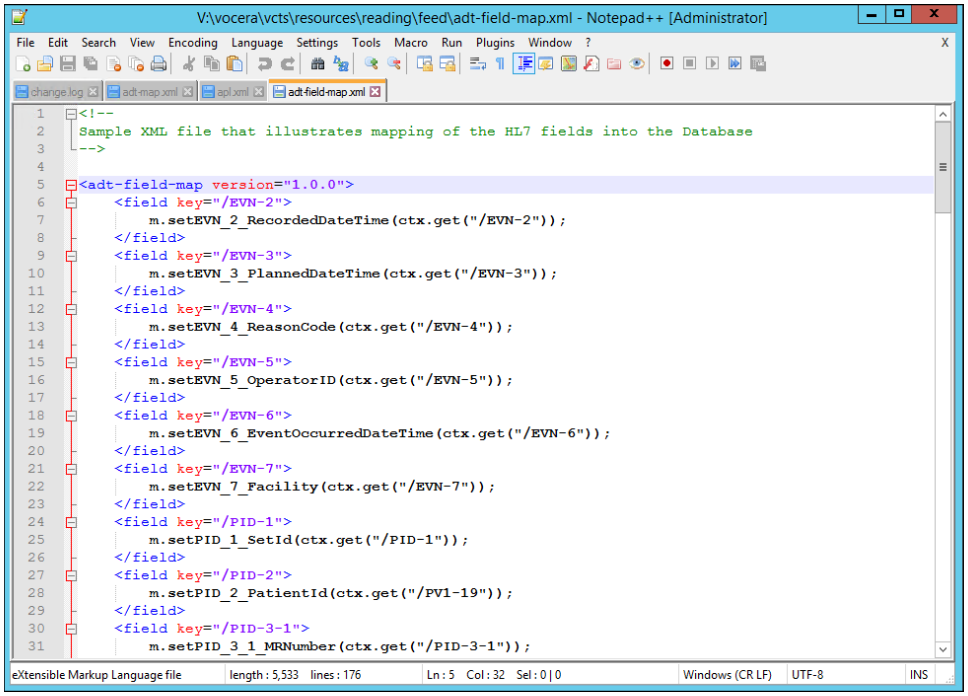
The *Integrating with an HL7 Feed* on page 112 appendix contains detailed information on ADT events and fields and the ways in which CTS expects to user them.

### How to Update the ADT Field Map

In some situations, the ADT end point supplies a value for an HL7 segment in a different field than CTS expects. If this situation occurs, you need to update the ADT field map to accommodate the behavior of the end point.

To update the ADT field map:

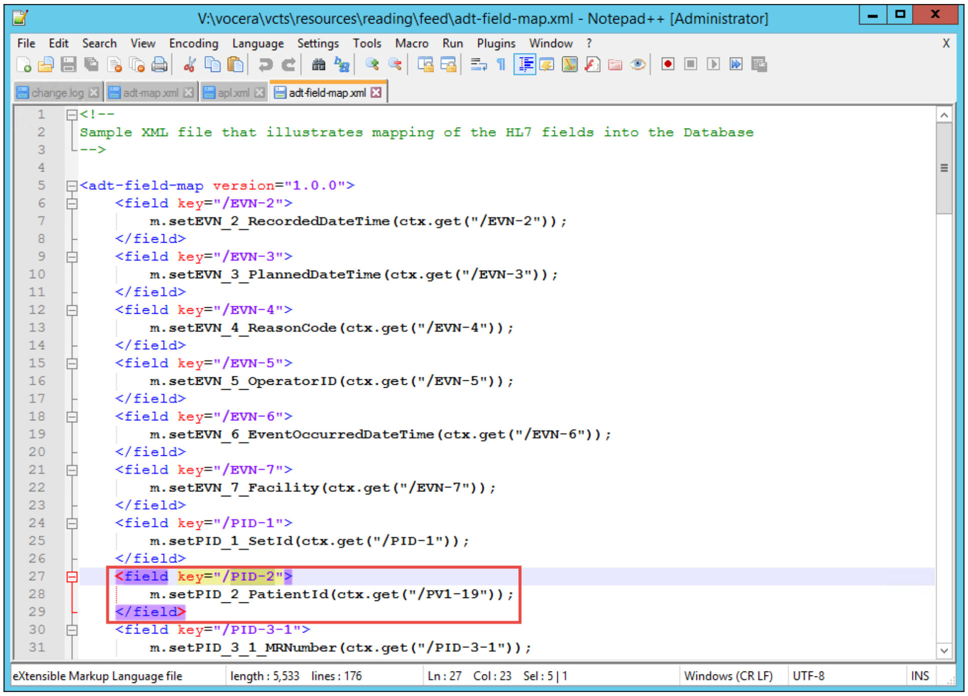
1. Copy the `adt-field-map.xml` file from `\vocera\vcts\samples` to the proper location in the `\vocera\vcts\resources` tree: either directly within `resources` or within the subdirectory of your choice.
2. Open the `adt-field-map.xml` file in a text editor (the open source *Notepad++* product is a convenient XML-aware editor if you do not have one).
   The ADT field map opens.


V:\vocera\vcts\resources\reading\feed\adt-field-map.xml - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log
adt-map.xml
apl.xml
adt-field-map.xml
eXtensible Markup Language file
length : 5,533 lines : 176
Ln : 5 Col : 32 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS

```
<!--
Sample XML file that illustrates mapping of the HL7 fields into the Database
-->

<adt-field-map version="1.0.0">
    <field key="/EVN-2">
        m.setEVN_2_RecordedDateTime(ctx.get("/EVN-2"));
    </field>
    <field key="/EVN-3">
        m.setEVN_3_PlannedDateTime(ctx.get("/EVN-3"));
    </field>
    <field key="/EVN-4">
        m.setEVN_4_ReasonCode(ctx.get("/EVN-4"));
    </field>
    <field key="/EVN-5">
        m.setEVN_5_OperatorID(ctx.get("/EVN-5"));
    </field>
    <field key="/EVN-6">
        m.setEVN_6_EventOccurredDateTime(ctx.get("/EVN-6"));
    </field>
    <field key="/EVN-7">
        m.setEVN_7_Facility(ctx.get("/EVN-7"));
    </field>
    <field key="/PID-1">
        m.setPID_1_SetId(ctx.get("/PID-1"));
    </field>
    <field key="/PID-2">
        m.setPID_2_PatientId(ctx.get("/PV1-19"));
    </field>
    <field key="/PID-3-1">
        m.setPID_3_1_MRNumber(ctx.get("/PID-3-1"));
```

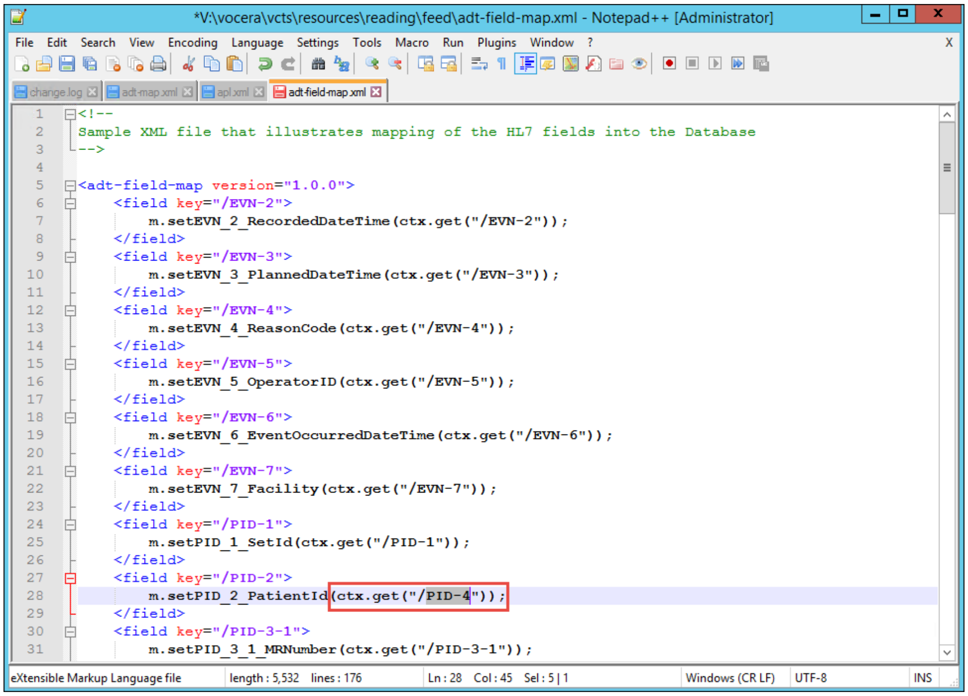
**3.** Navigate to the segment element that you need to modify.

For example, suppose the HL7 feed actually provides a value for the `PID-2` ("Patient ID") segment element in the `PID-4` ("Alternate Patient ID") element. In this situation, navigate to the `field` whose `key` is set to `PID-2`.


V:\vocera\vcts\resources\reading\feed\adt-field-map.xml - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log
adt-map.xml
apl.xml
adt-field-map.xml
eXtensible Markup Language file
length : 5,533 lines : 176
Ln : 27 Col : 23 Sel : 5 | 1
Windows (CR LF)
UTF-8
INS

```
<!--
Sample XML file that illustrates mapping of the HL7 fields into the Database
-->

<adt-field-map version="1.0.0">
    <field key="/EVN-2">
        m.setEVN_2_RecordedDateTime(ctx.get("/EVN-2"));
    </field>
    <field key="/EVN-3">
        m.setEVN_3_PlannedDateTime(ctx.get("/EVN-3"));
    </field>
    <field key="/EVN-4">
        m.setEVN_4_ReasonCode(ctx.get("/EVN-4"));
    </field>
    <field key="/EVN-5">
        m.setEVN_5_OperatorID(ctx.get("/EVN-5"));
    </field>
    <field key="/EVN-6">
        m.setEVN_6_EventOccurredDateTime(ctx.get("/EVN-6"));
    </field>
    <field key="/EVN-7">
        m.setEVN_7_Facility(ctx.get("/EVN-7"));
    </field>
    <field key="/PID-1">
        m.setPID_1_SetId(ctx.get("/PID-1"));
    </field>
    <field key="/PID-2">
        m.setPID_2_PatientId(ctx.get("/PV1-19"));
    </field>
    <field key="/PID-3-1">
        m.setPID_3_1_MRNumber(ctx.get("/PID-3-1"));
```

**4.** Set the value in the `ctx.get()` method for the `PID-2` key to `PID-4`.


*V:\vocera\vcts\resources\reading\feed\adt-field-map.xml - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log
adt-map.xml
api.xml
adt-field-map.xml

```
<!--
Sample XML file that illustrates mapping of the HL7 fields into the Database
-->

<adt-field-map version="1.0.0">
    <field key="/EVN-2">
        m.setEVN_2_RecordedDateTime(ctx.get("/EVN-2"));
    </field>
    <field key="/EVN-3">
        m.setEVN_3_PlannedDateTime(ctx.get("/EVN-3"));
    </field>
    <field key="/EVN-4">
        m.setEVN_4_ReasonCode(ctx.get("/EVN-4"));
    </field>
    <field key="/EVN-5">
        m.setEVN_5_OperatorID(ctx.get("/EVN-5"));
    </field>
    <field key="/EVN-6">
        m.setEVN_6_EventOccurredDateTime(ctx.get("/EVN-6"));
    </field>
    <field key="/EVN-7">
        m.setEVN_7_Facility(ctx.get("/EVN-7"));
    </field>
    <field key="/PID-1">
        m.setPID_1_SetId(ctx.get("/PID-1"));
    </field>
    <field key="/PID-2">
        m.setPID_2_PatientId(ctx.get("/PID-4"));
    </field>
    <field key="/PID-3-1">
        m.setPID_3_1_MRNumber(ctx.get("/PID-3-1"));
```

eXtensible Markup Language file
length : 5,532 lines : 176
Ln : 28 Col : 45 Sel : 5 | 1
Windows (CR LF)
UTF-8
INS

5. Save the file and close the editor.
6. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

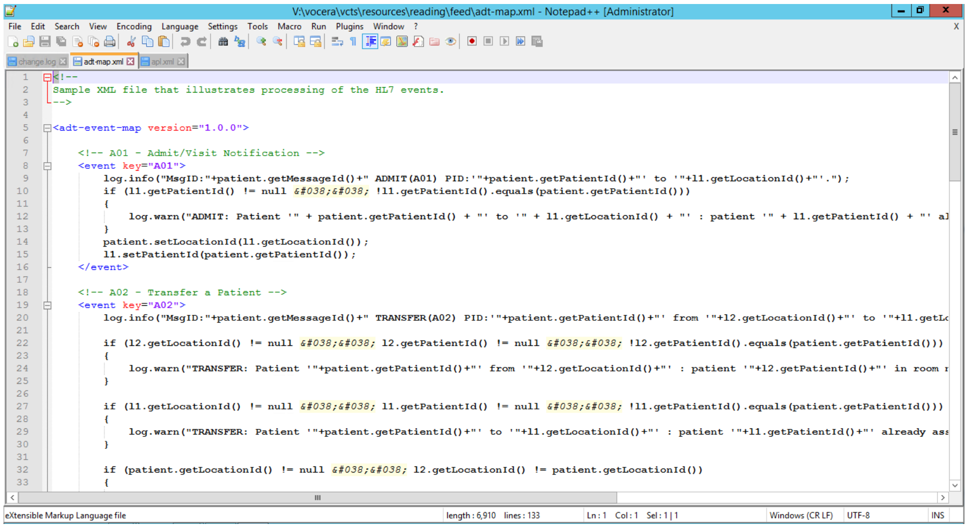
### How to Update the ADT Event Map

In some situations, the ADT end point may use an unexpected event type to perform an admit, transfer, discharge, or update action. If this situation occurs, you need to update the ADT event map to accommodate the behavior of the end point.

For example, suppose the HL7 end point uses both the **A02** and **A08** events to record a **Transfer** action (**A02** is typically used for **Transfer** and **A08** is typically used for **Update**). In this situation, you need to copy the logic for handling the transfer from the **A02** section of the event map to the **A08** section.

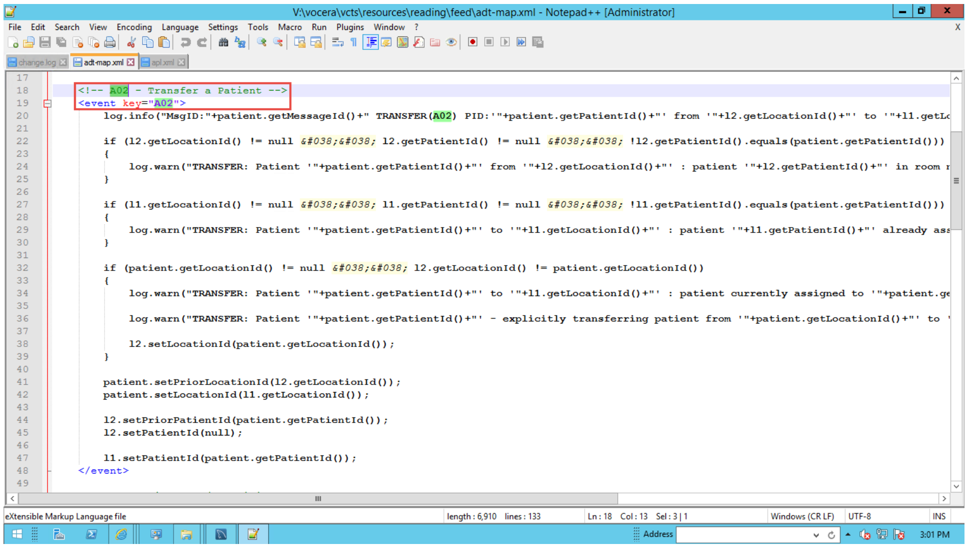
1. Copy the `adt-map.xml` file from `\vocera\vcts\samples` to the proper location in the `\vocera\vcts\resources` tree: either directly within `resources` or within the subdirectory of your choice.
2. Open the `adt-map.xml` file in a text editor (the open source *Notepad++* product is a convenient XML-aware editor if you do not have one).

   The ADT event map opens.


V:\vocera\vcts\resources\reading\feed\adt-map.xml - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log
adt-map.xml
apl.xml
eXtensible Markup Language file
length : 6,910 lines : 133
Ln : 1 Col : 1 Sel : 1 | 1
Windows (CR LF)
UTF-8
INS

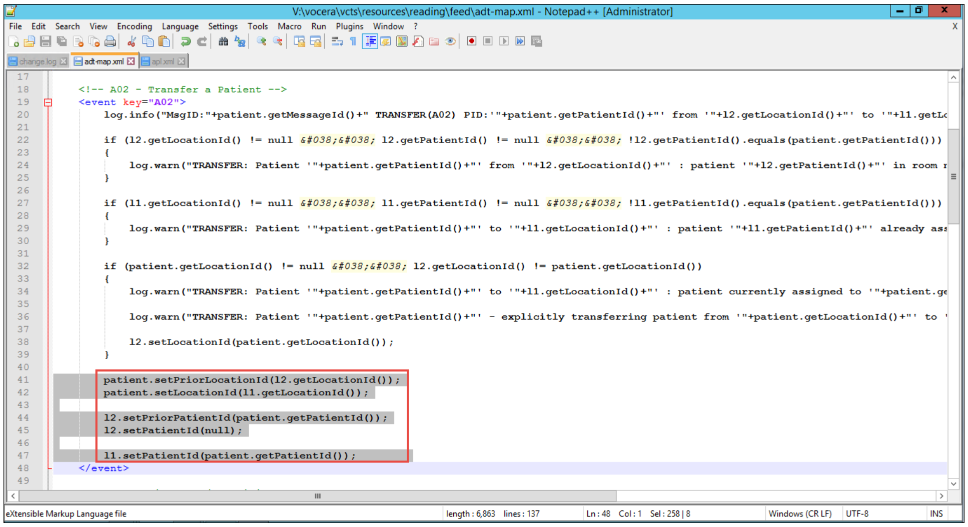
```
<!--
Sample XML file that illustrates processing of the HL7 events.
-->

<adt-event-map version="1.0.0">

    <!-- A01 - Admit/Visit Notification -->
    <event key="A01">
        log.info("MsgID:"+patient.getMessageId()+" ADMIT(A01) PID:'"+patient.getPatientId()+"' to '"+l1.getLocationId()+"'.");
        if (l1.getPatientId() != null &#038;&#038; !l1.getPatientId().equals(patient.getPatientId()))
        {
            log.warn("ADMIT: Patient '" + patient.getPatientId() + "' to '" + l1.getLocationId() + "' : patient '" + l1.getPatientId() + "' al
        }
        patient.setLocationId(l1.getLocationId());
        l1.setPatientId(patient.getPatientId());
    </event>

    <!-- A02 - Transfer a Patient -->
    <event key="A02">
        log.info("MsgID:"+patient.getMessageId()+" TRANSFER(A02) PID:'"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' to '"+l1.getLo

        if (l2.getLocationId() != null &#038;&#038; l2.getPatientId() != null &#038;&#038; !l2.getPatientId().equals(patient.getPatientId()))
        {
            log.warn("TRANSFER: Patient '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' : patient '"+l2.getPatientId()+"' in room r
        }

        if (l1.getLocationId() != null &#038;&#038; l1.getPatientId() != null &#038;&#038; !l1.getPatientId().equals(patient.getPatientId()))
        {
            log.warn("TRANSFER: Patient '"+patient.getPatientId()+"' to '"+l1.getLocationId()+"' : patient '"+l1.getPatientId()+"' already ass
        }

        if (patient.getLocationId() != null &#038;&#038; l2.getLocationId() != patient.getLocationId())
        {
```

3. Navigate to the **A02** event section of the map.


V:\vocera\vcts\resources\reading\feed\adt-map.xml - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log
adt-map.xml
apl.xml
eXtensible Markup Language file
length : 6,910 lines : 133
Ln : 18 Col : 13 Sel : 3 | 1
Windows (CR LF)
UTF-8
INS
Address
3:01 PM

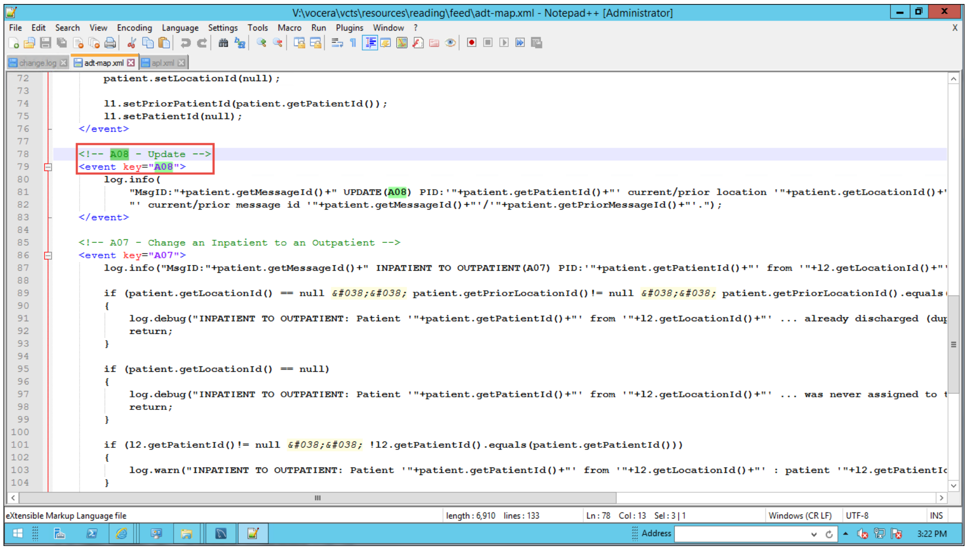
```

    <!-- A02 - Transfer a Patient -->
    <event key="A02">
        log.info("MsgID:"+patient.getMessageId()+" TRANSFER(A02) PID:'"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' to '"+l1.getLo

        if (l2.getLocationId() != null &#038;&#038; l2.getPatientId() != null &#038;&#038; !l2.getPatientId().equals(patient.getPatientId()))
        {
            log.warn("TRANSFER: Patient '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' : patient '"+l2.getPatientId()+"' in room r
        }

        if (l1.getLocationId() != null &#038;&#038; l1.getPatientId() != null &#038;&#038; !l1.getPatientId().equals(patient.getPatientId()))
        {
            log.warn("TRANSFER: Patient '"+patient.getPatientId()+"' to '"+l1.getLocationId()+"' : patient '"+l1.getPatientId()+"' already ass
        }

        if (patient.getLocationId() != null &#038;&#038; l2.getLocationId() != patient.getLocationId())
        {
            log.warn("TRANSFER: Patient '"+patient.getPatientId()+"' to '"+l1.getLocationId()+"' : patient currently assigned to '"+patient.ge

            log.warn("TRANSFER: Patient '"+patient.getPatientId()+"' - explicitly transferring patient from '"+patient.getLocationId()+"' to '

            l2.setLocationId(patient.getLocationId());
        }

        patient.setPriorLocationId(l2.getLocationId());
        patient.setLocationId(l1.getLocationId());

        l2.setPriorPatientId(patient.getPatientId());
        l2.setPatientId(null);

        l1.setPatientId(patient.getPatientId());
    </event>

```

4. Select all the logic statements within the opening and closing <event/> tags and copy them (do not copy the logging statements at this time).


V:\vocera\vcts\resources\reading\feed\adt-map.xml - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log
adt-map.xml
apl.xml
eXtensible Markup Language file
length : 6,863 lines : 137
Ln : 48 Col : 1 Sel : 258 | 8
Windows (CR LF)
UTF-8
INS

```

    <!-- A02 - Transfer a Patient -->
    <event key="A02">
        log.info("MsgID:"+patient.getMessageId()+" TRANSFER(A02) PID:'"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' to '"+l1.getLo

        if (l2.getLocationId() != null &#038;&#038; l2.getPatientId() != null &#038;&#038; !l2.getPatientId().equals(patient.getPatientId()))
        {
            log.warn("TRANSFER: Patient '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' : patient '"+l2.getPatientId()+"' in room r
        }

        if (l1.getLocationId() != null &#038;&#038; l1.getPatientId() != null &#038;&#038; !l1.getPatientId().equals(patient.getPatientId()))
        {
            log.warn("TRANSFER: Patient '"+patient.getPatientId()+"' to '"+l1.getLocationId()+"' : patient '"+l1.getPatientId()+"' already ass
        }

        if (patient.getLocationId() != null &#038;&#038; l2.getLocationId() != patient.getLocationId())
        {
            log.warn("TRANSFER: Patient '"+patient.getPatientId()+"' to '"+l1.getLocationId()+"' : patient currently assigned to '"+patient.ge

            log.warn("TRANSFER: Patient '"+patient.getPatientId()+"' - explicitly transferring patient from '"+patient.getLocationId()+"' to '

            l2.setLocationId(patient.getLocationId());
        }

        patient.setPriorLocationId(l2.getLocationId());
        patient.setLocationId(l1.getLocationId());

        l2.setPriorPatientId(patient.getPatientId());
        l2.setPatientId(null);

        l1.setPatientId(patient.getPatientId());
    </event>

```

5. Navigate to the **A08** event section of the map.

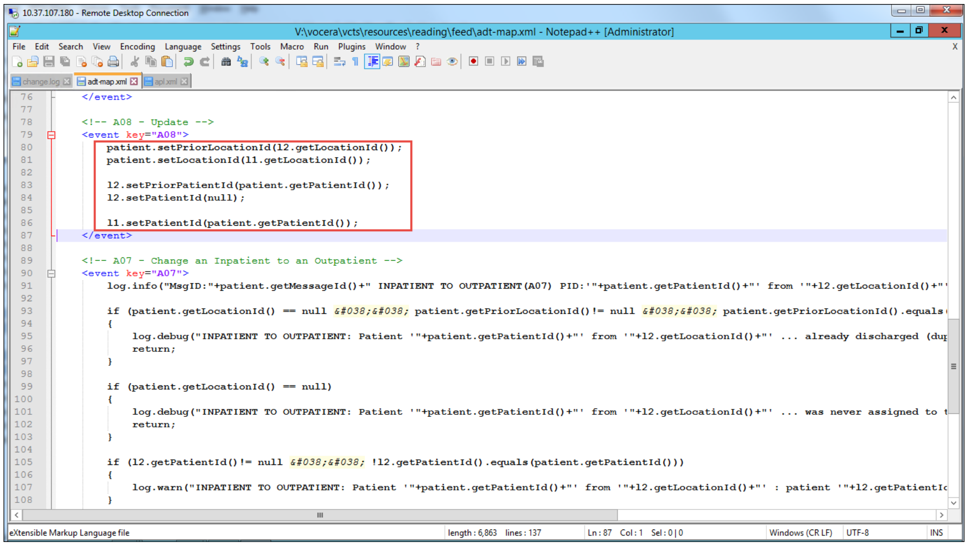


V:\vocera\vcts\resources\reading\feed\adt-map.xml - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log
adt-map.xml
apl.xml

```
        patient.setLocationId(null);

        l1.setPriorPatientId(patient.getPatientId());
        l1.setPatientId(null);
    </event>

    <!-- A08 - Update -->
    <event key="A08">
        log.info(
            "MsgID:"+patient.getMessageId()+" UPDATE(A08) PID:'"+patient.getPatientId()+"' current/prior location '"+patient.getLocationId()+'
            "' current/prior message id '"+patient.getMessageId()+"'/'"+patient.getPriorMessageId()+"'.");
    </event>

    <!-- A07 - Change an Inpatient to an Outpatient -->
    <event key="A07">
        log.info("MsgID:"+patient.getMessageId()+" INPATIENT TO OUTPATIENT(A07) PID:'"+patient.getPatientId()+"' from '"+l2.getLocationId()+"'

        if (patient.getLocationId() == null &#038;&#038; patient.getPriorLocationId()!= null &#038;&#038; patient.getPriorLocationId().equals
        {
            log.debug("INPATIENT TO OUTPATIENT: Patient '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' ... already discharged (dup
            return;
        }

        if (patient.getLocationId() == null)
        {
            log.debug("INPATIENT TO OUTPATIENT: Patient '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' ... was never assigned to t
            return;
        }

        if (l2.getPatientId()!= null &#038;&#038; !l2.getPatientId().equals(patient.getPatientId()))
        {
            log.warn("INPATIENT TO OUTPATIENT: Patient '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' : patient '"+l2.getPatientIc
        }
```

eXtensible Markup Language file length : 6,910 lines : 133 Ln : 78 Col : 13 Sel : 3 | 1 Windows (CR LF) UTF-8 INS

6. Paste the **A02** logic and logging within the A08 <event/> tags, replacing all the existing **A08** statements.



10.37.107.180 - Remote Desktop Connection
V:\vocera\vcts\resources\reading\feed\adt-map.xml - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log
adt-map.xml
apl.xml

```
    </event>

    <!-- A08 - Update -->
    <event key="A08">
        patient.setPriorLocationId(l2.getLocationId());
        patient.setLocationId(l1.getLocationId());

        l2.setPriorPatientId(patient.getPatientId());
        l2.setPatientId(null);

        l1.setPatientId(patient.getPatientId());
    </event>

    <!-- A07 - Change an Inpatient to an Outpatient -->
    <event key="A07">
        log.info("MsgID:"+patient.getMessageId()+" INPATIENT TO OUTPATIENT(A07) PID:'"+patient.getPatientId()+"' from '"+l2.getLocationId()+"'

        if (patient.getLocationId() == null &#038;&#038; patient.getPriorLocationId()!= null &#038;&#038; patient.getPriorLocationId().equals
        {
            log.debug("INPATIENT TO OUTPATIENT: Patient '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' ... already discharged (dup
            return;
        }

        if (patient.getLocationId() == null)
        {
            log.debug("INPATIENT TO OUTPATIENT: Patient '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' ... was never assigned to t
            return;
        }

        if (l2.getPatientId()!= null &#038;&#038; !l2.getPatientId().equals(patient.getPatientId()))
        {
            log.warn("INPATIENT TO OUTPATIENT: Patient '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"' : patient '"+l2.getPatientIc
        }
```

eXtensible Markup Language file length : 6,863 lines : 137 Ln : 87 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

**Note:** You may also copy and paste the logging statements from **A02** into **A08**; however, you should also replace any references to "Transfer" in the logging statements with "Update", so you can distinguish between **A02** and **A08** logging during troubleshooting.

7. Save the file and close the editor.
8. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

## Connecting to an HL7 Feed

Connect the CTS server to an HL7 feed of ADT records (Admissions, Discharges, and Transfers) from an EHR system.

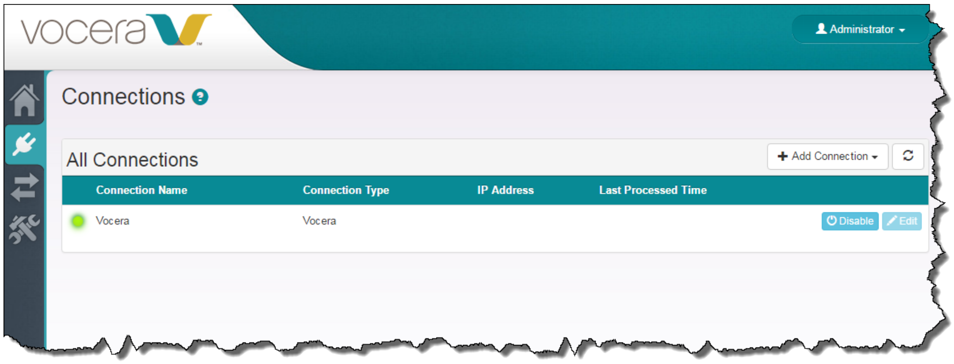
## How to Add an HL7 Connection

Specify the location of any normalizers and map files required to process HL7 data correctly, along with the additional parameters that allow the ADT system on the far end to communicate with the Vocera Care Team Sync server.

Use the following steps to connect the Vocera Care Team Sync server with your HL7 feed.

1. In the CTS Console, click the **Connections** icon ( 🔌 ) in the navigation bar on the left.

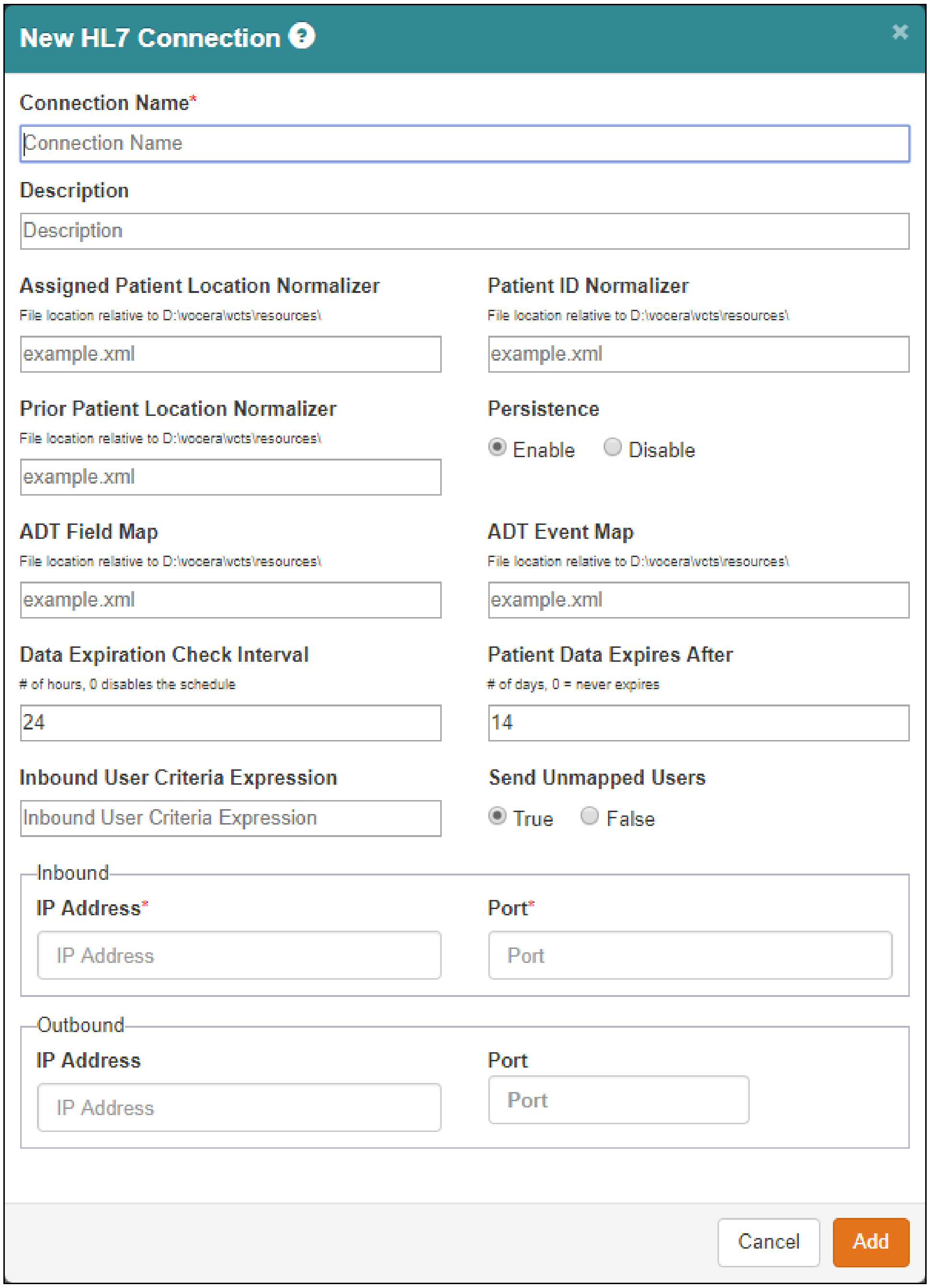
   The **Connections** screen appears.



| Connection Name | Connection Type | IP Address | Last Processed Time |
|---|---|---|---|
| Vocera | Vocera | | |

2. Click the **Add Connection** button and select **HL7** from the drop-down list that appears.

   The **New HL7 Connection** dialog box appears.



New HL7 Connection

Connection Name*
Connection Name

Description
Description

Assigned Patient Location Normalizer
File location relative to D:\vocera\vcts\resources\
example.xml

Patient ID Normalizer
File location relative to D:\vocera\vcts\resources\
example.xml

Prior Patient Location Normalizer
File location relative to D:\vocera\vcts\resources\
example.xml

Persistence
● Enable ○ Disable

ADT Field Map
File location relative to D:\vocera\vcts\resources\
example.xml

ADT Event Map
File location relative to D:\vocera\vcts\resources\
example.xml

Data Expiration Check Interval
# of hours, 0 disables the schedule
24

Patient Data Expires After
# of days, 0 = never expires
14

Inbound User Criteria Expression
Inbound User Criteria Expression

Send Unmapped Users
● True ○ False

Inbound
IP Address*
IP Address
Port*
Port

Outbound
IP Address
IP Address
Port
Port

**3.** Specify the HL7 connection parameters in the **New HL7 Connection** dialog box as follows:

| Field | Description |
|---|---|
| Connection Name | Specify a name for the connection to help you identify it in the future. This name appears as an identifier in other screens of the CTS Console. |
| Description | Optionally enter a more verbose description of the connection for your own information. The description does not appear on the **Connections** page, but it is visible for your reference if you edit the connection at a later time. |
| Assigned Patient Location Normalizer | The **Assigned Patient Location Normalizer** specifies the new location of the patient identified by the value in the **Patient ID Normalizer** field.<br>Enter the location of the normalizer relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the normalizers used by different connectors. |

| Field | Description |
|---|---|
| Patient ID Normalizer | The **Patient ID Normalizer** specifies the MRN or other patient ID whose location is tracked by the **Assigned Patient Location Normalizer** and the **Prior Patient Location Normalizer**<br>Enter the location of the normalizer relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the normalizers used by different connectors. |
| Prior Patient Location Normalizer | The **Prior Patient Location Normalizer** specifies the previous location of the patient identified by the value in the **Patient ID Normalizer** field.<br>Enter the location of the normalizer relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the normalizers used by different connectors. |
| Persistence | The value of the **Persistence** field determines whether assignments are actually made to the database by the HL7 connection.<br>• **Enabled** specifies that all assignments in the **Source of Truth** are made to the CTS database.<br>• **Disabled** specifies that assignments are not made to the CTS database; however, if the connection is successfully established, the HL7 feed is still received by the CTS system and recorded in the `hl7-*.log` files in the `\vocera\logs` directory.<br>Set **Persistence** to **Disabled** when you are testing to confirm that the connection to the HL7 endpoint is established correctly, and set **Persistence** to **Enabled** when you are ready to commit assignments. |
| ADT Field Map | The **ADT Field Map** associates standard fields in the HL7 feed with the field that CTS actually uses to retrieve its value.<br>Enter the location of the `adt-field-map.xml` file relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the files used by different connectors.<br>See *How to Update the ADT Field Map* on page 38 for additional information. |
| ADT Event Map | The **ADT Event Map** specifies in which HL7 field CTS can find admit, discharge, and transfer events.<br>Enter the location of the `adt-map.xml` file relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the files used by different connectors.<br>See *How to Update the ADT Event Map* on page 40 for additional information. |
| Data Expiration Check Interval | The **Data Expiration Check Interval** field specifies the interval in hours at which the system will check to see if there is expired data that needs to be purged. For example, set the interval to **24** to check for expired data once per day.<br>This field accepts values from 0-24, and the default is 24. If you set the value to zero (0), the data check never runs, and the data is never purged.<br>See *Managing HL7 Data Retention* on page 51 for more information. |
| Patient Data Expires After | The **Patient Data Expires After** field specifies the maximum number of days for which data is retained; after the expiration value is reached, the data is purged. For example, set the expiration value to **14** to keep a maximum of two weeks of data, or set the value to **31** to retain approximately one month of data.<br>This field accepts values from 0-365, and the default is 14. If you set the value to zero (0), the data never expires, and the data is never purged.<br>See *Managing HL7 Data Retention* on page 51 for more information. |
| Inbound User Criteria Expression | The **Inbound User Criteria Expression** optionally allows you to specify a template that maps "u-name" values (user IDs) from the external system to values in the **User ID** field in the Vocera Voice Server Administration Console.<br>Specify the full mapping expression in the **Inbound User Criteria Expression** field. You may create multi-line expressions on the single line in this field by using semicolon delimiters. |
| Send Unmapped Users | The **Send Unmapped Users** determines whether the Vocera Voice Server "optimistically" sends user IDs that are possibly unmapped to the external system. By default, this value is set to **True**, and the VS sends all users to the external system. |

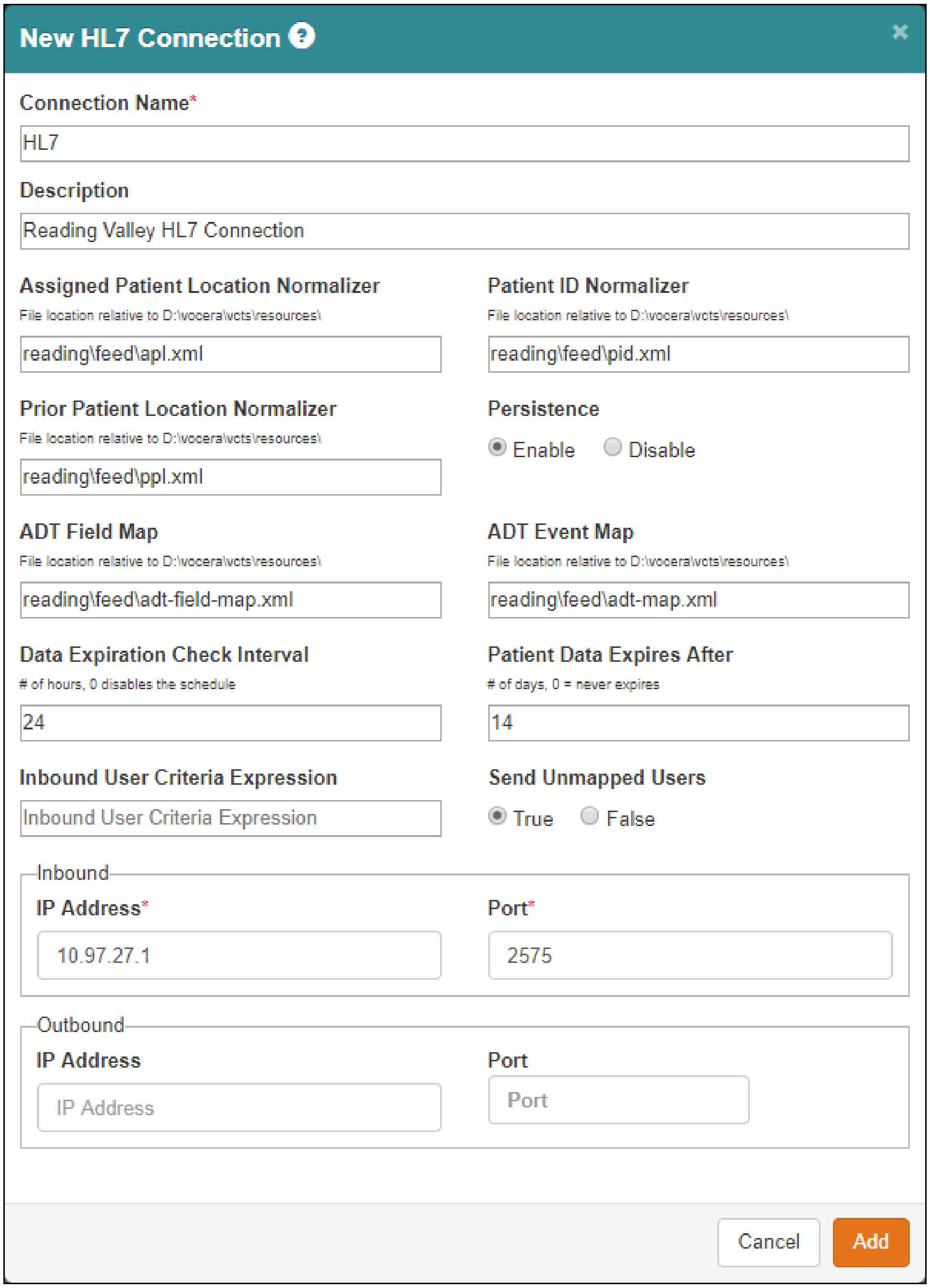
4. In the **Inbound** section of the **New HL7 Connection** dialog box, specify the following information:

| Field | Description |
|---|---|
| IP Address | Specify the IP address of the Vocera Care Team Sync server. |
| Port | Specify the port that the CTS server uses to listen for ADT messages. This port is typically 2575. |

5. In the **Outbound** section of the **New HL7 Connection** dialog box, specify the following information:

| Field | Description |
|---|---|
| IP Address | Specify the IP address or fully qualified domain name of the server that provides the HL7 feed (the far end). |
| Port | Specify the port on which the far end server listens for HL7 messages from the CTS server. |

When you are finished, the **New HL7 Connection** dialog box should look similar to the following:



New HL7 Connection

Connection Name*
HL7

Description
Reading Valley HL7 Connection

Assigned Patient Location Normalizer
File location relative to D:\vocera\vcts\resources\
reading\feed\apl.xml

Patient ID Normalizer
File location relative to D:\vocera\vcts\resources\
reading\feed\pid.xml

Prior Patient Location Normalizer
File location relative to D:\vocera\vcts\resources\
reading\feed\ppl.xml

Persistence
● Enable ○ Disable

ADT Field Map
File location relative to D:\vocera\vcts\resources\
reading\feed\adt-field-map.xml

ADT Event Map
File location relative to D:\vocera\vcts\resources\
reading\feed\adt-map.xml

Data Expiration Check Interval
# of hours, 0 disables the schedule
24

Patient Data Expires After
# of days, 0 = never expires
14

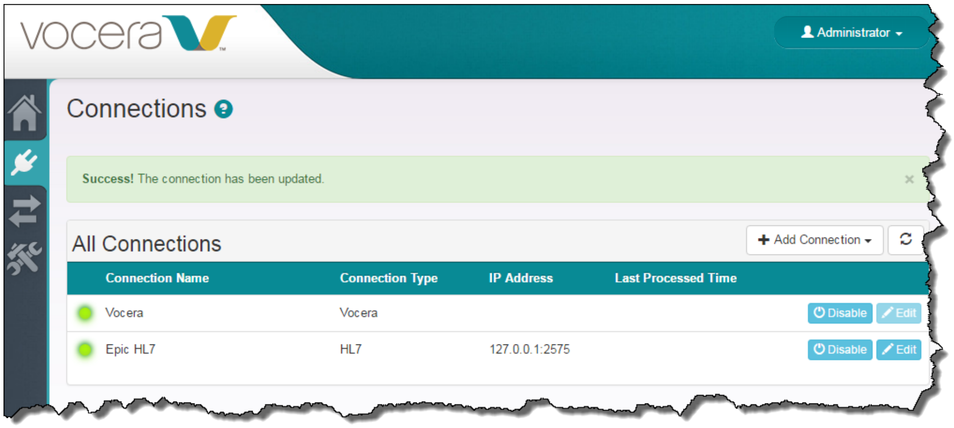
Inbound User Criteria Expression
Inbound User Criteria Expression

Send Unmapped Users
● True ○ False

Inbound
IP Address*
10.97.27.1
Port*
2575

Outbound
IP Address
IP Address
Port
Port

6. Click **Add**.
The **New HL7 Connection** dialog box closes, your changes are saved to the database, and the connection appears in the **All Connections** section of the **Connections** screen.



Connections

Success! The connection has been updated.

All Connections

| | Connection Name | Connection Type | IP Address | Last Processed Time | |
|---|---|---|---|---|---|
| ● | Vocera | Vocera | | | Disable Edit |
| ● | Epic HL7 | HL7 | 127.0.0.1:2575 | | Disable Edit |

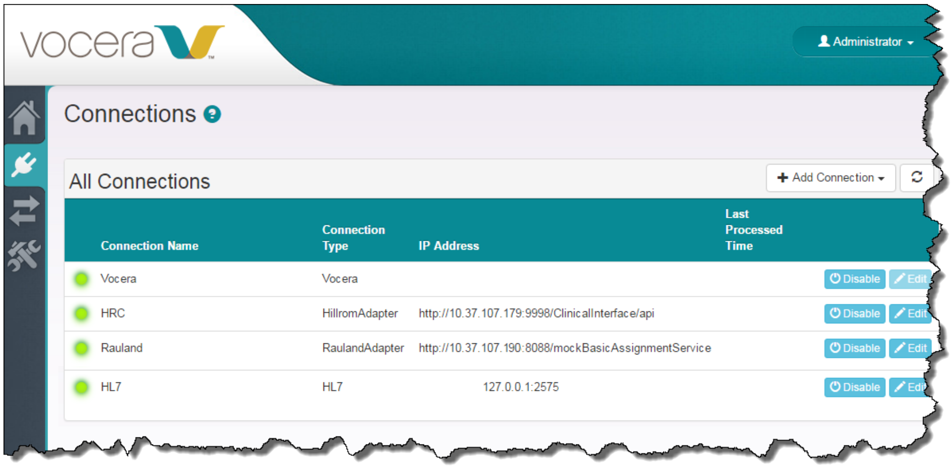
7. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

When the Vocera Care Team Sync-to-HL7 connection is set up correctly, a green light icon (enabled) appears next to the connection in the **Connection Name** column of the **Connections** screen.

In addition, you should start to see HL7 log files appear in the `\vocera\logs` directory of the Vocera Care Team Sync server. HL7 logs have the prefix `hl7` followed by the date/time. If HL7 logs are not being created, set **Persistence** to **Disabled** and the HL7 feed may be directed to the wrong server IP address or to the incorrect port.



Connections

All Connections

| | Connection Name | Connection Type | IP Address | Last Processed Time | |
|---|---|---|---|---|---|
| ● | Vocera | Vocera | | | Disable Edit |
| ● | HRC | HillromAdapter | http://10.37.107.179:9998/ClinicalInterface/api | | Disable Edit |
| ● | Rauland | RaulandAdapter | http://10.37.107.190:8088/mockBasicAssignmentService | | Disable Edit |
| ● | HL7 | HL7 | 127.0.0.1:2575 | | Disable Edit |

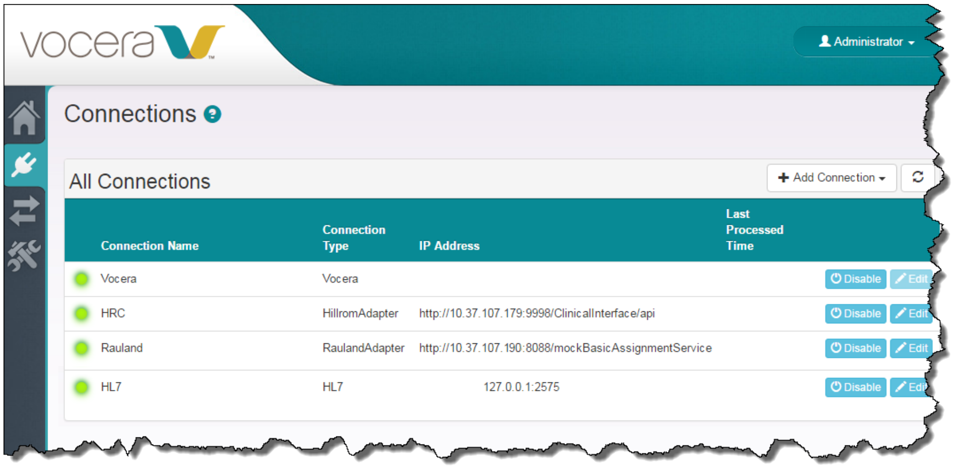
## How to Update an HL7 Connection

Edit the HL7 connection at any time to update the configuration parameters. After saving, you must restart the Vocera Vocera Care Team Sync service to allow changes to take effect.

Use the following steps to update the CTS connection to an HL7 endpoint.

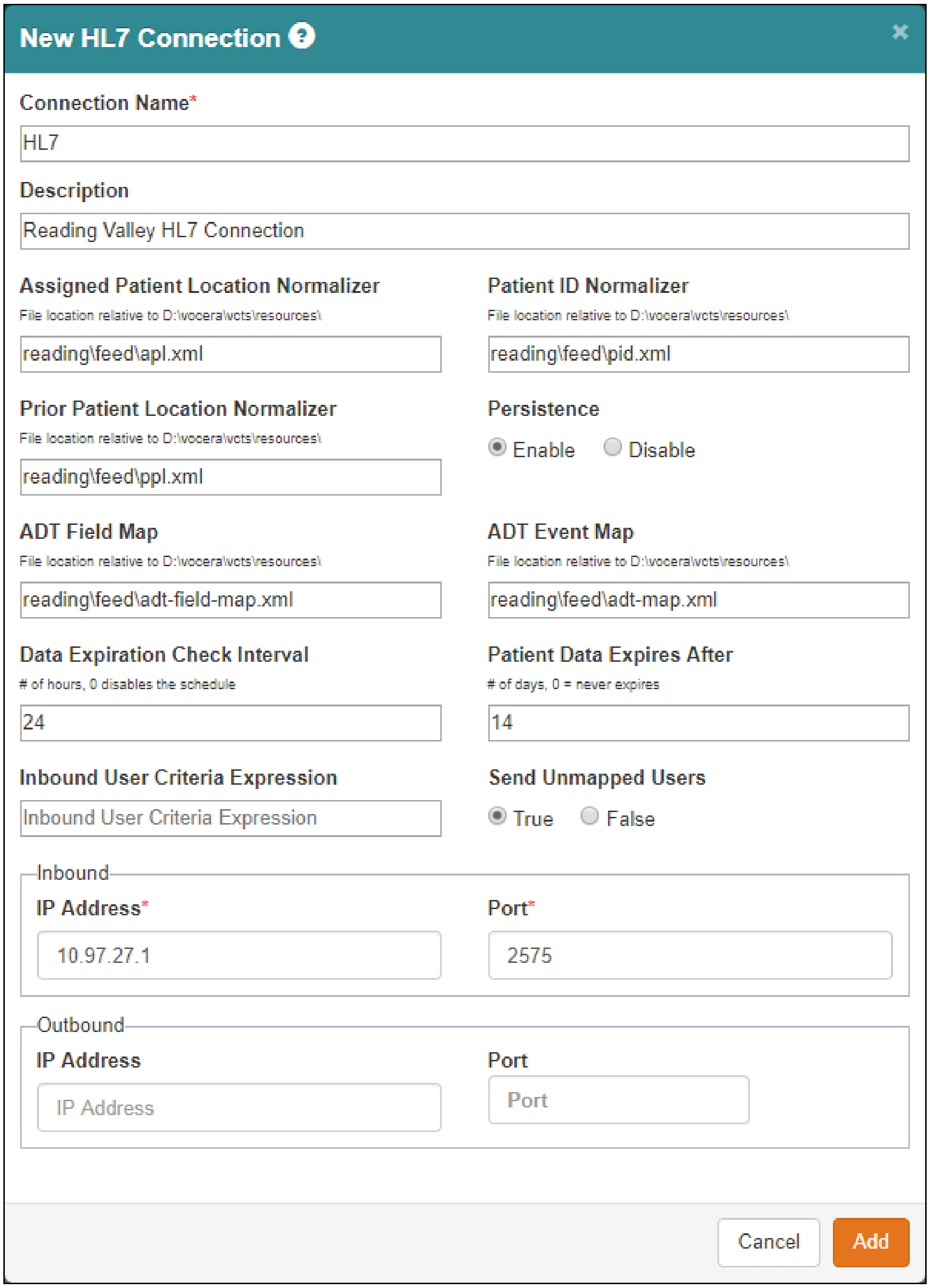
1. In the CTS Console, click the **Connections** icon ( ) in the navigation bar on the left.

   The **Connections** screen appears.



| | Connection Name | Connection Type | IP Address | Last Processed Time | |
|---|---|---|---|---|---|
| ● | Vocera | Vocera | | | Disable Edit |
| ● | HRC | HillromAdapter | http://10.37.107.179:9998/ClinicalInterface/api | | Disable Edit |
| ● | Rauland | RaulandAdapter | http://10.37.107.190:8088/mockBasicAssignmentService | | Disable Edit |
| ● | HL7 | HL7 | 127.0.0.1:2575 | | Disable Edit |

2. Click the **Edit** button to the right of the connection that you want to edit.
   The **Edit HL7 Connection** dialog box appears.



New HL7 Connection

Connection Name*

HL7

Description

Reading Valley HL7 Connection

Assigned Patient Location Normalizer

File location relative to D:\vocera\vcts\resources\

reading\feed\apl.xml

Patient ID Normalizer

File location relative to D:\vocera\vcts\resources\

reading\feed\pid.xml

Prior Patient Location Normalizer

File location relative to D:\vocera\vcts\resources\

reading\feed\ppl.xml

Persistence

● Enable ○ Disable

ADT Field Map

File location relative to D:\vocera\vcts\resources\

reading\feed\adt-field-map.xml

ADT Event Map

File location relative to D:\vocera\vcts\resources\

reading\feed\adt-map.xml

Data Expiration Check Interval

# of hours, 0 disables the schedule

24

Patient Data Expires After

# of days, 0 = never expires

14

Inbound User Criteria Expression

Inbound User Criteria Expression

Send Unmapped Users

● True ○ False

Inbound

IP Address*

10.97.27.1

Port*

2575

Outbound

IP Address

IP Address

Port

Port

3. Edit any fields that need to be updated.

   See the descriptions of the fields in *How to Add an HL7 Connection* on page 43.

4. Click **Save** to record your changes.

   The **Edit HL7 Connection** dialog box closes, and the connection changes are saved to the database.

5. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

### Managing HL7 Data Retention

Message data from an HL7 feed accumulates in the CTS database, and over time it affects the performance of the system. You can manage this performance impact by configuring purge criteria for the data retrieved by each HL7 connection.

The purge criteria specify the maximum number of days of HL7 message data to preserve in the database (**Patient Data Expires After**) and the frequency with which the system checks for expired data (**Data Expiration Check Interval**). You establish values for both criteria when you add or edit an HL7 connection; consequently, the settings may have independent values for each HL7 connection that you have in use, and you may maintain independent ranges of data for each connection.

The purge mechanism has an "offset" value that takes effect whenever the system is restarted; it prevents a purge from running while the system is still initializing. By default, the offset value is set to 5, providing a 5-minute buffer for the restarting system to stabilize before allowing a purge to run. If necessary, you can change this default value by updating the `hl7.purgeTask.startupOffsetMinutes` property of the `system_configuration` table in the MySQL Database.

Purged data is removed from the CTS database immediately; however, the data persists in the log files. If you need to restore an older range of data that has been purged from the database, you can use an HL7 "player" to re-play the original range of messages; the Vocera Care Team Sync application then receives the messages again and stores them in its database. Contact Vocera Customer Support for more information if you need to perform this procedure.

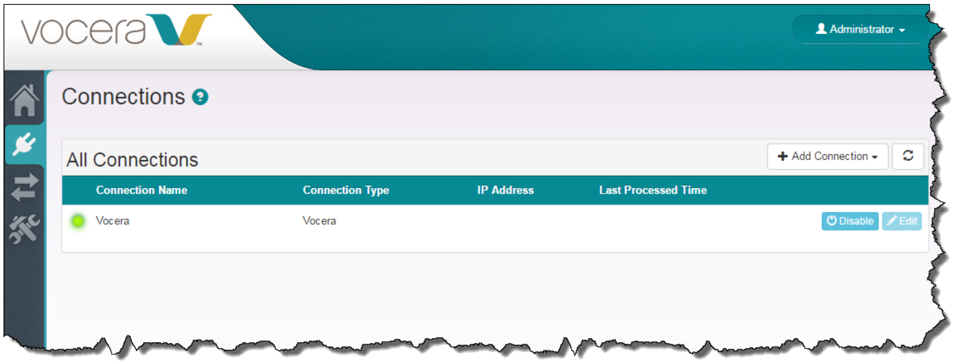
### How to Configure Database Purge Settings

Configure purge settings when you add or edit an HL7 connection.

Use the following steps to specify database purge settings for an HL7 connection:

1. In the CTS Console, click the **Connections** icon ( 🔌 ) in the navigation bar on the left.

   The **Connections** screen appears.



Connections

All Connections

| Connection Name | Connection Type | IP Address | Last Processed Time | |
|---|---|---|---|---|
| Vocera | Vocera | | | Disable Edit |

2. Do either of the following:
   - If you are updating an existing HL7 connection, edit it and set the **Patient Data Expires After** and **Data Expiration Check Interval** properties.

     See *How to Update an HL7 Connection* on page 48.
   - If you are creating a new HL7 connection, add it and set the **Patient Data Expires After** and **Data Expiration Check Interval** properties.

     See *How to Add an HL7 Connection* on page 43.
3. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

# Synchronizing Care Team Information

Synchronizing assignment data between other clinical staff assignment systems and the Vocera Voice Server or Staff Assignment provides a consistent, reliable source of information about a patient's current care team, so team members can easily identify and communicate with one another. Staff members can view this information in any assignment system and access it with Vocera work flows to leverage it for mobile communications.

When you synchronize assignments between systems, you need to decide which system has the most trusted assignments—Vocera or the external clinical assignment system. CTS refers to the most trusted system as the *source of truth*. The source of truth publishes assignments; other systems subscribe and update their assignments based on the values provided by the source of truth.

In certain situations, the Vocera Voice Server or Staff Assignment may have more current shift-based assignments than other clinical staff assignment systems. Other staff assignment systems may include accurate information about a patient's physicians, but they may also include nurses and other shift-based personnel who are no longer on the care team. In this situation, Staff Assignment may provide more accurate care team data for shift-based roles such as “Room 101 Nurse” and “Transport Team.”

In CTS, the source of truth is defined for the combination of site, unit, and role. This allows you to specify a granular source of truth. For example, another clinical staff assignment system may potentially be the source of truth for the "Emergency Department RN" role, while the Vocera Voice Server may potentially be used as the source of truth for a role such as "4 West PCA".

## How to Synchronize Care Team Information

When you synchronize assignments, you designate a **Source of Truth**—either Vocera or the foreign clinical staff assignment system—for patient care team role in each location of a unit, and then synchronize data with another system based on the values in the specified **Source of Truth**.

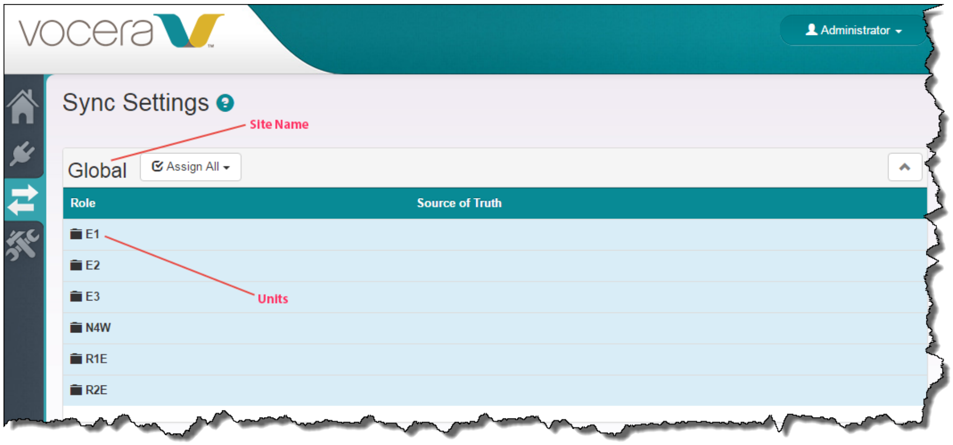
Before you begin, make sure that you have configured at least one connection and set up one or more units. See *Adding and Updating Connections* on page 27 and *How to Create a Unit* on page 74.

The output from the **Vocera Group Role Normalizer** (defined when you create a unit) appears as the site/unit/role combinations on the **Sync Settings** page.

**Note:** The output from the **Vocera Group Location Normalizer** (also defined when you create a unit) appears only in the `location` table, not in the user interface. This allows you to see the source of truth for any role at a glance; however, it also means that the source of truth is applied to every location within that role.

Use the following steps to synchronize care team data between your foreign assignment system and Vocera.

1. In the CTS Console, click the **Sync Settings** icon ( ⇄ ) in the navigation bar on the left.
   The **Sync Settings** page appears.



Sync Settings

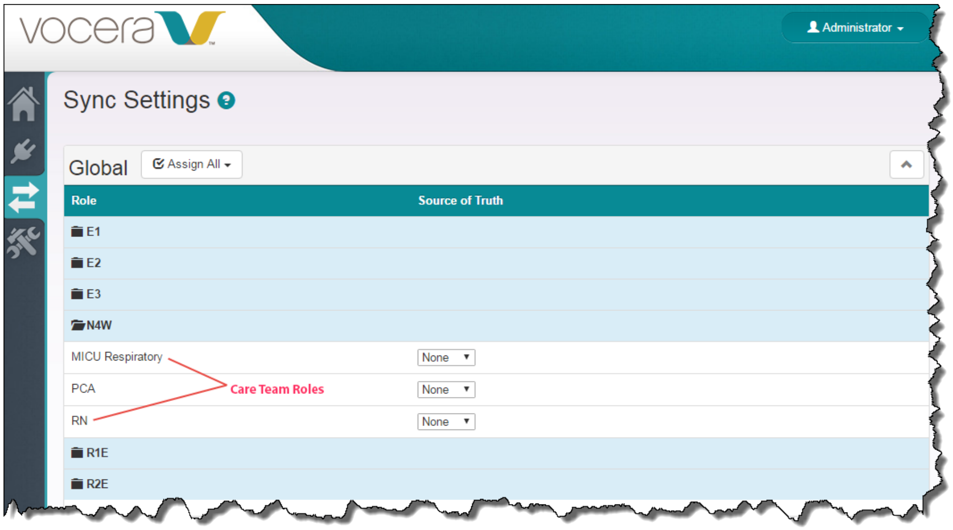
Global Assign All

| Role | Source of Truth |
|---|---|
| E1 | |
| E2 | |
| E3 | |
| N4W | |
| R1E | |
| R2E | |

The **Sync Settings** page displays the name of each site for which you have created units in an expanding/ collapsing list. The **Role** column within each site contains an entry for each unit you have created with an expanding folder icon next to it.

**Note:** If you have not yet created any units, the **Sync Settings** page displays a message stating "No sites have been set up yet".

2. Click the folder icon ( ■ ) next to one of the units.
   The row expands to display the care team roles in use by that unit.



Sync Settings

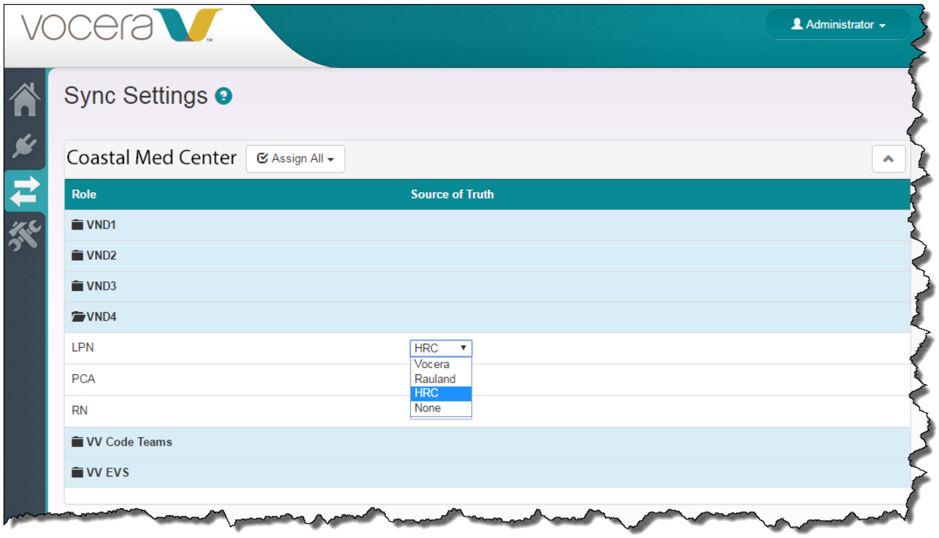
Global Assign All

| Role | Source of Truth |
|---|---|
| E1 | |
| E2 | |
| E3 | |
| N4W | |
| MICU Respiratory | None |
| PCA | None |
| RN | None |
| R1E | |
| R2E | |

The **Source of Truth** column displays the name of the data source that is used to synchronize the care team information for each role—either **None**, **Vocera**, or the **Connection Name** that you specified when you added the connection.

The value you specify in the **Source of Truth** column controls the direction of the synchronization between Vocera and your foreign staff assignment system. By default, the value is set to **None** to indicate that no synchronization is occurring.

3. Do either of the following:
   - Use the list that appears next to each role in the **Source of Truth** column and specify the data source that has the most current assignments for that unit and role combination.



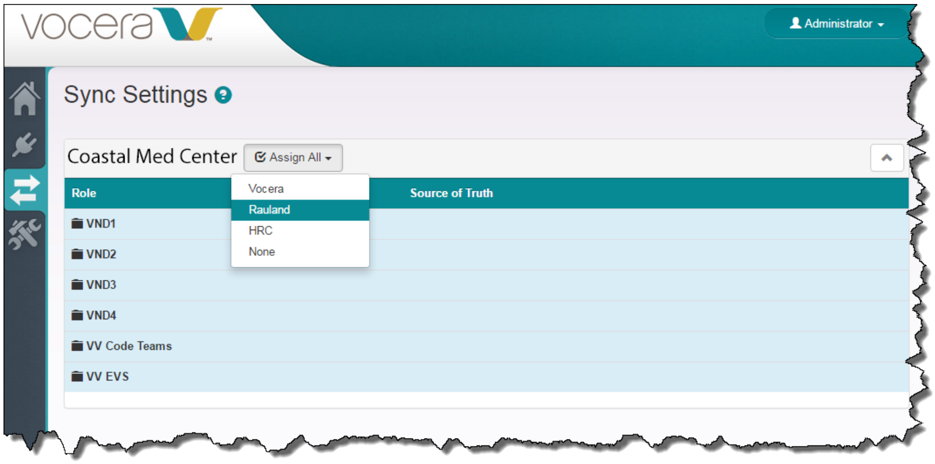
Sync Settings

Coastal Med Center

| Role | Source of Truth |
|---|---|
| VND1 | |
| VND2 | |
| VND3 | |
| VND4 | |
| LPN | HRC |
| PCA | |
| RN | |
| VV Code Teams | |
| VV EVS | |

Choosing this option allows you to specify a different source of truth for each unit and role combination in the site.

- Use the **Assign All** list that appears next to the site name and specify the data source that has the most current assignments.



Sync Settings

Coastal Med Center

| Role | Source of Truth |
|---|---|
| VND1 | |
| VND2 | |
| VND3 | |
| VND4 | |
| VV Code Teams | |
| VV EVS | |

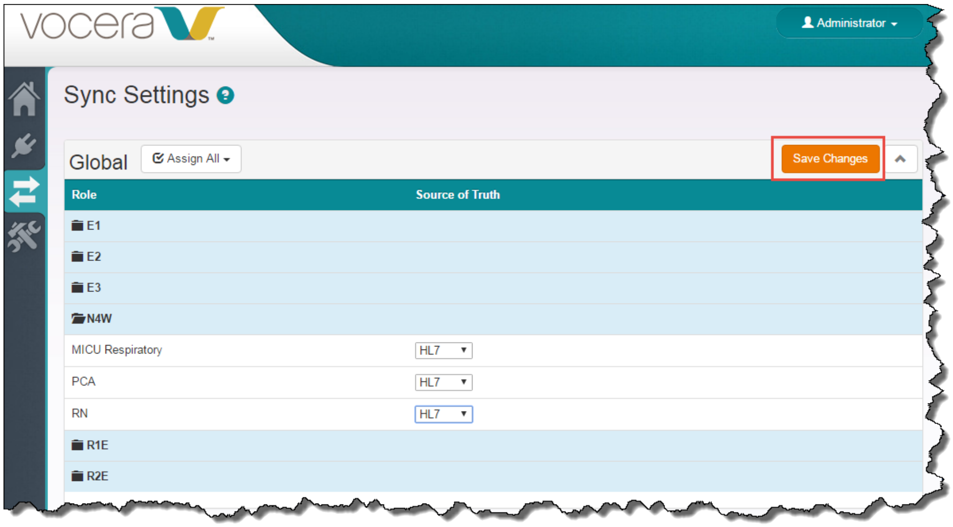
Choosing this option sets the source of truth for each unit and role combination in the site to the same data source.

**Important:** *Do not* specify a **Source of Truth** unless you are certain that you are ready to update either the data in your foreign assignment system or the data used by the Vocera Voice Server.

- Setting the **Source of Truth** to **Vocera** indicates that you believe the Vocera care assignments are more accurate than the foreign system's assignments; this setting will cause CTS to update the foreign system with the Vocera assignments.
- Setting the **Source of Truth** to the name of a foreign system indicates that you believe the foreign system's care assignments are more accurate than the Vocera assignments; this setting will cause CTS to update your Vocera database with the assignments from the foreign system.

If you are uncertain which assignments to trust, leave the **Source of Truth** set to **None**, which will prevent any synchronization from occurring.

When you change the **Source of Truth** for any role, the **Save Changes** button appears.



| Role | Source of Truth |
|---|---|
| E1 | |
| E2 | |
| E3 | |
| N4W | |
| MICU Respiratory | HL7 |
| PCA | HL7 |
| RN | HL7 |
| R1E | |
| R2E | |

4. When you are finished setting the **Source of Truth** for each role, click the **Save Changes** button.
   The changes are saved in the database.
5. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

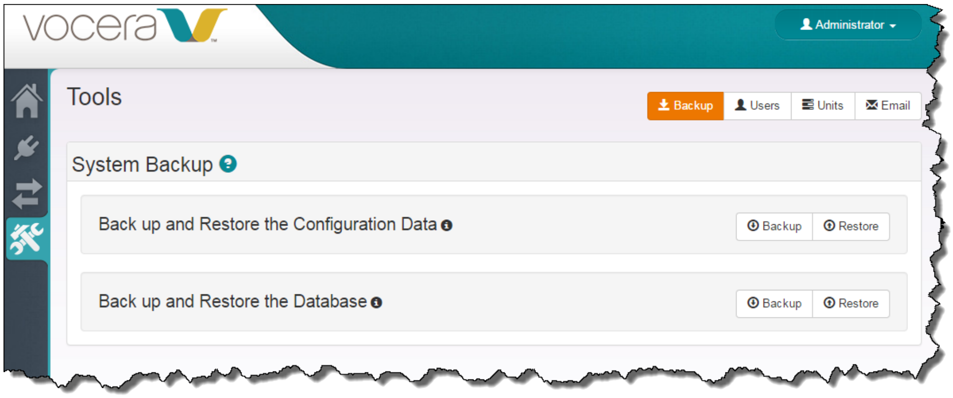
Your synchronization settings are saved, and CTS begins synchronizing data in the direction you specified by the **Source of Truth**.

**Note:** If you do not restart the Vocera CTS service, changes to the source of truth will take effect over time (for example, when shift change occurs). Restarting the service is recommended, but not required.

# Using CTS Tools

The **Tools** page in the CTS Console allows you to perform various administrative and ancillary tasks for your CTS deployment.

Navigate to the **Tools** page by clicking the **Tools** ( 🛠 ) icon in the navigation bar on the left side of the CTS Console.



Buttons in the top-right section of the CTS Console provide access to each of the tools. By default, the **System Backup** tab is displayed initially.

## Backing Up and Restoring Data

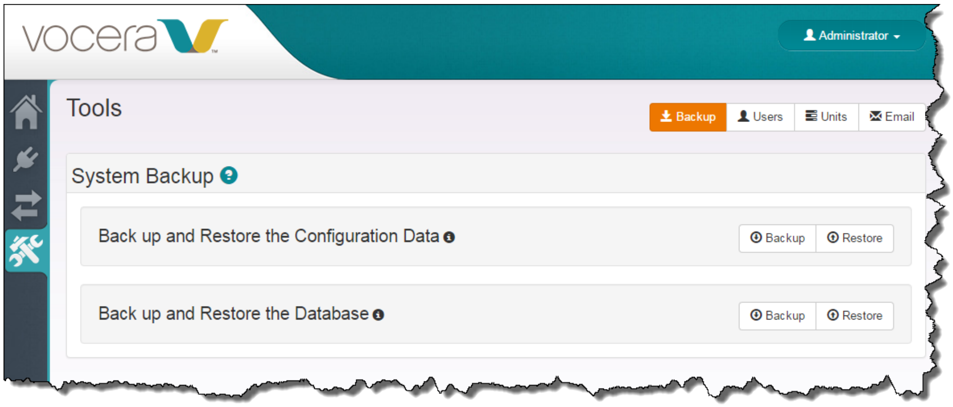
CTS allows you to back up your entire database (including data from your external connection) or just the configuration data for your system. Similarly, you can restore either the configuration data or the entire database from a backup at any time.

### How to Back Up Configuration Data

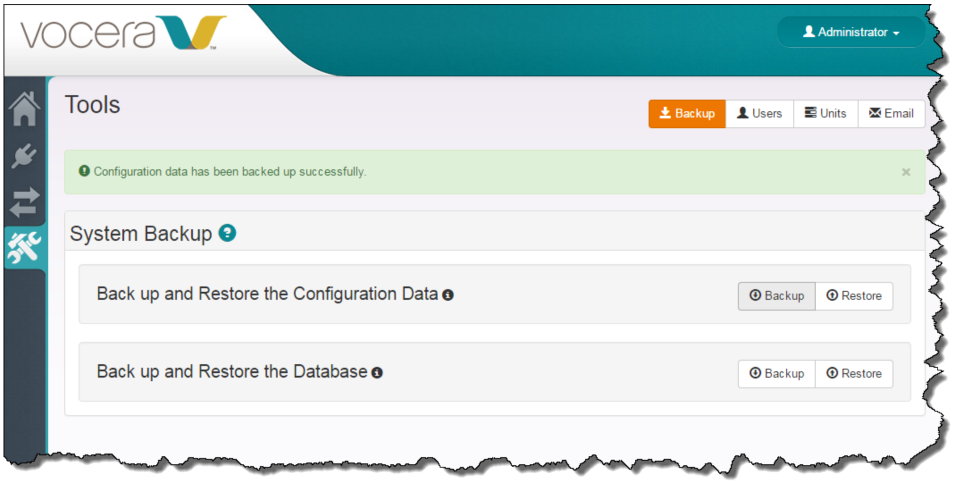
Back up your configuration data separately if you want to be able to restore your system configuration without restoring data received from an external connection.

Use the following steps to back up configuration data:

1. In the CTS Console, click the **Tools** icon ( 🛠 ) in the navigation bar on the left.
   The **Tools** page appears.



2. Click the **Backup** button in the **Back up and Restore the Configuration Data** section.
   The CTS Console displays the message, "Configuration data has been backed up successfully" when it is complete.


Configuration data has been backed up successfully.

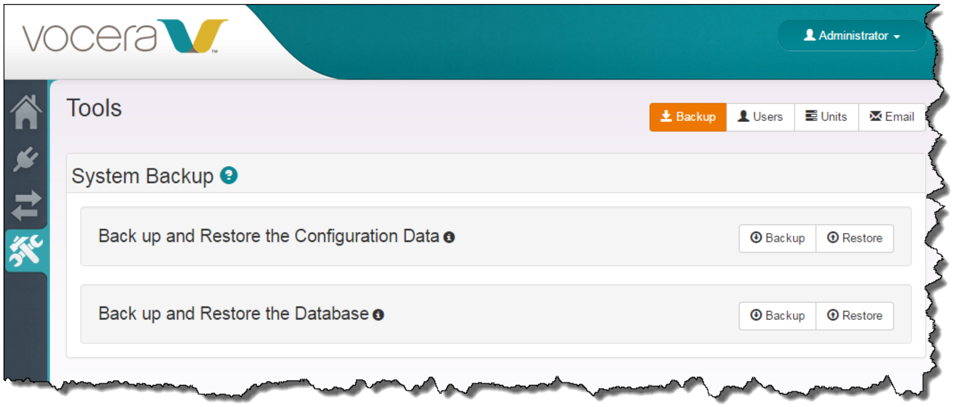
CTS creates a `.zip` file in the `\vocera\vcts\backup\configuration` directory, embedding the current date and time in the filename.

## How to Back Up the CTS Database

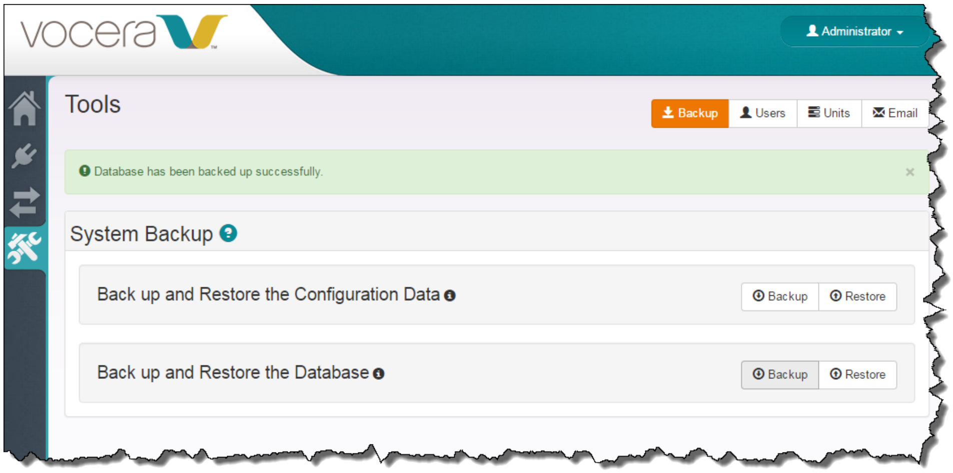
Back up the entire CTS database if you want the ability to restore all your data, including data received from an external connection.

Use the following steps to back up the entire database.

1. In the CTS Console, click the **Tools** icon ( 🛠 ) in the navigation bar on the left.
   The **Tools** page appears.



2. Click the **Backup** button in the **Back up and Restore the Database** section.
   The CTS Console displays the message, "Database has been backed up successfully" when it is complete.



Database has been backed up successfully.

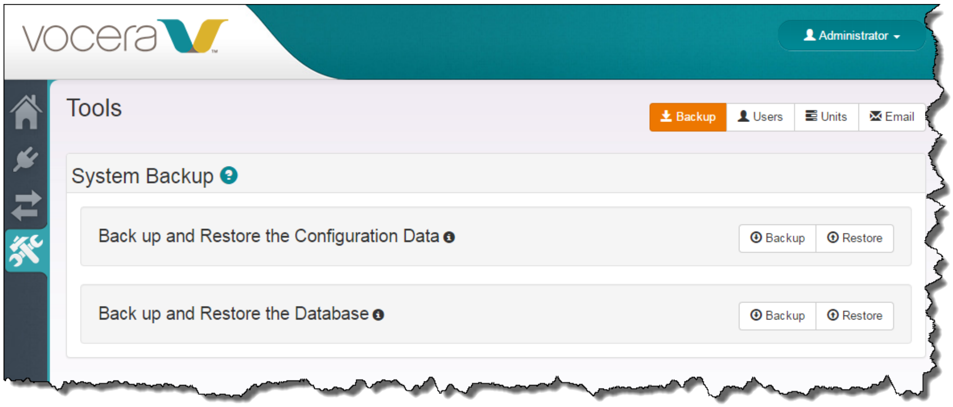
CTS creates a `.zip` file in the `\vocera\vcts\backup\database` directory, embedding the current date and time in the filename.

## How to Restore Configuration Data

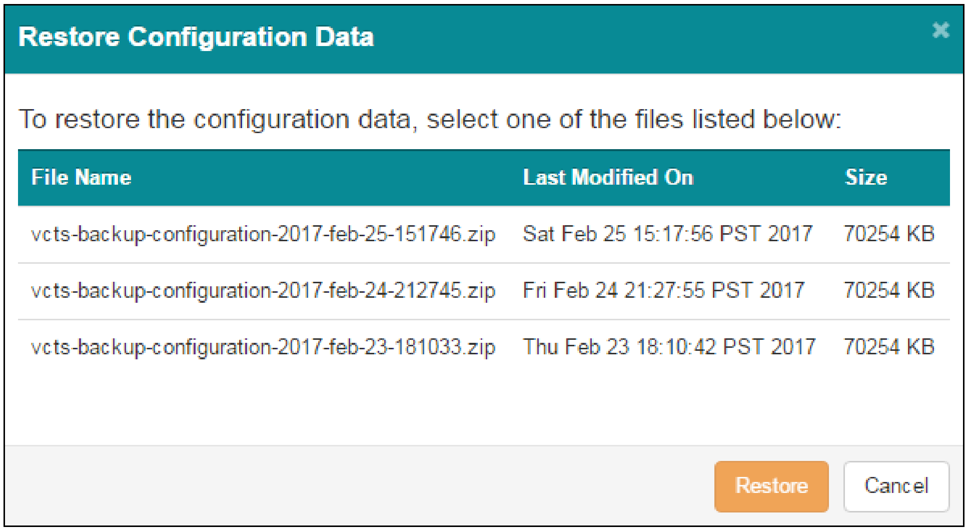
Optionally restore only your system configuration without restoring data received from another clinical staff assignment system.

Use the following steps to restore configuration data:

1. In the CTS Console, click the **Tools** icon ( 🛠 ) in the navigation bar on the left.
   The **Tools** page appears.



2. Click the **Restore** button in the **Back up and Restore the Configuration Data** section.
   The **Restore Configuration Data** dialog box appears.

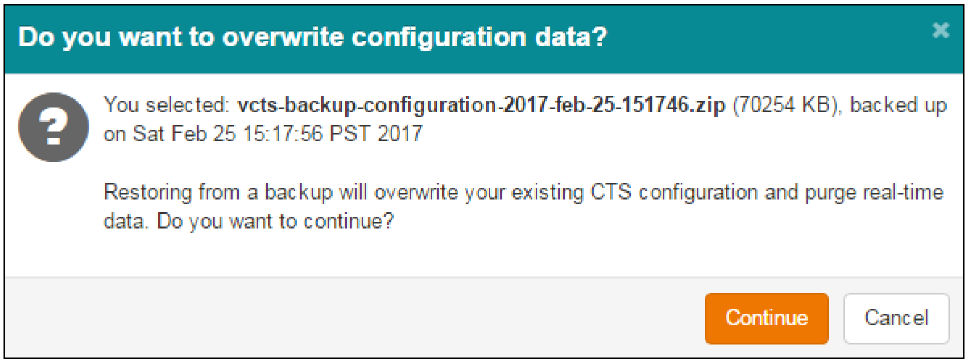


Restore Configuration Data

To restore the configuration data, select one of the files listed below:

| File Name | Last Modified On | Size |
|---|---|---|
| vcts-backup-configuration-2017-feb-25-151746.zip | Sat Feb 25 15:17:56 PST 2017 | 70254 KB |
| vcts-backup-configuration-2017-feb-24-212745.zip | Fri Feb 24 21:27:55 PST 2017 | 70254 KB |
| vcts-backup-configuration-2017-feb-23-181033.zip | Thu Feb 23 18:10:42 PST 2017 | 70254 KB |

3. Select a file and click **Restore**.
   A confirmation dialog box warns that you are overwriting configuration data.

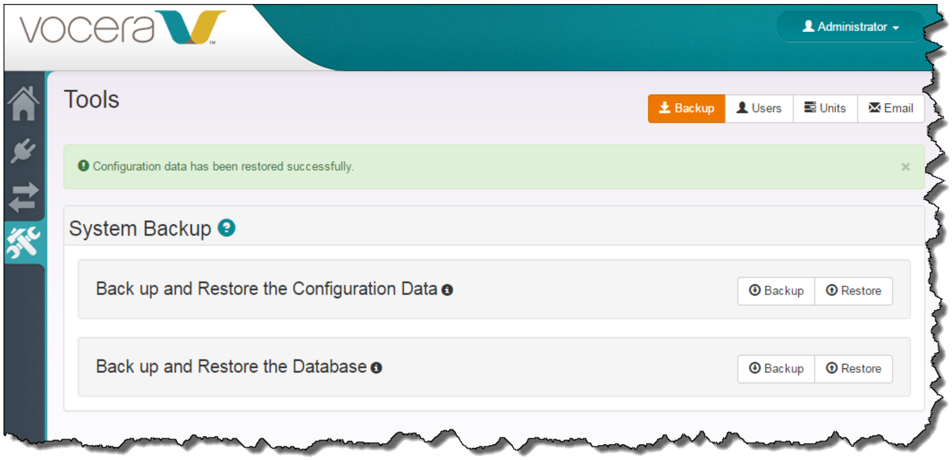


Do you want to overwrite configuration data?

You selected: **vcts-backup-configuration-2017-feb-25-151746.zip** (70254 KB), backed up on Sat Feb 25 15:17:56 PST 2017

Restoring from a backup will overwrite your existing CTS configuration and purge real-time data. Do you want to continue?

4. Click **Continue**.
   The CTS Console displays the message, "Configuration data has been restored successfully" when it is complete.



Configuration data has been restored successfully.

System Backup

Back up and Restore the Configuration Data

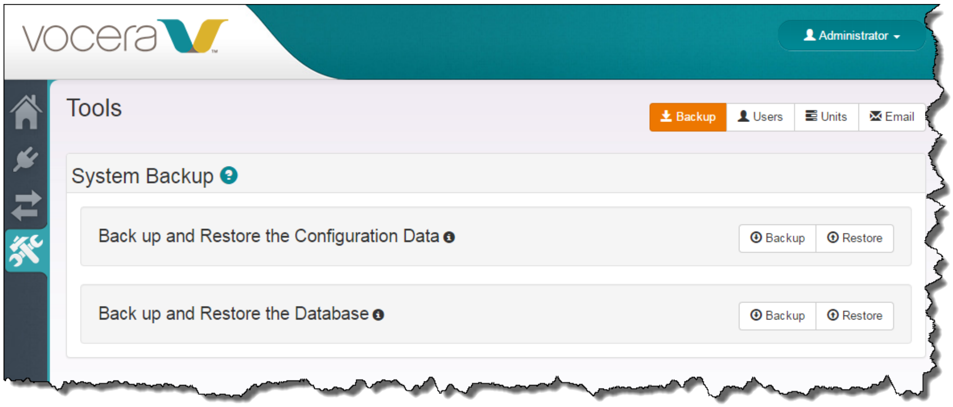
Back up and Restore the Database

### How to Restore the CTS Database

Optionally restore your entire database, including data received from another clinical staff assignment system.

Use the following steps to restore your database:

1. In the CTS Console, click the **Tools** icon ( 🛠 ) in the navigation bar on the left.
   The **Tools** page appears.

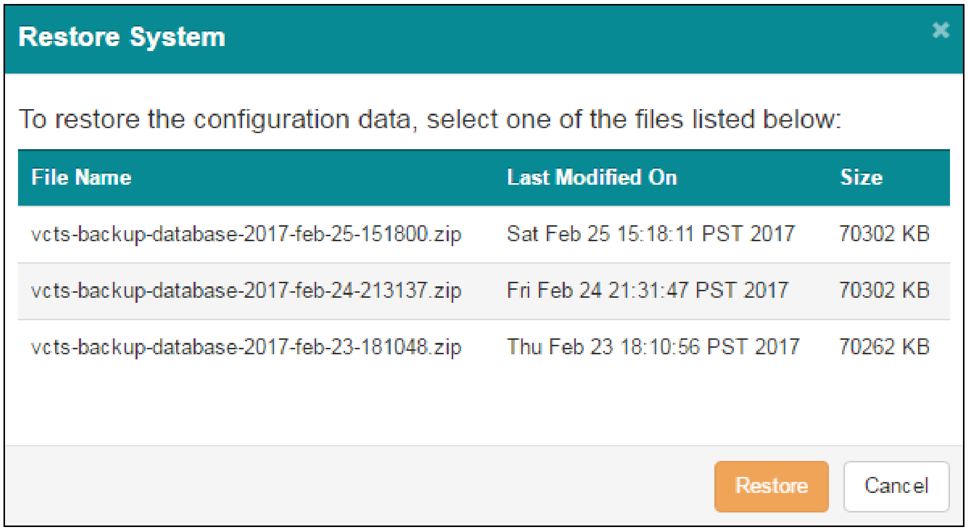


System Backup

Back up and Restore the Configuration Data

Back up and Restore the Database

2. Click the **Restore** button in the **Back up and Restore the Database** section.
   The **Restore System** dialog box appears.



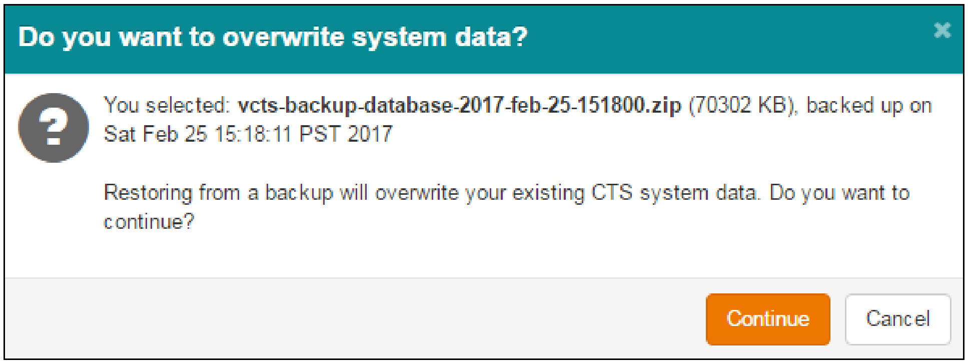
Restore System

To restore the configuration data, select one of the files listed below:

| File Name | Last Modified On | Size |
|---|---|---|
| vcts-backup-database-2017-feb-25-151800.zip | Sat Feb 25 15:18:11 PST 2017 | 70302 KB |
| vcts-backup-database-2017-feb-24-213137.zip | Fri Feb 24 21:31:47 PST 2017 | 70302 KB |
| vcts-backup-database-2017-feb-23-181048.zip | Thu Feb 23 18:10:56 PST 2017 | 70262 KB |

3. Select a file and click **Restore**.

   A confirmation dialog box warns that you are overwriting system data.



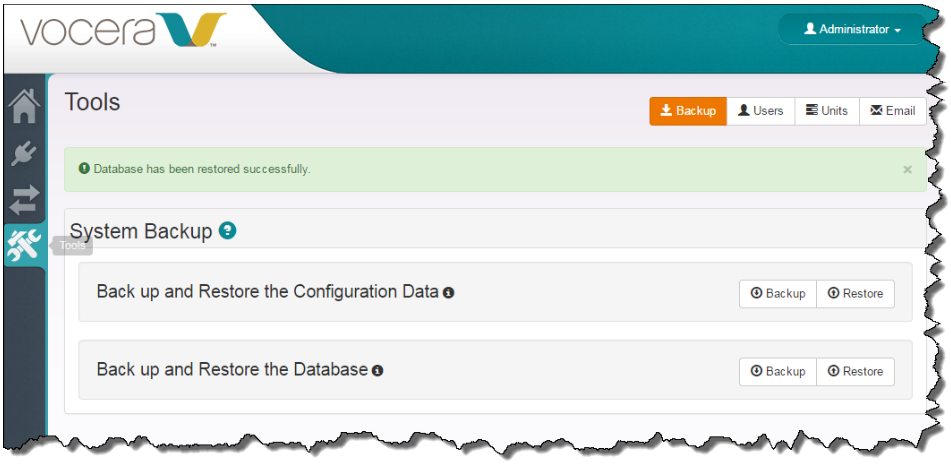
**Do you want to overwrite system data?**

You selected: **vcts-backup-database-2017-feb-25-151800.zip** (70302 KB), backed up on Sat Feb 25 15:18:11 PST 2017

Restoring from a backup will overwrite your existing CTS system data. Do you want to continue?

4. Click **Continue**.

   The CTS Console displays the message, "Database has been restored successfully" when it is complete.



Tools

Database has been restored successfully.

System Backup

Back up and Restore the Configuration Data

Back up and Restore the Database

## Mapping Users

Because the core functionality of the Vocera Care Team Sync product is to synchronize assignments between the Vocera Voice Server and a foreign clinical assignment system, you must associate or "map" the u-name values (user IDs) that are used by the two systems with each other.

When the user IDs in the two systems vary systematically, you can accomplish this task by creating a single expression in the **Inbound User Criteria Expression** field of the **New** or **Update Connection** dialog box. For example, if the foreign assignment system uses numeric values as user IDs and the Vocera Voice Server uses the same numeric value prefixed by a "u", the variation is systematic and may be captured by an expression in this connection definition.

**Note:** When the user IDs in the two systems are identical, you are required to use the **Inbound User Criteria Expression** field to map the users.

In other situations, the user ID convention implemented by the two systems is so different that no systematic mapping is possible. For example, if the foreign assignment system uses numeric values as user IDs and the Vocera Voice Server uses alphabetic user IDs based on the user name itself, the variation is so extreme that you will likely be unable to describe it in an expression.

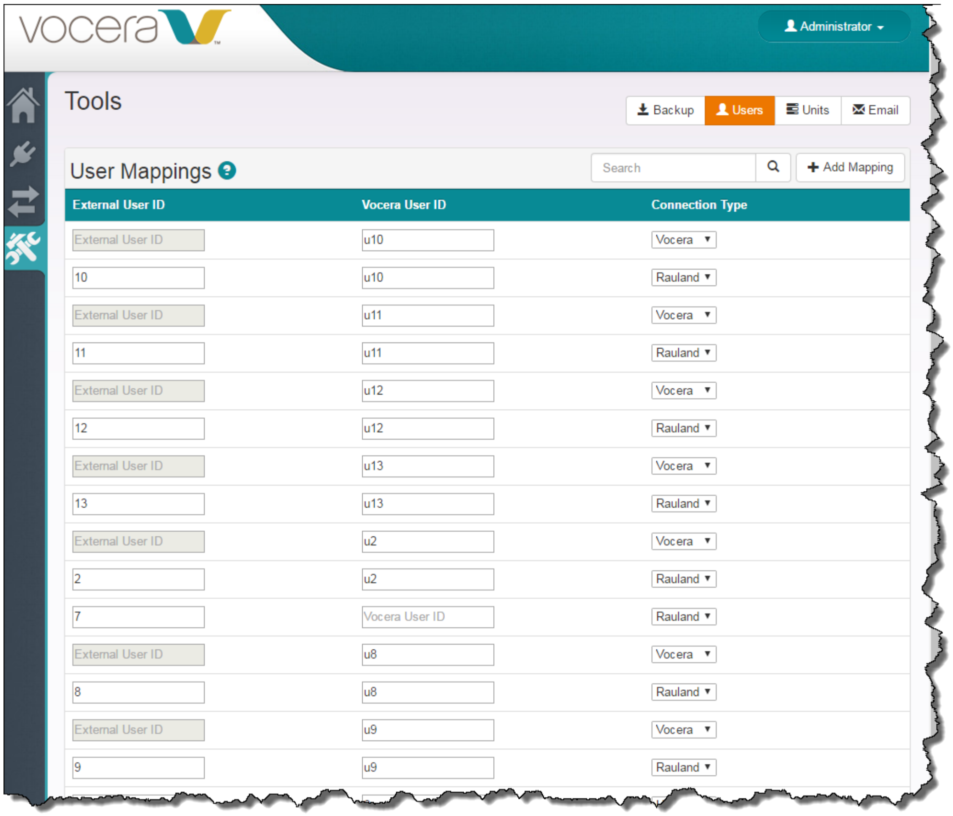
In all situations, you must map the user IDs in the two systems: either systematically through an expression when you define the connection, or manually through the CTS Console. In some cases, you may be able to map most users by way of the expression in the connection definition and use the **User Mappings** page to accommodate any exceptions to this general mapping rule.

**Note:** If you are comfortable with SQL, you may also create and import a CSV file that maps the user IDs in the two systems. If you choose this approach, create a three-column CSV file with values for the `voceraUserId`, `externalId`, and `assignmentClientSEQ` columns and use the **Table Data Import Wizard** in **MySQL Workbench** to import the values. See the **MySQL Workbench** documentation for further information.

## About User Mappings

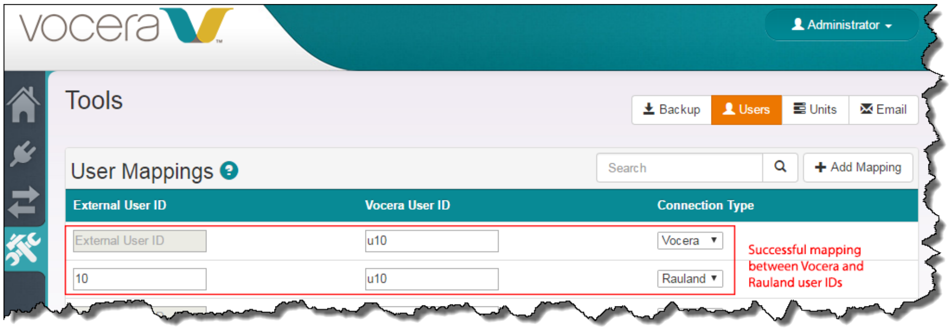
The **User Mappings** page in the **Tools** section of the CTS Console displays all the user mappings that already exist in the two systems. In addition, if you set **Send Unmapped Users** in the connection definition to true, CTS creates a new row for each unmapped user in the subscribed system, allowing you to map the user manually.

In the following illustration, the **User Mappings** page displays mappings for a Rauland connection where the Vocera Voice Server and Rauland Responder have user IDs that vary systematically—that is, the Rauland system uses numeric user IDs, and the VS uses the same ID prefixed with a "u".



| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| External User ID | u10 | Vocera |
| 10 | u10 | Rauland |
| External User ID | u11 | Vocera |
| 11 | u11 | Rauland |
| External User ID | u12 | Vocera |
| 12 | u12 | Rauland |
| External User ID | u13 | Vocera |
| 13 | u13 | Rauland |
| External User ID | u2 | Vocera |
| 2 | u2 | Rauland |
| 7 | Vocera User ID | Rauland |
| External User ID | u8 | Vocera |
| 8 | u8 | Rauland |
| External User ID | u9 | Vocera |
| 9 | u9 | Rauland |

A pair of rows (one for a **Connection Type** of Vocera and one for a foreign end point) with the same **Vocera User ID** defines a mapping for a single user, as shown in the following illustration.

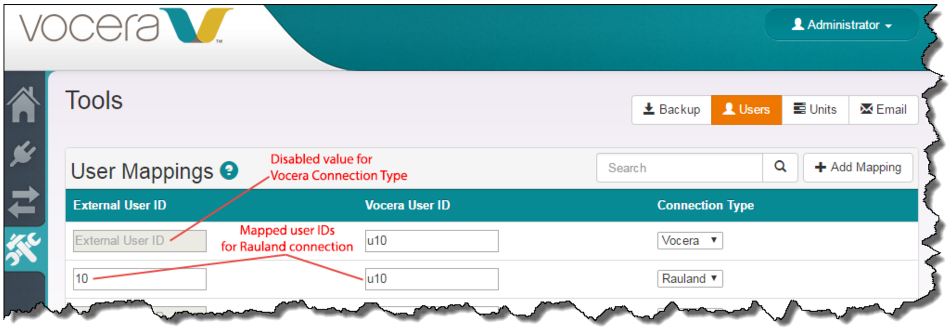


| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| External User ID | u10 | Vocera |
| 10 | u10 | Rauland |

Successful mapping between Vocera and Rauland user IDs

When the **Connection Type** is **Vocera**, the **Vocera User ID** column always displays the actual VS user ID, and the **External User ID** column always displays a disabled value. The user mapping is always in the context of the Vocera user, so the **External User ID** will always be disabled and without a value when the **Connection Type** is **Vocera**. When the **Connection Type** is set to the foreign end point, the values in the **Vocera User ID** column and the **External User ID** column specify the mapping that is currently being used.

Both the **External User ID** and the **Vocera User ID** must be unique for a given connection.

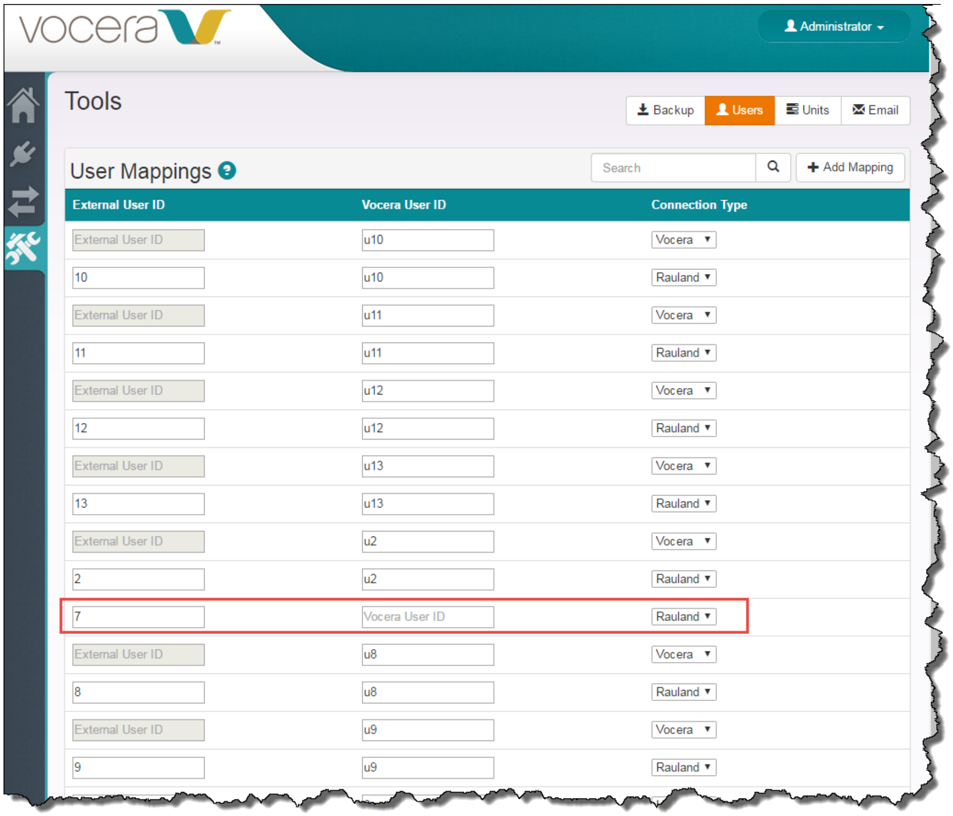


Disabled value for Vocera Connection Type

Mapped user IDs for Rauland connection

| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| External User ID | u10 | Vocera |
| 10 | u10 | Rauland |

When a user is not successfully mapped, the **External User ID** column displays a value and the **Vocera User ID** column displays a null value with place holder text, as shown in the following illustration.



| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| External User ID | u10 | Vocera |
| 10 | u10 | Rauland |
| External User ID | u11 | Vocera |
| 11 | u11 | Rauland |
| External User ID | u12 | Vocera |
| 12 | u12 | Rauland |
| External User ID | u13 | Vocera |
| 13 | u13 | Rauland |
| External User ID | u2 | Vocera |
| 2 | u2 | Rauland |
| 7 | Vocera User ID | Rauland |
| External User ID | u8 | Vocera |
| 8 | u8 | Rauland |
| External User ID | u9 | Vocera |
| 9 | u9 | Rauland |

You can use the **User Mappings** page to provide a mapping for these unmapped users. See *How to Edit a User Mapping* on page 67 for additional information.

Vocera Care Team Sync uses three criteria to attempt mapping of users between itself and a foreign end point, in the following order of preference:

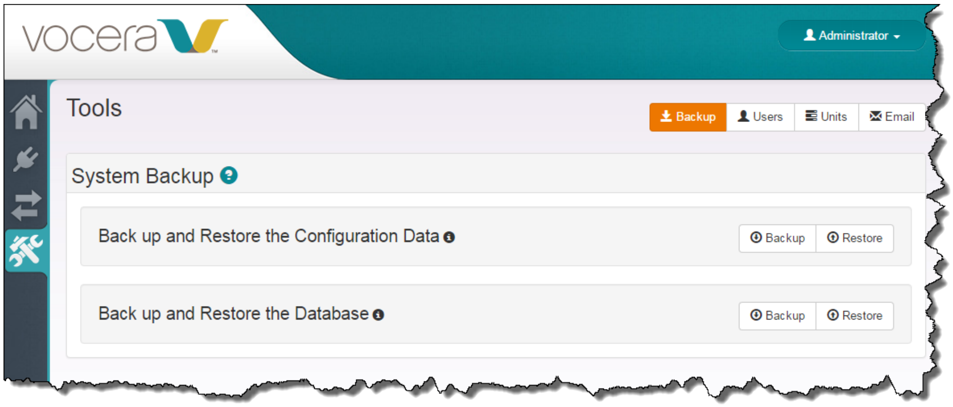
- An explicit combination of user IDs that is provided on the **User Mappings** page
- The **Inbound User Criteria Expression** field of the **New** or **Update Connection** dialog box
- The user name itself, if there is an exact and unique match with the first and last names

## How to Search for Users

You can search for a user ID on the **User Mappings** page to assist your navigation. When you search **User Mappings**, CTS filters the display of the page and shows only rows with user IDs that contain the value you provided.

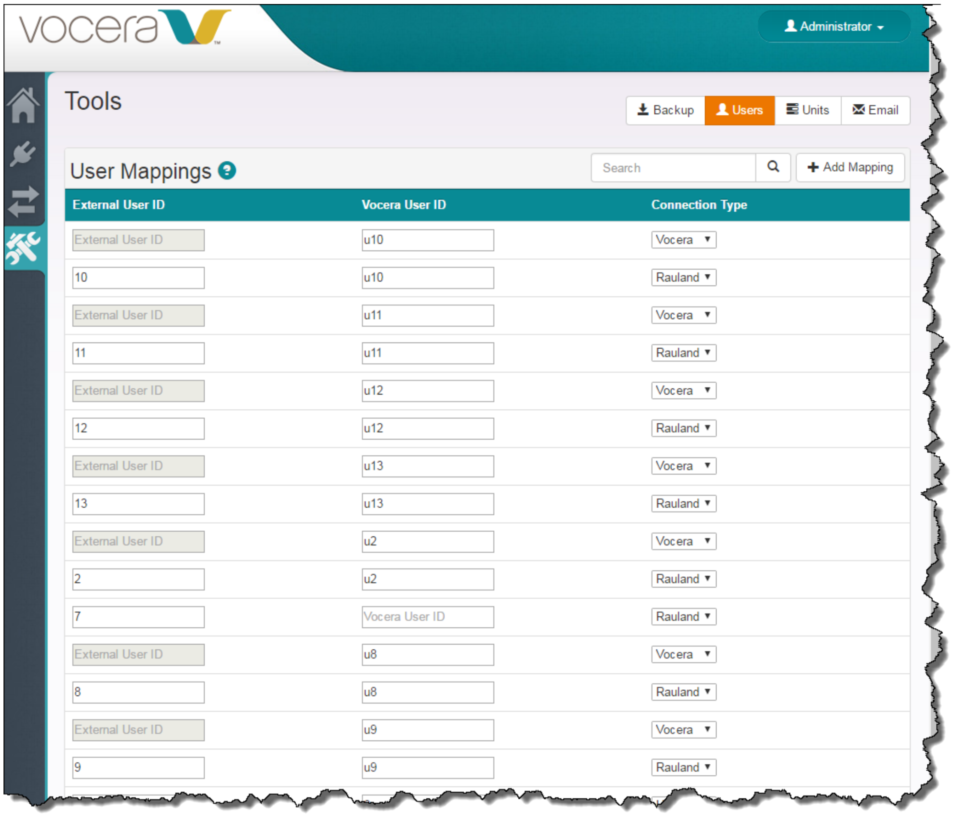
To search for a user ID:

1. In the CTS Console, click the **Tools** icon ( ) in the navigation bar on the left.
   The **Tools** page appears.



2. Click the **Users** tab in the top-right section of the page.

   The **User Mappings** page appears.

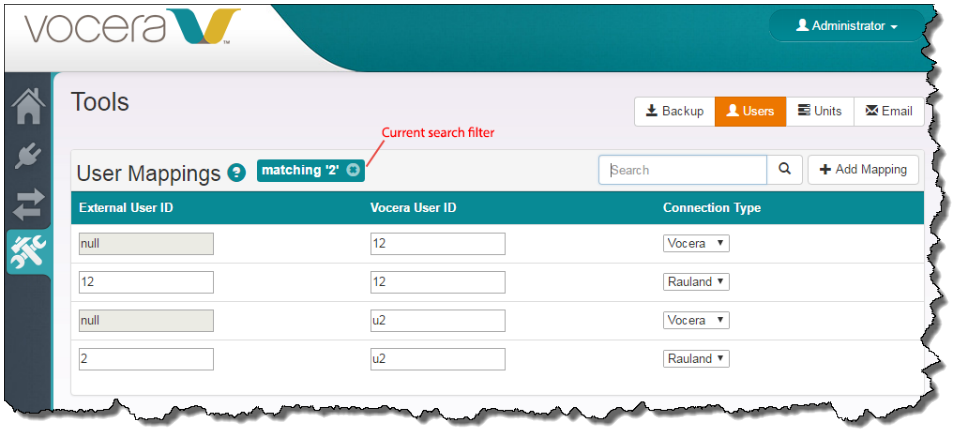


| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| External User ID | u10 | Vocera |
| 10 | u10 | Rauland |
| External User ID | u11 | Vocera |
| 11 | u11 | Rauland |
| External User ID | u12 | Vocera |
| 12 | u12 | Rauland |
| External User ID | u13 | Vocera |
| 13 | u13 | Rauland |
| External User ID | u2 | Vocera |
| 2 | u2 | Rauland |
| 7 | Vocera User ID | Rauland |
| External User ID | u8 | Vocera |
| 8 | u8 | Rauland |
| External User ID | u9 | Vocera |
| 9 | u9 | Rauland |

3. Enter any part of the user ID that you want to find in the **Search** box at the top-right of the page and click the magnifying glass icon.

   The search filter that you are using displays on the left next to the **User Mappings** title, and the page displays the filtered list of rows that match the value you provided.

   For example, if you search the above page for the value 2, CTS filters the display as shown in the following illustration.



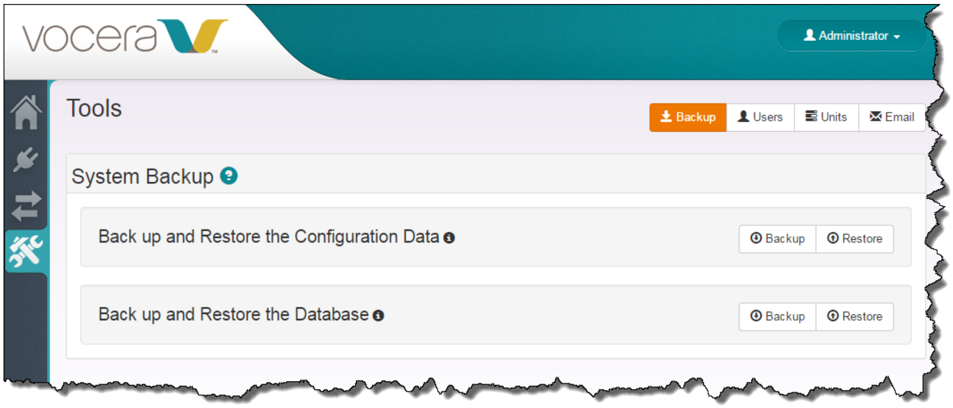
| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| null | 12 | Vocera |
| 12 | 12 | Rauland |
| null | u2 | Vocera |
| 2 | u2 | Rauland |

## How to Edit a User Mapping

You can use the **User Mappings** page to edit a user mapping at any time. The most common reason for updating values on this page is to specify mappings for unmapped users.

To provide a mapping for an unmapped user:

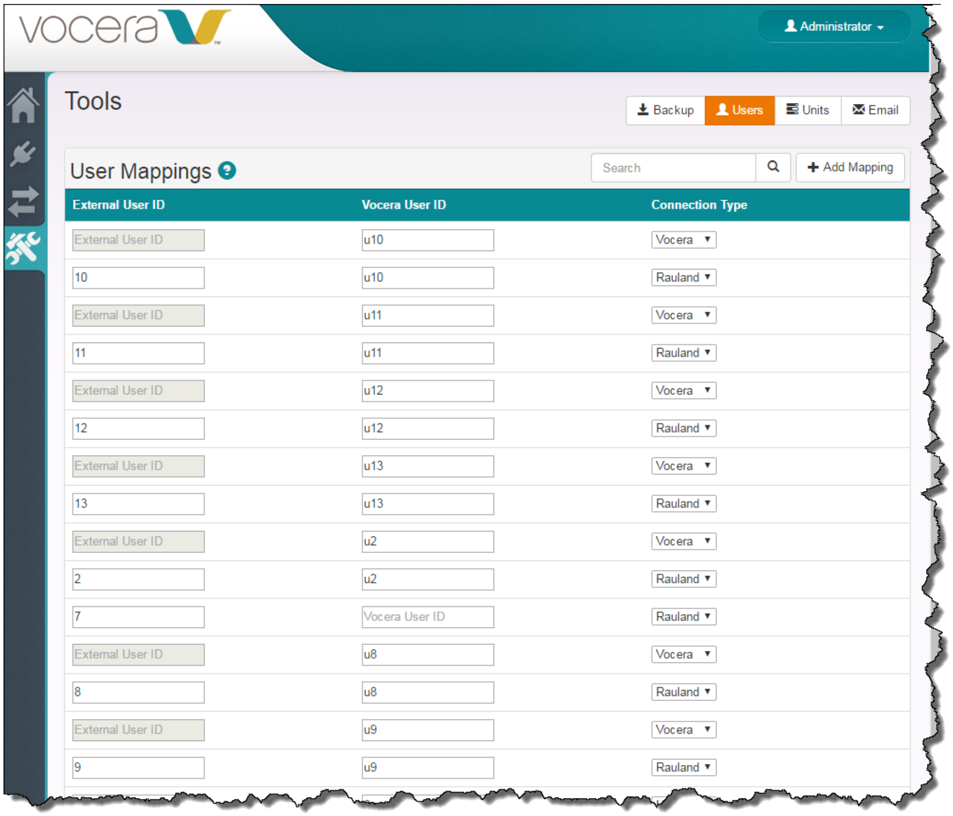
1. In the CTS Console, click the **Tools** icon ( 🛠 ) in the navigation bar on the left.
   The **Tools** page appears.



Back up and Restore the Configuration Data

Back up and Restore the Database

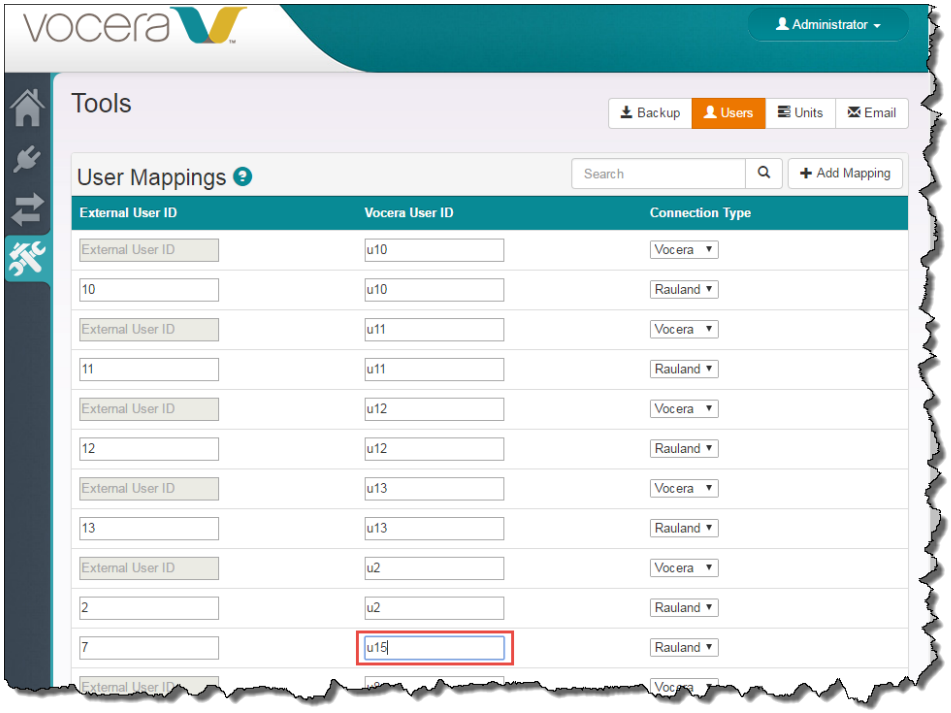
2. Click the **Users** tab in the top-right section of the page.
   The **User Mappings** page appears.



| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| External User ID | u10 | Vocera |
| 10 | u10 | Rauland |
| External User ID | u11 | Vocera |
| 11 | u11 | Rauland |
| External User ID | u12 | Vocera |
| 12 | u12 | Rauland |
| External User ID | u13 | Vocera |
| 13 | u13 | Rauland |
| External User ID | u2 | Vocera |
| 2 | u2 | Rauland |
| 7 | Vocera User ID | Rauland |
| External User ID | u8 | Vocera |
| 8 | u8 | Rauland |
| External User ID | u9 | Vocera |
| 9 | u9 | Rauland |

3. Click the value in the row you want to edit.

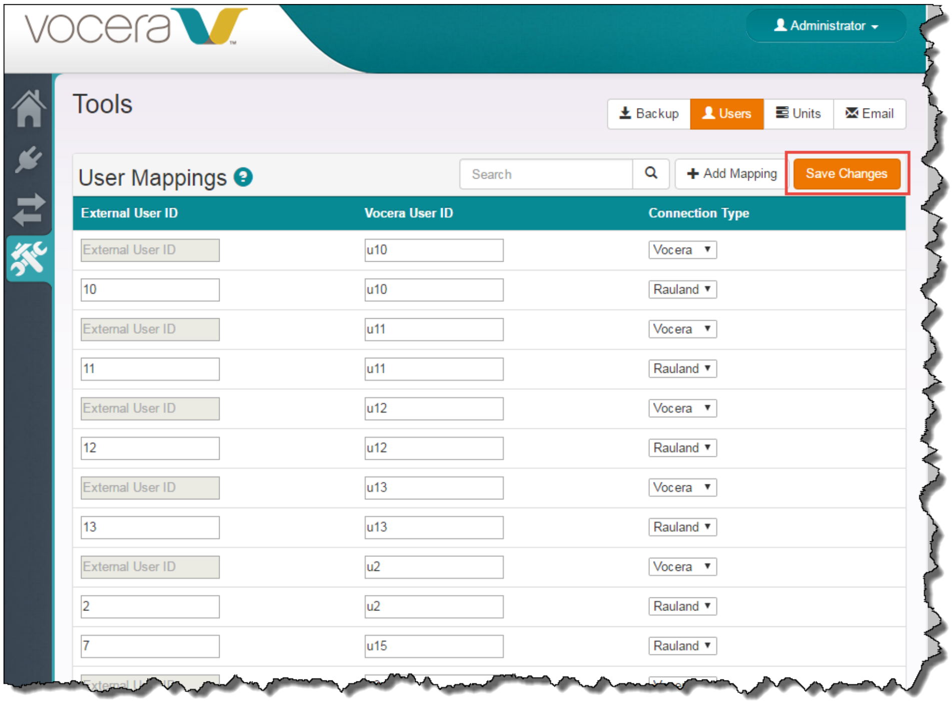
   For example, if you want to specify a mapping for the unmapped **External User ID** whose value is **7**, click the value in the **Vocera User ID** column next to it.
4. Enter the appropriate user ID in the **Vocera User ID** column.



| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| External User ID | u10 | Vocera |
| 10 | u10 | Rauland |
| External User ID | u11 | Vocera |
| 11 | u11 | Rauland |
| External User ID | u12 | Vocera |
| 12 | u12 | Rauland |
| External User ID | u13 | Vocera |
| 13 | u13 | Rauland |
| External User ID | u2 | Vocera |
| 2 | u2 | Rauland |
| 7 | u15 | Rauland |
| External User ID | u8 | Vocera |

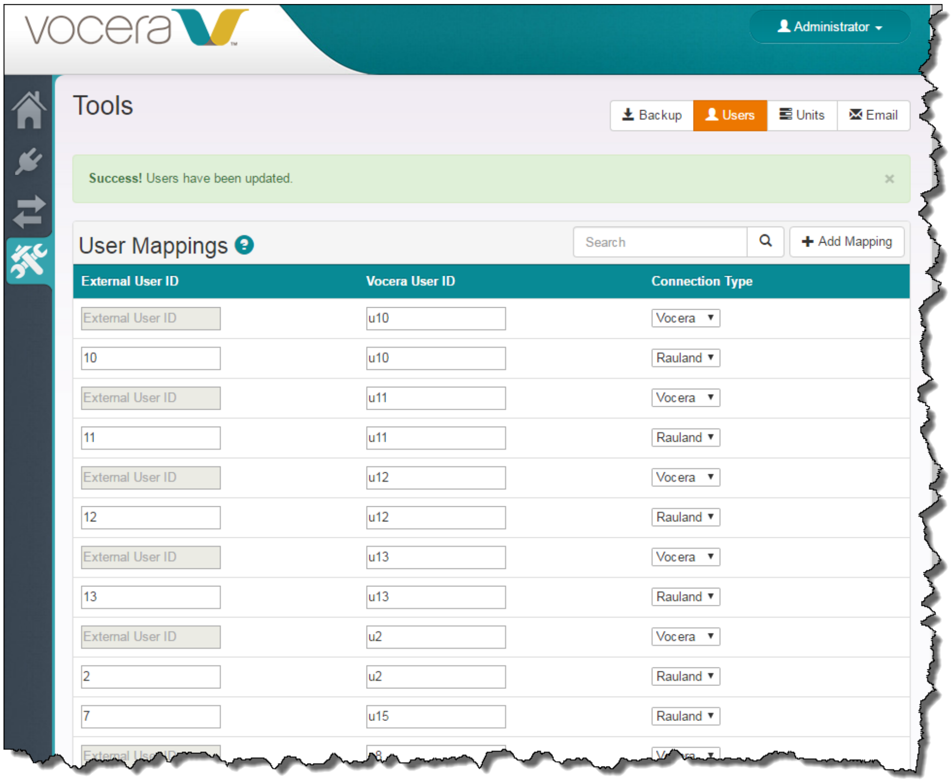
5. Tab away from the field.

   The **Save Changes** button appears in the top-right section of the page.



| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| External User ID | u10 | Vocera |
| 10 | u10 | Rauland |
| External User ID | u11 | Vocera |
| 11 | u11 | Rauland |
| External User ID | u12 | Vocera |
| 12 | u12 | Rauland |
| External User ID | u13 | Vocera |
| 13 | u13 | Rauland |
| External User ID | u2 | Vocera |
| 2 | u2 | Rauland |
| 7 | u15 | Rauland |

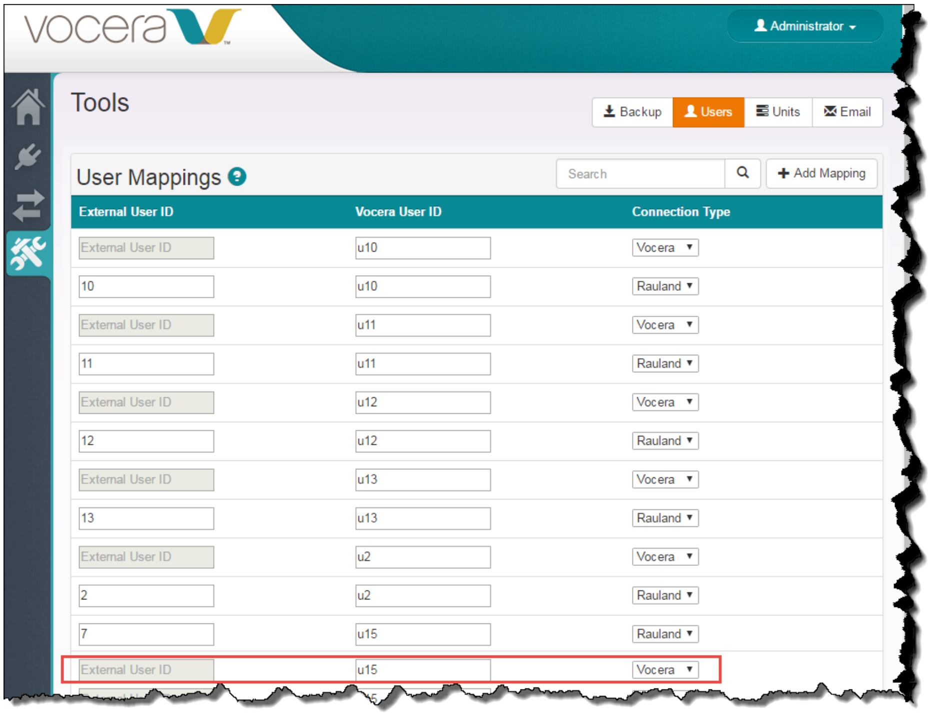
6. Click the **Save Changes** button.
   The page displays a success message in a banner across the top of the page.



**Success!** Users have been updated.

| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| External User ID | u10 | Vocera |
| 10 | u10 | Rauland |
| External User ID | u11 | Vocera |
| 11 | u11 | Rauland |
| External User ID | u12 | Vocera |
| 12 | u12 | Rauland |
| External User ID | u13 | Vocera |
| 13 | u13 | Rauland |
| External User ID | u2 | Vocera |
| 2 | u2 | Rauland |
| 7 | u15 | Rauland |

After the connection synchronizes users, CTS pulls in the Vocera user and displays it, as shown in the following illustration.



Tools

User Mappings

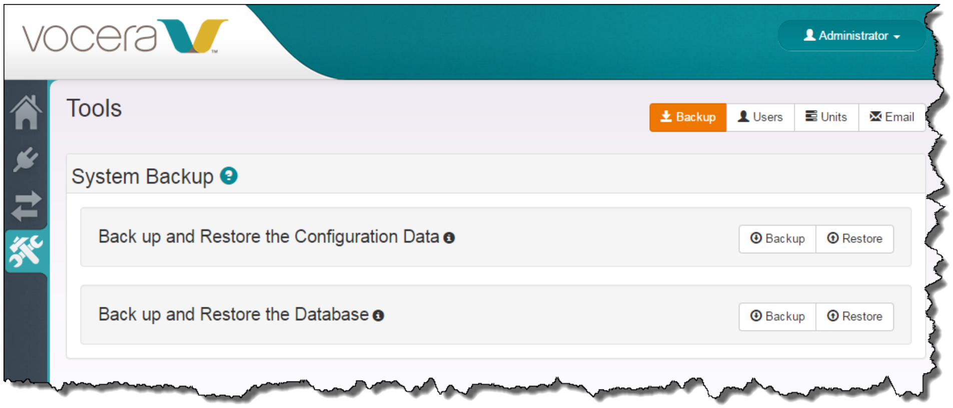
| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| External User ID | u10 | Vocera |
| 10 | u10 | Rauland |
| External User ID | u11 | Vocera |
| 11 | u11 | Rauland |
| External User ID | u12 | Vocera |
| 12 | u12 | Rauland |
| External User ID | u13 | Vocera |
| 13 | u13 | Rauland |
| External User ID | u2 | Vocera |
| 2 | u2 | Rauland |
| 7 | u15 | Rauland |
| External User ID | u15 | Vocera |

## How to Add a User Mapping

If neither the **Vocera User ID** nor the **External User ID** are already mapped by CTS for a specific connection, you can add that mapping to the system. When you add a mapping, you need to create two entries—one for the Vocera system, and one for the foreign end point.

To add a user mapping:

1. In the CTS Console, click the **Tools** icon ( 🛠 ) in the navigation bar on the left.
   The **Tools** page appears.



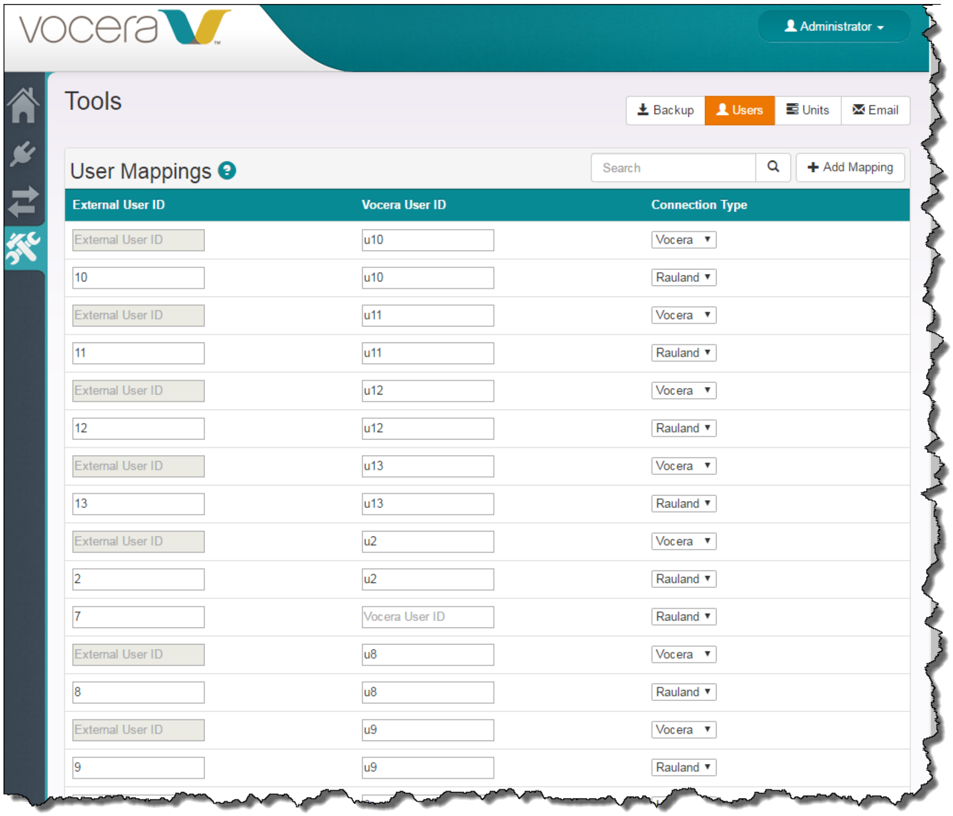
Tools

System Backup

Back up and Restore the Configuration Data

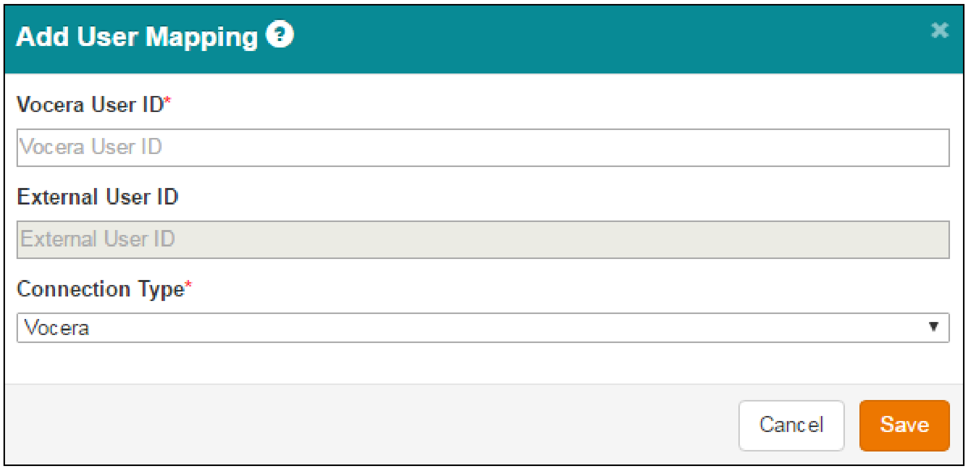
Back up and Restore the Database

2. Click the **Users** tab in the top-right section of the page.
   The **User Mappings** page appears.



| External User ID | Vocera User ID | Connection Type |
|---|---|---|
| External User ID | u10 | Vocera |
| 10 | u10 | Rauland |
| External User ID | u11 | Vocera |
| 11 | u11 | Rauland |
| External User ID | u12 | Vocera |
| 12 | u12 | Rauland |
| External User ID | u13 | Vocera |
| 13 | u13 | Rauland |
| External User ID | u2 | Vocera |
| 2 | u2 | Rauland |
| 7 | Vocera User ID | Rauland |
| External User ID | u8 | Vocera |
| 8 | u8 | Rauland |
| External User ID | u9 | Vocera |
| 9 | u9 | Rauland |

3. Create the mapping for the Vocera entry as follows:
   a. Click the **Add Mapping** button in the top-right section of the page.
      The **Add User Mapping** dialog box appears.



Add User Mapping

Vocera User ID*
Vocera User ID

External User ID
External User ID

Connection Type*
Vocera

   b. Set the **Connection Type** to **Vocera**.
   c. Set the **Vocera User ID** field to specify the ID that was assigned to the user on the Vocera Voice Server.
      This **Vocera User ID** must not yet be in use by CTS for the **Vocera** connection type.
   d. Click **Save**.

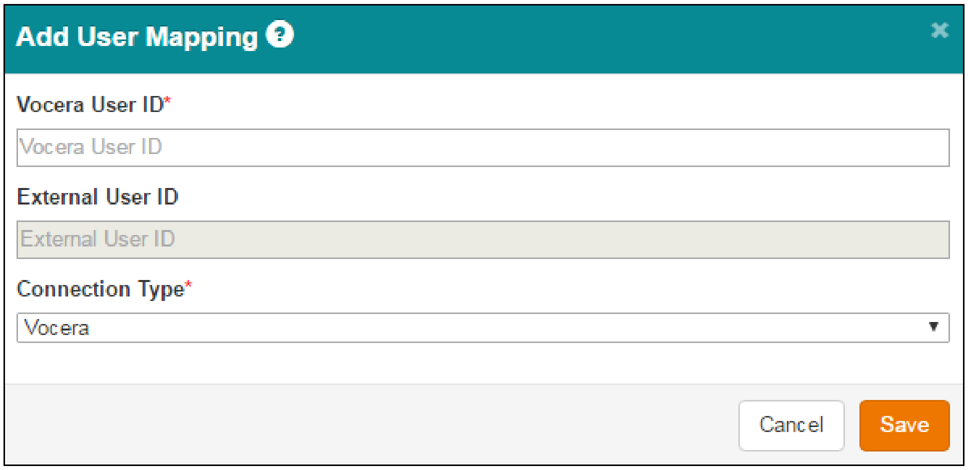
If the **Vocera User ID** is unique (that is, not already in use by CTS for the **Vocera** connection type), the dialog box closes, the mapping is created, and a success message appears in a banner across the top of the page.

If the user ID already exists, an error message appears in the dialog box, the mapping is not saved, and the dialog box remains open.

4. Create the mapping for the foreign system entry as follows:

   a. Click the **Add Mapping** button in the top-right section of the page.

      The **Add User Mapping** dialog box appears.



Add User Mapping

Vocera User ID*

Vocera User ID

External User ID

External User ID

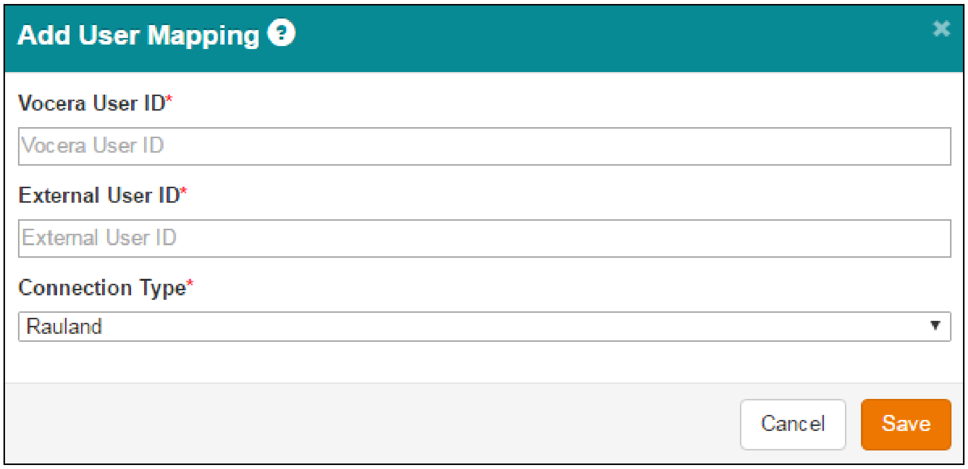
Connection Type*

Vocera

   b. Set the **Connection Type** to the foreign system.

      **Note:** The value that appears in the **Connection Type** list is the **Connection Name** that you provided when you defined the connection.

      The **External User ID** field appears enabled when the connection type is set to the foreign end point.



Add User Mapping

Vocera User ID*

Vocera User ID

External User ID*

External User ID

Connection Type*

Rauland

   c. Set the **Vocera User ID** field to specify the ID that was assigned to the user on the Vocera Voice Server.

      This **Vocera User ID** must not yet be in use by CTS for this connection type.

   d. Specify the ID that was assigned to the user on the foreign system.

      This **External User ID** must not yet be in use by CTS for this connection type.

      **Note:** If **Send Unmapped Users** is set to **True** in the connection definition, CTS imports that user and leaves it unmapped on the **User Mappings** page. See *How to Edit a User Mapping* on page 67 for information on how to complete this mapping.

   e. Click **Save**.

   If both the **Vocera User ID** and the **External User ID** are unique (that is, not already in use by CTS for this connection type), the dialog box closes, the mapping is created, and a success message appears in a banner across the top of the page.

   If either ID already exists, an error message appears in the dialog box, the mapping is not saved, and the dialog box remains open.

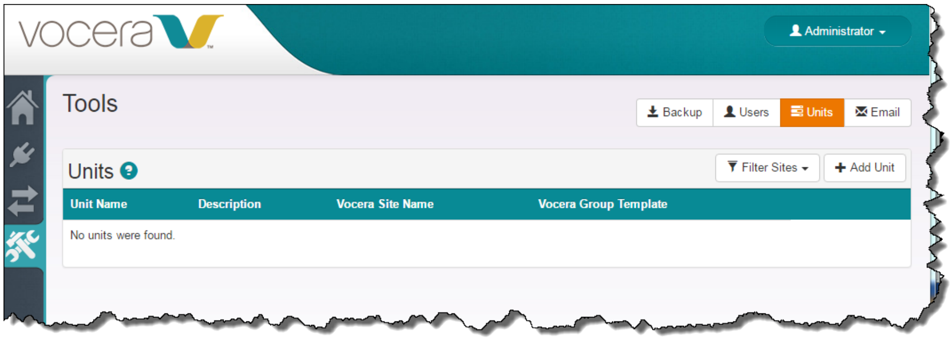
When you have created both and entry for the Vocera system and an entry for the foreign end point, the mapping is complete.

**Note:** The user IDs must exist on both the Vocera Voice Server and the foreign system for assignments to sync. If one or both of the user IDs do not exist, CTS will be unable to make assignments. CTS does not check to confirm these IDs exist; it simply saves whatever mapping you specify.

## Managing Units

When you define units in CTS, you provide templates to pre-filter groups from the Vocera Voice Server and normalizers to refine the output of the templates. The groups in the VS that the normalizers allow are processed and output as normalized roles and locations that are available in CTS.

The combination of Vocera Voice Server site name, CTS unit name, and normalized roles populates the **Sync Settings** page. Before you define units, both the **Sync Settings** page and the **Units** page are empty.

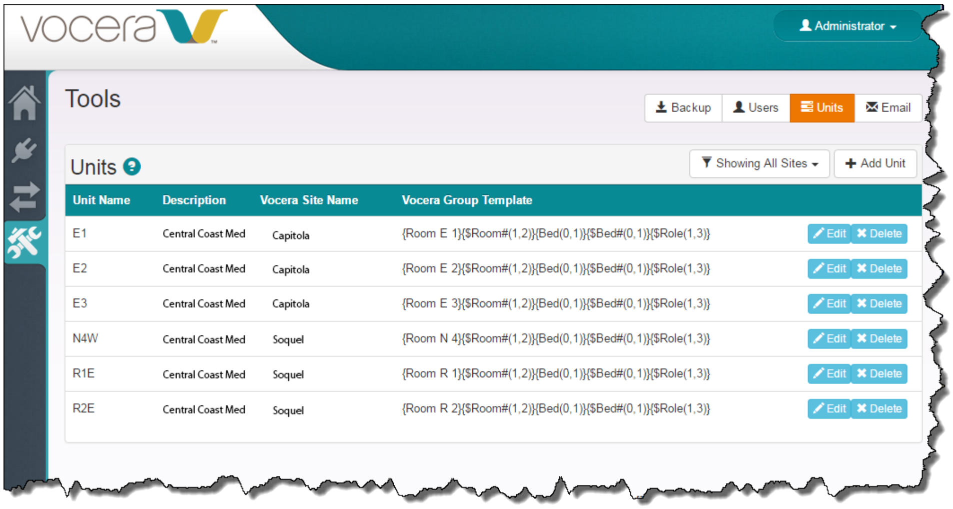


Tools

Units

| Unit Name | Description | Vocera Site Name | Vocera Group Template |
|---|---|---|---|
| No units were found. | | | |

After you define units, use the **Units** page to view a summary, edit, and create new units.

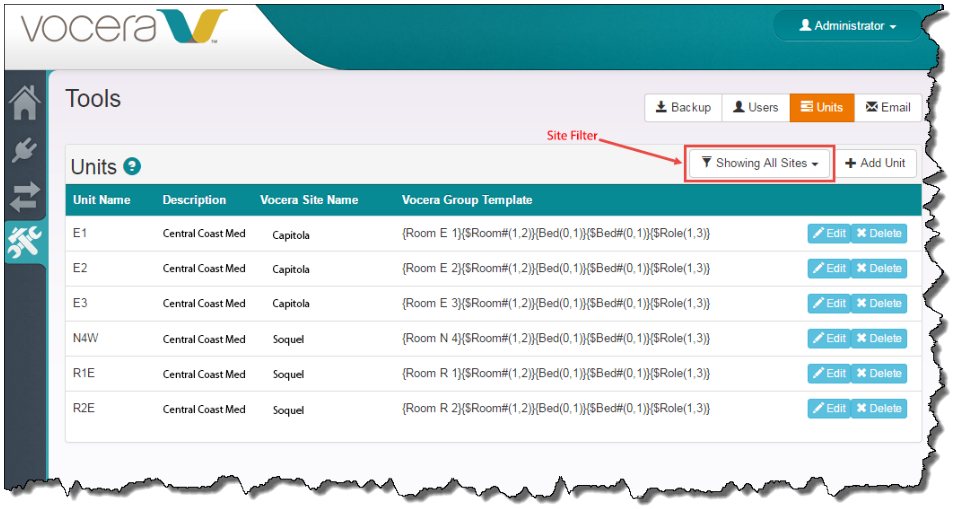


Tools

Units

| Unit Name | Description | Vocera Site Name | Vocera Group Template | |
|---|---|---|---|---|
| E1 | Central Coast Med | Capitola | {Room E 1}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} | Edit Delete |
| E2 | Central Coast Med | Capitola | {Room E 2}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} | Edit Delete |
| E3 | Central Coast Med | Capitola | {Room E 3}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} | Edit Delete |
| N4W | Central Coast Med | Soquel | {Room N 4}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} | Edit Delete |
| R1E | Central Coast Med | Soquel | {Room R 1}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} | Edit Delete |
| R2E | Central Coast Med | Soquel | {Room R 2}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} | Edit Delete |

Optionally use the **Site Filter** on the **Units** page to limit the display to a single site or to view all sites.



| Unit Name | Description | Vocera Site Name | Vocera Group Template |
|---|---|---|---|
| E1 | Central Coast Med | Capitola | {Room E 1}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} |
| E2 | Central Coast Med | Capitola | {Room E 2}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} |
| E3 | Central Coast Med | Capitola | {Room E 3}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} |
| N4W | Central Coast Med | Soquel | {Room N 4}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} |
| R1E | Central Coast Med | Soquel | {Room R 1}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} |
| R2E | Central Coast Med | Soquel | {Room R 2}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} |

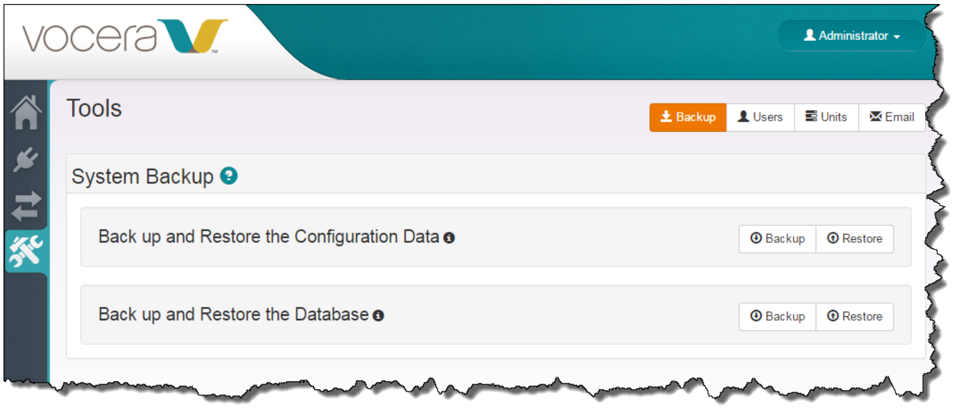
## How to Create a Unit

When you create a unit, you specify a Vocera Voice Server site name, define a CTS unit name, and provide a template and normalizers that filter role-and-location based group names from the VS and output normalized roles and locations.

To add a unit:

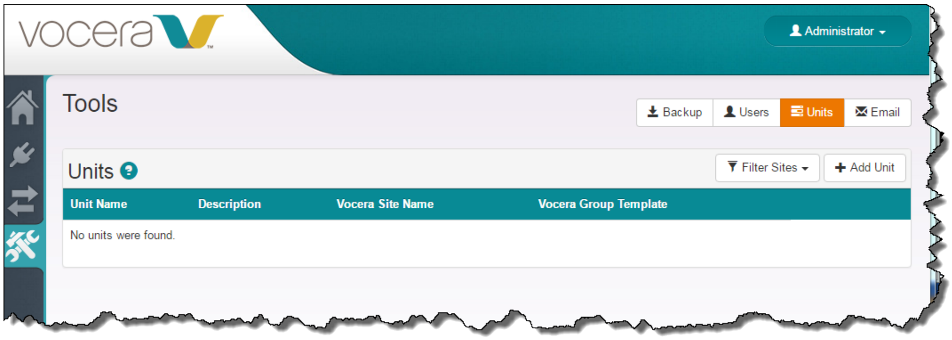
1. In the CTS Console, click the **Tools** icon ( 🛠 ) in the navigation bar on the left.

   The **Tools** page appears.



2. Click the **Units** button on the top right of the page.

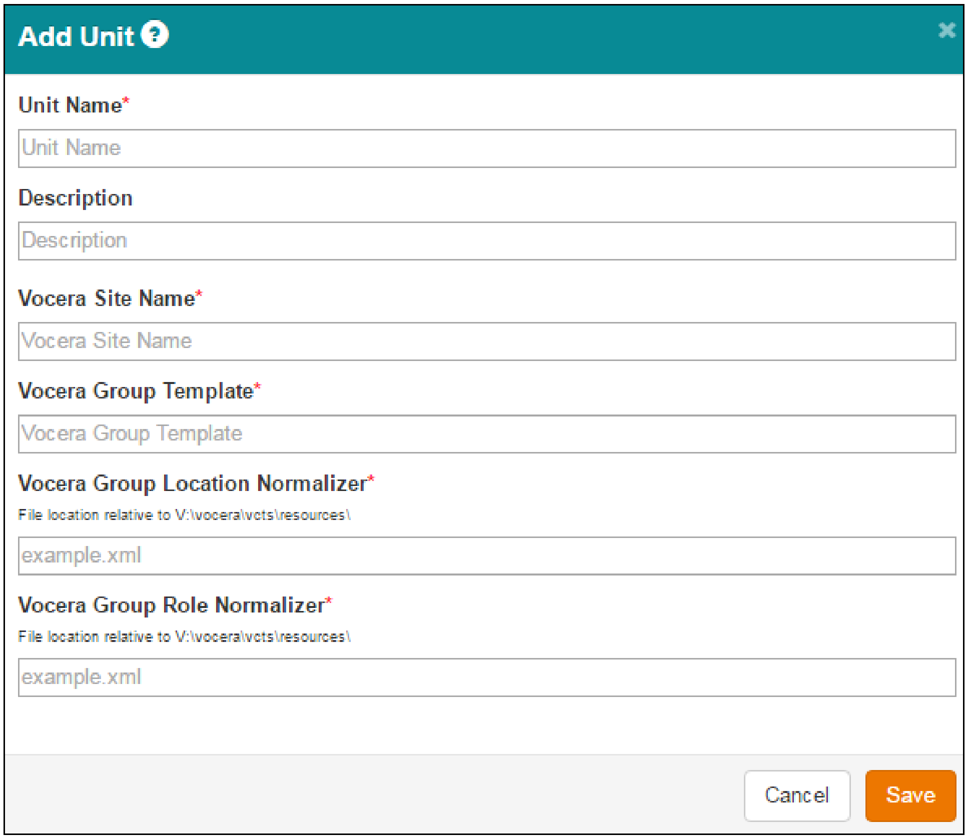
   The **Units** page appears.



| Unit Name | Description | Vocera Site Name | Vocera Group Template |
|---|---|---|---|
| No units were found. | | | |

3. Click the **Add Unit** button.

The **Add Unit** dialog box appears.



Add Unit

Unit Name*

Unit Name

Description

Description

Vocera Site Name*

Vocera Site Name

Vocera Group Template*

Vocera Group Template

Vocera Group Location Normalizer*

File location relative to V:\vocera\vcts\resources\

example.xml

Vocera Group Role Normalizer*

File location relative to V:\vocera\vcts\resources\

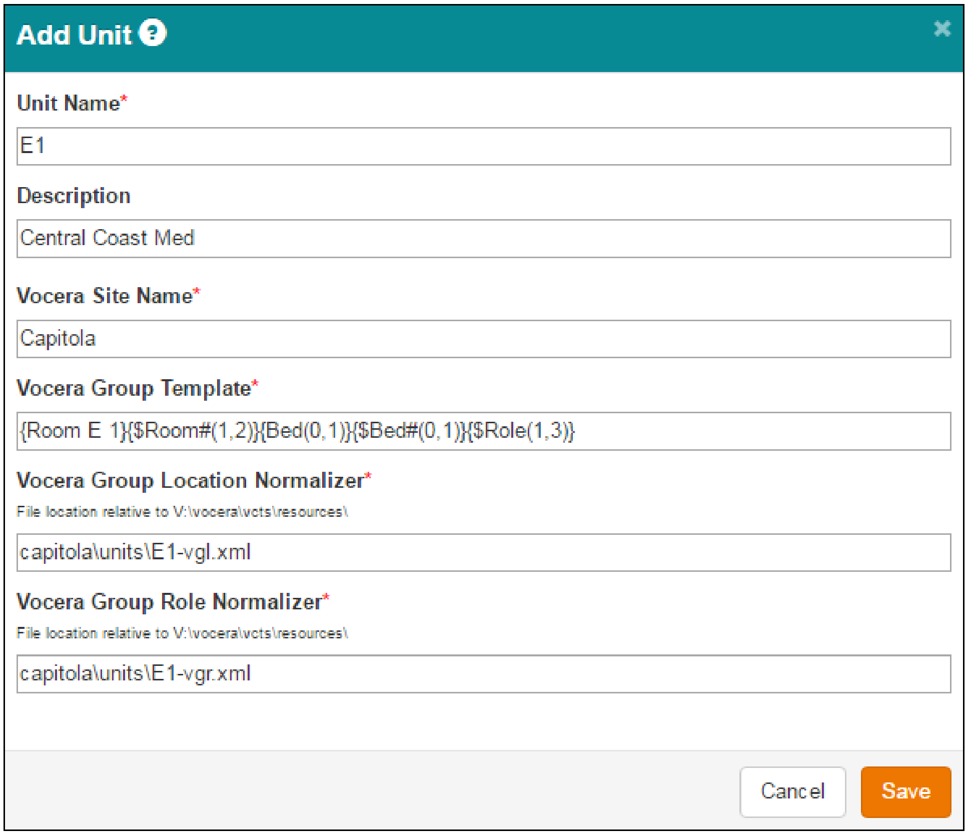
example.xml

4. Specify the unit definition in the **Add Unit** dialog box as follows:

| Field | Description |
|---|---|
| Unit Name | Specify the name of the unit you are defining. |
| Description | Optionally enter an additional description of the unit for your own information. This description is for your own reference only. |
| Vocera Site Name | Specify the name of the Vocera Voice Server site that contains the role-and-location based groups comprising this unit. Use the exact spelling that the VS uses so the unit is associated with the proper Vocera site. |
| Vocera Group Template | Provide a template to pre-filter groups from the Vocera Voice Server. The role and location normalizers use the group names that the template allows as input; these normalizers output the normalized roles and locations that populate the `role` and `location` tables in the CTS database.<br>See *Creating Group Templates for Units* on page 96. |
| Vocera Group Location Normalizer | The **Vocera Group Location Normalizer** parses role-and-location based group names from the Vocera Voice Server and outputs normalized locations. These locations are maintained in the CTS `locations` table, and they are not visible in the CTS user interface.<br>Enter the location of the normalizer relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the normalizers used by different sites.<br>See *Creating Group Normalizers for Units* on page 101. |

| Field | Description |
|---|---|
| Vocera Group Role Normalizer | The **Vocera Group Role Normalizer** parses role-and-location based group names from the Vocera Voice Server and outputs normalized roles. These roles are maintained in the CTS `roles` table, and they also appear as the site/unit/role combinations on the **Sync Settings** page.<br>Enter the location of the normalizer relative to the `\vocera\vcts\resources\` path on the CTS machine. You may optionally create a directory structure under `resources` to provide separate directories for the normalizers used by different sites.<br>See *Creating Group Normalizers for Units* on page 101. |

When you are finished, the dialog box should look similar to the following:



Add Unit

Unit Name*

E1

Description

Central Coast Med

Vocera Site Name*

Capitola

Vocera Group Template*

{Room E 1}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)}

Vocera Group Location Normalizer*

File location relative to V:\vocera\vcts\resources\

capitola\units\E1-vgl.xml

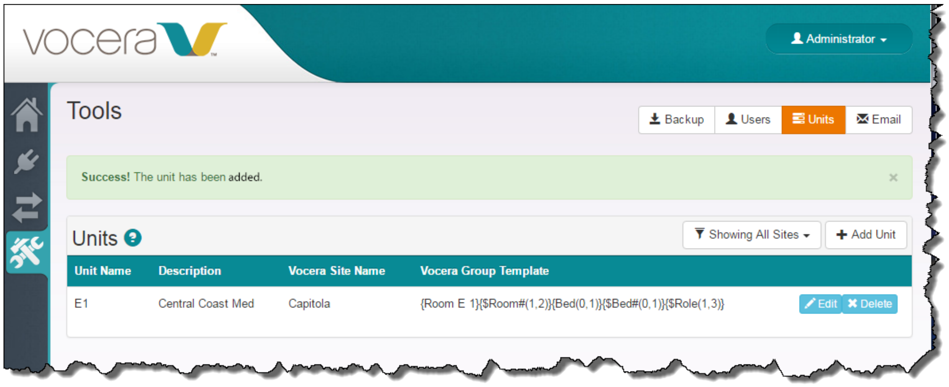
Vocera Group Role Normalizer*

File location relative to V:\vocera\vcts\resources\

capitola\units\E1-vgr.xml

5. Click **Save**.

The **Add Unit** dialog box closes, your changes are saved to the database, and the unit appears in the **Unit Name** section of the **Units** page. The page also displays a success message.



Tools

Success! The unit has been added.

Units

| Unit Name | Description | Vocera Site Name | Vocera Group Template | |
|---|---|---|---|---|
| E1 | Central Coast Med | Capitola | {Room E 1}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} | Edit Delete |

6. Continue adding unit and role combinations in this manner until you have defined all the VS groups whose assignments are managed by CTS.

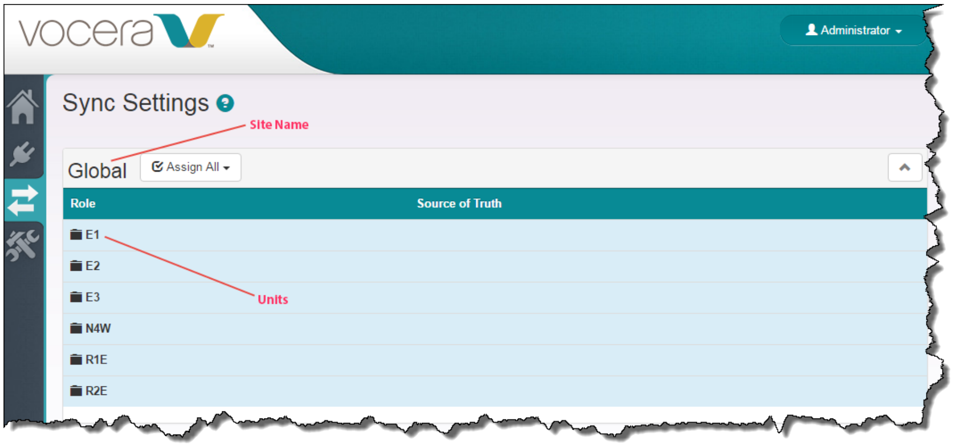
7. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

8. Click the **Sync Settings** icon ( ⇄ ) in the navigation bar on the left.

   The **Sync Settings** page appears.



Sync Settings

Site Name

Global

| Role | Source of Truth |
|---|---|
| E1 | |
| E2 | |
| E3 | |
| N4W | |
| R1E | |
| R2E | |

Units

9. Confirm that the units you created are visible on the **Sync Settings** page.

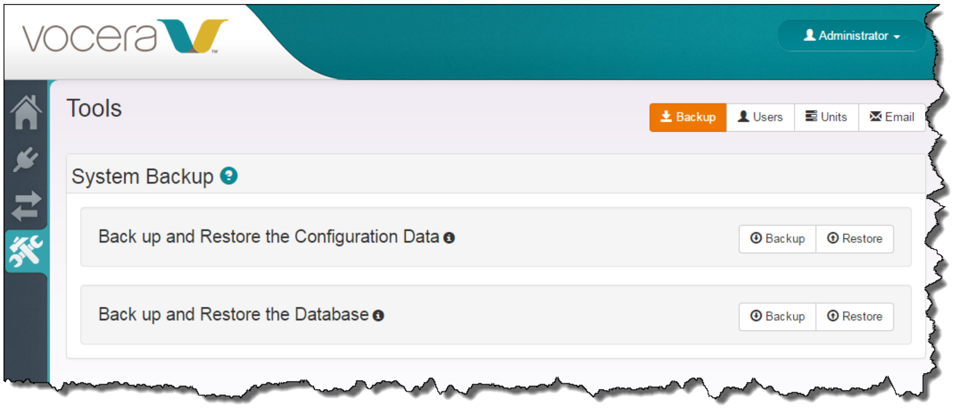
## How to Update a Unit

Edit a unit at any time to update its definition. After saving, you must restart the Vocera Vocera Care Team Sync service to allow changes to take effect.

To update the definition of a unit:

1. In the CTS Console, click the **Tools** icon ( 🛠 ) in the navigation bar on the left.

   The **Tools** page appears.



Tools

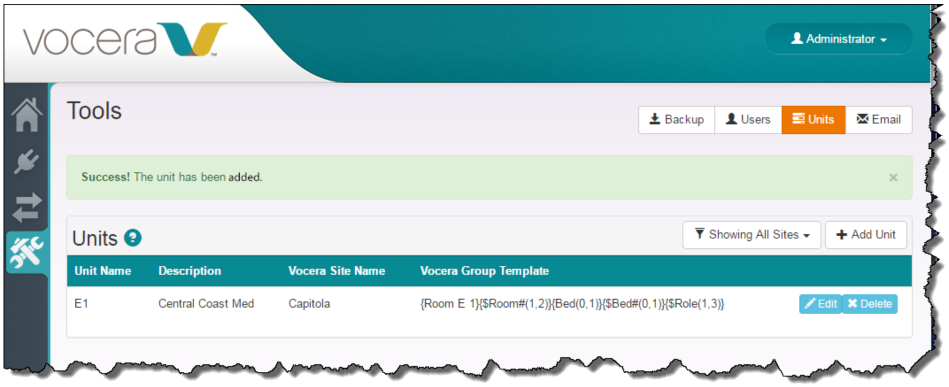
System Backup

Back up and Restore the Configuration Data

Back up and Restore the Database

2. Click the **Units** button on the top right of the page.

   The **Units** page appears.



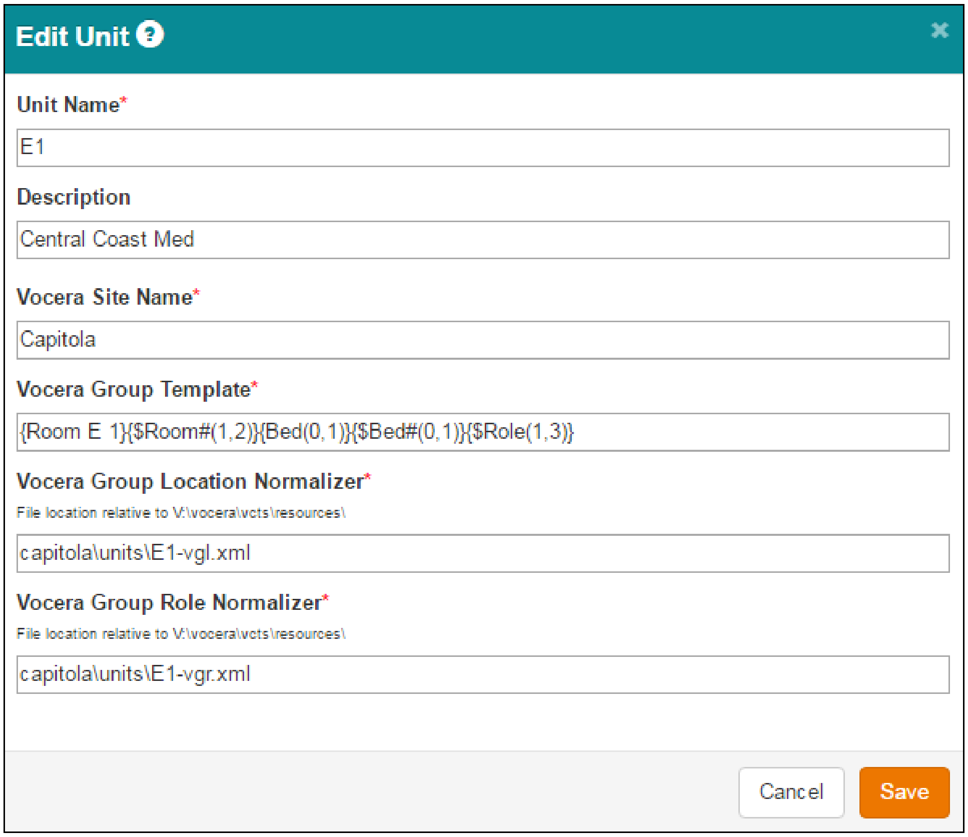
Success! The unit has been added.

Units

| Unit Name | Description | Vocera Site Name | Vocera Group Template |
|---|---|---|---|
| E1 | Central Coast Med | Capitola | {Room E 1}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} |

3. Click the **Edit** button to the right of the unit that you want to edit.

   The **Edit Unit** dialog box appears.



Edit Unit

Unit Name*

E1

Description

Central Coast Med

Vocera Site Name*

Capitola

Vocera Group Template*

{Room E 1}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)}

Vocera Group Location Normalizer*

File location relative to V:\vocera\vcts\resources\

capitola\units\E1-vgl.xml

Vocera Group Role Normalizer*

File location relative to V:\vocera\vcts\resources\

capitola\units\E1-vgr.xml

4. Edit any fields that need to be updated.

   See the descriptions of the fields in *How to Create a Unit* on page 74.

5. Click **Save** to record your changes.

   The **Edit Unit** dialog box closes, and the changes are saved to the database.

6. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

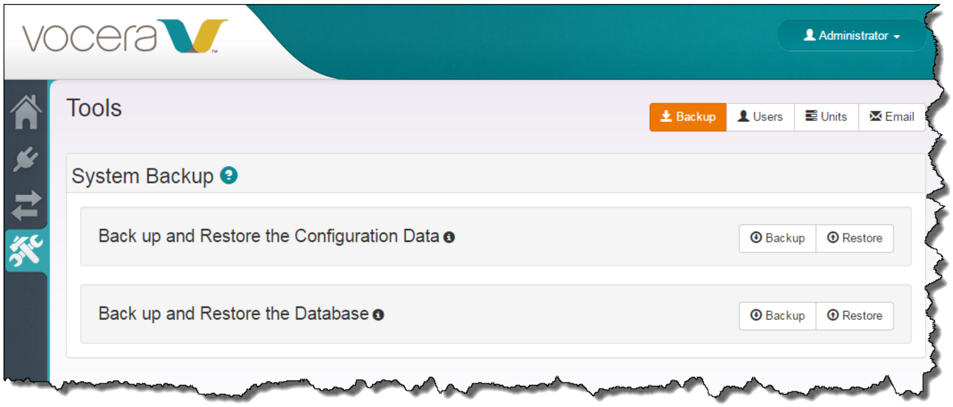
   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

### How to Delete a Unit

You can delete a unit if you determine that it is no longer necessary, or for any other reason, as long as no assignments have been made. After deleting, you must restart the Vocera Vocera Care Team Sync service to allow changes to take effect.

To delete a unit:

1. In the CTS Console, click the **Tools** icon ( 🛠 ) in the navigation bar on the left.
   The **Tools** page appears.

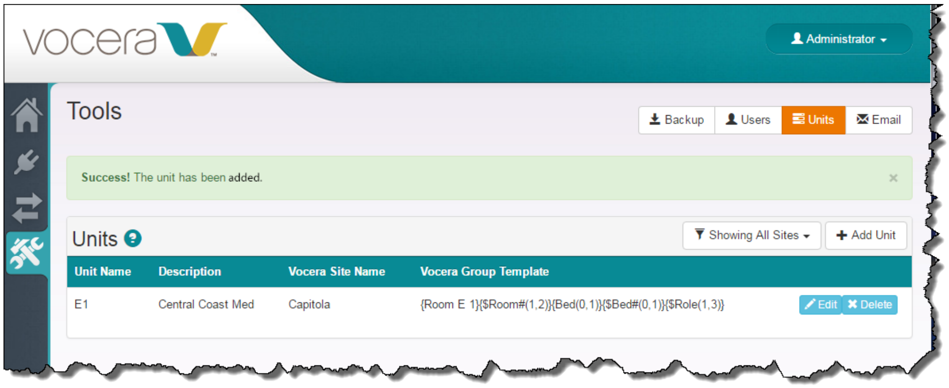


System Backup

Back up and Restore the Configuration Data

Back up and Restore the Database

2. Click the **Units** button on the top right of the page.
   The **Units** page appears.

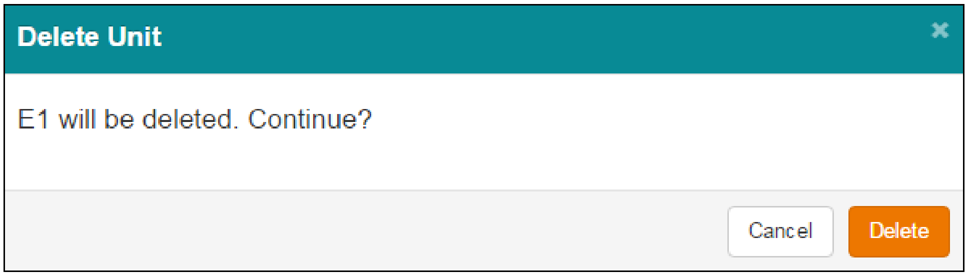


**Success!** The unit has been added.

Units

| Unit Name | Description | Vocera Site Name | Vocera Group Template | |
|---|---|---|---|---|
| E1 | Central Coast Med | Capitola | {Room E 1}{$Room#(1,2)}{Bed(0,1)}{$Bed#(0,1)}{$Role(1,3)} | Edit Delete |

3. Click the **Delete** button to the right of the unit that you want to edit.
   A dialog box appears and asks you to confirm the deletion.



Delete Unit

E1 will be deleted. Continue?

4. Click **Delete** to confirm.
   The confirmation dialog box closes, and the changes are saved to the database.
5. Restart the Vocera CTS service.
   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

## Configuring Email

The CTS system can send alert messages to administrators to notify them about failed connections. You can configure this capability by using the **Email Alert Settings** page to specify information about your email server and the frequency with which CTS sends alerts.

The situation that triggers an email alert is slightly different for each type of connection:

| Connection Type | Alert Condition |
|---|---|
| Rauland and Hill-Rom | For Rauland and Hill-Rom connections, the foreign system publishes a web service; CTS attempts to subscribe to the web service when the Windows Vocera CTS service starts or when CTS misses three consecutive heartbeats with the foreign system.<br>CTS sends an alert at the specified frequency as long as these subscription attempts fail. |
| HL7 | For HL7 connections, the foreign system publishes a web service and actively attempts to connect to CTS.<br>CTS sends an alert at the specified frequency if it determines that the connection is down. |
| Vocera Voice Server VAI Connection | For VAI connections, CTS attempts to connect to the Vocera Voice Server when the Windows Vocera CTS service starts or when CTS misses three consecutive heartbeats with the Vocera Voice Server.<br>CTS sends an alert if it fails to connect to VAI when the Windows Vocera CTS service starts. |

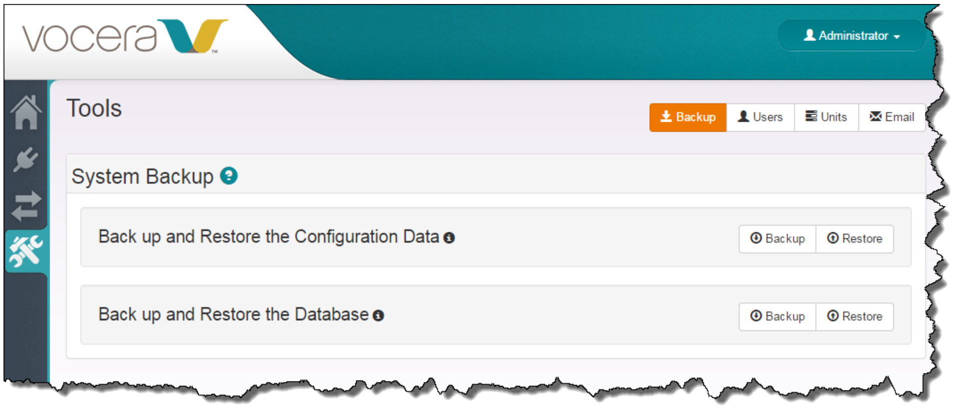
### How to Set Up Email

Email configuration allows the CTS to send alert messages to any email address, notifying you when a connection has failed. You can choose whether to receive an alert a single time only, or configure the frequency with which you receive the alert messages.

Use the following steps to configure the email integration:

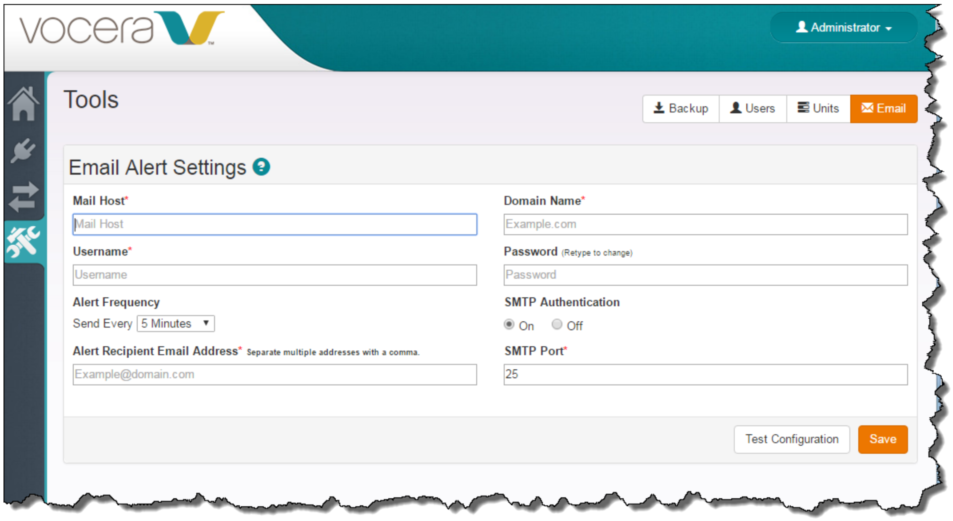
1. In the CTS Console, click the **Tools** icon ( 🛠 ) in the navigation bar on the left.

   The **Tools** page appears.



2. Click the **Email** button on the top right of the page.

   The **Email Alert Settings** page appears.



Tools

Email Alert Settings

Mail Host* Mail Host

Domain Name* Example.com

Username* Username

Password (Retype to change) Password

Alert Frequency
Send Every 5 Minutes ▼

SMTP Authentication
● On ○ Off

Alert Recipient Email Address* Separate multiple addresses with a comma.
Example@domain.com

SMTP Port* 25

3. Enter the email integration parameters that allow CTS to communicate with your mail server as follows:

| Field | Description |
|---|---|
| Mail Host | Enter the name of the POP or IMAP server used for outgoing mail in the **Mail Host** field. For example: `mail.yourcompany.com`. |
| Domain Name | In the **Domain Name** field, specify the domain name used in email addresses at your site. CTS uses this value to complete any email addresses in the **Username** or **Alert Recipient Email Address** field that do not have a domain name. |
| Username | Enter the user name or address used to login to the outgoing mail server in the **Username** field. |
| Password | In the **Password** field, enter the password that provides the account permissions to send email on behalf of CTS. |
| Alert Frequency | In the **Alert Frequency** field, use the list to set the interval at which email alerts are sent. CTS continues to send alerts until the connection is restored. |
| SMTP Authentication | Check **SMTP Authentication** if your mail server requires its subscribers to provide authentication when sending an email message. |
| Alert Recipient Email Address | In the **Alert Recipient Email Address** field, enter an email address to receive alert messages that the CTS issues. Separate multiple addresses with a comma.<br>If this field is empty, you won't get any alerts. |
| SMTP Port | If **SMTP Authentication** is checked, specify the port that is used for the authentication request. |

4. Click **Save** to save your configuration settings to the database.
5. Click **Test Configuration** to send a test message to the specified default alert recipient.

   If the configuration is successful, you receive an email at the **Alert Recipient Email Address** account. If the configuration is not correct, the screen displays the error message, "Oops! Unable to send test email".

## Receiving Alert Emails

Administrators receive a single email message when you test the email configuration, and they also receive an email at the frequency that you specify for each connection to CTS that fails.

When you test the email configuration, the accounts specified in the **Alert Recipient Email Address** field on the **Email Alert Settings** page each receive a message with the following information:

| Field | Description |
|---|---|
| Subject | CTS Test Email Notification |

| Field | Description |
|---|---|
| Message | This is a test email notification from the Vocera CTS server. |

If any connection fails after the email integration is established, the accounts specified in the **Alert Recipient Email Address** field receive a message for each connection that fails containing information similar to the following:

| Field | Description |
|---|---|
| Subject | CTS Alert - Client HRC Failed to Subscribe |
| Message | CTS System Alert - Subscribe Failed<br>2017-02-28 10:28:51 PST<br>Client HRC failed to subscribe. This has occurred 30 consecutive times. The client will attempt to resubscribe. Please check the system logs for more details. |

The name of the client, date/time, and number of failures are variables that are updated by the system each time an alert message is sent.

# Troubleshooting

This section shows you how to perform several typical troubleshooting tasks.

## Confirming Assignment Sync

When you are ready to test your system, confirm that assignments are syncing correctly by viewing the expected assignments on the target system—either the Vocera Voice Server or the foreign clinical assignment system. If you do not see the assignments, work backwards from the target system to the CTS database to the CTS logs to see if assignments appear anywhere.

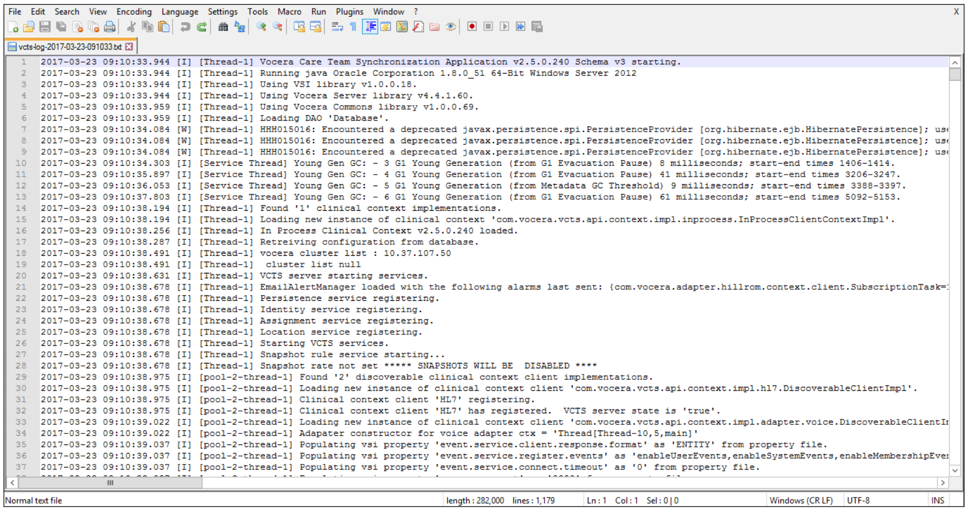
### How to Confirm Connections and Assignments in the Logs

Use the logs on the CTS machine to confirm that CTS has successfully connected with the foreign clinical assignment system and that it is receiving assignments.

CTS maintains logs in the `\vocera\logs` directory. The date and time that the each log file was created is specified in the file name as `vcts-log-<date>-<time>.txt`. CTS creates a new log at midnight every night, as long as an assignment client is connected.

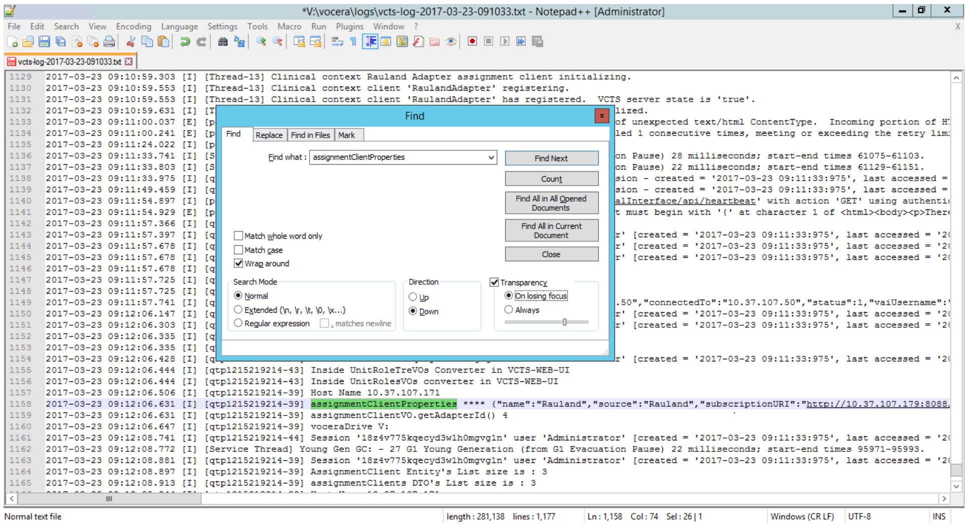
To confirm connections and assignments in the CTS logs:

1. On the CTS machine, navigate to the `\vocera\logs` directory and open the most recent log file in a text editor.



```
2017-03-23 09:10:33.944 [I] [Thread-1] Vocera Care Team Synchronization Application v2.5.0.240 Schema v3 starting.
2017-03-23 09:10:33.944 [I] [Thread-1] Running java Oracle Corporation 1.8.0_51 64-Bit Windows Server 2012
2017-03-23 09:10:33.944 [I] [Thread-1] Using VSI library v1.0.0.18.
2017-03-23 09:10:33.944 [I] [Thread-1] Using Vocera Server library v4.4.1.60.
2017-03-23 09:10:33.959 [I] [Thread-1] Using Vocera Commons library v1.0.0.69.
2017-03-23 09:10:33.959 [I] [Thread-1] Loading DAO 'Database'.
2017-03-23 09:10:34.084 [W] [Thread-1] HHH015016: Encountered a deprecated javax.persistence.spi.PersistenceProvider [org.hibernate.ejb.HibernatePersistence]; use
2017-03-23 09:10:34.084 [W] [Thread-1] HHH015016: Encountered a deprecated javax.persistence.spi.PersistenceProvider [org.hibernate.ejb.HibernatePersistence]; use
2017-03-23 09:10:34.084 [W] [Thread-1] HHH015016: Encountered a deprecated javax.persistence.spi.PersistenceProvider [org.hibernate.ejb.HibernatePersistence]; use
2017-03-23 09:10:34.303 [I] [Service Thread] Young Gen GC: - 3 G1 Young Generation (from G1 Evacuation Pause) 8 milliseconds; start-end times 1406-1414.
2017-03-23 09:10:35.897 [I] [Service Thread] Young Gen GC: - 4 G1 Young Generation (from G1 Evacuation Pause) 41 milliseconds; start-end times 3206-3247.
2017-03-23 09:10:36.053 [I] [Service Thread] Young Gen GC: - 5 G1 Young Generation (from Metadata GC Threshold) 9 milliseconds; start-end times 3388-3397.
2017-03-23 09:10:37.803 [I] [Service Thread] Young Gen GC: - 6 G1 Young Generation (from G1 Evacuation Pause) 61 milliseconds; start-end times 5092-5153.
2017-03-23 09:10:38.194 [I] [Thread-1] Found '1' clinical context implementations.
2017-03-23 09:10:38.194 [I] [Thread-1] Loading new instance of clinical context 'com.vocera.vcts.api.context.impl.inprocess.InProcessClientContextImpl'.
2017-03-23 09:10:38.256 [I] [Thread-1] In Process Clinical Context v2.5.0.240 loaded.
2017-03-23 09:10:38.287 [I] [Thread-1] Retreiving configuration from database.
2017-03-23 09:10:38.491 [I] [Thread-1] vocera cluster list : 10.37.107.50
2017-03-23 09:10:38.491 [I] [Thread-1]  cluster list null
2017-03-23 09:10:38.631 [I] [Thread-1] VCTS server starting services.
2017-03-23 09:10:38.678 [I] [Thread-1] EmailAlertManager loaded with the following alarms last sent: {com.vocera.adapter.hillrom.context.client.SubscriptionTask=
2017-03-23 09:10:38.678 [I] [Thread-1] Persistence service registering.
2017-03-23 09:10:38.678 [I] [Thread-1] Identity service registering.
2017-03-23 09:10:38.678 [I] [Thread-1] Assignment service registering.
2017-03-23 09:10:38.678 [I] [Thread-1] Location service registering.
2017-03-23 09:10:38.678 [I] [Thread-1] Starting VCTS services.
2017-03-23 09:10:38.678 [I] [Thread-1] Snapshot rule service starting...
2017-03-23 09:10:38.678 [I] [Thread-1] Snapshot rate not set ***** SNAPSHOTS WILL BE  DISABLED ****
2017-03-23 09:10:38.975 [I] [pool-2-thread-1] Found '2' discoverable clinical context client implementations.
2017-03-23 09:10:38.975 [I] [pool-2-thread-1] Loading new instance of clinical context client 'com.vocera.vcts.api.context.impl.hl7.DiscoverableClientImpl'.
2017-03-23 09:10:38.975 [I] [pool-2-thread-1] Clinical context client 'HL7' registering.
2017-03-23 09:10:38.975 [I] [pool-2-thread-1] Clinical context client 'HL7' has registered.  VCTS server state is 'true'.
2017-03-23 09:10:39.022 [I] [pool-2-thread-1] Loading new instance of clinical context client 'com.vocera.vcts.api.context.impl.adapter.voice.DiscoverableClientI
2017-03-23 09:10:39.022 [I] [pool-2-thread-1] Adapater constructor for voice adapter ctx = 'Thread[Thread-10,5,main]'
2017-03-23 09:10:39.037 [I] [pool-2-thread-1] Populating vsi property 'event.service.client.response.format' as 'ENTITY' from property file.
2017-03-23 09:10:39.037 [I] [pool-2-thread-1] Populating vsi property 'event.service.register.events' as 'enableUserEvents,enableSystemEvents,enableMembershipEve
2017-03-23 09:10:39.037 [I] [pool-2-thread-1] Populating vsi property 'event.service.connect.timeout' as '0' from property file.
```

2. Search for the string `assignmentClientProperties`.



*V:\vocera\logs\vcts-log-2017-03-23-091033.txt - Notepad++ [Administrator]

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
vcts-log-2017-03-23-091033.txt
Find
Find Replace Find in Files Mark
Find Next
Count
Find All in All Opened Documents
Find All in Current Document
Close

Find what : assignmentClientProperties

☐ Match whole word only
☐ Match case
☑ Wrap around

Search Mode: ● Normal ○ Extended (\n, \r, \t, \0, \x...) ○ Regular expression ☐ . matches newline

Direction: ○ Up ● Down

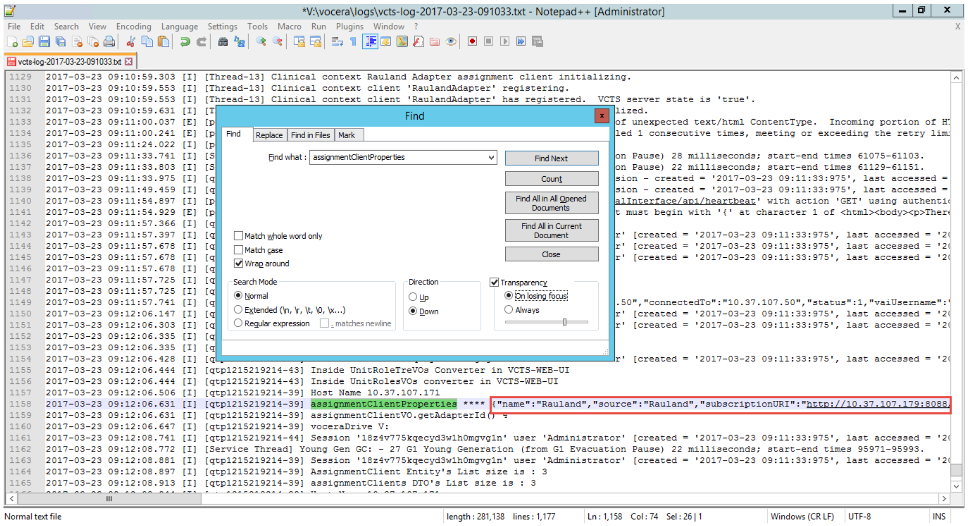
☑ Transparency: ● On losing focus ○ Always

```
2017-03-23 09:10:59.303 [I] [Thread-13] Clinical context Rauland Adapter assignment client initializing.
2017-03-23 09:10:59.553 [I] [Thread-13] Clinical context client 'RaulandAdapter' registering.
2017-03-23 09:10:59.553 [I] [Thread-13] Clinical context client 'RaulandAdapter' has registered.  VCTS server state is 'true'.
2017-03-23 09:10:59.631 [I] [T                                          lized.
2017-03-23 09:11:00.037 [E] [p                                          of unexpected text/html ContentType.  Incoming portion of H
2017-03-23 09:11:00.241 [E] [p                                          led 1 consecutive times, meeting or exceeding the retry lim
2017-03-23 09:11:24.022 [I] [p
2017-03-23 09:11:33.741 [I] [S                                          on Pause) 28 milliseconds; start-end times 61075-61103.
2017-03-23 09:11:33.803 [I] [S                                          on Pause) 22 milliseconds; start-end times 61129-61151.
2017-03-23 09:11:33.975 [I] [q                                          sion - created = '2017-03-23 09:11:33:975', last accessed =
2017-03-23 09:11:49.459 [I] [q                                          sion - created = '2017-03-23 09:11:33:975', last accessed =
2017-03-23 09:11:54.897 [I] [p                                          alInterface/api/heartbeat' with action 'GET' using authenti
2017-03-23 09:11:54.929 [E] [p                                          t must begin with '{' at character 1 of <html><body><p>There
2017-03-23 09:11:57.366 [I] [q
2017-03-23 09:11:57.397 [I] [q                                          r' [created = '2017-03-23 09:11:33:975', last accessed = '2
2017-03-23 09:11:57.678 [I] [q                                          r' [created = '2017-03-23 09:11:33:975', last accessed = '2
2017-03-23 09:11:57.678 [I] [q                                          r' [created = '2017-03-23 09:11:33:975', last accessed = '2
2017-03-23 09:11:57.678 [I] [q
2017-03-23 09:11:57.725 [I] [q
2017-03-23 09:11:57.725 [I] [q
2017-03-23 09:11:57.741 [I] [q                                          .50","connectedTo":"10.37.107.50","status":1,"vaiUsername":'
2017-03-23 09:12:06.147 [I] [q                                          r' [created = '2017-03-23 09:11:33:975', last accessed = '2
2017-03-23 09:12:06.303 [I] [q                                          r' [created = '2017-03-23 09:11:33:975', last accessed = '2
2017-03-23 09:12:06.335 [I] [q
2017-03-23 09:12:06.335 [I] [q
2017-03-23 09:12:06.428 [I] [q                                          r' [created = '2017-03-23 09:11:33:975', last accessed = '2
2017-03-23 09:12:06.444 [I] [qtp1215219214-43] Inside UnitRoleTreVOs Converter in VCTS-WEB-UI
2017-03-23 09:12:06.444 [I] [qtp1215219214-43] Inside UnitRolesVOs converter in VCTS-WEB-UI
2017-03-23 09:12:06.506 [I] [qtp1215219214-39] Host Name 10.37.107.171
2017-03-23 09:12:06.631 [I] [qtp1215219214-39] assignmentClientProperties **** {"name":"Rauland","source":"Rauland","subscriptionURI":"http://10.37.107.179:8088,
2017-03-23 09:12:06.631 [I] [qtp1215219214-39] assignmentClientVO.getAdapterId() 4
2017-03-23 09:12:06.647 [I] [qtp1215219214-39] voceraDrive V:
2017-03-23 09:12:08.741 [I] [qtp1215219214-44] Session '18z4v775kqecyd3w1h0mgvg1n' user 'Administrator' [created = '2017-03-23 09:11:33:975', last accessed = '2
2017-03-23 09:12:08.772 [I] [Service Thread] Young Gen GC: - 27 G1 Young Generation (from G1 Evacuation Pause) 22 milliseconds; start-end times 95971-95993.
2017-03-23 09:12:08.881 [I] [qtp1215219214-39] Session '18z4v775kqecyd3w1h0mgvg1n' user 'Administrator' [created = '2017-03-23 09:11:33:975', last accessed = '2
2017-03-23 09:12:08.897 [I] [qtp1215219214-39] AssignmentClient Entity's List size is : 3
2017-03-23 09:12:08.913 [I] [qtp1215219214-39] assignmentClients DTO's List size is : 3
```

Normal text file | length : 281,138 lines : 1,177 | Ln : 1,158 Col : 74 Sel : 26 | 1 | Windows (CR LF) | UTF-8 | INS

3. Confirm that the logs reference the connection you are looking for.

   For example, for a Rauland connection you should see the name, source, subscription URI, and other values you provided in the connection definition.



*V:\vocera\logs\vcts-log-2017-03-23-091033.txt - Notepad++ [Administrator]

Find what : assignmentClientProperties

☐ Match whole word only
☐ Match case
☑ Wrap around

Search Mode: ● Normal ○ Extended (\n, \r, \t, \0, \x...) ○ Regular expression ☐ . matches newline

Direction: ○ Up ● Down

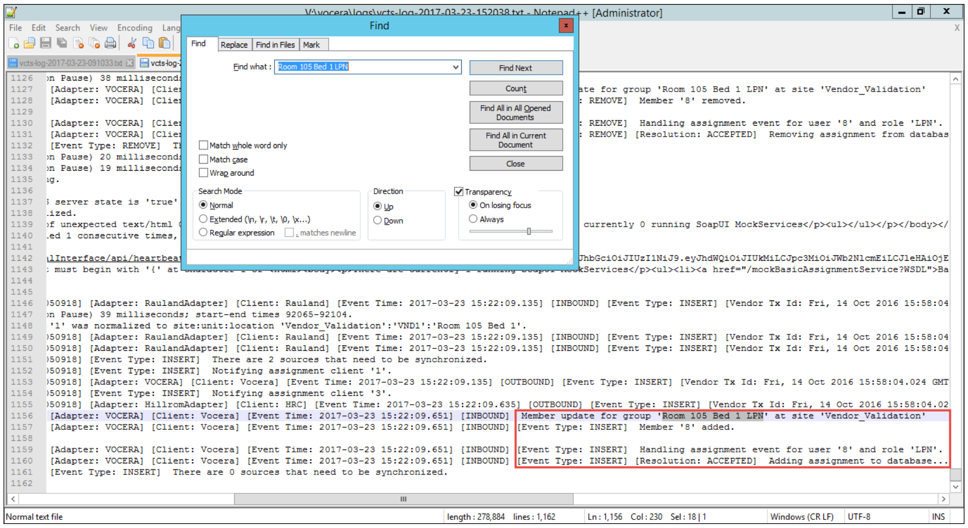
☑ Transparency: ● On losing focus ○ Always

2017-03-23 09:12:06.631 [I] [qtp1215219214-39] assignmentClientProperties **** {"name":"Rauland","source":"Rauland","subscriptionURI":"http://10.37.107.179:8088,

Normal text file | length : 281,138 lines : 1,177 | Ln : 1,158 Col : 74 Sel : 26 | 1 | Windows (CR LF) | UTF-8 | INS

4. Search for a role that the connection should have assigned, and confirm that you see an action such as "Member update" and a Resolution such as "Accepted"

   For example, if you are expecting an assignment for `Room 105 Bed 1 LPN`, you should see entries such as the following in the logs:



V:\vocera\logs\vcts-log-2017-03-23-152038.txt - Notepad++ [Administrator]

Find what : Room 105 Bed 1 LPN

□ Match whole word only
□ Match case
□ Wrap around

Search Mode: ● Normal ○ Extended (\n, \r, \t, \0, \x...) ○ Regular expression □ . matches newline

Direction: ● Up ○ Down

☑ Transparency: ● On losing focus ○ Always

```
n Pause) 38 milliseconds
 [Adapter: VOCERA] [Clie  ate for group 'Room 105 Bed 1 LPN' at site 'Vendor_Validation'
 [Adapter: VOCERA] [Clie  : REMOVE]  Member '8' removed.

 [Adapter: VOCERA] [Clie  : REMOVE]  Handling assignment event for user '8' and role 'LPN'.
 [Adapter: VOCERA] [Clie  : REMOVE] [Resolution: ACCEPTED]  Removing assignment from databas
 [Event Type: REMOVE]  Th
n Pause) 20 milliseconds
n Pause) 19 milliseconds
g.

5 server state is 'true'
ized.
of unexpected text/html (  currently 0 running SoapUI MockServices</p><ul></ul></p></body></
ed 1 consecutive times,

lInterface/api/heartbeat  JhbGciOiJIUzI1NiJ9.eyJhdWQiOiJIUkMiLCJpc3MiOiJWb2NlcmEiLCJleHAiOjE
: must begin with '{' at  kServices</p><ul><li><a href="/mockBasicAssignmentService?WSDL">Ba


50918] [Adapter: RaulandAdapter] [Client: Rauland] [Event Time: 2017-03-23 15:22:09.135] [INBOUND] [Event Type: INSERT] [Vendor Tx Id: Fri, 14 Oct 2016 15:58:04
n Pause) 39 milliseconds; start-end times 92065-92104.
'1' was normalized to site:unit:location 'Vendor_Validation':'VND1':'Room 105 Bed 1'.
50918] [Adapter: RaulandAdapter] [Client: Rauland] [Event Time: 2017-03-23 15:22:09.135] [INBOUND] [Event Type: INSERT] [Vendor Tx Id: Fri, 14 Oct 2016 15:58:04
50918] [Adapter: RaulandAdapter] [Client: Rauland] [Event Time: 2017-03-23 15:22:09.135] [INBOUND] [Event Type: INSERT] [Vendor Tx Id: Fri, 14 Oct 2016 15:58:04
50918] [Event Type: INSERT]  There are 2 sources that need to be synchronized.
50918] [Event Type: INSERT]  Notifying assignment client '1'.
50918] [Adapter: VOCERA] [Client: Vocera] [Event Time: 2017-03-23 15:22:09.135] [OUTBOUND] [Event Type: INSERT] [Vendor Tx Id: Fri, 14 Oct 2016 15:58:04.024 GMT
50918] [Event Type: INSERT]  Notifying assignment client '3'.
50918] [Adapter: HillromAdapter] [Client: HRC] [Event Time: 2017-03-23 15:22:09.635] [OUTBOUND] [Event Type: INSERT] [Vendor Tx Id: Fri, 14 Oct 2016 15:58:04.02
 [Adapter: VOCERA] [Client: Vocera] [Event Time: 2017-03-23 15:22:09.651] [INBOUND]  Member update for group 'Room 105 Bed 1 LPN' at site 'Vendor_Validation'
 [Adapter: VOCERA] [Client: Vocera] [Event Time: 2017-03-23 15:22:09.651] [INBOUND] [Event Type: INSERT]  Member '8' added.

 [Adapter: VOCERA] [Client: Vocera] [Event Time: 2017-03-23 15:22:09.651] [INBOUND] [Event Type: INSERT]  Handling assignment event for user '8' and role 'LPN'.
 [Adapter: VOCERA] [Client: Vocera] [Event Time: 2017-03-23 15:22:09.651] [INBOUND] [Event Type: INSERT] [Resolution: ACCEPTED]  Adding assignment to database...
 [Event Type: INSERT]  There are 0 sources that need to be synchronized.
```

Normal text file    length : 278,884    lines : 1,162    Ln : 1,156    Col : 230    Sel : 18 | 1    Windows (CR LF)    UTF-8    INS

File
Edit
Search
View
Encoding
Lang
vcts-log-2017-03-23-091033.txt
vcts-log-
Find
Replace
Find in Files
Mark
Find Next
Count
Find All in All Opened Documents
Find All in Current Document
Close

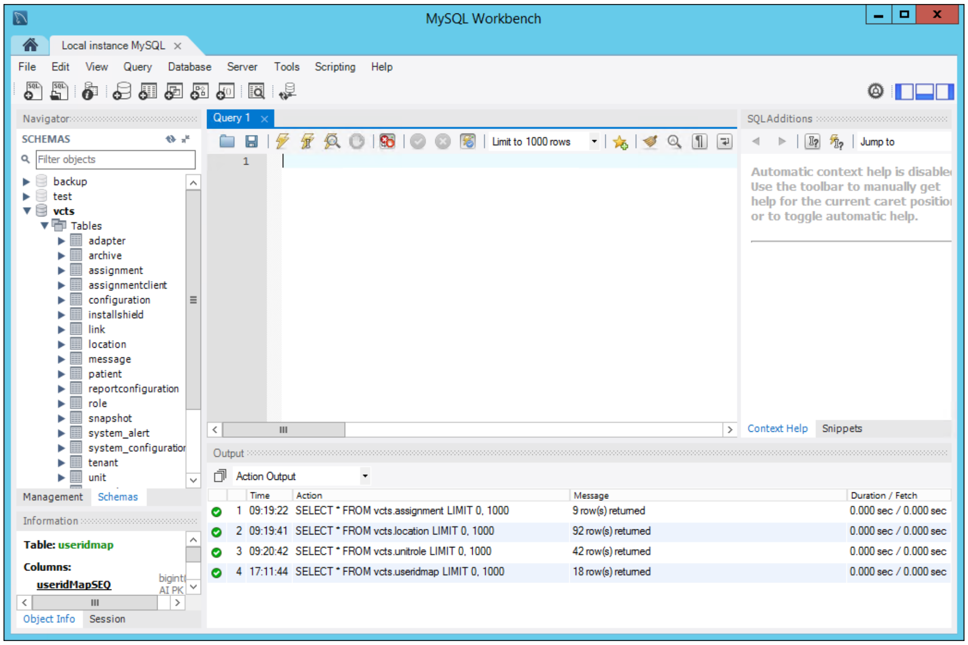
## How to Confirm Assignments in the CTS Database

Assignments that are visible in the logs should also be visible in the CTS database. If assignments appear in the logs but not in the CTS database, make sure that the user ID is mapped correctly between the source and target systems. This topic shows you how to confirm that expected assignments appear in the database.

To confirm assignments in the CTS database:

1. On the CTS machine, use the Windows interface to launch MySQL Workbench. If prompted, log into the `root` account with the password `vocera`.

   The MySQL Workbench user interface appears.



Automatic context help is disable
Use the toolbar to manually get
help for the current caret positio
or to toggle automatic help.

Table: useridmap

Columns:

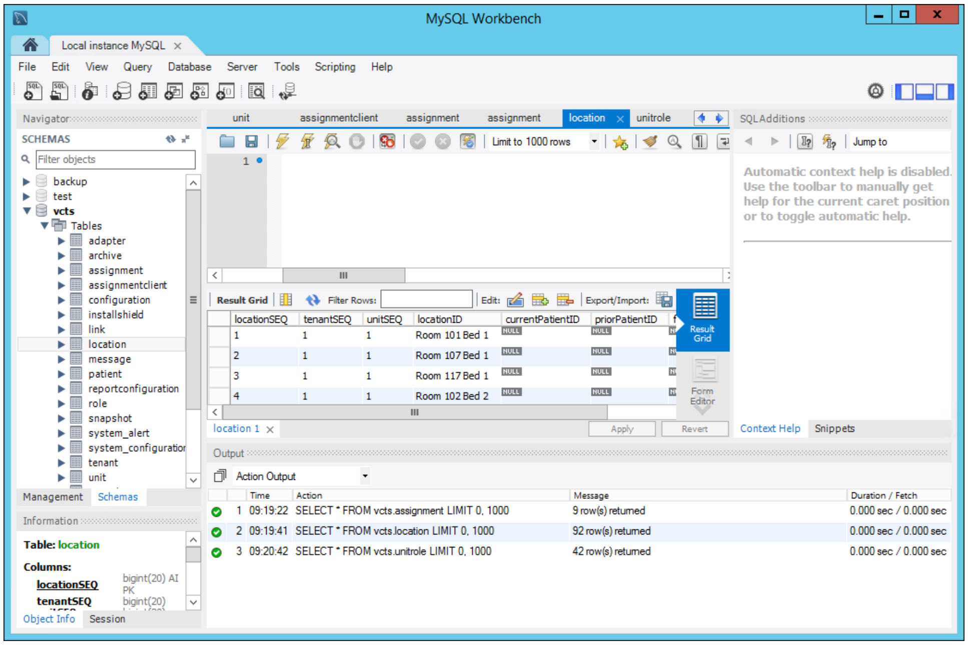
useridMapSEQ    bigint
AI PK

Action Output

|  |  | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|---|
| ✓ | 1 | 09:19:22 | SELECT * FROM vcts.assignment LIMIT 0, 1000 | 9 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ | 2 | 09:19:41 | SELECT * FROM vcts.location LIMIT 0, 1000 | 92 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ | 3 | 09:20:42 | SELECT * FROM vcts.unitrole LIMIT 0, 1000 | 42 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ | 4 | 17:11:44 | SELECT * FROM vcts.useridmap LIMIT 0, 1000 | 18 row(s) returned | 0.000 sec / 0.000 sec |

2. Right-click the `location` table and choose **Select Rows - Limit 1000**.

   A grid showing values in the rows and columns of the `location` table appears.



Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

| locationSEQ | tenantSEQ | unitSEQ | locationID | currentPatientID | priorPatientID |
|---|---|---|---|---|---|
| 1 | 1 | 1 | Room 101 Bed 1 | NULL | NULL |
| 2 | 1 | 1 | Room 107 Bed 1 | NULL | NULL |
| 3 | 1 | 1 | Room 117 Bed 1 | NULL | NULL |
| 4 | 1 | 1 | Room 102 Bed 2 | NULL | NULL |

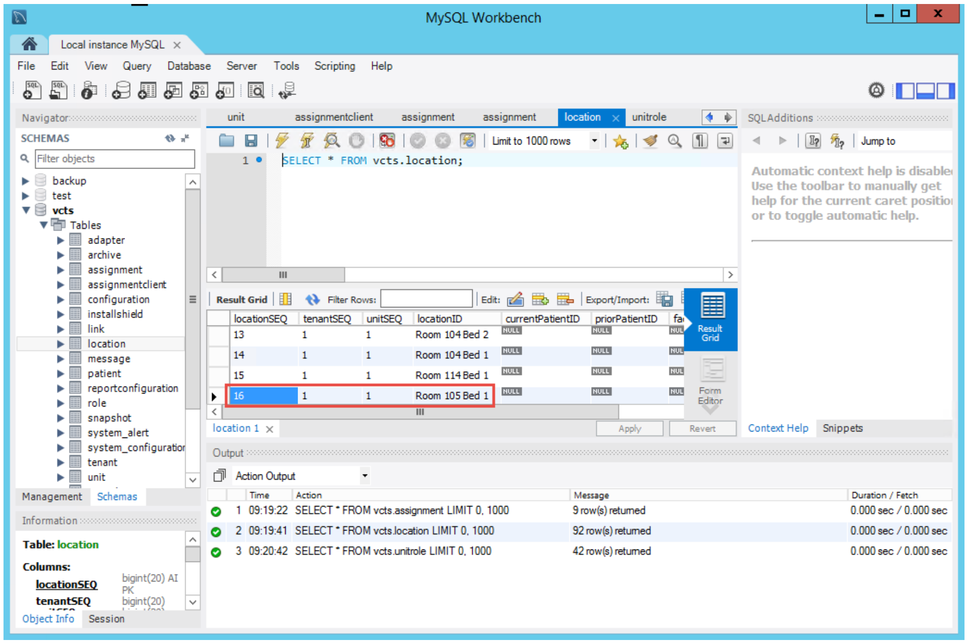
```
   Time      Action                                        Message              Duration / Fetch
1  09:19:22  SELECT * FROM vcts.assignment LIMIT 0, 1000   9 row(s) returned    0.000 sec / 0.000 sec
2  09:19:41  SELECT * FROM vcts.location LIMIT 0, 1000     92 row(s) returned   0.000 sec / 0.000 sec
3  09:20:42  SELECT * FROM vcts.unitrole LIMIT 0, 1000     42 row(s) returned   0.000 sec / 0.000 sec
```

Table: location

Columns:
locationSEQ bigint(20) AI PK
tenantSEQ bigint(20)

3. Scroll down in the table until you see a `locationID` that the connection should have assigned, and make a note of the `locationSEQ` value.

   For example, if you are expecting an assignment for the `Room 105 Bed 1` location, note that `locationSEQ` is set to 16:



```
SELECT * FROM vcts.location;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

| locationSEQ | tenantSEQ | unitSEQ | locationID | currentPatientID | priorPatientID |
|---|---|---|---|---|---|
| 13 | 1 | 1 | Room 104 Bed 2 | NULL | NULL |
| 14 | 1 | 1 | Room 104 Bed 1 | NULL | NULL |
| 15 | 1 | 1 | Room 114 Bed 1 | NULL | NULL |
| 16 | 1 | 1 | Room 105 Bed 1 | NULL | NULL |

```
   Time      Action                                        Message              Duration / Fetch
1  09:19:22  SELECT * FROM vcts.assignment LIMIT 0, 1000   9 row(s) returned    0.000 sec / 0.000 sec
2  09:19:41  SELECT * FROM vcts.location LIMIT 0, 1000     92 row(s) returned   0.000 sec / 0.000 sec
3  09:20:42  SELECT * FROM vcts.unitrole LIMIT 0, 1000     42 row(s) returned   0.000 sec / 0.000 sec
```

Table: location

Columns:
locationSEQ bigint(20) AI PK
tenantSEQ bigint(20)

4. Right-click the `assignment` table, choose **Select Rows - Limit 1000**, navigate to the `locationSEQ` whose value is set as you expected, and make a note of the `unitRoleSEQ` and `userId` values.

   The `userId` value uniquely identifies the user who has been assigned to this location. The `unitRoleSEQ` value allows you to determine the role that the user was assigned to.

   For example, navigate to the `locationSEQ` whose value is set to `16` and note that `unitRoleSEQ` is set to `1` and `userId` is set to `8`.

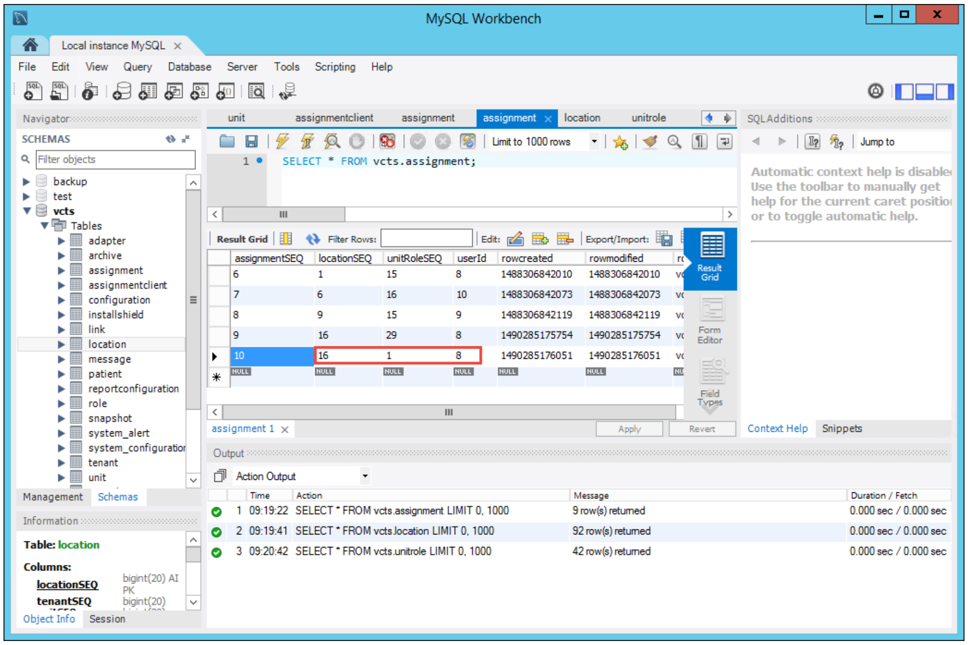


Table: location

Columns:
locationSEQ bigint(20) AI PK
tenantSEQ bigint(20)

```
1  SELECT * FROM vcts.assignment;
```

| assignmentSEQ | locationSEQ | unitRoleSEQ | userId | rowcreated | rowmodified | rc |
|---|---|---|---|---|---|---|
| 6 | 1 | 15 | 8 | 1488306842010 | 1488306842010 | vc |
| 7 | 6 | 16 | 10 | 1488306842073 | 1488306842073 | vc |
| 8 | 9 | 15 | 9 | 1488306842119 | 1488306842119 | vc |
| 9 | 16 | 29 | 8 | 1490285175754 | 1490285175754 | vc |
| 10 | 16 | 1 | 8 | 1490285176051 | 1490285176051 | vc |
| NULL | NULL | NULL | NULL | NULL | NULL | NU |

assignment 1

| | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|
| 1 | 09:19:22 | SELECT * FROM vcts.assignment LIMIT 0, 1000 | 9 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 09:19:41 | SELECT * FROM vcts.location LIMIT 0, 1000 | 92 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 09:20:42 | SELECT * FROM vcts.unitrole LIMIT 0, 1000 | 42 row(s) returned | 0.000 sec / 0.000 sec |

5. Right-click the `unitrole` table, choose **Select Rows - Limit 1000**, navigate to the `unitRoleSEQ` whose value is set as you expected, and make a note of the `role` value.

   This value should be the same as the role you intended to assign.

   For example, navigate to the `unitRoleSEQ` whose value is set to `1` and note that `role` is set to `LPN`.

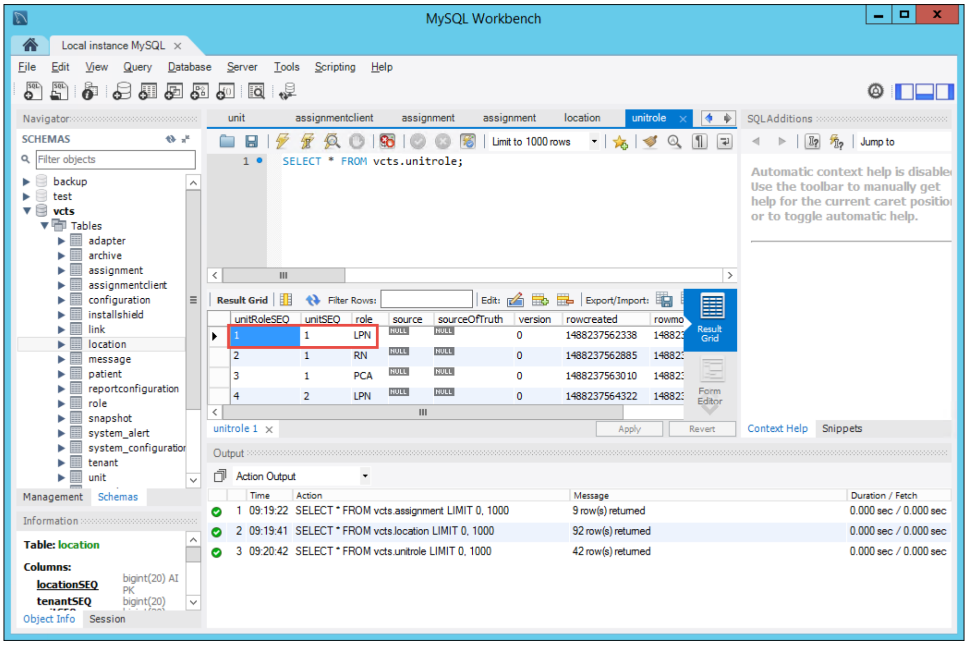


Table: location

Columns:
locationSEQ bigint(20) AI PK
tenantSEQ bigint(20)

```
1  SELECT * FROM vcts.unitrole;
```

| unitRoleSEQ | unitSEQ | role | source | sourceOfTruth | version | rowcreated | rowmo |
|---|---|---|---|---|---|---|---|
| 1 | 1 | LPN | NULL | NULL | 0 | 1488237562338 | 14882: |
| 2 | 1 | RN | NULL | NULL | 0 | 1488237562885 | 14882: |
| 3 | 1 | PCA | NULL | NULL | 0 | 1488237563010 | 14882: |
| 4 | 2 | LPN | NULL | NULL | 0 | 1488237564322 | 14882: |

unitrole 1

| | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|
| 1 | 09:19:22 | SELECT * FROM vcts.assignment LIMIT 0, 1000 | 9 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 09:19:41 | SELECT * FROM vcts.location LIMIT 0, 1000 | 92 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 09:20:42 | SELECT * FROM vcts.unitrole LIMIT 0, 1000 | 42 row(s) returned | 0.000 sec / 0.000 sec |

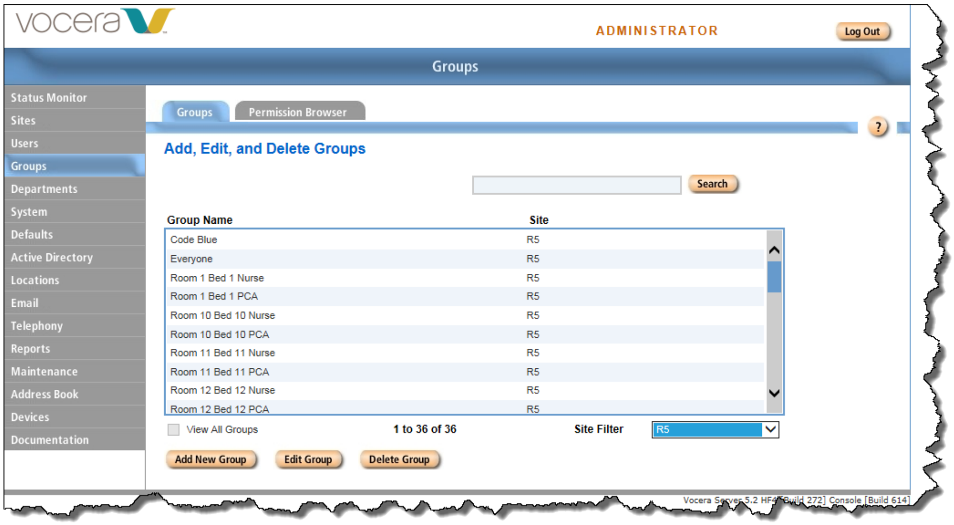
## How to Confirm Assignments in the Vocera Voice Server

If the foreign clinical system is the **Source of Truth**, the Vocera Voice Server is the target system and should be receiving assignments. Check the VS to confirm that expected assignments are being made.

To confirm assignments in the Vocera Voice Server:

1. Log into the VS Administration Console and click **Groups** in the navigation bar on the left.

   The **Groups** page appears.



Add, Edit, and Delete Groups

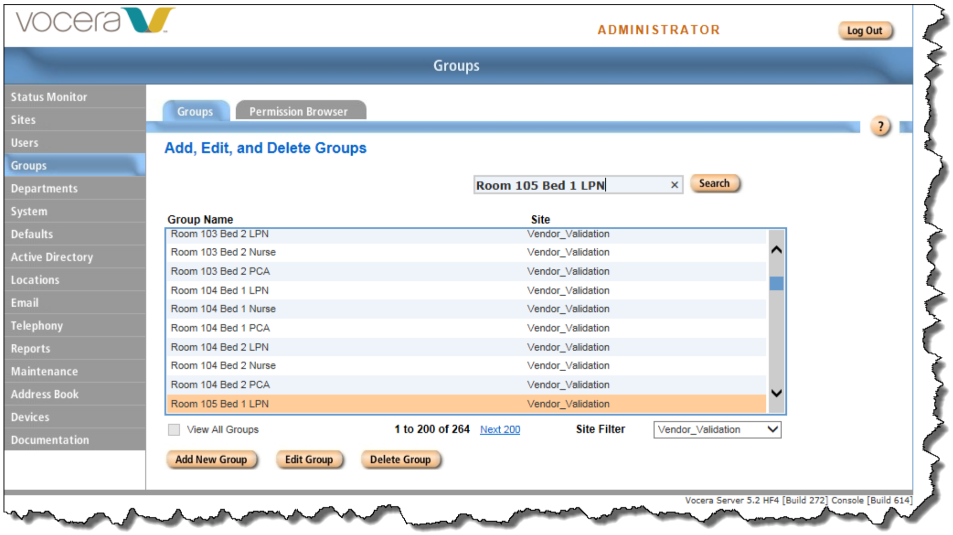
| Group Name | Site |
|---|---|
| Code Blue | R5 |
| Everyone | R5 |
| Room 1 Bed 1 Nurse | R5 |
| Room 1 Bed 1 PCA | R5 |
| Room 10 Bed 10 Nurse | R5 |
| Room 10 Bed 10 PCA | R5 |
| Room 11 Bed 11 Nurse | R5 |
| Room 11 Bed 11 PCA | R5 |
| Room 12 Bed 12 Nurse | R5 |
| Room 12 Bed 12 PCA | R5 |

□ View All Groups 1 to 36 of 36 Site Filter R5

Vocera Server 5.2 HF4 [Build 272] Console [Build 614]

2. In the search box, enter the name of a role-and-location based group that should have received an assignment and click **Search**.

   The group appears and is selected in the scrolling list of groups.



Add, Edit, and Delete Groups

Room 105 Bed 1 LPN

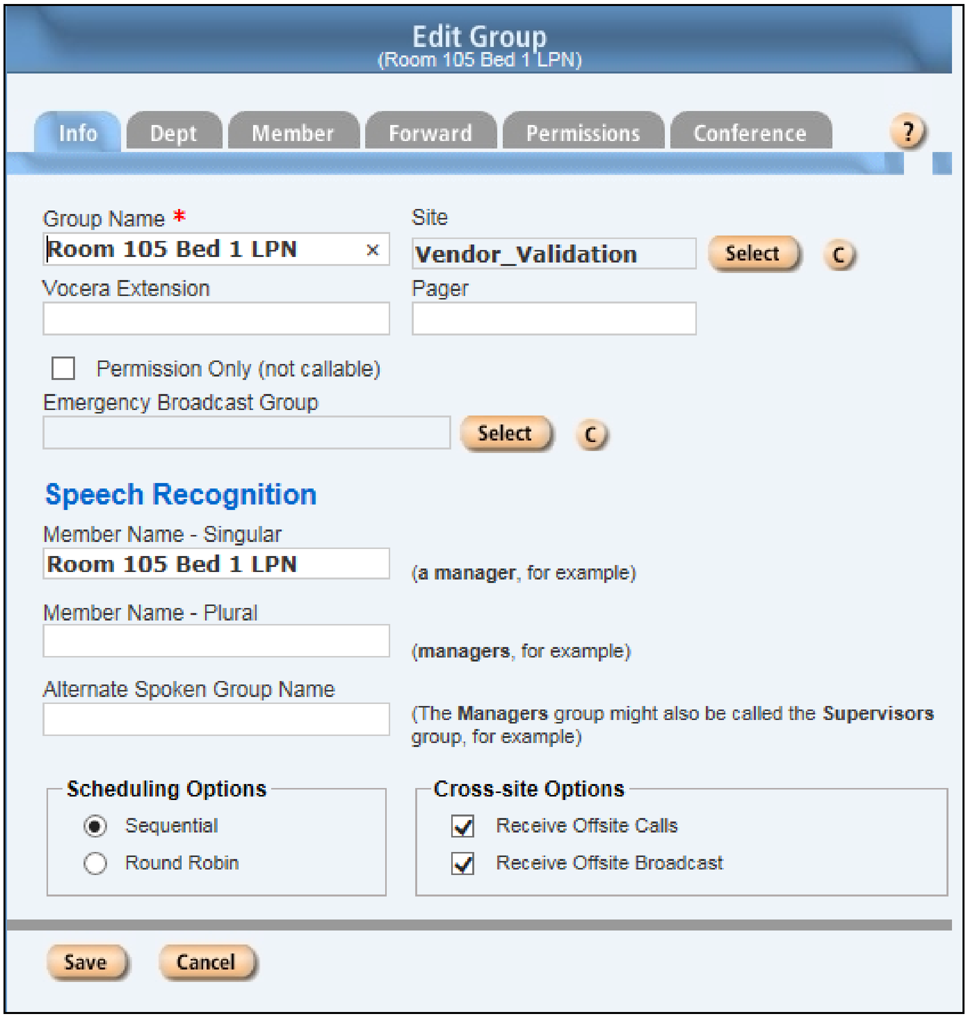
| Group Name | Site |
|---|---|
| Room 103 Bed 2 LPN | Vendor_Validation |
| Room 103 Bed 2 Nurse | Vendor_Validation |
| Room 103 Bed 2 PCA | Vendor_Validation |
| Room 104 Bed 1 LPN | Vendor_Validation |
| Room 104 Bed 1 Nurse | Vendor_Validation |
| Room 104 Bed 1 PCA | Vendor_Validation |
| Room 104 Bed 2 LPN | Vendor_Validation |
| Room 104 Bed 2 Nurse | Vendor_Validation |
| Room 104 Bed 2 PCA | Vendor_Validation |
| Room 105 Bed 1 LPN | Vendor_Validation |

□ View All Groups 1 to 200 of 264 Next 200 Site Filter Vendor_Validation

Vocera Server 5.2 HF4 [Build 272] Console [Build 614]

3. Click **Edit**.

   The **Edit Group** dialog box appears.



Edit Group
(Room 105 Bed 1 LPN)

Group Name * Room 105 Bed 1 LPN

Site Vendor_Validation

Vocera Extension

Pager

□ Permission Only (not callable)

Emergency Broadcast Group

Speech Recognition

Member Name - Singular Room 105 Bed 1 LPN (a **manager**, for example)

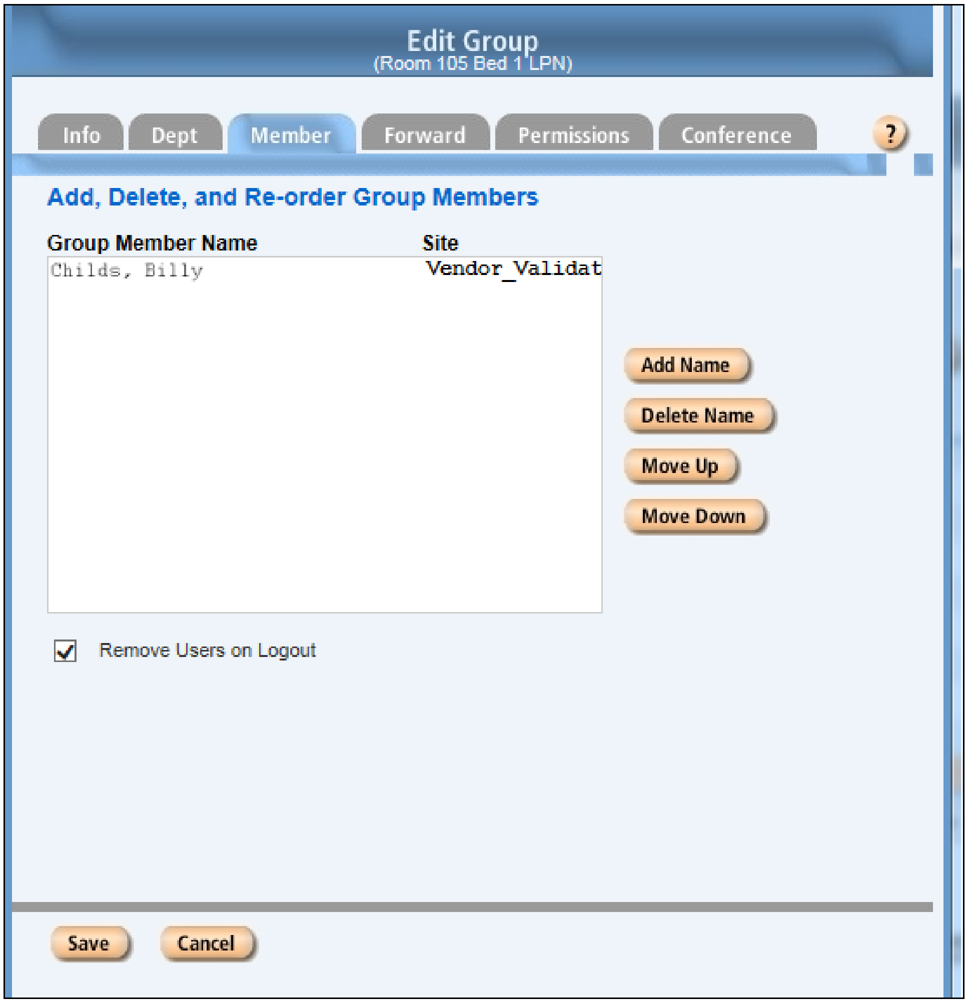
Member Name - Plural (**managers**, for example)

Alternate Spoken Group Name (The **Managers** group might also be called the **Supervisors** group, for example)

Scheduling Options
- ● Sequential
- ○ Round Robin

Cross-site Options
- ☑ Receive Offsite Calls
- ☑ Receive Offsite Broadcast

4. Click the **Members** tab.
   The expected assignment should be visible in **Members** list.



Edit Group
(Room 105 Bed 1 LPN)

Add, Delete, and Re-order Group Members

| Group Member Name | Site |
|---|---|
| Childs, Billy | Vendor_Validat |

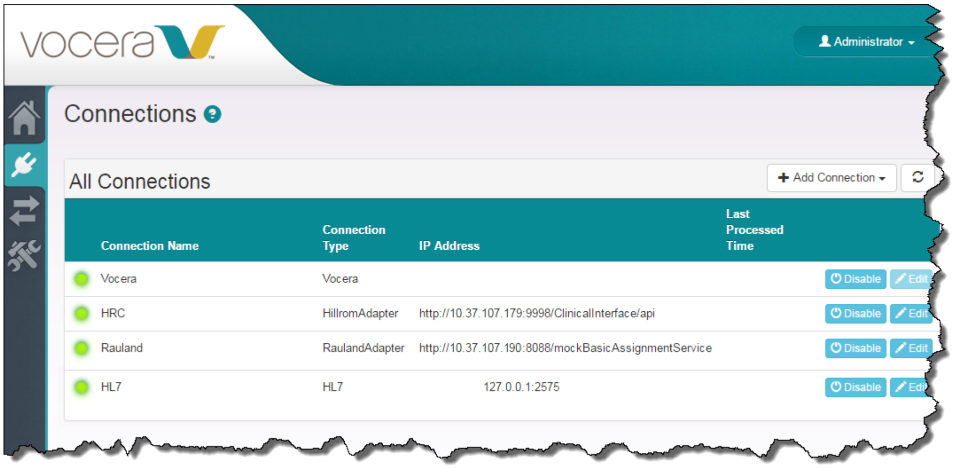
☑ Remove Users on Logout

## How to Disable a Connection

You can disable a connection if you need to stop publishing or subscribing to data for any reason.

To disable a connection:

1. In the CTS Console, click the **Connections** icon ( 🔌 ) in the navigation bar on the left.
   The **Connections** screen appears.



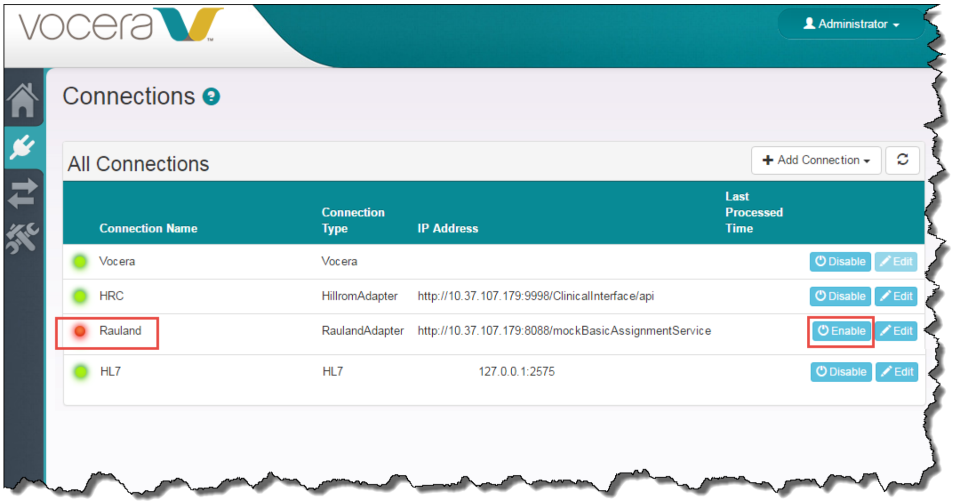
Connections

All Connections

| Connection Name | Connection Type | IP Address | Last Processed Time | |
|---|---|---|---|---|
| Vocera | Vocera | | | Disable Edit |
| HRC | HillromAdapter | http://10.37.107.179:9998/ClinicalInterface/api | | Disable Edit |
| Rauland | RaulandAdapter | http://10.37.107.190:8088/mockBasicAssignmentService | | Disable Edit |
| HL7 | HL7 | 127.0.0.1:2575 | | Disable Edit |

2. Click the **Disable** button that is in line with any connection.

   A red light icon (disabled) appears next to the connection in the **Connection Name** column, and the **Disable** button toggles to show **Enable**.



| | Connection Name | Connection Type | IP Address | Last Processed Time | |
|---|---|---|---|---|---|
| 🟢 | Vocera | Vocera | | | Disable Edit |
| 🟢 | HRC | HillromAdapter | http://10.37.107.179:9998/ClinicalInterface/api | | Disable Edit |
| 🔴 | Rauland | RaulandAdapter | http://10.37.107.179:8088/mockBasicAssignmentService | | Enable Edit |
| 🟢 | HL7 | HL7 | 127.0.0.1:2575 | | Disable Edit |

3. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

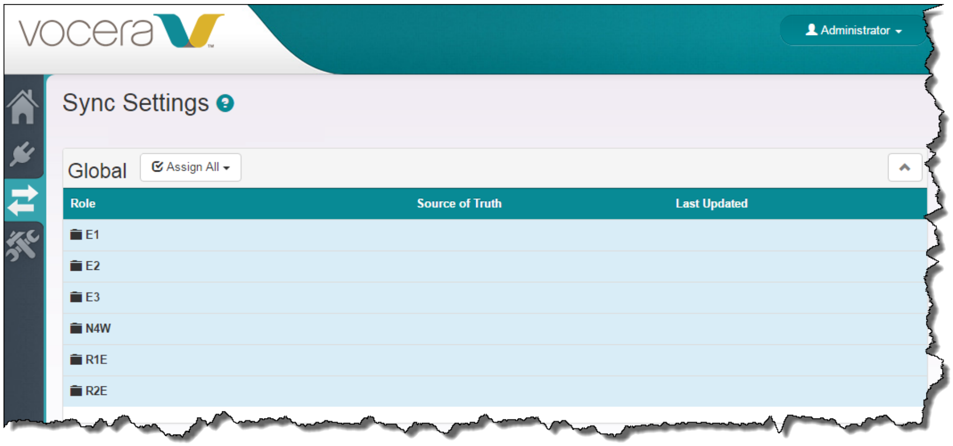
## How to Stop Syncing

When you stop synchronizing data, the source of truth no longer sends care team personnel updates to the subscribing system. If you need to stop synchronizing data for any reason, set the **Source of Truth** to **None** and restart the Vocera CTS service. This technique is useful when you temporarily want to prevent updates from occurring without affecting configuration parameters.

To stop synchronizing data:

1. In the CTS Console, click the **Sync Settings** icon ( ⇄ ) in the navigation bar on the left.

   The **Sync Settings** page appears.



| Role | Source of Truth | Last Updated |
|---|---|---|
| 📁 E1 | | |
| 📁 E2 | | |
| 📁 E3 | | |
| 📁 N4W | | |
| 📁 R1E | | |
| 📁 R2E | | |

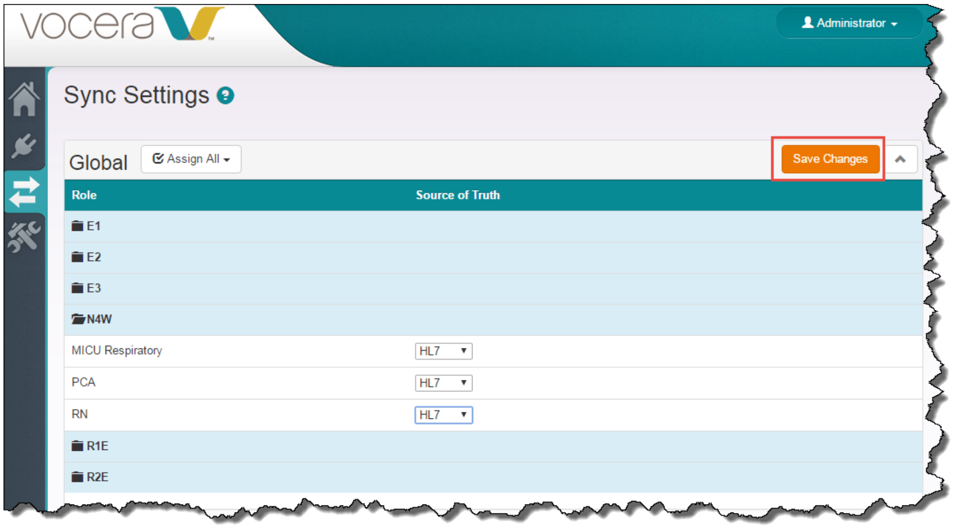
2. Do either of the following:

   - Click the folder icon ( 📁 ) next to the appropriate unit to display its roles, and then set the **Source of Truth** for the appropriate role to **None**.

  Perform this action for other unit and role combinations, if necessary.

- Use the **Assign All** list that appears next to the site name and set the **Source of Truth** to **None**. This action sets the **Source of Truth** for each unit and role combination in the site to **None**.

The **Save Changes** button appears.



| Role | Source of Truth |
|---|---|
| E1 | |
| E2 | |
| E3 | |
| N4W | |
| MICU Respiratory | HL7 |
| PCA | HL7 |
| RN | HL7 |
| R1E | |
| R2E | |

3. Click the **Save Changes** button.
   The changes are saved in the database, but assignment synchronization does not take place until you restart the Vocera CTSservice.
4. Restart the Vocera CTS service.

   Open the **Windows Services Control Panel**, navigate to the Vocera CTS service, right-click it, and choose **Restart**.

   The system reads the connection parameters and related information into memory, and any changes that you have made take effect.

## How to Change the Default Database Restore Time

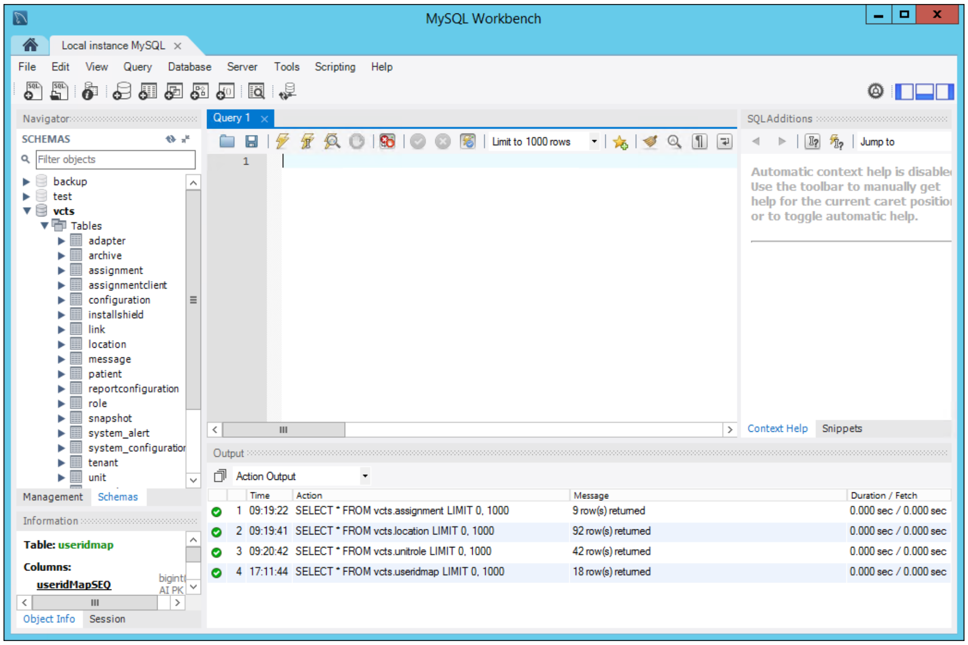
In certain situations, restoring an extremely large backup file may time out, resulting in an error message stating `The backup could not be restored within the max permitted time of 15 minutes`. If necessary, you can increase the default time out value for a restore to let the process continue.

**Note:** Make sure you contact Vocera Customer Support before changing this time out value.

To change the default database restore time out:

1. On the CTS machine, use the Windows interface to launch MySQL Workbench. If prompted, log into the `root` account with the password `vocera`.

   The MySQL Workbench user interface appears.



Automatic context help is disable
Use the toolbar to manually get
help for the current caret positio
or to toggle automatic help.

Table: useridmap

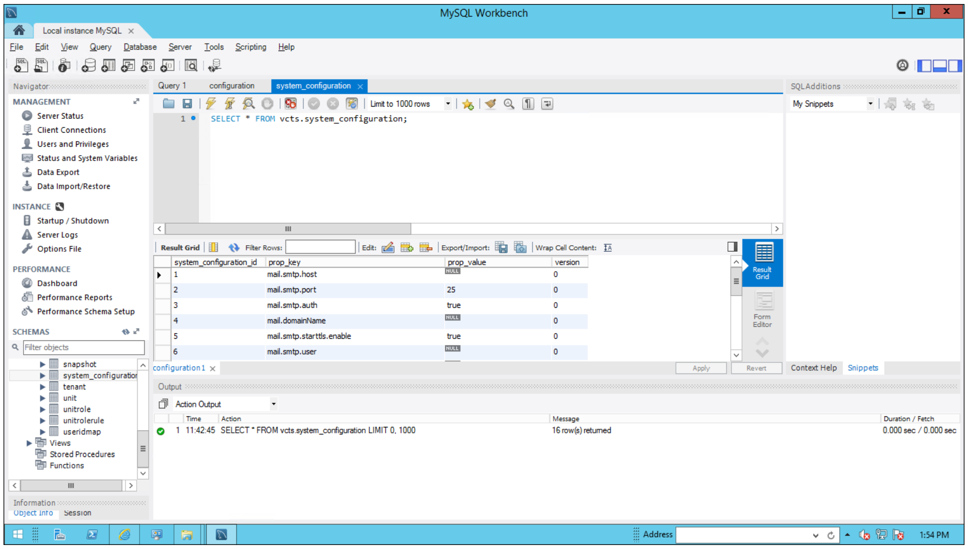
Columns:

useridMapSEQ bigint AI PK

|  | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|
| 1 | 09:19:22 | SELECT * FROM vcts.assignment LIMIT 0, 1000 | 9 row(s) returned | 0.000 sec / 0.000 sec |
| 2 | 09:19:41 | SELECT * FROM vcts.location LIMIT 0, 1000 | 92 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 09:20:42 | SELECT * FROM vcts.unitrole LIMIT 0, 1000 | 42 row(s) returned | 0.000 sec / 0.000 sec |
| 4 | 17:11:44 | SELECT * FROM vcts.useridmap LIMIT 0, 1000 | 18 row(s) returned | 0.000 sec / 0.000 sec |

2. Right-click the `system_configuration` table and choose **Select Rows - Limit 1000**.

   A grid showing values in the rows and columns of the `system_configuration` table appears.

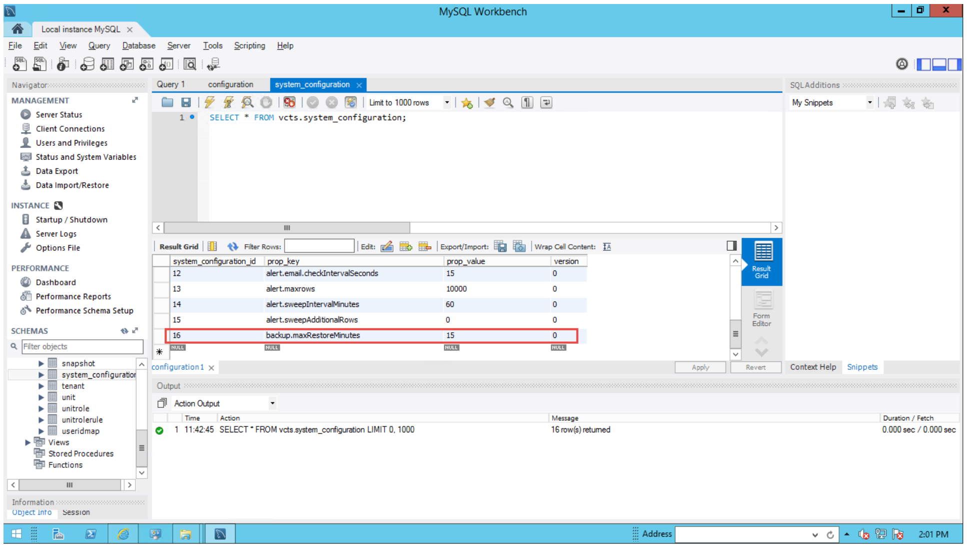


```
SELECT * FROM vcts.system_configuration;
```

| system_configuration_id | prop_key | prop_value | version |
|---|---|---|---|
| 1 | mail.smtp.host | NULL | 0 |
| 2 | mail.smtp.port | 25 | 0 |
| 3 | mail.smtp.auth | true | 0 |
| 4 | mail.domainName | NULL | 0 |
| 5 | mail.smtp.starttls.enable | true | 0 |
| 6 | mail.smtp.user | NULL | 0 |

|  | Time | Action | Message | Duration / Fetch |
|---|---|---|---|---|
| 1 | 11:42:45 | SELECT * FROM vcts.system_configuration LIMIT 0, 1000 | 16 row(s) returned | 0.000 sec / 0.000 sec |

3. Scroll down in the table until you see the `prop_key` whose value is `backup.maxRestoreMinutes`.



```
SELECT * FROM vcts.system_configuration;
```

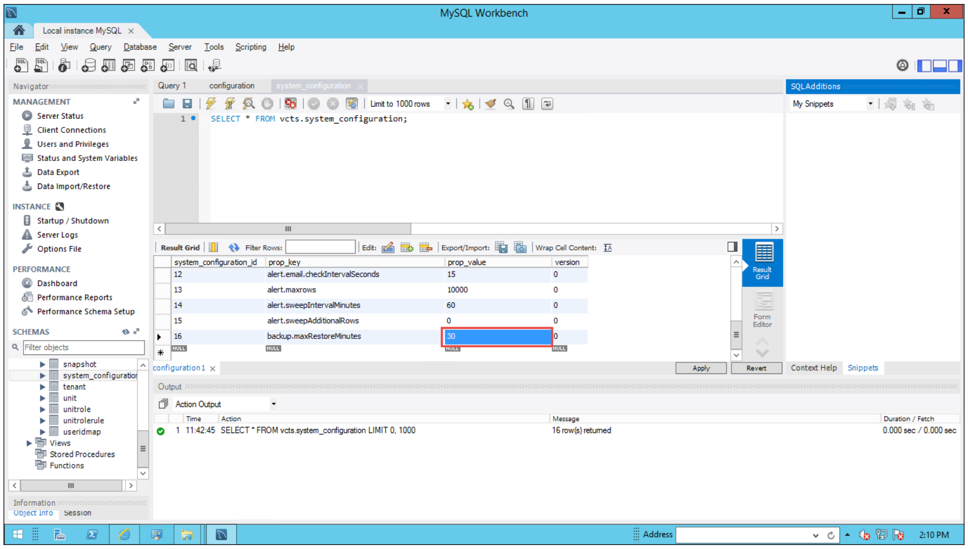
| system_configuration_id | prop_key | prop_value | version |
|---|---|---|---|
| 12 | alert.email.checkIntervalSeconds | 15 | 0 |
| 13 | alert.maxrows | 10000 | 0 |
| 14 | alert.sweepIntervalMinutes | 60 | 0 |
| 15 | alert.sweepAdditionalRows | 0 | 0 |
| 16 | backup.maxRestoreMinutes | 15 | 0 |
| NULL | NULL | NULL | NULL |

```
   Time      Action                                              Message              Duration / Fetch
1  11:42:45  SELECT * FROM vcts.system_configuration LIMIT 0, 1000  16 row(s) returned  0.000 sec / 0.000 sec
```

4. In the `backup.maxRestoreMinutes` row, click the value in the `prop_value` column. By default, this value is `15`.

   The value is selected, and the table is placed in edit mode.

5. Delete the existing value, enter a replacement value, and click away from the cell to deselect it.



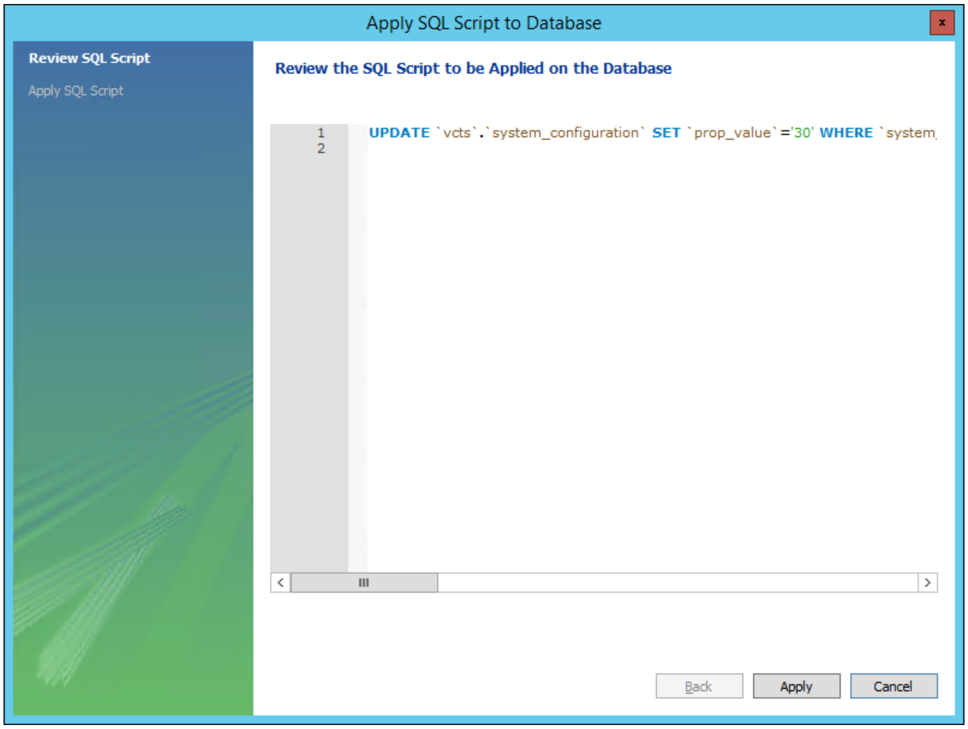
```
SELECT * FROM vcts.system_configuration;
```

| system_configuration_id | prop_key | prop_value | version |
|---|---|---|---|
| 12 | alert.email.checkIntervalSeconds | 15 | 0 |
| 13 | alert.maxrows | 10000 | 0 |
| 14 | alert.sweepIntervalMinutes | 60 | 0 |
| 15 | alert.sweepAdditionalRows | 0 | 0 |
| 16 | backup.maxRestoreMinutes | 30 | 0 |
| NULL | NULL | NULL | NULL |

```
   Time      Action                                              Message              Duration / Fetch
1  11:42:45  SELECT * FROM vcts.system_configuration LIMIT 0, 1000  16 row(s) returned  0.000 sec / 0.000 sec
```

6. Click the **Apply** button.

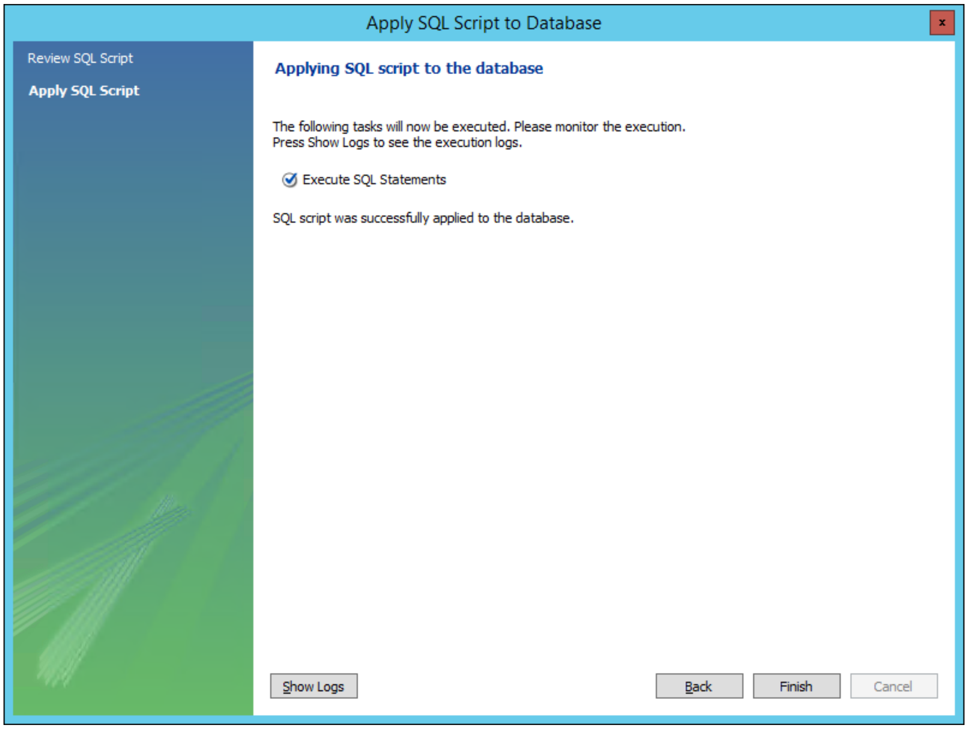
   The **Apply SQL Script to Database** window appears, displaying an SQL script with the proposed change.



Review the SQL Script to be Applied on the Database

```
UPDATE `vcts`.`system_configuration` SET `prop_value`='30' WHERE `system_
```

7. Review the proposed change and then click **Apply** in the **Apply SQL Script to Database** window.
   The **Apply SQL Script to Database** window displays a message indicating it will run the SQL script.



Applying SQL script to the database

The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

☑ Execute SQL Statements

SQL script was successfully applied to the database.

8. Click the **Finish** button.
   The new value is posted to the database.

# Appendixes

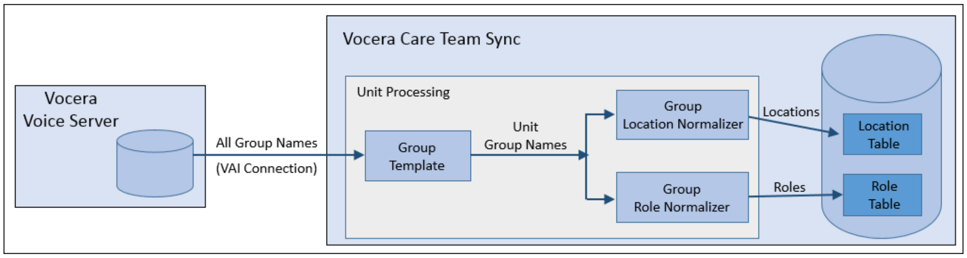
These topics provide some supplementary and reference information for Vocera Care Team Sync.

## Creating Group Templates and Normalizers for Units

When you add a unit in CTS, you define specialized components called *group templates* and *group normalizers* that process group names provided by the Vocera Voice Server. This section provides background on these components, explains why they are necessary, and shows you how to create them.

Group templates use a simplified regular expression notation to filter Vocera Voice Server groups, allowing only role-and-location based groups into CTS management. Group normalizers use a Java regular expression notation to process the names output by the templates, providing a set of allowed location names and role names for the CTS database.

The following diagram shows the relationship between the Vocera Voice Server and the templates and normalizers that process group names for CTS.



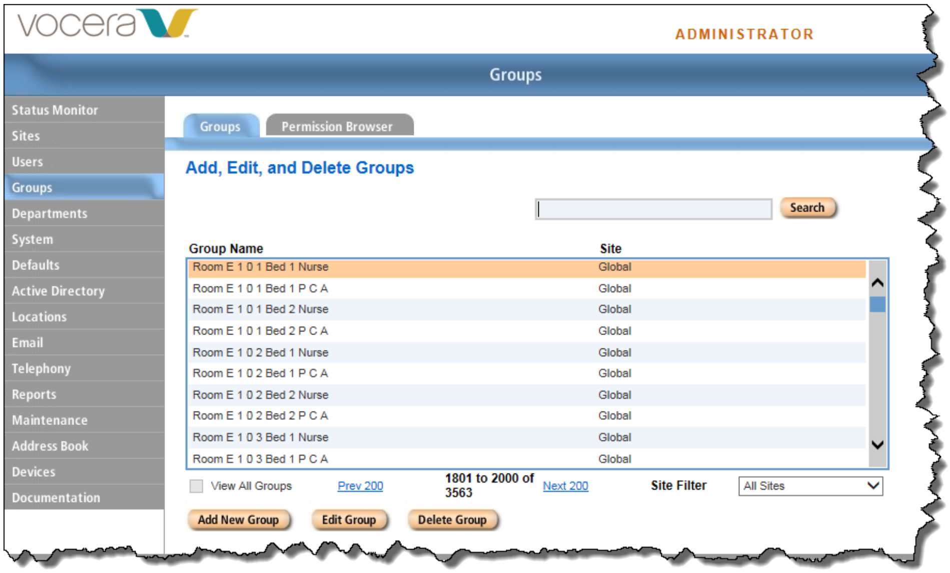
### Creating Group Templates for Units

The group template is a filter that you specify in the unit definition; it allows a set of Vocera Voice Server role-and-location based groups that you will manage in CTS, and it filters out all other groups. The output of the template is further processed by the group normalizers that are also referenced in the unit definition.

### About Group Naming Conventions

Role-based groups for specific locations are usually named in a predictable pattern in the Vocera Voice Server so users can remember how to reference them in voice commands. For example, it is fairly common to see a naming convention such as **Room** *RoomNumber* **Bed** *BedNumber RoleName*, where the fields in the convention have either constant values or vary in a predictable pattern.

This pattern is used as a naming convention for specific role-and-location based group names such as the following shown in the VS Administration Console:

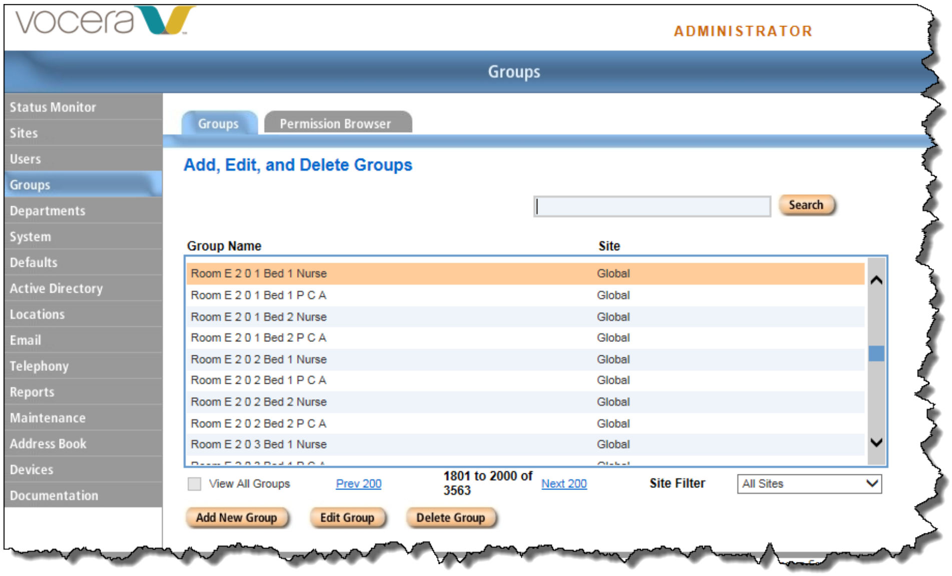


Add, Edit, and Delete Groups

| Group Name | Site |
|---|---|
| Room E 1 0 1 Bed 1 Nurse | Global |
| Room E 1 0 1 Bed 1 P C A | Global |
| Room E 1 0 1 Bed 2 Nurse | Global |
| Room E 1 0 1 Bed 2 P C A | Global |
| Room E 1 0 2 Bed 1 Nurse | Global |
| Room E 1 0 2 Bed 1 P C A | Global |
| Room E 1 0 2 Bed 2 Nurse | Global |
| Room E 1 0 2 Bed 2 P C A | Global |
| Room E 1 0 3 Bed 1 Nurse | Global |
| Room E 1 0 3 Bed 1 P C A | Global |

1801 to 2000 of 3563

The naming convention is often replicated from one unit to another, since hospital units often have a systematic structure. For example, The E2 unit could have the same convention:



Add, Edit, and Delete Groups

| Group Name | Site |
|---|---|
| Room E 2 0 1 Bed 1 Nurse | Global |
| Room E 2 0 1 Bed 1 P C A | Global |
| Room E 2 0 1 Bed 2 Nurse | Global |
| Room E 2 0 1 Bed 2 P C A | Global |
| Room E 2 0 2 Bed 1 Nurse | Global |
| Room E 2 0 2 Bed 1 P C A | Global |
| Room E 2 0 2 Bed 2 Nurse | Global |
| Room E 2 0 2 Bed 2 P C A | Global |
| Room E 2 0 3 Bed 1 Nurse | Global |

1801 to 2000 of 3563

These systematic naming conventions can be captured efficiently in a template by using a simplified form of regular expressions.

## Regular Expressions for Unit Templates

Because role-based group names are so systematic, you can use templates to filter out the Vocera Voice Server groups whose assignments you will not maintain in CTS. These templates are specified with a simplified regular expression notation when you add a unit.

*About Group Naming Conventions* on page 96 describes the patterns that are typically found within the names of role-based groups on the Vocera Voice Server. It is easier to visualize the individual components of this systematic pattern by representing the group name data in tables such as the following:

*Table 6: Room, bed, and role names for the E1 unit*

| Room | RoomNumber | Bed | BedNumber | RoleName |
|---|---|---|---|---|
| Room | E 1 0 1 | Bed | 1 | Nurse |
| Room | E 1 0 1 | Bed | 1 | PCA |
| Room | E 1 0 1 | Bed | 2 | Nurse |
| Room | E 1 0 1 | Bed | 2 | PCA |
| Room | E 1 0 2 | Bed | 1 | Nurse |
| Room | E 1 0 2 | Bed | 1 | PCA |
| Room | E 1 0 2 | Bed | 2 | Nurse |
| Room | E 1 0 2 | Bed | 2 | PCA |

*Table 7: Room, bed, and role names for the E2 unit*

| Room | RoomNumber | Bed | BedNumber | RoleName |
|---|---|---|---|---|
| Room | E 2 0 1 | Bed | 1 | Nurse |
| Room | E 2 0 1 | Bed | 1 | PCA |
| Room | E 2 0 1 | Bed | 2 | Nurse |
| Room | E 2 0 1 | Bed | 2 | PCA |
| Room | E 2 0 2 | Bed | 1 | Nurse |
| Room | E 2 0 2 | Bed | 1 | PCA |
| Room | E 2 0 2 | Bed | 2 | Nurse |
| Room | E 2 0 2 | Bed | 2 | PCA |

In the above tables, notice that the values in columns 1 and 3 do not vary—they are *constants*. Notice also that while the values in columns 2 and 4 vary, they do so systematically—column 4 contains a numeric value, and column 2 contains a string and a numeric value. Finally, the values in column 5 vary, and it contains only strings.

The template regular expression notation accommodates all these variations. Templates may contain three types of parts, as shown in the following table:

*Table 8: Examples of Template Parts*

| Template Part Type | Template Part Syntax | Template Examples | Allows |
|---|---|---|---|
| Constant | {*Value*} | `{ICU}`<br>`{Room}` | ICU<br>Room |
| String | {$*VarName*} | `{$Role}` | Nurse<br>PCA |
| Numeric | {$*VarName*#} | `{$Room#}` | 1<br>2 |

As you can see from the examples in the above table, values enclosed in curly braces ( `{ }` ) are template parts. Values that are prefixed with a dollar sign ( `$` ) are variables, and values appended with a hash ( `#` ) are numeric parts.

Numeric and string template parts may contain up to two modifiers enclosed in parentheses, shown as `(m,n)` in the syntax table below. In this syntax notation, the modifiers are defined as follows:

- *m* specifies whether the part is optional (m=0) or required (m=1).

- *n* specifies the maximum number of words in the part, where a word is a value delimited by spaces. *n* may vary from 0 to 9.

### *Table 9: Examples of Template Parts with Modifiers*

| Template Part Type | Template Part Syntax | Template Examples | Allows |
|---|---|---|---|
| String | {$*VarName*(*m*,*n*)} | `{$Role(1,2)}` | Nurse<br>PCA *(but not "P C A")* |
| Numeric | {$*VarName*#(*m*,*n*)} | `{$Room#(1,3)}` | 1<br>5 1 7 |

Modifiers for numeric and string template parts may include an optional range, specified as follows.

- 'x' 'y'... 'z' specifies a list of allowed strings in String Template parts. Use single quotes to enclose each allowed string.
- [*xxx*-*yyy*] specifies a range of zero through three digits in Numeric Template parts. Use square brackets to enclose this range.

### *Table 10: Examples of Template Parts with Modifiers and Ranges*

| Template Part Type | Template Part Syntax | Template Examples | Allows |
|---|---|---|---|
| String | {$*VarName*('*x*' '*y*'... '*z*',*m*,*n*)} | `{$Role('Nurse' 'PCA',1,2)}` | Nurse or PCA only (not LPN or "P C A") |
| Numeric | {$*VarName*#([*xxx*-*yyy*],*m*,*n*)} | `{$Room#([32-45],1,3)}` | 32 to 45, inclusive |

## Unit Template Examples

This topic provides examples of unit templates with a brief commentary on some of their usage subtleties.

The following table provides some simple examples of templates without modifiers or ranges. When modifiers are not specified, each part is assumed to be required and allows only a single word.

### *Table 11: Examples of Templates with Required Parts and Single Words*

| Template | Allows | Filters Out |
|---|---|---|
| {4 West} {Room} {$Room#} {$Role} | 4 West Room 101 Nurse<br>4 West Room 102 PCA<br>4 West Room 301 Nurse | 5 West Room 101 Nurse<br>4 West Room 101<br>4 West Room 1 0 1 Nurse |

In the previous example, notice that the template filters out the `4 West Room 1 0 1 Nurse` group, even though it allows the group named `4 West Room 101 Nurse`. Because the template does not use a modifier, the input room number must be a single word.

The following table provides examples of template parts that use modifiers to further restrict the allowed group names.

### *Table 12: Examples of Templates with Modifiers*

| Template | Allows | Filters Out |
|---|---|---|
| {E D} {Pod} {$Pod#(1,2)} {$Role(1,3)} | E D Pod 1 0 Nurse<br>E D Pod 1 P C A<br>E D Pod 1 0 1 Nurse | ED Pod 1 0 Nurse<br>E R Pod 1 0 P C A |

In the above example, notice that while `E D Pod 1 0 1 Nurse` is allowed by the template, it is not allowed as a Pod whose number is "101" and a Role whose name is "Nurse"; instead, it allows a group name with Pod number "10" and with a role named "1 Nurse". Because the `{$Pod#(1,2)}` modifier constrains the Pod number to two words, the initial "1 0" is interpreted as the number; however, the {$Role(1,3)} modifier allows up to three words, and the next two words of input ("1 Nurse") are identified as the Role. A template with a pre-filter such as this requires additional processing in the group name normalizer to achieve the expected behavior.

Finally, the following table provides sample templates with both modifiers and ranges to further limit allowed group names.

### *Table 13: Examples of Templates with Modifiers and Ranges*

| Template | Allows | Filters Out |
|---|---|---|
| {4 West} {Room} {$Room#([400-499],1,3)} {$Role('Nurse' 'P C A',1,3)} | 4 West Room 400 Nurse<br>4 West Room 4 0 0 Nurse<br>4 West Room 4 0 0 P C A | 4 West Room 100 Nurse<br>4 West Room 400 PCA |

In the previous example, notice that the template allows a group named `4 West Room 4 0 0 Nurse`, even though the Room number is "4 0 0" and the range is set to 400-499. A numeric range ignores the spaces between the digits in the input value and allows the group name.

Also in the previous example, notice that the template filters out a group named `4 West Room 400 PCA`, even though the range includes the "P C A" string. A string range allows only exact matches for character spacing, although it ignores the case of the characters.

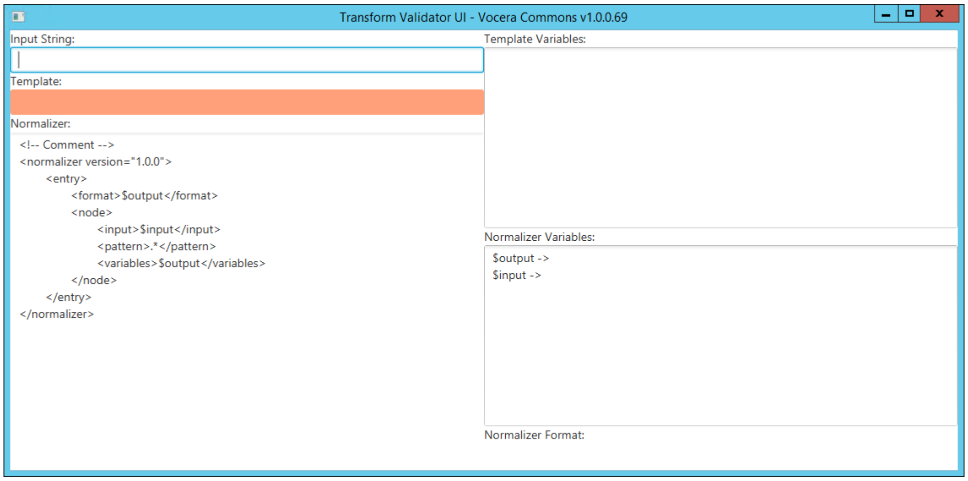
## How to Test a Unit Template

The regular expressions that you specify when you define unit templates may be too permissive—allowing unwanted values instead of filtering them out—or too restrictive—filtering out valid values that you want to accept. Use the **Transform Validator** tool when you create unit templates to make sure the regular expressions you specify allow and filter the appropriate values, before referencing the templates in a unit definition.

To test a unit template in the **Transform Validator**:

1. In **Windows Explorer**, navigate to the `\vocera\vcts\tools\TransformValidator\bin`.
2. Run the `TransformValidator.bat` file.

   The **Transform Validator** appears.

   ```
   <!-- Comment -->
   <normalizer version="1.0.0">
         <entry>
               <format>$output</format>
               <node>
                     <input>$input</input>
                     <pattern>.*</pattern>
                     <variables>$output</variables>
               </node>
         </entry>
   </normalizer>
   ```

   ```
   $output ->
   $input ->
   ```

3. In the **Template** field, enter the template that will filter the list of group names from the Vocera Voice Server database, allowing only the role-based group names for the unit you are defining.

4. In the **Input String** field, enter the name of an actual role-based group from the unit.

   The **Transform Validator** passes this string to the template to see if the template allows it as a group name. One of the following situations occurs::

   - If the template filters the group name successfully, the **Transform Validator** displays the value of each template part in the **Template Variables** pane.
   - If the template is unable to filter the group name, the **Template** field is highlighted in orange and the **Template Variables** pane remains empty.

5. If necessary, edit the template definition until the template allows the group name in the **Input String** field.
6. After one group name is allowed, try other group names from the unit you are defining and make sure they are also allowed. Adjust the template if necessary until it allows a representative sample of group names from the unit.
7. Enter group names from other units that have similar naming conventions to make sure that the template correctly filters them out.

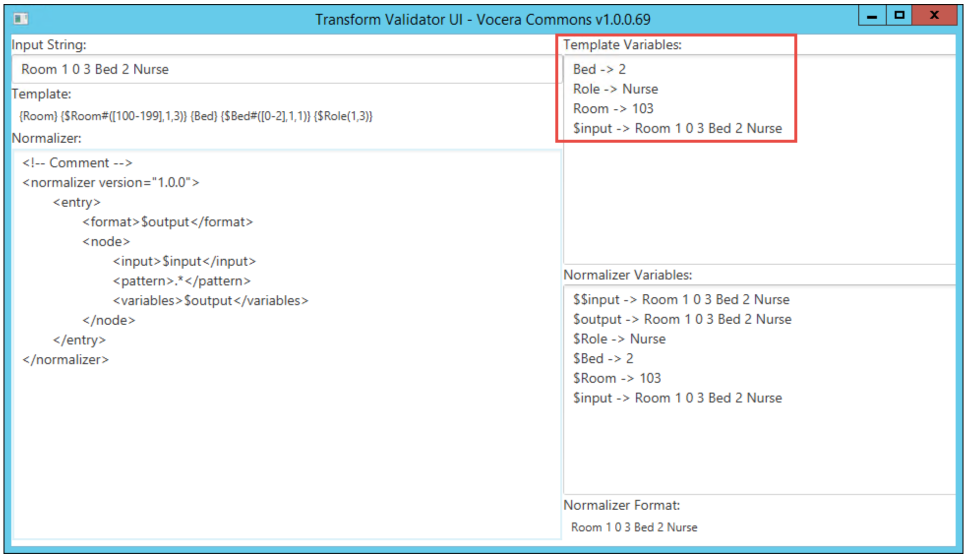
   When you are finished, the template successfully allows all the role based groups from the unit you are defining. Copy the template definition from the Transform Validator and paste it into the **Vocera Group Template** field in either the **Add Unit** or **Update Unit** dialog box.

   See *How to Create a Unit* on page 74 and *How to Update a Unit* on page 77.

For example, suppose you wanted to test a **Template** and an **Input String** using the values shown in the following table:

| Element | Value |
|---|---|
| Template | `{Room} {$Room#([100#199],1,3)} {Bed} {$Bed#([1-2],0,1)} {$Role(1,3)}` |
| Input String | `Room 1 0 3 Bed 2 Nurse` |

After entering the **Template** and **Input String** in the **Template Validator**, the **Template Variables** section shows that the variable values are assigned as expected.



Input String:

Room 1 0 3 Bed 2 Nurse

Template:

{Room} {$Room#([100-199],1,3)} {Bed} {$Bed#([0-2],1,1)} {$Role(1,3)}

Normalizer:

```
<!-- Comment -->
<normalizer version="1.0.0">
    <entry>
        <format>$output</format>
        <node>
            <input>$input</input>
            <pattern>.*</pattern>
            <variables>$output</variables>
        </node>
    </entry>
</normalizer>
```

Template Variables:

```
Bed -> 2
Role -> Nurse
Room -> 103
$input -> Room 1 0 3 Bed 2 Nurse
```

Normalizer Variables:

```
$$input -> Room 1 0 3 Bed 2 Nurse
$output -> Room 1 0 3 Bed 2 Nurse
$Role -> Nurse
$Bed -> 2
$Room -> 103
$input -> Room 1 0 3 Bed 2 Nurse
```

Normalizer Format:

Room 1 0 3 Bed 2 Nurse

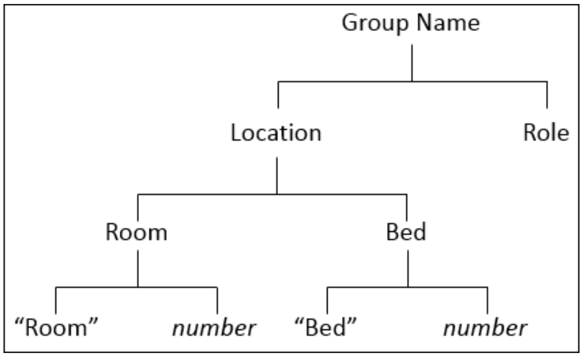
## Creating Group Normalizers for Units

The **Vocera Group Location Normalizer** and the **Vocera Group Role Normalizer** process the output of the **Vocera Group Template**, respectively providing a set of allowed location names and role names. The group normalizers are XML documents that use Java regular expressions to process the template output, providing a set of normalized location names and role names for the CTS database.

### About Group Normalizer Markup

The **Group Location** and **Group Role** normalizers use a small set of XML elements to parse the role-and-location groups that are output by the **Vocera Group Template**.

You can think of a group that is input to the localizer as a branching tree of nodes. For example, a template such as `{Room} {$Room#([100#199],1,3)} {Bed} {$Bed#} {$Role(1,3)}` allows a group name such as `Room 1 0 3 Bed 2 Nurse`; you can visualize this group as a branching tree with a structure similar to the following:



The structured hierarchical syntax of XML allows you to process structured input such as group names effectively. The XML `<node/>` elements in the **Group Location** and **Group Role** normalizers process each of the nodes in this branching tree and allow the normalizers to output normalized values for location and role.

The **Group Location** and **Group Role** normalizers use the XML syntax shown in the following table.

***Table 14: Normalizer Syntax***

| XML Element | Description |
|---|---|
| `<normalizer/>` | Top-level element with an optional `version` attribute for your own reference. |
| `<entry/>` | An element that processes a single chunk of data—either a location or a role. Both `<format/>` and `<node/>` elements are allowed within an `<entry/>` element. |
| `<format/>` | An immediate child of the `<entry/>` element, `<format/>` specifies either the location or role that is output by the normalizer. This output is assembled with the processing done by each of the individual nodes.<br>For example, a location normalizer may use a `<format/>` element such as the following to specify that its output is "Room" followed by a number and "Bed" followed by a number; in this example, Room 103 Bed 2 would be an allowed location:<br>`<format>Room $room Bed $bed</format>`<br>The `<format/>` element always determines the final output of the normalizer, even if `$role` or `$location` are defined in the `<variables/>` element. |
| `<node/>` | An immediate child of the `<entry/>` element or of another `<node/>`, each `<node/>` processes one or more variables in the input template. For example, the top-level node in a location normalizer may evaluate the input group, and nested nodes may evaluate room (`$Room`) and bed (`$Bed`) variables.<br>Each <node> element is a parent of `<input>`, `<pattern>`, and `<variable/>` elements. |
| `<input/>` | An immediate child of the `<node/>` element, `<input/>` specifies which variable from the template its parent `<node/>` is processing. |

| XML Element | Description |
|---|---|
| `<pattern/>` | An immediate child of the `<node/>` element, `<pattern/>` uses Java regular expressions to specify the pattern of data allowed into the node.<br>Every regular expression group in the `<pattern/>` element must correspond to a variable in the `<variables/>` element. If the `<node/>` specifies more `<pattern/>` groups than `<variables/>`, the normalizer produces an error. |
| `<variables/>` | An immediate child of the `<node/>` element, `<variables/>` specifies one or more variables that are refined by the parent `<node/>` processing. This resolved value of the `<node/>` element is passed to the `<format/>` element, which in turn uses the value as part of its output.<br>If the processing in any `<node/>` fails, the `<variables/>` element is not assigned a return value. |

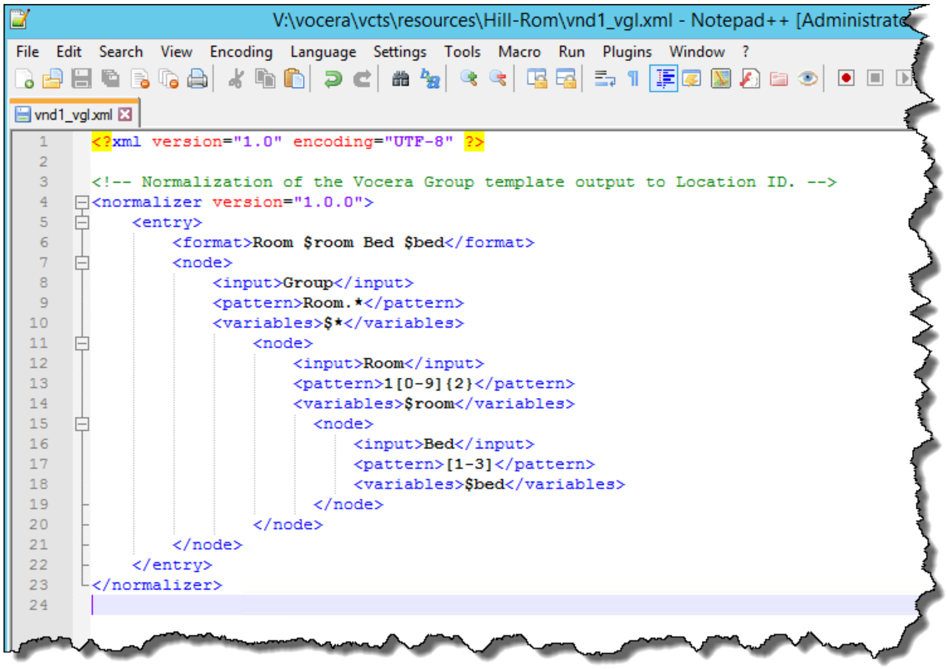
## Unit Group Location Normalizer Example

This example steps through each section of a **Vocera Group Location Normalizer** to examine how it processes an input group and transforms it into the desired normalized output.

This example uses the input template, sample input group, and desired normalized output shown in the following table. Notice that the input template allows a three-word group number such as `1 0 3`; however, the spaces between the numbers must be removed from the normalized output.

| Element | Value |
|---|---|
| Input Template | `{Room} {$Room#([100#199],1,3)} {Bed} {$Bed#([1-2],0,1)} {$Role(1,3)}` |
| Input Group | `Room 1 0 3 Bed 2 Nurse` |
| Normalized Output | `Room 103 Bed 2` |

The following illustration provides an example of a normalizer that can accept an input template and input group such as those shown in the above table and provide normalized output in the desired format for storage in the CTS `locations` table.



```
<?xml version="1.0" encoding="UTF-8" ?>

<!-- Normalization of the Vocera Group template output to Location ID. -->
<normalizer version="1.0.0">
    <entry>
        <format>Room $room Bed $bed</format>
        <node>
            <input>Group</input>
            <pattern>Room.*</pattern>
            <variables>$*</variables>
                <node>
                    <input>Room</input>
                    <pattern>1[0-9]{2}</pattern>
                    <variables>$room</variables>
                      <node>
                          <input>Bed</input>
                          <pattern>[1-3]</pattern>
                          <variables>$bed</variables>
                      </node>
                </node>
        </node>
    </entry>
</normalizer>
```

The top-level node (shown in the following code fragment) allows any group whose name begins with the word "Room" to proceed in the normalizer for further processing:

```
<node>
```

```
   <input>Group</input>
   <pattern>Room.*</pattern>
   <variables>$*</variables>
   ...
```

Since our sample input group `Room 1 0 3 Bed 2 Nurse` meets this criterion, the node processes it. This node outputs any template variable that begins with a `$`, because the value of the `<variables/>` element is defined as `$*`. In this case, the three template variables `$Room`, `$Bed`, and `$Role` are output from this node.

The nested `<node/>` element shown in the following code fragment processes the `$Room` variable, as defined by the `<input/>` element, filtering for a three-digit number that begins with `1` (defined by the `<pattern/>` element):

```
<node>
    <input>Room</input>
    <pattern>1[0-9]{2}</pattern>
    <variables>$room</variables>
    ...
```

Because the sample input group meets both of these criteria, the node assigns the value `103` to the `$room` output variable for the `<format/>` element to consume.

The nested `<node/>` element shown in the following code fragment processes the `$Bed` variable, as defined by the `<input/>` element, filtering for the number `1`, `2`, or `3` (defined by the `<pattern/>` element):

```
<node>
   <input>Bed</input>
   <pattern>[1-3]</pattern>
   <variables>$bed</variables>
   </node>
```

Because the sample input group meets both of these criteria, the node assigns the value `2` to the `$bed` output variable for the `<format/>` element to consume.

The processing for this `<entry/>` element is now complete, and the `<format>Room $room Bed $bed</format>` element now evaluates to `Room 103 Bed 2`, which is stored in the `locations` table of the CTS database.

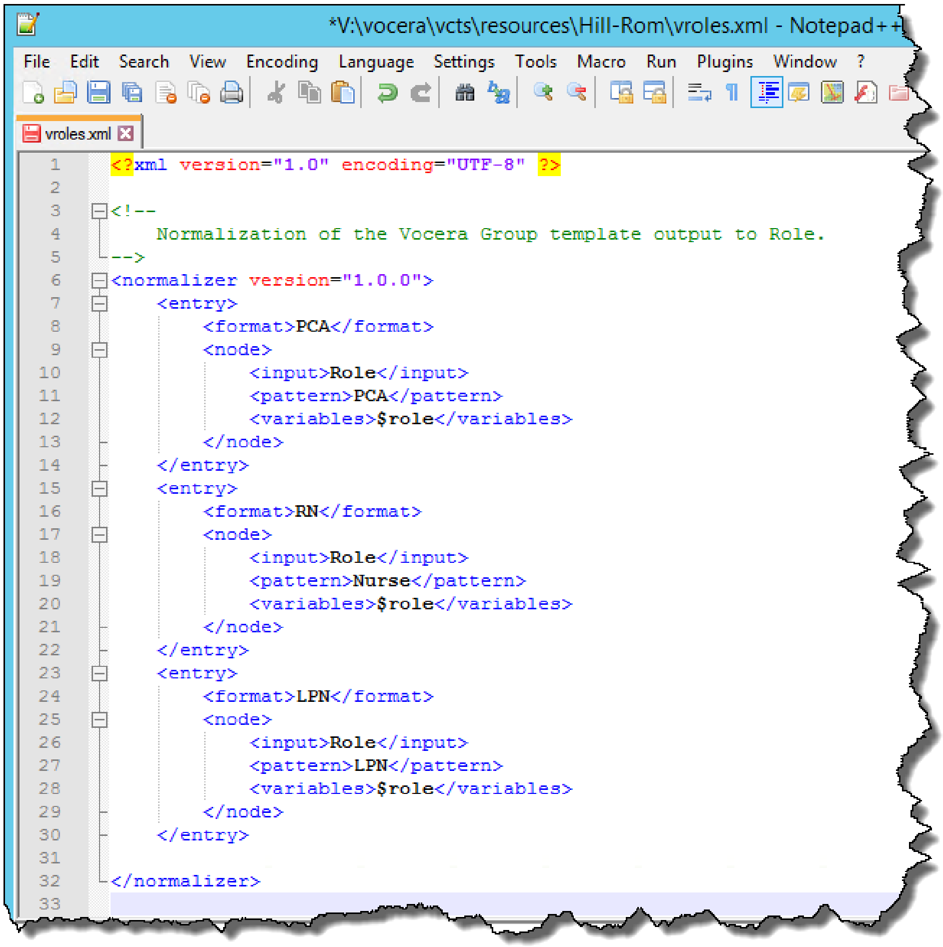
## Unit Group Role Normalizer Example

This example steps through each section of a **Vocera Group Role Normalizer** to examine how it processes an input group and transforms it into the desired normalized output.

This example uses the input template, sample input group, and desired normalized output shown in the following table. Notice that the input template allows a role such as `Nurse`; however, this value must be transformed to `RN` in the normalized output.

| Element | Value |
|---|---|
| Input Template | `{Room} {$Room#([100#199],1,3)} {Bed} {$Bed#([1-2],0,1)} {$Role(1,3)}` |
| Input Group | `Room 1 0 3 Bed 2 Nurse` |
| Normalized Output | `RN` |

The following illustration provides an example of a normalizer that can accept an input template and input group such as those shown in the above table and provide normalized output in the desired format for storage in the CTS `roles` table.


*V:\vocera\vcts\resources\Hill-Rom\vroles.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
vroles.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<!--
    Normalization of the Vocera Group template output to Role.
-->
<normalizer version="1.0.0">
    <entry>
        <format>PCA</format>
        <node>
            <input>Role</input>
            <pattern>PCA</pattern>
            <variables>$role</variables>
        </node>
    </entry>
    <entry>
        <format>RN</format>
        <node>
            <input>Role</input>
            <pattern>Nurse</pattern>
            <variables>$role</variables>
        </node>
    </entry>
    <entry>
        <format>LPN</format>
        <node>
            <input>Role</input>
            <pattern>LPN</pattern>
            <variables>$role</variables>
        </node>
    </entry>

</normalizer>
```

Each `<entry/>` element in this normalizer allows a `$Role` variable from the template (defined by the `<input>` element) whose value is filtered by a very specific name (defined in each `<pattern>` element): one of `PCA`, `Nurse`, or `LPN`.

Since our sample input role `Nurse` meets this criterion, the second `<entry/>` element processes it. In this situation, however, the output format specified within the `<entry/>` element is `<format>RN</format>`, transforming the output for the `roles` table in the database to the normalized value `RN`.

**Note:** If the normalized value Nurse is desired, you can define the `<format/>` element as `$role` or simply `Nurse`.

The normalizer in the above graphic also illustrates "pass through" values—that is, values that are input to the normalizer and passed through as output without any modification. The first and last `<entry/>` elements in the graphic allow a specific `<pattern/>` string and pass the same string through as unmodified output in the `<format/>` element.

### How to Test a Unit Group Normalizer

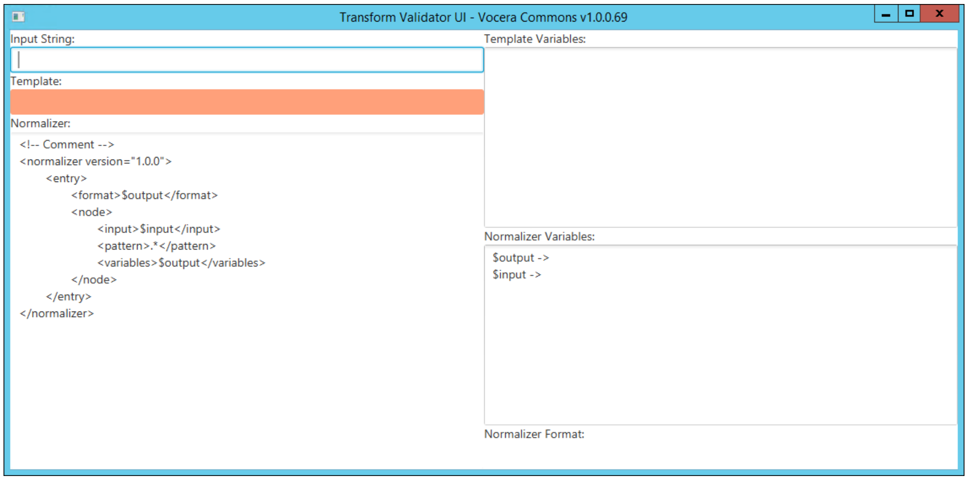
As with unit templates, you can use the **Transform Validator** tool to test group normalizers that you create for units. Use this tool to make sure that the logic you specify allows and filters values appropriately before referencing the normalizer in a unit definition.

To test a unit group normalizer in the **Transform Validator**:

1. In **Windows Explorer**, navigate to the `\vocera\vcts\tools\TransformValidator\bin`.

2. Run the `TransformValidator.bat` file.

   The **Transform Validator** appears.



Transform Validator UI - Vocera Commons v1.0.0.69

Input String:

Template:

Normalizer:

```
<!-- Comment -->
<normalizer version="1.0.0">
    <entry>
        <format>$output</format>
        <node>
            <input>$input</input>
            <pattern>.*</pattern>
            <variables>$output</variables>
        </node>
    </entry>
</normalizer>
```

Template Variables:

Normalizer Variables:

```
$output ->
$input ->
```

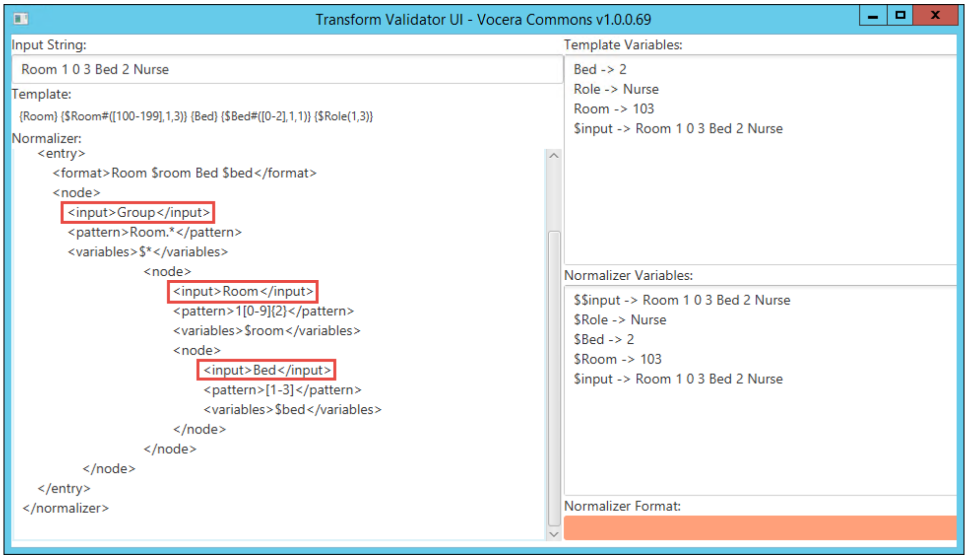
Normalizer Format:

3. Enter data to use as input to the normalizer in the **Template** and **Input String** fields and make sure that the **Template Variables** field shows expected and appropriate values.

   See *How to Test a Unit Template* on page 100.

4. Copy the code for your normalizer from your text editor and paste it into the **Normalizer** section of the **Transform Validator**.

   The **Normalizer Format** field appears highlighted in orange due to minor discrepancies between the `<input/>` variables used in the actual normalizer and the variables expected by the **Transform Validator**.



Transform Validator UI - Vocera Commons v1.0.0.69

Input String:

Room 1 0 3 Bed 2 Nurse

Template:

{Room} {$Room#([100-199],1,3)} {Bed} {$Bed#([0-2],1,1)} {$Role(1,3)}

Normalizer:

```
    <entry>
      <format>Room $room Bed $bed</format>
      <node>
        <input>Group</input>
        <pattern>Room.*</pattern>
        <variables>$*</variables>
                <node>
                    <input>Room</input>
                    <pattern>1[0-9]{2}</pattern>
                    <variables>$room</variables>
                    <node>
                        <input>Bed</input>
                        <pattern>[1-3]</pattern>
                        <variables>$bed</variables>
                    </node>
                </node>
        </node>
    </entry>
  </normalizer>
```

Template Variables:

```
Bed -> 2
Role -> Nurse
Room -> 103
$input -> Room 1 0 3 Bed 2 Nurse
```

Normalizer Variables:

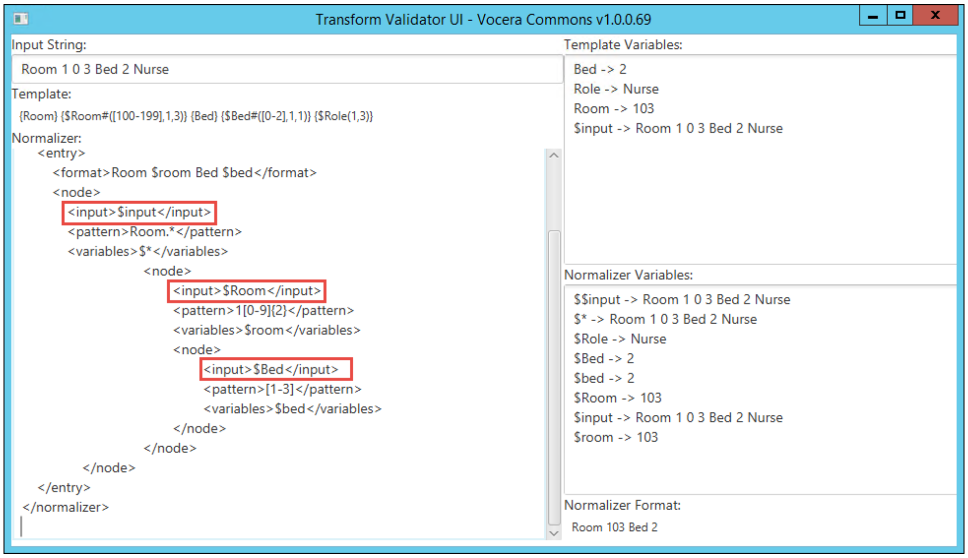
```
$$input -> Room 1 0 3 Bed 2 Nurse
$Role -> Nurse
$Bed -> 2
$Room -> 103
$input -> Room 1 0 3 Bed 2 Nurse
```

Normalizer Format:

5. Edit the variables in the Normalizer field of the **Transform Validator** as follows:

   - In the top-level node, replace the special `Group` variable with `$input`.
   - In the other nodes, prefix the `<input/>` variables with a `$` character.

   When you are finished, the **Normalizer** field should look similar to the following:



Transform Validator UI - Vocera Commons v1.0.0.69

Input String:

Room 1 0 3 Bed 2 Nurse

Template:

{Room} {$Room#([100-199],1,3)} {Bed} {$Bed#([0-2],1,1)} {$Role(1,3)}

Normalizer:

```
      <entry>
         <format>Room $room Bed $bed</format>
         <node>
           <input>$input</input>
           <pattern>Room.*</pattern>
           <variables>$*</variables>
                          <node>
                            <input>$Room</input>
                            <pattern>1[0-9]{2}</pattern>
                            <variables>$room</variables>
                            <node>
                                <input>$Bed</input>
                                <pattern>[1-3]</pattern>
                                <variables>$bed</variables>
                            </node>
                          </node>
               </node>
      </entry>
  </normalizer>
```

Template Variables:

```
Bed -> 2
Role -> Nurse
Room -> 103
$input -> Room 1 0 3 Bed 2 Nurse
```

Normalizer Variables:

```
$$input -> Room 1 0 3 Bed 2 Nurse
$* -> Room 1 0 3 Bed 2 Nurse
$Role -> Nurse
$Bed -> 2
$bed -> 2
$Room -> 103
$input -> Room 1 0 3 Bed 2 Nurse
$room -> 103
```

Normalizer Format:

Room 103 Bed 2

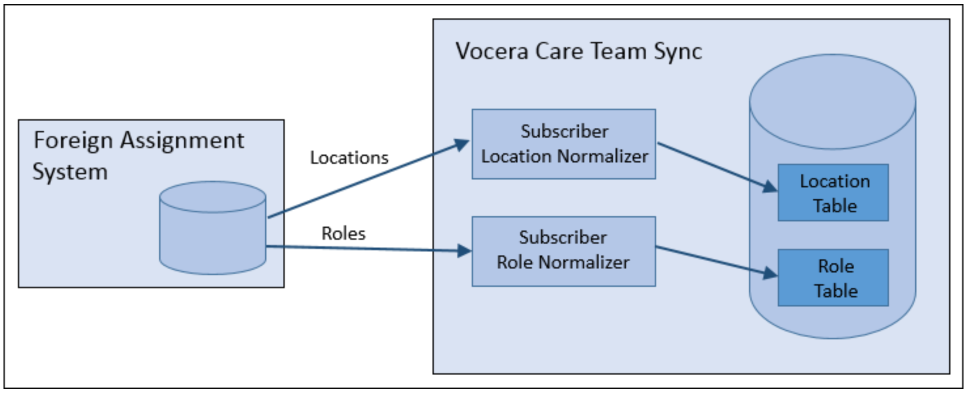
6. Confirm that the **Normalizer Variables** field shows the assigned values you expect and the **Normalizer Format** field displays the expected normalized location or role.
7. If necessary, edit the code you provided in the **Normalizer** field until your normalizer provides the expected output.
8. Copy and paste any changes you made in the **Normalizer** field of the **Transform Validator** into the normalizer in your text editor and save it.

## Creating Normalizers for Connections

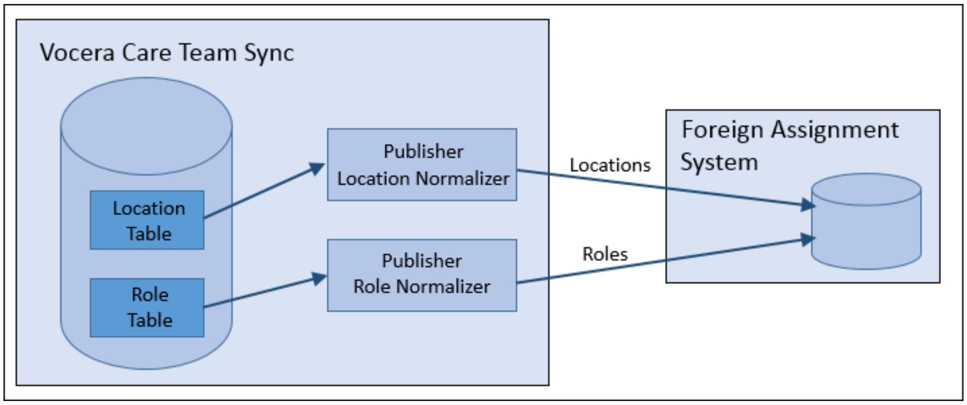
Connection normalizers share a number of features with unit normalizers. Similar to unit normalizers, connection normalizers process groups into their role and location components; both types of normalizers also use the same XML markup. The overall purpose of connection normalizers is different from that of unit normalizers, however. Whereas unit normalizers take Vocera groups as input and provide normalized role and location output for the CTS database, connection normalizers transform role and location data from the convention used by the foreign end point to the CTS convention, or vice versa.

When CTS is the source of truth, it is the *publisher* of assignments; when the foreign end point is the source of truth, CTS is the *subscriber*. Consequently, a *publisher normalizer* transforms either CTS roles or locations into the role or location representation required by the foreign end point. A *subscriber normalizer* transforms the foreign roles or locations into the representation required by the CTS.

The following diagram shows the relationship between CTS and the foreign clinical assignment system when the foreign system is the source of truth.


Vocera Care Team Sync
Foreign Assignment System
Locations
Roles
Subscriber Location Normalizer
Subscriber Role Normalizer
Location Table
Role Table

The following diagram shows the relationship between CTS and the foreign clinical assignment system when CTS is the source of truth.


Vocera Care Team Sync
Location Table
Role Table
Publisher Location Normalizer
Publisher Role Normalizer
Locations
Roles
Foreign Assignment System

## About Connection Normalizer Markup

The connection normalizers use the same small set of XML elements that the unit normalizers use. With this XML markup, connection normalizers translate location and role references between CTS and the foreign clinical assignment system.

Connection normalizers use the XML syntax shown in the following table.

### *Table 15: Normalizer Syntax*

| XML Element | Description |
|---|---|
| `<normalizer/>` | Top-level element with an optional `version` attribute for your own reference. |
| `<entry/>` | An element that processes a single chunk of data—either a location or a role. Both `<format/>` and `<node/>` elements are allowed within an `<entry/>` element. |
| `<format/>` | An immediate child of the `<entry/>` element, `<format/>` specifies either the location or role that is output by the normalizer. This output is assembled with the processing done by each of the individual nodes.<br>For example, a location normalizer may use a `<format/>` element such as the following to specify that its output is "Room" followed by a number and "Bed" followed by a number; in this example, Room 103 Bed 2 would be an allowed location:<br>`<format>Room $room Bed $bed</format>`<br>The `<format/>` element always determines the final output of the normalizer, even if `$role` or `$location` are defined in the `<variables/>` element. |
| `<node/>` | An immediate child of the `<entry/>` element or of another `<node/>`, each `<node/>` processes one or more variables in the input template. For example, the top-level node in a location normalizer may evaluate the input group, and nested nodes may evaluate room (`$Room`) and bed (`$Bed`) variables.<br>Each <node> element is a parent of `<input>`, `<pattern>`, and `<variable/>` elements. |
| `<input/>` | An immediate child of the `<node/>` element, `<input/>` specifies which variable from the template its parent `<node/>` is processing. |
| `<pattern/>` | An immediate child of the `<node/>` element, `<pattern/>` uses Java regular expressions to specify the pattern of data allowed into the node.<br>Every regular expression group in the `<pattern/>` element must correspond to a variable in the `<variables/>` element. If the `<node/>` specifies more `<pattern/>` groups than `<variables/>`, the normalizer produces an error. |
| `<variables/>` | An immediate child of the `<node/>` element, `<variables/>` specifies one or more variables that are refined by the parent `<node/>` processing. This resolved value of the `<node/>` element is passed to the `<format/>` element, which in turn uses the value as part of its output.<br>If the processing in any `<node/>` fails, the `<variables/>` element is not assigned a return value. |

## Subscriber Location Normalizer Example

This example steps through every section of a **Subscriber Location Normalizer** to examine how it processes an input location and transforms it into the desired output location convention. In this example, the foreign system is the publisher and CTS is the subscriber, so data is translated from the foreign system's location convention to the CTS convention.

The foreign assignment system sends data to CTS in the web service request. In this example, an element of the web request looks similar to the following:

```
<Location>
   <LocationID>23<LocationID>
   <UnitZone>VND 1<UnitZone>
   <Room>RM 103<Room>
   <Bed>1<Bed>
   <Facility>Vendor_Validation<Facility>
<Location>
<Role>PCT</Role>
```

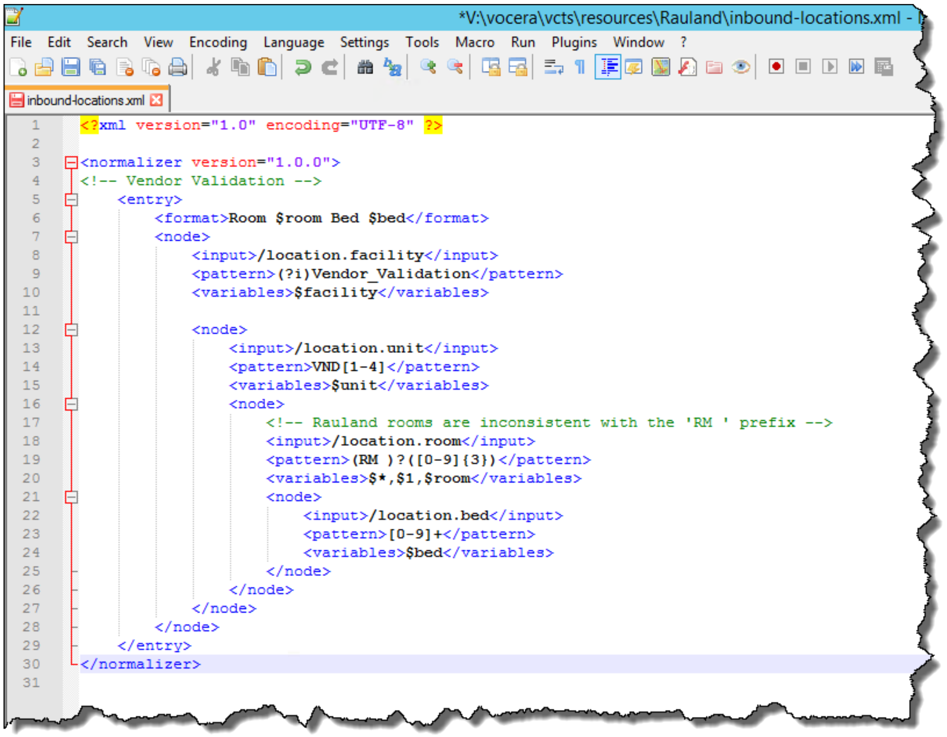
CTS expects data from the web request in this format, and the adapter knows how to assign these values to variables that the normalizer understands. CTS parses the data it receives in this web request, automatically maps it to the following fields, and provides it as input to the normalizer:

| Web Service Field | Input Variable | Value |
|---|---|---|
| `<Facility/>` | `/location.facility` | `Vendor_Validation` |
| `<UnitZone/>` | `/location.unit` | `VND 1` |
| `<Room/>` | `/location.room` | `RM 103` |
| `<Bed/>` | `/location.bed` | `1` |
| `<Role/>` | `/role` | `PCT` |

The `/location` and `/role` prefixes indicate that the values of these variables are assigned by the foreign system. Alternatively, the `$` prefix for a variable indicates that its value is being processed and assigned by the normalizer. Consequently, the normalizer in this example needs to accept the following input and transform it into the following normalized format that is used by CTS:

| Node Input | Input Value | Node Output | Output Value |
|---|---|---|---|
| `/location.facility` | `Vendor_Validation` | `$facility` | `Vendor_Validation` |
| `/location.unit` | `VND 1` | `$unit` | `VND 1` |
| `/location.room` | `RM 103` | `$*`<br>`$1`<br>`$room` | `RM 103`<br>`RM`<br>`103` |
| `/location.bed` | `1` | `$bed` | `1` |

The following illustration provides an example of a normalizer that can accept the input in the format shown in the above table and provide output in the normalized format that is used in the CTS.



*V:\vocera\vcts\resources\Rauland\inbound-locations.xml -
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
inbound-locations.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<normalizer version="1.0.0">
<!-- Vendor Validation -->
    <entry>
        <format>Room $room Bed $bed</format>
        <node>
            <input>/location.facility</input>
            <pattern>(?i)Vendor_Validation</pattern>
            <variables>$facility</variables>

            <node>
                <input>/location.unit</input>
                <pattern>VND[1-4]</pattern>
                <variables>$unit</variables>
                <node>
                    <!-- Rauland rooms are inconsistent with the 'RM ' prefix -->
                    <input>/location.room</input>
                    <pattern>(RM )?([0-9]{3})</pattern>
                    <variables>$*,$1,$room</variables>
                    <node>
                        <input>/location.bed</input>
                        <pattern>[0-9]+</pattern>
                        <variables>$bed</variables>
                    </node>
                </node>
            </node>
        </node>
    </entry>
</normalizer>

```

The four `<node/>` elements in this example successively process the four parts of a location from a Rauland Responder system: facility, unit, room, and bed.

The first node (shown in the following code fragment) takes as input a string identified by the Rauland system as the facility and allows it if it matches the string `<pattern/>` `Vendor_Validation`; the `(?i)` flag makes this pattern case-insensitive.

```
<node>
   <input>/location.facility</input>
   <pattern>(?i)Vendor_Validation</pattern>
   <variables>$facility</variables>
   ...
```

Since our sample input facility meets this criterion, the node processes it and assigns the input directly to the `$facility` variable without transforming it. Because the `<format/>` element (the output of the normalizer) does not handle `$facility`, this data is discarded.

The nested `<node/>` element shown in the following code fragment processes the `<input>` identified as a unit by the Rauland system and allows it if the unit begins with the literal `VND` and is followed by any of the numbers `1` through `4`.

```
<node>
   <input>/location.unit</input>
   <pattern>VND[1-4]</pattern>
   <variables>$unit</variables>
   ...
```

Since our sample input unit meets this criterion, the node processes it and assigns the input directly to `$unit`, again without transforming it. As with `$facility`, the `<format/>` element does not handle `$unit` and the data is discarded.

The nested `<node/>` element shown in the following code fragment processes the `<input/>` identified as a room by the Rauland system. The `?` quantifier (once or not at all) is necessary because Rauland sometimes sends a room as the string `RM` followed by a space and a three-digit number; other times, the Rauland system just passes the three-digit number.

```
<node>
   <input>/location.room</input>
   <pattern>(RM )?([0-9]{3})</pattern>
   <variables>$*,$1,$room</variables>
   ...
```

In the previous code snippet, the `<pattern/>` element comprises three regular expression groups:

- The entire expression `(RM )?([0-9]{3})`, whose value is assigned to the `$*` variable
- The `(RM )?` group, whose value is assigned to the `$1` variable
- The `([0-9]{3})` group, whose value is assigned to the $room variable

The `([0-9]{3})` group allows a three-digit number, each digit of which can be any of the numbers `0` through `9`. Our sample input `RM 103` is allowed by the `<node/>` and the value `103` is assigned to the `$room` variable and passed to the `<format/>` output.

The final `<node/>` element, which is shown in the following code fragment, processes the `<input/>` identified as a bed by the Rauland system.

```
<node>
   <input>/location.bed</input>
   <pattern>[0-9]+</pattern>
   <variables>$bed</variables>
   </node>
```

The `[0-9]` part of the `<pattern/>` allows any single digit in the range of `0` through `9`; the `+` quantifier (one or more times) is necessary because some beds in this system are multiple digit numbers. Our sample input `1` is allowed by the `<node/>` and the value `1` is assigned to the `bed` variable and passed to the `<format/>` output.

The processing for this `<entry/>` is complete, and the `<format>` element now evaluates to `Room 103 Bed 1`, transformed for CTS consumption from the original normalizer input of `Vendor_Validation VND 1 RM 103 1`.

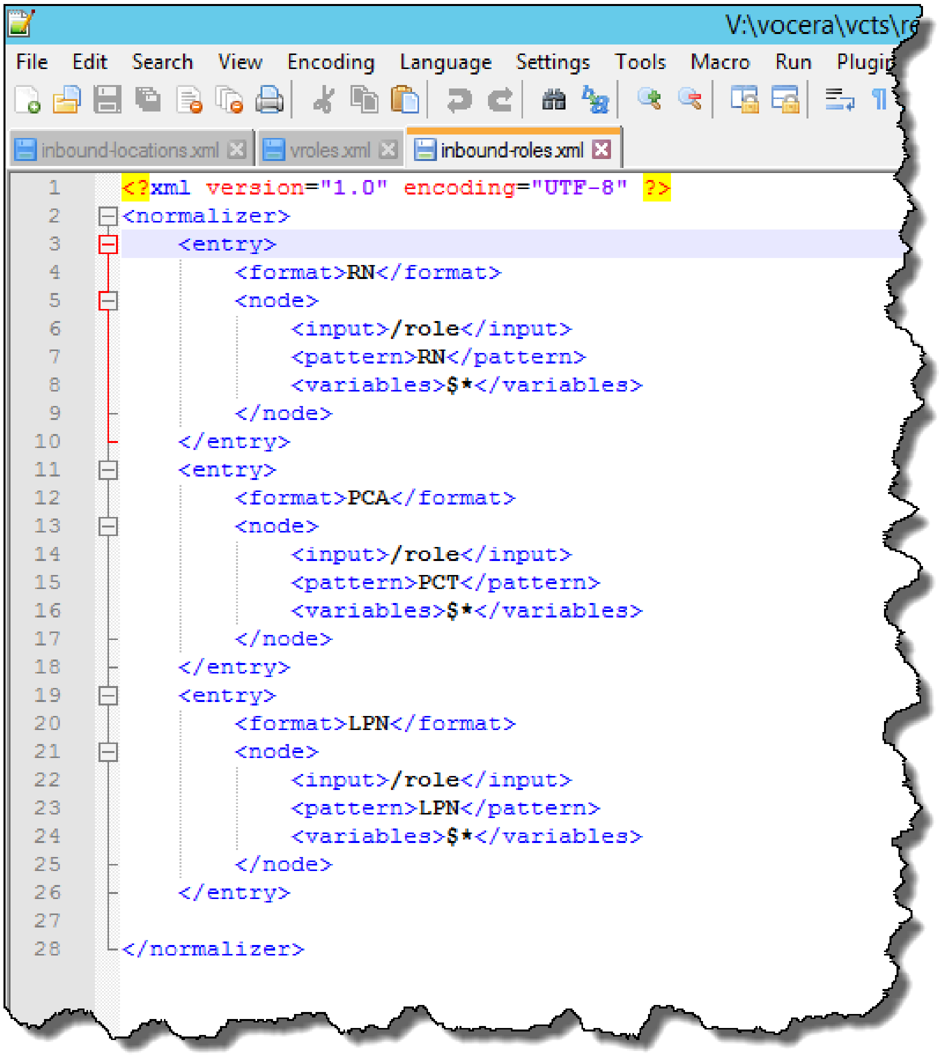
### Subscriber Role Normalizer Example

This example steps through each section of a **Subscriber Role Normalizer** to examine how it processes an input role and transforms it into the desired output convention. In this example, the foreign system is the publisher and CTS is the subscriber, so data is translated from the foreign system's role convention to the CTS convention.

This example uses the sample input role and desired normalized output shown in the following table. Notice that the input allows a role such as `PCT`; however, this value must be transformed to `PCA` in the normalized output.

| Element | Value |
|---|---|
| Input Role | `PCT` |
| Normalized Output | `PCA` |

The following illustration provides an example of a normalizer that can accept an input role such as the one shown in the above table and provide normalized output in the desired format for storage in the CTS `roles` table.



```
<?xml version="1.0" encoding="UTF-8" ?>
<normalizer>
    <entry>
        <format>RN</format>
        <node>
            <input>/role</input>
            <pattern>RN</pattern>
            <variables>$*</variables>
        </node>
    </entry>
    <entry>
        <format>PCA</format>
        <node>
            <input>/role</input>
            <pattern>PCT</pattern>
            <variables>$*</variables>
        </node>
    </entry>
    <entry>
        <format>LPN</format>
        <node>
            <input>/role</input>
            <pattern>LPN</pattern>
            <variables>$*</variables>
        </node>
    </entry>

</normalizer>
```

Each `<entry/>` element in this normalizer takes as input a string that is identified by the Rauland system as the role and whose value is filtered by a very specific name (defined in each `<pattern>` element): one of `RN`, `PCT`, or `LPN`.

Since our sample input role `PCT` meets this criterion, the second `<entry/>` element processes it. In this situation, however, the output format specified within the `<entry/>` element is `<format>PCA</format>`, transforming the output into the normalized value `PCA` for the `roles` table in the CTS database.

The normalizer in the above graphic also illustrates "pass through" values—that is, values that are input to the normalizer and passed through as output without any modification. The first and last `<entry/>` elements in the graphic allow a specific `<pattern/>` string and pass the same string through as unmodified output in the `<format/>` element.

## Integrating with an HL7 Feed

### Overview of HL7 Integration

The Vocera EHR Care Team Synchronization Admission, Discharge and Transfer (ADT) interface is used to collect ADT information from a hospital's EHR/ADT systems. The Vocera Care Team Sync (CTS) application uses this data to retrieve patient and care team data. It then updates the care team with information from the shift-based current assignments of nurses and assistants stored in the Vocera Staff Assignment application.

This process requires a two-way ADT interface. To ensure that the care team data reflects the current staff assignments being made at the hospital, Vocera requires that this interface be real time as opposed to a batch interface. As ADT messages are received, patient data is stored in the Vocera room-based tables for use within the product.

## Normalizing and Mapping ADT Data

This section describes how to normalize and map HL7 ADT data from the external feed.

### Normalizing and Mapping Workflow

To prepare your CTS system to integrate with your foreign care team management system, you must iteratively analyze the two systems and prepare the appropriate normalization and mapping rules that allow the two systems to share data with one another.

The following table describes the expected workflow.

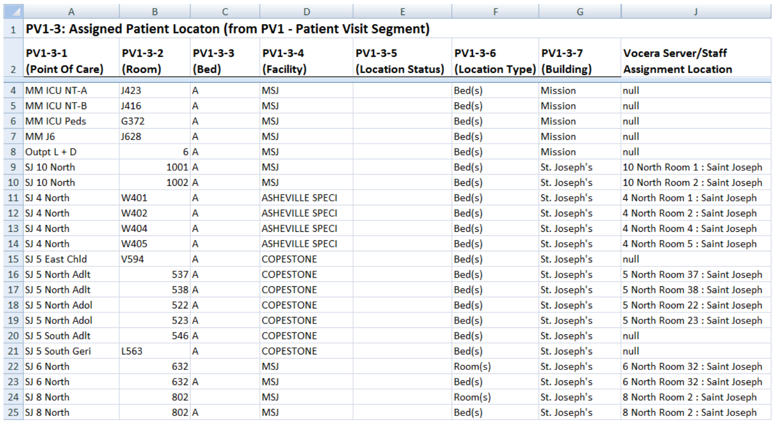
***Table 16: Workflow for normalization and mapping***

| Task | Description |
|---|---|
| 1. Collect HL7 ADT data from the external feed. | Before analyzing data from the feed, make sure you collect several weeks of sample data from the external system. |
| 2. Create a normalization table. | The table should show how locations should be transformed from one system to the other. |
| 3. Write normalization rules. | In an XML file, use regular expressions to transform the inbound location values so they are consistent with Vocera Voice Server/Staff Assignment group names. |
| 4. Write ADT mapping rules. | In an XML file, Java expression language to define the logic for each ADT event that Vocera supports. |
| 5. Test the normalization and mapping rules by processing several HL7 messages. | You can dump the database and then reprocess HL7 messages that CTS has already received from the feed. |
| 6. Check the results, and make appropriate corrections. | You likely won't get the normalization and mapping rules perfect the first time. If so, edit the rules and reprocess HL7 messages. |
| 7. When you're ready, move the CTS from staging to production. | If the transformations are working as designed, you're ready to move CTS into a production phase, connecting it to a live HL7 feed. |

### Sample Normalization Table

Create a normalization table to help you analyze and understand the data you are receiving in the HL7 feed.

The following figure shows a sample normalization table for CTS for segments PV1-3 from the HL7. The last column shows the Vocera Voice Server/Staff Assignment group name. This normalization table could be created by a Vocera customer or by Vocera Professional Services staff after collecting and analyzing the customer's HL7 data.



| | A | B | C | D | E | F | G | J |
|---|---|---|---|---|---|---|---|---|
| 1 | PV1-3: Assigned Patient Locaton (from PV1 - Patient Visit Segment) | | | | | | | |
| 2 | PV1-3-1 (Point Of Care) | PV1-3-2 (Room) | PV1-3-3 (Bed) | PV1-3-4 (Facility) | PV1-3-5 (Location Status) | PV1-3-6 (Location Type) | PV1-3-7 (Building) | Vocera Server/Staff Assignment Location |
| 4 | MM ICU NT-A | J423 | A | MSJ | | Bed(s) | Mission | null |
| 5 | MM ICU NT-B | J416 | A | MSJ | | Bed(s) | Mission | null |
| 6 | MM ICU Peds | G372 | A | MSJ | | Bed(s) | Mission | null |
| 7 | MM J6 | J628 | A | MSJ | | Bed(s) | Mission | null |
| 8 | Outpt L + D | 6 | A | MSJ | | Bed(s) | Mission | null |
| 9 | SJ 10 North | 1001 | A | MSJ | | Bed(s) | St. Joseph's | 10 North Room 1 : Saint Joseph |
| 10 | SJ 10 North | 1002 | A | MSJ | | Bed(s) | St. Joseph's | 10 North Room 2 : Saint Joseph |
| 11 | SJ 4 North | W401 | A | ASHEVILLE SPECI | | Bed(s) | St. Joseph's | 4 North Room 1 : Saint Joseph |
| 12 | SJ 4 North | W402 | A | ASHEVILLE SPECI | | Bed(s) | St. Joseph's | 4 North Room 2 : Saint Joseph |
| 13 | SJ 4 North | W404 | A | ASHEVILLE SPECI | | Bed(s) | St. Joseph's | 4 North Room 4 : Saint Joseph |
| 14 | SJ 4 North | W405 | A | ASHEVILLE SPECI | | Bed(s) | St. Joseph's | 4 North Room 5 : Saint Joseph |
| 15 | SJ 5 East Chld | V594 | A | COPESTONE | | Bed(s) | St. Joseph's | null |
| 16 | SJ 5 North Adlt | 537 | A | COPESTONE | | Bed(s) | St. Joseph's | 5 North Room 37 : Saint Joseph |
| 17 | SJ 5 North Adlt | 538 | A | COPESTONE | | Bed(s) | St. Joseph's | 5 North Room 38 : Saint Joseph |
| 18 | SJ 5 North Adol | 522 | A | COPESTONE | | Bed(s) | St. Joseph's | 5 North Room 22 : Saint Joseph |
| 19 | SJ 5 North Adol | 523 | A | COPESTONE | | Bed(s) | St. Joseph's | 5 North Room 23 : Saint Joseph |
| 20 | SJ 5 South Adlt | 546 | A | COPESTONE | | Bed(s) | St. Joseph's | null |
| 21 | SJ 5 South Geri | L563 | A | COPESTONE | | Bed(s) | St. Joseph's | null |
| 22 | SJ 6 North | 632 | | MSJ | | Room(s) | St. Joseph's | 6 North Room 32 : Saint Joseph |
| 23 | SJ 6 North | 632 | A | MSJ | | Bed(s) | St. Joseph's | 6 North Room 32 : Saint Joseph |
| 24 | SJ 8 North | 802 | | MSJ | | Room(s) | St. Joseph's | 8 North Room 2 : Saint Joseph |
| 25 | SJ 8 North | 802 | A | MSJ | | Bed(s) | St. Joseph's | 8 North Room 2 : Saint Joseph |

## Normalizing ADT Data

CTS allows you to normalize ADT data by specifying regular expressions that search and replace patient location and patient ID values in the HL7 feed so they are suitable for your Vocera system. A regular expression is a string that is used to find and manipulate text.

If you are unfamiliar with regular expressions, Vocera recommends that you brush up on them before tackling the task of normalizing patient locations and patient IDs. Here are just a few books that will help you learn how to use regular expressions:

- *Introducing Regular Expressions - by Michael Fitzgerald*
- *SAMS Teach Yourself Regular Expressions in 10 Minutes - by Ben Forta*
- *Mastering Regular Expressions - by Jeffrey E.F. Friedl*

There are also several regular expression tutorials and cheat sheets that you can find online.

Why would you need to normalize patient location values? For two reasons:

1. Epic HL7 patient location values do not conform to the conventions for Vocera Staff Assignment group names.
2. Some Epic HL7 patient locations are in hospital units not managed by Vocera Staff Assignment, and therefore all messages related to those locations can be ignored.

To normalize patient location and patient ID values, CTS requires that you create XML files with the following names in the `\vocera\vcts\resources` folder:

***Table 17: XML files for normalizing patient location and patient ID***

| XML file | Description |
|---|---|
| apl.xml | Active (current) patient location |
| ppl.xml | Prior patient location |
| pid.xml | Patient ID |

Here's a sample normalizer file for the active patient location (`apl.xml`):

***Example 1: Active patient location normalizer***

```
<normalizer>
  <entry>
    <format>$1 Room $2 : Saint Joseph</format>
    <node>
      <input>/PV1-3-1</input>
      <pattern>SJ ([0-9]+ North).*</pattern>
      <variables>$*,$1</variables>
      <node>
        <input>/PV1-3-2</input>
        <pattern>^.*([0-9]{2})$</pattern>
        <variables>$*,$2</variables>
        <node>
          <input>$2</input>
          <pattern>0*([0-9]+)</pattern>
          <variables>$*,$2</variables>
        </node>
      </node>
    </node>
    <node>
      <input>/PV1-3-1</input>
      <pattern>SJ (ICU).*</pattern>
      <variables>$*,$1</variables>
      <node>
        <input>/PV1-3-2</input>
        <pattern>CC([0-9]+)</pattern>
        <variables>$*,$2</variables>
        <node>
          <input>$2</input>
          <pattern>0*([0-9]+)</pattern>
          <variables>$*,$2</variables>
        </node>
      </node>
    </node>
  </entry>
</normalizer>
```

### Mapping ADT Events

As CTS processes the HL7 feed, it encounters ADT events that trigger updates of care team information.

The following ADT events are recognized by CTS:

***Table 18: ADT events***

| ADT event type | Event type description | Action |
|---|---|---|
| A01 | Admit/Visit Notification | Admit |
| A02 | Transfer a Patient | Transfer |
| A03 | Discharge/End Visit | Discharge |
| A06 | Change an Outpatient to an Inpatient | Admit |
| A07 | Change an Inpatient to an Outpatient | Discharge |
| A08 | Update Patient Information | Update |
| A13 | Cancel Discharge/End Visit | Update |

By default, all other ADT events are ignored by CTS. However, you can configure CTS to map other ADT events as necessary so long as the information is relevant to the Vocera system.

The ADT mapping file is an XML file named `adt-map.xml`. The file must be saved in the `\vocera\vcts\resources` folder in order for CTS to load it.

## ADT Mapping Methods

When you define an ADT mapping rule, you write the rule using Java Expression Language. The purpose of the ADT mapping rule is to handle the ADT event and process the data appropriately by setting the current and prior location for each patient and the current and prior patient for each location.

To perform these operations, CTS exposes the following Java methods you can use in an ADT mapping rule.

***Table 19: ADT mapping rule methods***

| Object | Method | Description |
|---|---|---|
| Patient | `String getLocationId();` | Returns the current location ID. |
| | `String getPriorLocationId();` | Returns the prior location ID. |
| | `String getPatientId();` | Returns the patient ID. |
| | `void setLocationId(String locationId);` | Sets the location ID. |
| | `void setPriorLocationId(String locationId);` | Sets the prior location ID. |
| Location | `String getLocationId();` | Returns the current location ID. |
| | `void setPatientId(String patientId);` | Sets the patient ID. |
| | `void setPriorPatientId(String patientId);` | Sets the prior patient ID. |

## ADT Event Mapping Examples

This section provides several examples of ADT event mappings from the `adt-map.xml` file.

The file has the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<adt-event-map>
  <event key="event_id"> <-- Define mapping rule here --></event>
  <event key="event_id"> <-- Define mapping rule here --></event>
  <event key="event_id"> <-- Define mapping rule here --></event>
  <event key="event_id"> <-- Define mapping rule here --></event>
</adt-event-map>
```

### *Example 2: A01 event mapping example*

### *Example 2: A02 event mapping example*

### *Example 2: A03 event mapping example*

```
<!-- A01 - Admit/Visit Notification -->
<event key="A01">
  //Log the A01 event.
  log.info("ADMIT: Patient '"+patient.getPatientId()+"'
  to '"+l1.getLocationId()+"'.");
  //If patient is already assigned to that location, log a warning.
  if (l1.getPatientId() != null &#038;&#038;
  !l1.getPatientId().equals(patient.getPatientId()))
  {
    log.warn("ADMIT: Patient '" + patient.getPatientId() + "'
    to '" + l1.getLocationId() + "' : patient '" +
    l1.getPatientId() + "' already assigned this location!");
  }
  //Set the patient location
  patient.setLocationId(l1.getLocationId());

  //Set the patient ID for the location
  l1.setPatientId(patient.getPatientId());
</event>
```

**Note:** In the above example, &#038 is the ASCII code for the ampersand character (&).

```
<!-- A02 - Transfer a Patient -->
<event key="A02">
  //Log the A02 event.
  log.info("TRANSFER: Patient '"+patient.getPatientId()+"'
    from '"+l2.getLocationId()+"' to '"+l1.getLocationId()+"'.");
  //If patient in prior location is not equal to the transferred patient,
  //log a warning.
  if (l2.getLocationId()!= null &#038;&#038;
    l2.getPatientId() != null &#038;&#038;
    !l2.getPatientId().equals(patient.getPatientId()))
  {
    log.warn("TRANSFER: Patient '"+patient.getPatientId()+"'
      from '"+l2.getLocationId()+"' : patient '"+l2.getPatientId()+"'
      in room not equal to transfer!");
  }
  //If patient is already assigned to that location, log a warning.
  if (l1.getLocationId()!= null &#038;&#038;
    l1.getPatientId() != null &#038;&#038;
    !l1.getPatientId().equals(patient.getPatientId()))
  {
    log.warn("TRANSFER: Patient '"+patient.getPatientId()+"'
      to '"+l1.getLocationId()+"' : patient '"+l1.getPatientId()+"'
      already assigned this location!");
  }
  //Set patient's prior location
  patient.setPriorLocationId(l2.getLocationId());

  //Set patient's new location
  patient.setLocationId(l1.getLocationId());

  //Set prior patient ID for prior location
  l2.setPriorPatientId(patient.getPatientId());

  //Clear patient ID for prior location
  l2.setPatientId(null);

  //Set patient ID for new location
  l1.setPatientId(patient.getPatientId());
</event>
```

```
<!-- A03 - Discharge/End Visit -->
<event key="A03">
  //Log the A03 event.
  log.info("DISCHARGE: Patient '"+patient.getPatientId()+"'
    from '"+l1.getLocationId()+"'.");

  //If patient already discharged, do nothing.
  if (patient.getLocationId() == null &#038;&#038;
    patient.getPriorLocationId()!= null &#038;&#038;
    patient.getPriorLocationId().equals(l1.getLocationId()))
  {
    log.debug("DISCHARGE: Patient '"+patient.getPatientId()+"'
    from '"+l1.getLocationId()+"' ... already discharged (duplicate
    discharge or outpatient disposition).");
    return;
  }

  //If patient never assigned to location, do nothing.
  if (patient.getLocationId() == null)
  {
    log.debug("DISCHARGE: Patient '"+patient.getPatientId()+"'
      from '"+l1.getLocationId()+"' ... was never assigned to the room.");
    return;
  }

  //If patient in location not equal to discharge patient,
  //log a warning.
  if (l1.getPatientId()!= null &#038;&#038;
    !l1.getPatientId().equals(patient.getPatientId()))
  {
    log.warn("DISCHARGE: Patient '"+patient.getPatientId()+"'
      from '"+l1.getLocationId()+"' : patient '"+l1.getPatientId()+"'
      in room not equal to discharge!");
  }

  //Set patient's prior location
  patient.setPriorLocationId(l1.getLocationId());

  //Clear patient's location
  patient.setLocationId(null);

  //Set prior patient ID for the location
  l1.setPriorPatientId(patient.getPatientId());

  //Clear patient ID for location
  l1.setPatientId(null);
</event>
```

### *Example 3: A06 event mapping example*

### *Example 3: A07 event mapping example*

### *Example 3: A08 event mapping example*

### *Example 3: A13 event mapping example*

```
<!-- A06 - Change an Outpatient to an Inpatient -->
<event key="A06">
  //Log the A06 event.
  log.info("OUTPATIENT TO INPATIENT: Patient '"+patient.getPatientId()+"'
    to '"+l1.getLocationId()+"'.");
  //If patient is already assigned to that location, log a warning.
  if (l1.getPatientId() != null &#038;&#038;
    !l1.getPatientId().equals(patient.getPatientId()))
  {
    log.warn("OUTPATIENT TO INPATIENT: Patient '" +
      patient.getPatientId() + "' to '" + l1.getLocationId() + "'
      : patient '" + l1.getPatientId() + "' already assigned this location!");
  }
  //Set patient's location
  patient.setLocationId(l1.getLocationId());

  //Set patient ID for location
  l1.setPatientId(patient.getPatientId());
</event>
```

```
<!-- A07 - Change an Inpatient to an Outpatient -->
<event key="A07">
  //Log the A07 event.
  log.info("INPATIENT TO OUTPATIENT Patient '"+patient.getPatientId()+"'
    from '"+l2.getLocationId()+"'.");
  //If patient already discharged, do nothing.
  if (patient.getLocationId() == null &#038;&#038;
    patient.getPriorLocationId()!= null &#038;&#038;
    patient.getPriorLocationId().equals(l2.getLocationId()))
  {
    log.debug("INPATIENT TO OUTPATIENT: Patient
    '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"'
    ... already discharged (duplicate discharge or outpatient
    disposition).");
    return;
  }
  //If patient never assigned to location, do nothing.
  if (patient.getLocationId() == null)
  {
    log.debug("INPATIENT TO OUTPATIENT: Patient
    '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"'
    ... was never assigned to the room.");
    return;
  }
  //If patient in location not equal to discharge patient,
  //log a warning.
  if (l2.getPatientId()!= null &&
    !l2.getPatientId().equals(patient.getPatientId()))
  {
    log.warn("INPATIENT TO OUTPATIENT: Patient
      '"+patient.getPatientId()+"' from '"+l2.getLocationId()+"'
      : patient '"+l2.getPatientId()+"' in room not equal to
      discharge!");
  }

  //Set patient's prior location
  patient.setPriorLocationId(l2.getLocationId());

  //Clear patient's location
  patient.setLocationId(null);

  //Set prior patient ID for location
  l2.setPriorPatientId(patient.getPatientId());

  //Clear patient ID for location
  l2.setPatientId(null);
```

```
</event>
```

```
<!-- A08 - Update -->
<event key="A08">
  //Log the A08 event.
  log.info(
    "UPDATE: Patient '"+patient.getPatientId()+"'
      current/prior location '"+patient.getLocationId()+
      "'/'"+patient.getPriorLocationId()+
      "' current/prior message id '"+patient.getMessageId()+
      "'/'"+patient.getPriorMessageId()+"'.");

  //Set patient's location
  patient.setLocationId(l1.getLocationId());

  //Set patient ID for location
  l1.setPatientId(patient.getPatientId());

</event>
```

```
<!-- A13 - Cancel Discharge -->
<event key="A13">
  //Log the A13 event.
  log.info("CANCEL DISCHARGE: Patient '"+patient.getPatientId()+
    "' from '"+l1.getLocationId()+"'.");
  //If patient is already assigned to that location, log a warning.
  if (l1.getPatientId() != null &#038;&#038;
    !l1.getPatientId().equals(patient.getPatientId()))
  {
    log.warn("CANCEL DISCHARGE: Patient '" + patient.getPatientId() +
      "' from '" + l1.getLocationId() + "' : patient '" +
      l1.getPatientId() + "' already assigned this location!");
  }

  //Set patient's location
  patient.setLocationId(l1.getLocationId());

  //Set patient ID for location
  l1.setPatientId(patient.getPatientId());
</event>
```

### Inbound ADT Interface

This section describes the inbound (data flow from the ADT system to Vocera) ADT interface of the Vocera Care Team Sync product.

### Inbound ADT Message Requirements

This section describes the segments and event types of an inbound message that are relevant to the CTS.

### Message Segments

Vocera uses data from specific HL7 message segments disregards segments and fields that are not used within the product.

Vocera uses data from the following HL7 message segments:

- MSH – Message Header

- EVN – Event Type
- PID – Patient Identification
- PV1 – Patient Visit
- PV2 – Patient Visit – Additional Information
- ROL – Roles for Care Team of patient

Vocera interprets an empty field (||) as indicating no new changes (leave current value intact) and fields containing double quotes (|""|) as an indication nothing should be stored for the field. See *Vocera Data Requirements* on page 121to learn about Vocera requirements for each segment.

## Event Types

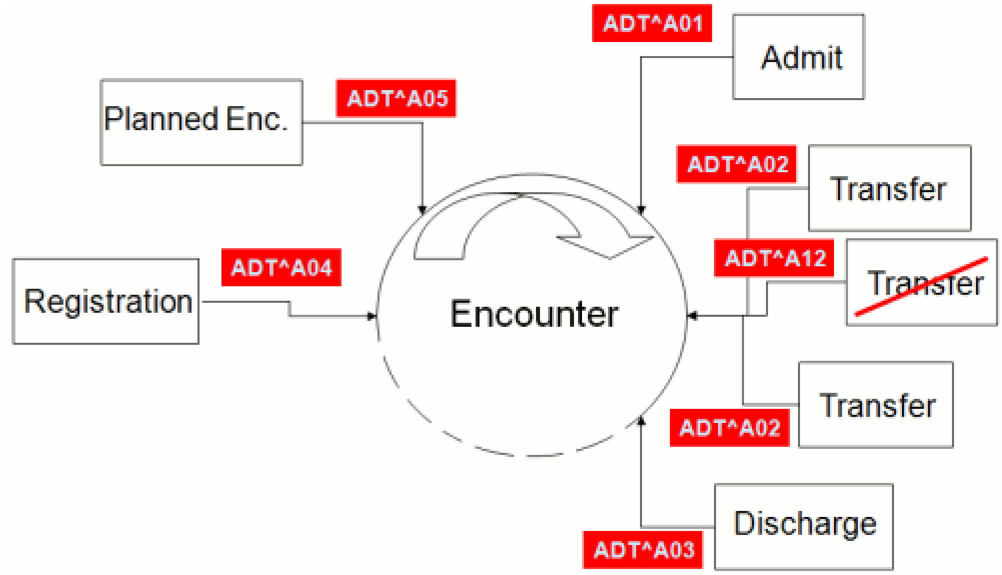
By default, Vocera recognizes specific ADT event types and ignores all other ADT events. You can customize Vocera and map other ADT events if necessary.

The following ADT event types are recognized by Vocera.

***Table 20: Event types***

| Event Type | Event Type Description | Action |
|---|---|---|
| A01 | Admit/Visit Notification | Admit |
| A02 | Transfer a Patient | Transfer |
| A03 | Discharge/End Visit | Discharge |
| A06 | Change an Outpatient to an Inpatient | Admit |
| A07 | Change an Inpatient to an Outpatient | Discharge |
| A08 | Update Patient Information | Update |
| A13 | Cancel Discharge/End Visit | Update |

By default, all other ADT events are ignored by CTS. However, you can configure CTS to map other ADT events as necessary so long as the information is relevant to the Vocera system.



## Vocera Data Requirements

This section describes the data elements that Vocera requires in the ADT data feed.

## Patients

A patient should be uniquely identifiable across the entire health system (especially in multi-facility health systems) as well as across multiple visits/admissions. Typically health systems choose to use a patient account number, MRN, or other field as the patient identifier. The identifier must be unique. Failure to provide a unique identifier will result in improper capture of patient visits/admissions across health system facilities as well as multiple visits over time.

Vocera looks for a unique patient identifier in the following segment elements:

- PID-2: Patient ID
- PID-3.1: Patient Identifier List – ID Number
- PID-4: Alternate Patient ID
- PID-18.1: Patient Account Number – ID Number
- PID-19: SSN Number – Patient

In addition, Vocera can be configured to use a combination of any elements in the PID segment for unique patient identification including using the ‘assigning authority’ element in cases of multi-facility health systems where the patient identifier is locally unique to a single facility.

## Location – Unit/Department

Vocera supports hospital locations at the unit/department level. The list of meaningful units/departments is specific to a facility and additional consideration must be given for multi-facility health systems as duplicate patient locations can exist across different facilities.

Vocera uses the patient location elements in the PV1 segment to identify location. Specifically, PV1_3 (Assigned Patient Location) and PV1_6 (Prior Patient Location). The identifiable unit information must be present within the 3 components of patient location ‘point of care’, ‘room’, and ‘bed’.

Proper identification of patient event locations is critical for proper retracing of patient movement as well as consistent performance analytics. Locations that are used for transfers, both internal facility as well as inpatient-to-outpatient and back, are also important for purposes of proper patient visit demarcation.

Per HL7 recommendations in cases of multi-facility health systems, we require the use of ‘facility’, the optional forth component of the patient location element to uniquely identify the healthcare facility containing the patient location.

## Patient Visits

Vocera uses ADT events to derive patient visit information. Vocera relies on being able to uniquely identify patients to properly identify a single inpatient visit/admission/encounter.

An inpatient visit/admission/encounter is comprised of a single admission data followed by a single discharge date. Same day admissions must also be captured.

Improper identification of patient visits will result in number of admissions/visits in Vocera not correlating closely to actual number of admissions/visits across the health system. It will also result in Vocera showing incorrect admission duration (i.e., the duration between admit date and discharge date) for a given visit.

Patient visits will be derived from related chronological patient events (or ADT events) for a given unique patient. A patient visit will start with a patient admit event (or equivalent) and end with a patient discharge event (or equivalent). For example, the most simple scenario will consists of A01 (Admit/Visit Notification) event chronologically followed by A03 (Discharge/End Visit) event for a unique patient.

A health system’s use of ADT event types for representing physical patient event is often idiosyncratic and requires close collaboration between Experia Health and the health system for proper identification of patient visits.

## ADT Acknowledgement Message

Vocera sends acknowledgement messages to the sending system upon receipt of valid messages.

The acknowledgment messages sent by Vocera are in the following format:

- MSH – Message Header
- EVN – Event Type
- PID – Patient Identification
- PV1 – Patient Visit
- PV2 – Patient Visit – Additional Information
- ROL – Roles for Care Team of patient

***Table 21: MSH – message header segment***

| SEQ | LEN | Element Name |
|---|---|---|
| 1 | 1 | Field Separator |
| 2 | 4 | Encoding Characters |
| 3 | 3 | Sending Application |
| 4 | 15 | Sending Facility |
| 7 | TS | Date/Time of Message |
| 9.1 | 3 | Message Type |
| 9.2 | 3 | Message Type (Event) |
| 10 | INT | Message Control ID |
| 11 | 1 | Processing ID |
| 12 | 3 | Version ID |

***Table 22: MSA – message acknowledgement segment***

| SEQ | LEN | Element Name |
|---|---|---|
| 1 | 2 | Acknowledgment Code |
| 2 | 20 | Message Control ID |

## Initial Upload

Vocera does not require an initial upload of data.  As records are received, Vocera will begin writing the appropriate records to the patient tables.

It is not required that a patient admit event (A01) precede other event types.

## Configuration/Support

By default, Vocera assigns TCP/IP port 2575 to listen for connections from the sending EHR system. Vocera tests the system to ensure that HL7 ADT messages are received appropriately.

Vocera does not include a separate test environment for CTS. Prior to moving the system to a production environment, you can purge messages received prior to a specified date.

## Segment Element Recommendation

This section presents a subset of segment elements (and sub-elements) that Vocera ADT interface engine will use if data is present. Refer to the *HL7 Messaging Standard Version 2.7.1* from ANSI for specific details around HL7 message implementation guidelines including element definition as well as format.

Check the box beside the each element provided. If information is provided in an element other than the one listed, fill in the segment and field in a blank element column.

### *Table 23: ADT event information*

| Information Needed | Element | Element |
|---|---|---|
| Field Separator | MSH-1 | |
| Encoding Characters | MSH-2 | |
| Sending Application | MSH-3 | |
| Sending Facility | MSH-4 | |
| Date/Time of Message | MSH-7 | |
| Message Type | MSH-9.1 | |
| Message Type (Event) | MSH-9.2 | |
| Message Control ID | MSH-10 | |
| Processing ID | MSH-11 | |
| Version ID | MSH-12 | |
| Event Type Code | EVN-1 | |
| Recorded Date/Time | EVN-2 | |
| Date/Time Planned Event | EVN-3 | |
| Event Reason Code – Identifier | EVN-4.1 | |
| Event Occurred | EVN-6 | |
| Event Facility – Namespace ID | EVN-7.1 | |

### *Table 24: Unique patient identifier – Mark all fields provided in feed and choose a unique Patient Identifier*

| Element Name | Element | Unique |
|---|---|---|
| Patient Identifier List – ID Number | PID-3.1 | |
| Alternate Patient ID | PID-4 | |
| Patient ID | PID-2 | |
| Patient Account Number – ID Number | PID-18.1 | |
| SSN Number – Patient | PID-19 | |

### *Table 25: Patient information*

| Information Needed | Element | Element |
|---|---|---|
| Patient Name – Family Name | PID-5.1 | |
| Patient Name – Given Name | PID-5.2 | |
| Patient Name – Second and Further Given Names | PID-5.3 | |
| Patient Name – Suffix (e.g., JR or III) | PID-5.4 | |

| Information Needed | Element | Element |
|---|---|---|
| Date/Time of Birth | PID-7 | |
| Administrative Sex | PID-8 | |
| Race | PID-10 | |
| Patient Street Address | PID-11.1 | |
| Patient Street Address | PID-11.2 | |
| Patient City | PID-11.3 | |
| Patient State or Province | PID-11.4 | |
| Patient  Zip or Postal Code | PID-11.5 | |
| Patient Country | PID-11.6 | |
| Home Phone | PID-13 | |
| Business Phone | PID-14 | |
| Marital Status | PID-16 | |

***Table 26: Unique visit identifier – Mark all fields provided in feed and choose a unique Patient Visit Identifier***

| Information Needed | Element | Unique |
|---|---|---|
| Visit Number | PV1-19 | |
| Alternate Visit ID | PV1-50 | |
| Patient Account Number – ID Number | PID-18.1 | |

***Table 27: Patient visit information***

| Information Needed | Element | Element |
|---|---|---|
| Assigned Patient Location – Point of Care | PV1-3.1 | |
| Assigned Patient Location – Room | PV1-3.2 | |
| Assigned Patient Location – Bed | PV1-3.3 | |
| Assigned Patient Location - Facility | PV1-3.4 | |
| Admission Type | PV1-4 | |
| Pre-Admit Number | PV1-5 | |
| Prior Patient Location | PV1-6 | |
| Attending Doctor – Person Identifier | PV1-7.1 | |
| Attending Doctor – Family Name | PV1-7.2.1 | |
| Attending Doctor – Given Name | PV1-7.3 | |
| Referring Doctor – Person Identifier | PV1-8.1 | |
| Referring Doctor – Family Name | PV1-8.2.1 | |
| Referring Doctor – Given Name | PV1-8.3 | |
| Consulting Doctor – Person Identifier | PV1-9.1 | |
| Consulting Doctor – Family Name | PV1-9.2.1 | |
| Consulting Doctor – Given Name | PV1-9.3 | |

| Information Needed | Element | Element |
|---|---|---|
| Re-Admission Indicator | PV1-13 | |
| Admit Source | PV1-14 | |
| Admitting Doctor – Person Identifier | PV1-17.1 | |
| Admitting Doctor – Family Name | PV1-17.2.1 | |
| Admitting Doctor – Given Name | PV1-17.3 | |
| Patient Type | PV1-18 | |
| Visit Number | PV1-19 | |
| Financial Class | PV1-20 | |
| Discharge Disposition | PV1-36 | |
| Discharge To Location | PV1-37 | |
| Admit Date/Time | PV1-44 | |
| Discharge Date/Time | PV1-45 | |
| Alternate Visit ID | PV1-50 | |
| Visit Indicator | PV1-51 | |
| Admit Reason | PV2-3 | |
| Transfer Reason | PV2-4 | |
| Expected Admit Date/Time | PV2-8 | |
| Expected Discharge Date/Time | PV2-9 | |
| Estimated Length of Inpatient Stay | PV2-10 | |
| Actual Length of Inpatient Stay | PV2-11 | |
| Expected Discharge Disposition | PV2-27 | |

***Table 28: Diagnosis information (optional)***

| Information Needed | Element | Element |
|---|---|---|
| Set ID – DG1 | DG1-1 | |
| Diagnosis Coding Method | DG1-2 | |
| Diagnosis Code – DG1 | DG1-3 | |
| Diagnosis Description | DG1-4 | |
| Diagnosis Date/Time | DG1-5 | |
| Diagnosis Type | DG1-6 | |
| Major Diagnostic Category | DG1-7 | |
| Diagnostic Related Group | DG1-8 | |

***Table 29: Diagnosis related group (optional)***

| Information Needed | Element | Element |
|---|---|---|
| Diagnostic Related Group | DG1-1 | |

***Table 30: Treatment team (required)***

| Information Needed | Element | Element |
|---|---|---|
| Role Instance ID (EI) | ROL-1 | |
| Action Code (ID) | ROL-2 | |
| Role-ROL (CE) | ROL-3 | |
| Role Person (XCN) | ROL-4 | |
| Role Begin Date/Time (TS) | ROL-5 | |
| Role End Date/Time (TS) | ROL-6 | |
| Role Duration (CE) | ROL-7 | |
| Role Action Reason (CE) | ROL-8 | |
| Provider Type (CE) | ROL-9 | |
| Organization Unit Type (CE) | ROL-10 | |
| Office/Home Address/Birthplace (XAD) | ROL-11 | |
| Phone (XTN) | ROL-12 | |

## Outbound ADT Interface

This section describes the outbound (data flow from Vocera to the ADT system) ADT interface of the Vocera Care Team Sync product.

### Use Cases

The care team assignments maintained by the Vocera Staff Assignment application are typically more accurate than the EHR assignments, because the Vocera assignments are updated with each shift. The examples in this section demonstrate some common situations where you may desire to update the EHR assignments from the Vocera database.

### Sample Message for Use Cases

Each use case in this section is based upon a single sample message.

Assume that the following message is the last UPDATE message received via HL7 from the ADT feed by the Vocera Care Team Sync application:

```
MSH|^~\&|REG_UPDATE|TRH_TST||Primary|20140827085804|IPRN|ADT^A08|137084|P|2.3
EVN|A08|20140827085803||REG_UPDATE|Model User^INPATIENT^NURSE^^^^^^TRHSA^^^^^TRH
PID|1||3007601^^^INV^MR||CAMPBELL^JESSICA||19751121|F|||^^^^^USA^P|||||||
100000023169||||||||||||N
PV1|1|INP|N4W^N422^01^TRH^R^^^^^^DEPID|ER|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^
   ^^PROVID|||MED||||Home/Work|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^^^PROVID||27139|
   P|||||||||||||||||||||Confirmed Admission|^^^TRH||20140731085952|||152488.74|||2
PV2||ICU||||||20140731|20140827|||Hospital Encounter|||||||||n|N
ROL|1||Surgeon|
010537^CASEY^GREG^D^^^^^INV^^^^PROVID~340483^CASEY^GREG^D^^^^^EPC^^^^PROVID|
20140419084647||||||525 RACE
   ST^SUITE 3220^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
```

```
ROL|2||Anesthesiolo|
005894^SHOSTAK^IGOR^^^^^^INV^^^^PROVID~341957^SHOSTAK^IGOR^^^^^^EPC^^^^PROVID|
20140419150838||||||525
   RACE ST^^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
ROL|3||Hospitalist|
010782^LANG^DONALD^J^^^^^INV^^^^PROVID~342061^LANG^DONALD^J^^^^^EPC^^^^PROVID|
20140419143134||||||525
   RACE ST^^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
ROL|4||RN|343872^PERKINS^GERALD^D^^^^^EPC^^^^PROVID|20140419032010|20140419103431
ROL|5||RN|344192^SOUTHARD^NIHAET^^^^^^EPC^^^^PROVID|20140419184945|20140420080000
ROL|6||RN|344439^SHIVELY^MICHELLE^^^^^^EPC^^^^PROVID|20140419160615
ROL|7||PCA|346726^CARDINAL^DAPHNE^A^^^^^EPC^^^^PROVID|20140419230415|20140420062713
ROL|8||PCA|347000^ZART^SHERYL^L^^^^^EPC^^^^PROVID|20140420070025
ROL|9||PCA|350051^WAYLAND^SASHANTAA^D^^^^^EPC^^^^PROVID|20140419163505|20140419232342
ROL|10||Consult MD|3040044^HOSPITALISTS^GEN^^^^^^EPC^^^^PROVID|20140419134012
ROL|11||Consult MD|3040054^GEN ORTHOPEDIC ASSOCIATE OF MY FACILITY^^^^^^^EPC^^^^PROVID|
20140419073248|20140420162447|||||525
   RACE ST^SUITE 3220^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
```

### Remove a Provider from the Active Care Team

This example shows you how to remove a provider who is currently on the active care team from that care team.

1. User is identified by the Care Team Sync application with SER 344439: `344439^SHIVELY^MICHELLE^^^^^^EPC^^^^PROVID|20140419160615`
2. Logs off Vocera at 7:30PM on Aug 27 ending her shift. Vocera automatically removes her from the Room 22 Bed 1 group within Vocera.
3. CTS sends an HL7 message with the last instance of each provider seen for the role RN, however for this provider the end time is set.

The following message is sent by CTS to the Epic system.

```
MSH|^~\&|VCTS_SYNC|VCTS_TST|||20140827193000.000-0400||ADT^A08|445|P|2.3
EVN|A08|20140827193000.000-0400||VCTS_SYNC|^INTERFACE^VCTS^^^^^^TRHSA
PID|1||3007601^^^INV^MR||CAMPBELL^JESSICA||19751121|F|||^^^^^USA^P|||||||
100000023169||||||||||||N
PV1|1|INP|N4W^N422^01^TRH^R^^^^^^DEPID|ER|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^
   ^^PROVID|||MED||||Home/Work|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^^^PROVID||27139|
   P|||||||||||||||||||||Confirmed Admission|^^^TRH||20140731085952|||152488.74|||2
PV2||ICU||||||20140731|20140827|||Hospital Encounter|||||||||n|N
ROL|4||RN|343872^^^^^^^^EPC^^^^PROVID|20140419032010|20140419103431
ROL|5||RN|344192^^^^^^^^EPC^^^^PROVID|20140419184945|20140420080000
ROL|6||RN|344439^^^^^^^^EPC^^^^PROVID|20140419160615|20140827193000
```

Only the providers in the role we are updating are sent. Also, notice that CTS includes only the SER number, and not the provider name, in the outbound message.

This affects the care team for the patient in the EPIC system which has the side effect of sending back an UPDATE message which should reflect the updated care team. The following message is received by the Vocera Care Team Sync application from EPIC:

```
MSH|^~\&|REG_UPDATE|TRH_TST||Primary|20140827193005|IPRN|ADT^A08|140000|P|2.3
EVN|A08|20140827193005||REG_UPDATE|^INTERFACE^VCTS^^^^^^TRHSA
PID|1||3007601^^^INV^MR||CAMPBELL^JESSICA||19751121|F|||^^^^^USA^P|||||||
100000023169||||||||||||N
PV1|1|INP|N4W^N422^01^TRH^R^^^^^^DEPID|ER|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^
   ^^PROVID|||MED||||Home/Work|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^^^PROVID||27139|
   P|||||||||||||||||||||Confirmed Admission|^^^TRH||20140731085952|||152488.74|||2
PV2||ICU||||||20140731|20140827|||Hospital Encounter|||||||||n|N
ROL|1||Surgeon|
010537^CASEY^GREG^D^^^^^INV^^^^PROVID~340483^CASEY^GREG^D^^^^^EPC^^^^PROVID|
20140419084647||||||525 RACE
   ST^SUITE 3220^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
ROL|2||Anesthesiolo|
005894^SHOSTAK^IGOR^^^^^^INV^^^^PROVID~341957^SHOSTAK^IGOR^^^^^^EPC^^^^PROVID|
20140419150838||||||525
   RACE ST^^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
ROL|3||Hospitalist|
010782^LANG^DONALD^J^^^^^INV^^^^PROVID~342061^LANG^DONALD^J^^^^^EPC^^^^PROVID|
20140419143134||||||525
   RACE ST^^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
ROL|4||RN|343872^PERKINS^GERALD^D^^^^^EPC^^^^PROVID|20140419032010|20140419103431
ROL|5||RN|344192^SOUTHARD^NIHAET^^^^^^EPC^^^^PROVID|20140419184945|20140420080000
ROL|6||RN|344439^SHIVELY^MICHELLE^^^^^^EPC^^^^PROVID|20140419160615|20140827193000
ROL|7||PCA|346726^CARDINAL^DAPHNE^A^^^^^EPC^^^^PROVID|20140419230415|20140420062713
ROL|8||PCA|347000^ZART^SHERYL^L^^^^^EPC^^^^PROVID|20140420070025
ROL|9||PCA|350051^WAYLAND^SASHANTAA^D^^^^^EPC^^^^PROVID|20140419163505|20140419232342
ROL|10||Consult MD|3040044^HOSPITALISTS^GEN^^^^^^EPC^^^^PROVID|20140419134012
ROL|11||Consult MD|3040054^GEN ORTHOPEDIC ASSOCIATE OF MY FACILITY^^^^^^^EPC^^^^PROVID|
20140419073248|20140420162447|||||525
   RACE ST^SUITE 3220^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
```

Notice the new UPDATE message has been updated with the change.

### Add a New Provider to the Active Care Team

This example shows you how to add a provider who has never been on the care team to the active care team.

1. User is identified by the Care Team Sync application with SER 346133:
   `346133^UNDERWOOD^HELANE^K^^^^^EPC^^^^PROVID`
2. Logs onto Vocera and is assigned Room 22 Bed 1 in the N 4 West Unit at 9:30AM on Aug 27.
3. CTS sends an HL7 message with the last instance of each provider seen for the role RN, adding a new ROL instance for this new provider not yet in the care team history.

The following message is sent by CTS to the Epic system.

```
MSH|^~\&|VCTS_SYNC|VCTS_TST|||20140827093000.000-0400||ADT^A08|345|P|2.3
```

```
EVN|A08|20140827093000.000-0400||VCTS_SYNC|^INTERFACE^VCTS^^^^^^TRHSA
PID|1||3007601^^^INV^MR||CAMPBELL^JESSICA||19751121|F|||^^^^^USA^P|||||||
100000023169||||||||||||N
PV1|1|INP|N4W^N422^01^TRH^R^^^^^^DEPID|ER|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^
^^PROVID|||MED||||Home/Work|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^^^PROVID||27139|
P|||||||||||||||||||||Confirmed Admission|^^^TRH||20140731085952|||152488.74|||2
PV2||ICU||||||20140731|20140827|||Hospital Encounter|||||||||n|N
ROL|4||RN|343872^^^^^^^^EPC^^^^PROVID|20140419032010|20140419103431
ROL|5||RN|344192^^^^^^^^EPC^^^^PROVID|20140419184945|20140420080000
ROL|6||RN|344439^^^^^^^^EPC^^^^PROVID|20140419160615
ROL|12||RN|346133^^^^^^^^EPC^^^^PROVID|20140827093000
```

Only the providers in the role we are updating are sent. Also, notice that CTS includes only the SER number, and not the provider name, in the outbound message.

This affects the membership of the care team for the patient in the EPIC system, which has the side effect of sending back an UPDATE message that should reflect the updated care team. The following message is received by the Vocera Care Team Sync application from EPIC:

```
MSH|^~\&|REG_UPDATE|TRH_TST||Primary|20140827093005|IPRN|ADT^A08|139000|P|2.3
EVN|A08|20140827093005||REG_UPDATE|^INTERFACE^VCTS^^^^^^TRHSA
PID|1||3007601^^^INV^MR||CAMPBELL^JESSICA||19751121|F|||^^^^^USA^P|||||||
100000023169||||||||||||N
PV1|1|INP|N4W^N422^01^TRH^R^^^^^^DEPID|ER|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^
   ^^PROVID|||MED||||Home/Work|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^^^PROVID||27139|
   P|||||||||||||||||||||Confirmed Admission|^^^TRH||20140731085952|||152488.74|||2
PV2||ICU||||||20140731|20140827|||Hospital Encounter|||||||||n|N
ROL|1||Surgeon|
010537^CASEY^GREG^D^^^^^INV^^^^PROVID~340483^CASEY^GREG^D^^^^^EPC^^^^PROVID|
20140419084647||||||525 RACE
   ST^SUITE 3220^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
ROL|2||Anesthesiolo|
005894^SHOSTAK^IGOR^^^^^^INV^^^^PROVID~341957^SHOSTAK^IGOR^^^^^^EPC^^^^PROVID|
20140419150838||||||525
   RACE ST^^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
ROL|3||Hospitalist|
010782^LANG^DONALD^J^^^^^INV^^^^PROVID~342061^LANG^DONALD^J^^^^^EPC^^^^PROVID|
20140419143134||||||525
   RACE ST^^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
ROL|4||RN|343872^PERKINS^GERALD^D^^^^^EPC^^^^PROVID|20140419032010|20140419103431
ROL|5||RN|344192^SOUTHARD^NIHAET^^^^^^EPC^^^^PROVID|20140419184945|20140420080000
ROL|6||RN|344439^SHIVELY^MICHELLE^^^^^^EPC^^^^PROVID|20140419160615
ROL|7||PCA|346726^CARDINAL^DAPHNE^A^^^^^EPC^^^^PROVID|20140419230415|20140420062713
ROL|8||PCA|347000^ZART^SHERYL^L^^^^^EPC^^^^PROVID|20140420070025
ROL|9||PCA|350051^WAYLAND^SASHANTAA^D^^^^^EPC^^^^PROVID|20140419163505|20140419232342
ROL|10||Consult MD|3040044^HOSPITALISTS^GEN^^^^^^EPC^^^^PROVID|20140419134012
ROL|11||Consult MD|3040054^GEN ORTHOPEDIC ASSOCIATE OF MY FACILITY^^^^^^^EPC^^^^PROVID|
20140419073248|20140420162447||||||525
   RACE ST^SUITE 3220^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
```

```
ROL|12||RN|346133^UNDERWOOD^HELANE^K^^^^^EPC^^^^PROVID|20140827093000
```

Notice the new UPDATE message has been updated with the change.

### Add a Former Provider to the Active Care Team Again

This example shows you how to add a provider who has previously been a member of the care team to the active care team.

1. User is identified by the Care Team Sync application with SER 344192: `344192^SOUTHARD^NIHAET^^^^^^EPC^^^^PROVID`
2. Logs onto Vocera and is assigned Room 22 Bed 1 in the N 4 West Unit at 9:30AM on Aug 27.
3. CTS sends an HL7 message with the last instance of each provider seen for the role RN, however for this provider, that has been seen before, it adds a new role instance.

The following message is sent by CTS to the Epic system.

```
MSH|^~\&|VCTS_SYNC|VCTS_TST|||20140827093000.000-0400||ADT^A08|345|P|2.3
EVN|A08|20140827093000.000-0400||VCTS_SYNC|^INTERFACE^VCTS^^^^^^TRHSA
PID|1||3007601^^^INV^MR||CAMPBELL^JESSICA||19751121|F|||^^^^^USA^P||||||
100000023169||||||||||||N
PV1|1|INP|N4W^N422^01^TRH^R^^^^^^DEPID|ER|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^
   ^^PROVID|||MED||||Home/Work|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^^^PROVID||27139|
   P|||||||||||||||||||||Confirmed Admission|^^^TRH||20140731085952|||152488.74|||2
PV2||ICU||||||20140731|20140827|||Hospital Encounter|||||||||n|N
ROL|4||RN|343872^^^^^^^^EPC^^^^PROVID|20140419032010|20140419103431
ROL|6||RN|344439^^^^^^^^EPC^^^^PROVID|20140419160615
ROL|12||RN|344192^^^^^^^^EPC^^^^PROVID|20140827093000
```

As in the earlier use cases, only the providers in the role we are updating are sent. Also, notice that CTS includes only the SER number, and not the provider name, in the outbound message. Finally, notice that CTS does not include the previous ROL instance, ID 5, for the provider.

This affects the care team for the patient in the EPIC system which has the side effect of sending back an UPDATE message which should reflect the updated care team. The following message is received by the Vocera Care Team Sync application from EPIC:

```
MSH|^~\&|REG_UPDATE|TRH_TST||Primary|20140827093005|IPRN|ADT^A08|139000|P|2.3
EVN|A08|20140827093005||REG_UPDATE|^INTERFACE^VCTS^^^^^^TRHSA
PID|1||3007601^^^INV^MR||CAMPBELL^JESSICA||19751121|F|||^^^^^USA^P||||||
100000023169||||||||||||N
PV1|1|INP|N4W^N422^01^TRH^R^^^^^^DEPID|ER|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^
   ^^PROVID|||MED||||Home/Work|||
019917^SANTA^SHELLY^J^^^^^INV^^^^PROVID~342809^SANTA^SHELLY^J^^^^^EPC^^^^PROVID||27139|
   P|||||||||||||||||||||Confirmed Admission|^^^TRH||20140731085952|||152488.74|||2
PV2||ICU||||||20140731|20140827|||Hospital Encounter|||||||||n|N
ROL|1||Surgeon|
010537^CASEY^GREG^D^^^^^INV^^^^PROVID~340483^CASEY^GREG^D^^^^^EPC^^^^PROVID|
20140419084647||||||525 RACE
   ST^SUITE 3220^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
```

```
ROL|2||Anesthesiolo|
005894^SHOSTAK^IGOR^^^^^^INV^^^^PROVID~341957^SHOSTAK^IGOR^^^^^^EPC^^^^PROVID|
20140419150838|||||525
   RACE ST^^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
ROL|3||Hospitalist|
010782^LANG^DONALD^J^^^^^INV^^^^PROVID~342061^LANG^DONALD^J^^^^^EPC^^^^PROVID|
20140419143134|||||525
   RACE ST^^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
ROL|4||RN|343872^PERKINS^GERALD^D^^^^^EPC^^^^PROVID|20140419032010|20140419103431
ROL|5||RN|344192^SOUTHARD^NIHAET^^^^^^EPC^^^^PROVID|20140419184945|20140420080000
ROL|6||RN|344439^SHIVELY^MICHELLE^^^^^^EPC^^^^PROVID|20140419160615
ROL|7||PCA|346726^CARDINAL^DAPHNE^A^^^^^EPC^^^^PROVID|20140419230415|20140420062713
ROL|8||PCA|347000^ZART^SHERYL^L^^^^^EPC^^^^PROVID|20140420070025
ROL|9||PCA|350051^WAYLAND^SASHANTAA^D^^^^^EPC^^^^PROVID|20140419163505|20140419232342
ROL|10||Consult MD|3040044^HOSPITALISTS^GEN^^^^^^EPC^^^^PROVID|20140419134012
ROL|11||Consult MD|3040054^GEN ORTHOPEDIC ASSOCIATE OF MY FACILITY^^^^^^^EPC^^^^PROVID|
20140419073248|20140420162447|||||525
   RACE ST^SUITE 3220^SAN JOSE^CA^95126^^O|
(123)456-7890^^PH^^^123^4567890~(123)456-7890^^FX^^^123^4567890
ROL|12||RN|344192^SOUTHARD^NIHAET^^^^^^EPC^^^^PROVID|20140827093000
```

Notice the new UPDATE message has been updated with the change.

## ADT Event and HL7 Role Reference

### ADT Event Types

This appendix provides a complete list of ADT event types.

***Table 31: ADT event types***

| Event Type | Event Type Description | Action |
|---|---|---|
| A01 | ADMIT A PATIENT | Admit |
| A02 | TRANSFER A PATIENT | Transfer |
| A03 | DISCHARGE A PATIENT | Discharge |
| A04 | REGISTER A PATIENT | Admit |
| A05 | PREADMIT A PATIENT | Admit |
| A06 | TRANSFER AN OUTPATIENT TO INPATIENT | Admit |
| A07 | TRANSFER AN INPATIENT TO OUTPATIENT | Discharge |
| A08 | UPDATE PATIENT INFORMATION | |
| A09 | PATIENT DEPARTING | Discharge |
| A10 | PATIENT ARRIVING | |
| A11 | CANCEL ADMIT | Discharge |
| A12 | CANCEL TRANSFER | |
| A13 | CANCEL DISCHARGE | |
| A14 | PENDING ADMIT | |
| A15 | PENDING TRANSFER | |

| Event Type | Event Type Description | Action |
|---|---|---|
| A16 | PENDING DISCHARGE | |
| A17 | SWAP PATIENTS | Transfer |
| A18 | MERGE PATIENT INFORMATION | |
| A19 | PATIENT, QUERY | |
| A20 | NURSING/CENSUS APPLICATION UPDATES | |
| A21 | LEAVE OF ABSENCE - OUT (LEAVING) | |
| A22 | LEAVE OF ABSENCE - IN (RETURNING) | |
| A23 | DELETE A PATIENT RECORD | |
| A24 | LINK PATIENT INFORMATION | |
| A25 | CANCEL PENDING DISCHARGE | |
| A26 | CANCEL PENDING TRANSFER | |
| A27 | CANCEL PENDING ADMIT | |
| A28 | ADD PERSON INFORMATION | |
| A29 | DELETE PERSON INFORMATION | |
| A30 | MERGE PERSON INFORMATION | |
| A31 | UPDATE PERSON INFORMATION | |
| A32 | CANCEL PATIENT ARRIVING | |
| A33 | CANCEL PATIENT DEPARTING | |
| A34 | MERGE PATIENT INFORMATION - PATIENT ID ONLY | |
| A35 | MERGE PATIENT INFORMATION - ACCOUNT NUMBER ONLY | |
| A36 | MERGE PATIENT INFORMATION - PATIENT ID AND ACCOUNT NUMBER | |
| A37 | UNLINK PATIENT INFORMATION | |

### ROL – Role Information

The optional `ROL` segment contains patient treatment team members at either the patient and/or the encounter-levels.

This segment has the following Retrieves:

- ROL-1: Role Instance ID (EI)
- ROL-2: Action Code (ID)
- ROL-3: Role-ROL (CE)
- ROL-4: Role Person (XCN)
- ROL-5: Role Begin Date/Time (TS)
- ROL-6: Role End Date/Time (TS)
- ROL-7: Role Duration (CE)
- ROL-8: Role Action Reason (CE)
- ROL-9: Provider Type (CE)
- ROL-10: Organization Unit Type (CE)
- ROL-11: Office/Home Address/Birthplace (XAD)
- ROL-12: Phone (XTN)

### ROLE-3-Role

The `ROLE-3-Role` field indicates involvement with the activity being transmitted.

The data type of this field is CE, whose components are as follows:

***Table 32: ROLE-3-Role components***

| Seq | Len | DT | Usage | Cardinality | TBL# | Element Name |
|---|---|---|---|---|---|---|
| 1 | 250 | ST | R | [1..1] | | Identifier |
| 2 | 250 | ST | X | [0..0] | | Text |
| 3 | 250 | ST | X | [0..0] | | Name of Coding System |
| 4 | 250 | ST | X | [0..0] | | Alternate Identifier |
| 5 | 250 | ST | X | [0..0] | | Alternate Text |
| 6 | 250 | ST | X | [0..0] | | Name of Alternate Coding System |

## IP Port Usage

The following table indicates the ports used by the CTS server for IP communication. If a firewall separates the CTS from your organization's network, make sure these ports are open for communication. For more information about IP ports used by Vocera, see the “IP Port Usage” appendix in the *Vocera Infrastructure Planning Guide*.

***Table 33: CTS IP port usage***

| Description | Protocol | Port No |
|---|---|---|
| Browser ⇔ Jetty HTTP Server Signaling | TCP | 443 (for SSL) |
| CTS listener port (HL7 ADT stream) | TCP/MLLP | 2575 |
| CTS JMX Listener | TCP | 9998 |