Reporting Guide for Cisco Unified Customer Voice Portal Release 9.0(1)

First Published: July 06, 2012

Last Modified: July 06, 2015

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

Purpose

This guide documents the Cisco Unified Customer Voice Portal (Unified CVP) reporting functionality and the reporting server.

It explains:

- How to deploy Unified CVP report templates with the Cisco Unified Intelligence Center (Unified Intelligence Center) reporting application.
- The templates that are installed by Unified CVP for import into Unified Intelligence Center.
- The reporting database schema.
- Reporting best practices.

Audience

This guide assists Contact Center managers, Unified CVP system managers, Cisco Unified Contact Center Enterprise (Unified CCE) system managers, VoIP technical experts, and Interactive Voice Response (IVR) application developers.

The expectation is that readers of this guide have a general understanding of Unified CVP and Unified Intelligence Center software and are familiar with the installation, setup, and configuration of both.

Organization

This guide is divided into the following chapters:

Chapter	Description
Reporting Server	Provides an introductory discussion of the reporting server and points to documents that detail its sizing, installation, and configuration.
Cisco Unified Intelligence Center Reporting Application	Explains how to work with Unified CVP templates in the Unified IC reporting product.
Cisco Unified Customer Voice Portal Templates for Cisco Unified Intelligence Center	Presents the Unified CVP templates that can be imported into—and generated from—Unified IC and how they are populated from the reporting database.
Database Schema	Documents the reporting database schema.
Database Management	Discusses concepts to keep in mind while managing the database, such as data retention, backup, and purge.
Reporting Best Practices	Provides a list of best practices.

Related Documentation

Troubleshooting--Point your browser to www.docwiki.cisco.com to read troubleshooting information for Unified CVP and Unified IC. This publicly-available site is offered to all and can be edited by anyone who has a cisco.com user ID and password.

To locate Unified CVP and Unified Intelligence Center tips, click the icon for Cisco Products. Then look under the letter U.

Planning your Unified CVP solution is an important part of the process in setting up Unified CVP. Cisco recommends that you read the *Unified Customer Voice Portal (CVP) Solution Reference Network Design Guide (SRND)* before configuring your Unified CVP solution.

Cisco.com - You can find the complete documentation set for Unified CVP on www.cisco.com.

The documentation set includes:

- installation and upgrade guides
- configuration and administration guides
- planning guides

The documentation set for Unified Intelligence Center is at this location.

Conventions

This manual uses the following conventions:

Convention	Description
boldface font	Boldface font is used to indicate commands, such as user entries, keys, buttons, and folder and submenu names. For example: • Choose **Edit** > **Find**. • Click **Finish**.
italic font	Italic font is used to indicate the following: • To introduce a new term. Example: A *skill group* is a collection of agents who share similar skills. • For emphasis. Example: *Do not* use the numerical naming convention. • A syntax value that the user must replace. Example: IF (*condition, true-value, false-value*) • A book title. Example: See the *Reporting Guide for Cisco Unified Customer Voice Portal* .
`window font`	Window font, such as Courier, is used for the following: • Text as it appears in code or that the window displays. Example: `<html><title>Cisco Systems,Inc. </title></html>`
`< >`	Angle brackets are used to indicate the following: • For arguments where the context does not allow italic, such as ASCII output. • A character string that the user enters but that does not appear on the window such as a password.

Documentation and Service Requests

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

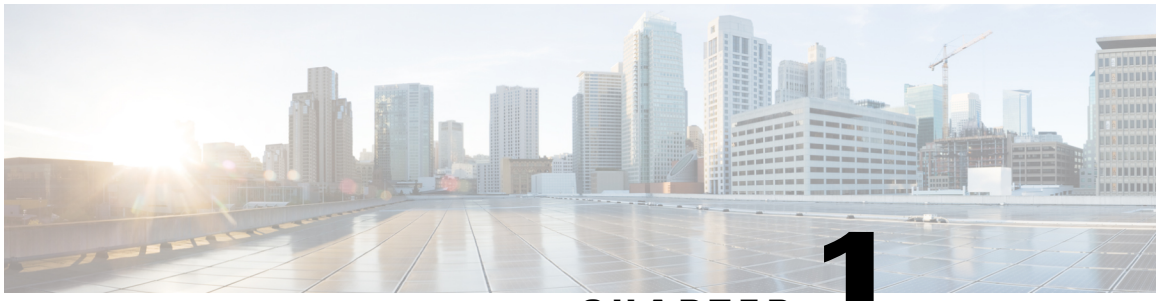
Subscribe to *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.

Documentation Feedback

You can provide comments about this document by sending e-mail to the following address:

ccbu_docfeedback@cisco.com

We appreciate your comments.



CHAPTER 1

Reporting Server

Reporting Service

Reporting is an optional component for Cisco Unified Customer Voice Portal (CVP) installation.

Select **Reporting** during the installation process to install the Reporting Server, which is comprised of the reporting service and the reporting database.

The reporting service receives reporting data from the Interactive Voice Response (IVR) service, the Session Initiation Protocol (SIP) service (if used), and the VoiceXML (VXML) server and transforms and writes this data to the Informix reporting database to provide historical reporting in a call center environment.

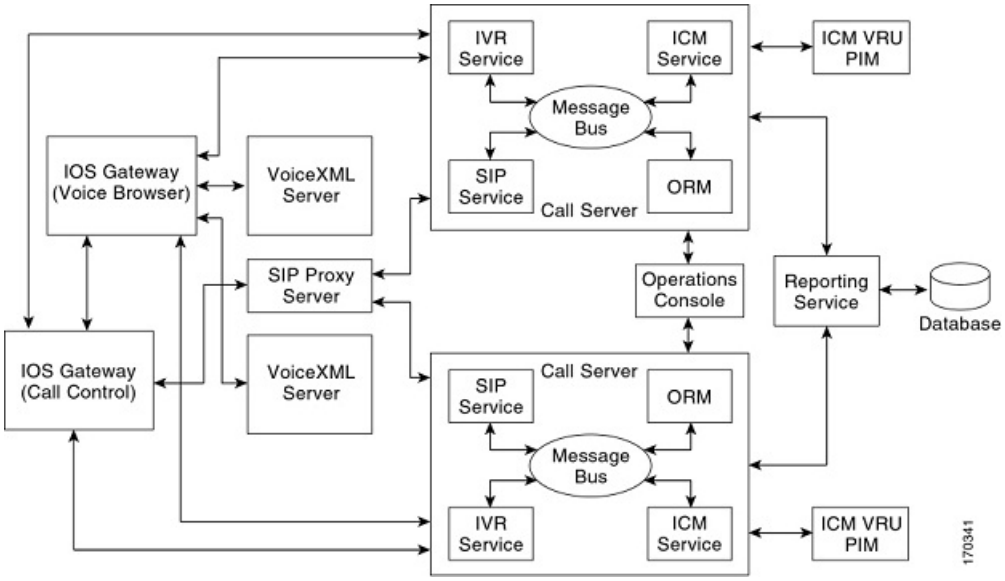
Reporting data includes summary information about call activity, which assists call center managers in reviewing and managing daily operations. It can also include operational detail data for various IVR applications.

Reporting Architecture

The following diagram shows the Unified CVP architecture. For clarity, the diagram separates the reporting service and the database.

Note The connection of the Operations Console to the call server through an OAMP Resource Manager (ORM) is simply indicative, because the ORM is invisible to the end user. An ORM is co-located with each managed Unified CVP component, and the Operations Console is connected to each component.

Figure 1: CVP Architecture



A call server is a physical machine on which the IVR service, the SIP service, and the Cisco Unified Intelligent Contact Management (ICM) service can reside. The call server uses a central messaging bus to allow each service to communicate.

The reporting service connects to the message bus through either an in-process plug-in or an out-of-process plug-in, depending on whether the reporting service resides in the same Java Virtual Machine (JVM) with the message bus system.

The service listens to all messages passing through the message bus and captures both call-state change messages sent from SIP and IVR services or reporting messages sent from a VXML Server.

The reporting service then parses those messages, converts them into batches of applicable Structured Query language (SQL) statements, and executes them into an SQL database using the Java Database Connectivity (JDBC) Application provisioning interface (API).

The reporting service can also receive and process Unified CVP messages related to Unified CVP system administrative tasks, such as turning on or off debugging and querying statistics. As the Figure 1 shows, the reporting service can be shared by multiple Call Servers that belong to the same Unified CVP deployment.

Note A deployment needs only one reporting server. During temporary database outages, messages are buffered to file and are inserted into the database after the database comes back on line. The amount of time that messages can be buffered depends on the system throughput. See Reporting User.

If your environment uses more than one reporting server, be aware that:

- Each Call Server and each VXML Server can be associated with only one reporting server.
- Reports cannot span multiple Informix databases.

Although Unified CVP does not have a native reporting engine, its installation includes reporting templates designed for use with the Unified Intelligence Center (Unified IC) reporting application. You can import these templates into Unified IC and run them from the Unified IC interface.

Cisco Unified Customer Voice Portal Reporting Server Deployment Options and Sizing

You can find the Unified CVP reporting solution deployment options, together with related sizing requirements in the *Cisco Unified Customer Voice Portal Design Guide* at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-implementation-design-guides-list.html.

Cisco Unified Customer Voice Portal Reporting Server Installation and Upgrade

Explanations and procedures regarding the installation and upgrade of the Unified CVP reporting server are documented in the Installation and Upgrade Guide for Cisco Unified Customer Voice Portal.

Topics in the Installation and Upgrade guide include:

- Installing the reporting component
- Specifying the reporting password
- Excluding the reporting server from anti-virus software port blocking
- Applying a license file to the reporting server
- Changing licensing information for the reporting server
- Upgrading the reporting server
- Adding reporting capability to the VXML Server
- Backing up and purging of the reporting database

Cisco Unified Customer Voice Portal Reporting Server Setup

You can find explanations and procedures regarding the configuration and maintenance of the Unified CVP Reporting server in the *Administration Guide for Cisco Unified Customer Voice Portal* and in the *Configuration Guide for Cisco Unified Customer Voice Portal*.

Topics in the operations console help and in the Configuration guide include:

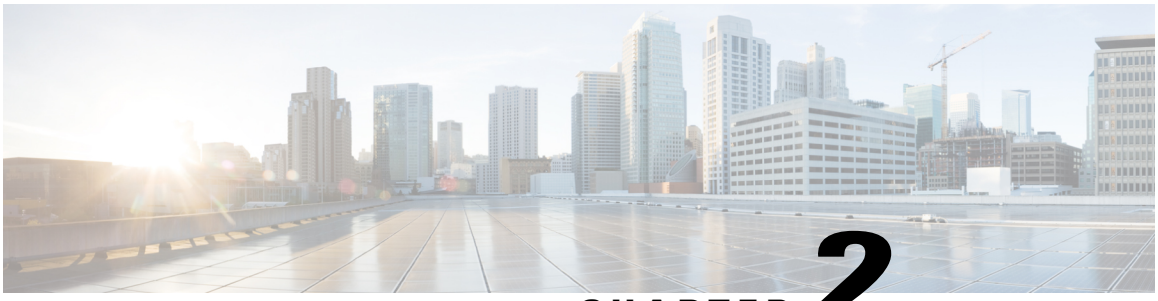
- Reporting server statistics
- Adding a reporting server

- Editing a reporting server
- Deleting a reporting server
- Finding a reporting server
- Applying a license to a reporting server
- Configuring a VoiceXML server for reporting (adding and editing)
- Applying inclusive and exclusive VoiceXML filters for reporting
- Transferring a file to multiple devices

Database Maintenance

Through the Operations Console, Unified CVP provides access to database maintenance and enables you to perform administrative tasks such as backups and purges.

See *Configuration Guide for Cisco Unified Customer Voice Portal* for details on database operations.



CHAPTER 2

Cisco Unified Intelligence Center Reporting Application

This release offers Cisco-designed reporting templates for Cisco Unified Customer Voice Portal (Unified CVP). They are installed with Unified CVP, and you can import them into the Cisco Unified Intelligence Center (Unified Intelligence Center) reporting application.

Although it is possible to run reports with other reporting tools, such as Crystal Reports, and to customize the Unified CVP reports, your Cisco support provider can assist you only with the Unified CVP templates that are installed with Unified CVP and then imported into the Unified Intelligence Center reporting application.

Use the Unified Intelligence Center reporting application to import, run, Save As (clone), and modify the imported templates; to schedule and print the Unified CVP reports; and export them to Microsoft Excel.

If you have a Premium license for Unified Intelligence Center, you can also modify the SQL query for the report definition, and design custom reports and report definitions using Save As.

This chapter explains how to set up the Unified Intelligence Center application to run Unified CVP report templates. It includes an overview of working with reports in Unified Intelligence Center. You can find full details on installing, configuring, and operating Unified Intelligence Center in the Cisco Unified Intelligence Center user documentation section of Cisco.com.

- Basic Cisco Unified Intelligence Center Reporting Concepts, page 17
- Report Functions, page 19
- Cisco Unified Intelligence Center Reporting User Roles, page 30

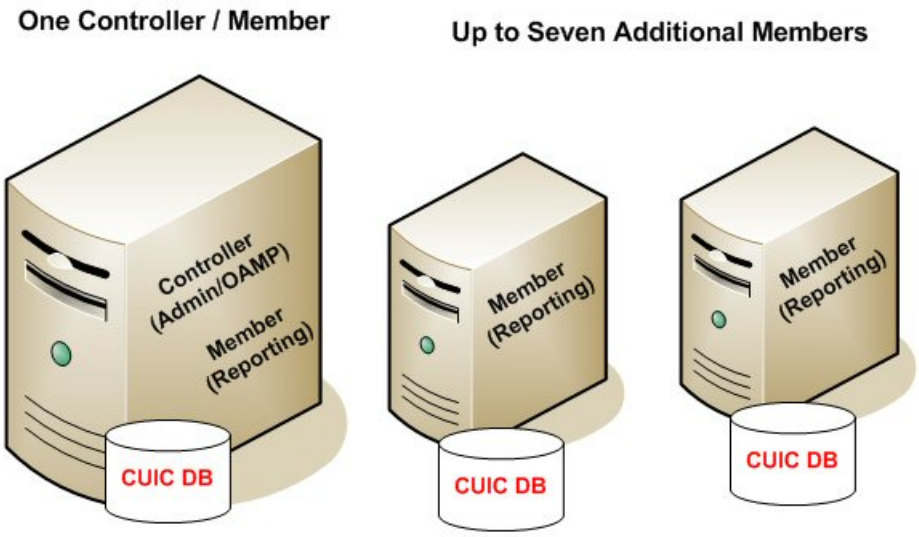
Cisco Unified Intelligence Center Installation and Setup

Who can install and set up Unified Intelligence Center: any user in your organization.

Unified Intelligence Center is installed by DVD media on a Cisco Unified Voice Operating System (VOS) platform. VOS is an appliance or *closed box* model and does not support navigation into, or manipulation of, the file system.

Unified Intelligence Center can be installed as a standalone server or as a cluster of a maximum of eight server nodes. Unified Intelligence Center consists of one mandatory publisher node (called the *Controller*) and a maximum of seven subscriber nodes (called *Members*). The Controller node includes a Member; thus a deployment can consist of a Controller alone.

Figure 2: Unified IC Cluster



Unified Intelligence Center has two web interfaces:

- The **Administration** application

  The administration application is the OAMP interface that provides application-level configuration for Unified Intelligence Center. This interface is offered on the Controller node only and is used by Unified Intelligence Center superusers.

- The Unified Intelligence Center **Reporting** application

  The Unified Intelligence Center reporting application is the interface for reporting users who can have various user roles pertinent to reporting.

You can find the installation instructions in the *Installation Guide for Cisco Unified Intelligence Center*.

Acquire License Online

Who can acquire a license online: any user in your organization.

After you install the Unified Intelligence Center Controller node, publisher and before you install any Member nodes the subscriber, you must contact Cisco to request a Unified Intelligence Center license.

Procedure

Step 1 To acquire the license file, go to the Cisco Product License Registration website at this URL: https://tools.cisco.com/SWIFT/LicensingUI/Home.

Step 2 If you do not have a Product Authorization Key (PAK), click the available licenses link.

Step 3 Scroll to Unified communications and click **Cisco Unified Intelligence Center**.

Step 4 Enter your MAC Address, accept the agreement, and enter your Registrant Information.
The MAC Address appears online at the end of the installation. If you need to find the MAC Address again, follow these steps to obtain it:

1. Sign in to the server node, using the credentials of the System Administration user.
2. Enter this CLI command: show status.

Step 5 Follow prompts to complete the registration windows.
You will receive an email from Cisco that contains your license file as an attachment. The file format is *.lic.

Step 6 Save the license file in a location where the System Application User can access it.

Warning Save a backup copy of this file. You can open a *.lic. file to look at it, but *do not make any changes to it*. Changing the file invalidates the license.

Step 7 Apply the license.

Unified Intelligence Center has four license types: Demo, Lab/Trial, Standard, and Premium. See the *Installation and Upgrade Guide for Cisco Unified Intelligence Center* at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-intelligence-center/products-installation-guides-list.html for an explanation of each license type.

What to Do Next

Sign In to Administration Console, on page 7

Sign In to Administration Console

Who can sign in to the administration console: The System Application User who is the default Superuser.

To upload the license, you must sign in to the Unified Intelligence Center Administration Console. This is the OAMP interface for Unified Intelligence Center. The first person who signs in to the Administration application must do so using the user ID and password that were defined for the System Application User during the installation. This user is the initial Superuser for Unified Intelligence Center Administration.

Procedure

Step 1 Enter this URL: `http://<HOST ADDRESS>/oamp`, where HOST ADDRESS is the IP address or hostname of your Controller node.

Step 2 Enter the System Application User ID and password that you defined during installation.

Upload License

Who can upload the license: The System Application User who is the default Superuser.

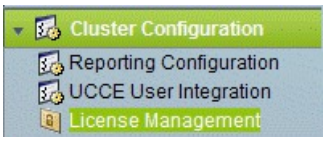
As soon as the System Application User signs in, the user must upload the license file. The file is uploaded to the Controller publisher node and, within a minute, is automatically replicated to all nodes in the cluster.

The partner must obtain a unique license and apply it to the imported Unified Intelligence Center servers at the customer site.

Procedure

Step 1 In Cisco Unified Intelligent Center Administration, choose **Cluster Configuration** > **License Management**. to open the **License File Management** page.

Figure 3: License File Management



Step 2 Click **Browse**.

Step 3 Navigate to the location where the *.lic file was saved.

Step 4 Click **Apply License** to load the license.
A message appears indicating that the license file was uploaded successfully and will be distributed to other nodes (if any) in the cluster in approximately one minute.

Note The databases are polled once a minute for changes. The license replication is not immediate but occurs within a minute.

What to Do Next

Create Reporting Users, on page 8

Create Reporting Users

Who can create a user:

- Initially, the System Application User who is the default Superuser.
- Eventually, any Superuser.

Unified CVP reporting users can sign in to Unified Intelligence Center only if they exist in the Administration console as Superusers or if Active Directory (AD) is configured in the Unified Intelligence Center Administration console for their domain:

- Superusers who are added are considered to be IP Multimedia Subsystem (IMS) users.
- Users who are authenticated through Active Directory are considered to be Lightweight Directory Access Protocol (LDAP) users. For more information, see Configure Active Directory Server, on page 9.

Both IMS users and LDAP users can log in to Unified Intelligence Center reporting and are restricted to the limited Login User role until the Unified Intelligence Center reporting security administrator gives them additional roles and flags them as active users.

Although you can create a user on the Unified Intelligence Center User List page, an entry on the User List is not sufficient for that user to sign in to the Unified Intelligence Center. One reason to create users on the User List page is to expedite the permissions for users before their Active Directory domain is configured.

Create Superusers

Procedure

Step 1 Log in to the Cisco Unified Intelligence Center Administration Console `(http://{hostname}/oamp)`.

Step 2 Navigate to **Admin User Management > Admin User Management** to open the Users page.

Step 3 Click **Add New** to add and configure a new user or click an existing username to edit the configuration for that user.
This page has three tabs: General, Credentials, and Policy. For information about completing these tabs, see *Administration Console User Guide for Cisco Unified Intelligence Center* at http://www.cisco.com/en/US/products/ps9755/prod_maintenance_guides_list.html or the Administration console online help.

Step 4 Click **Save**.

Configure Active Directory Server

Fields on the Active Directory tab configure the Active Directory server to authenticate reporting users as they log in to the Unified Intelligence Center Web application.

You must configure Active Directory for the Unified ICM/CC supervisors so that they can sign in as Unified Intelligence Center Reporting users.

Note Cisco Unified Intelligence Center uses LDAP V2 which does not support all Unicode characters that are used in the first name or surname of LDAP users.

Active Directory is not used to authenticate Administration Super Users. These Super Users can only be authenticated through the local database. The first Super User is added during installation. All other Super Users are added through the Admin User Management interface, and their credentials are encrypted into the local database.

To navigate to this page, choose **Cluster Configuration** > **Reporting Configuration** and select the Active Directory tab.

Table 1: Fields on This Tab

Field	Description
Host Address and Port for Primary Active Directory Server	Provide the Host name or IP address and the port of the Primary Active Directory server. The port defaults to 389.
Host Name and Port for Redundant Active Directory Server	Provide the Host name or IP address and the port of the Redundant Active Directory server. The port defaults to 389.
Use SSL	Check these boxes if you want the connection from the Unified device to the Active Directory connection to be encrypted with SSL while doing authentication.
Manager Distinguished Name	Enter the Manager Distinguished Name used to login to the Active Directory server, for example, on a default installation of Microsoft AD: `CN=Administrator, CN=users, DC=MYSERVER, DC=COM`. Replace *MYSERVER* and *COM* with your respective hostname. **Note** If users other than the LDAP administrator, is configured as Manager Distinguished Name in the OAMP LDAP configurations, they should have the following rights: **1** User search permissions on the domain. **2** Read access to the user objects and their attributes. **3** Read access to the base DN **4** Permission to bind to LDAP.
Manager Password	Enter the Active Directory manager password.
Confirm Manager Password	Confirm the Active Directory manager password.
User Search Base	Specify the user search base. For example, on a default installation of Microsoft AD, `CN=users, DC=MYSERVER, DC=COM`, replace *MYSERVER* and *COM* with your respective hostname. **Note** This example assumes you placed the users in the USERS subtree of AD. If you created a new organizational unit within your subtree, then the syntax would be: `OU=MYUSERS, DC=MYSERVER, DC=COM`. Note that it is "OU=MYUSERS" instead of "CN=MYUSERS".

Field	Description
Attribute for User ID	Whenever a user logs in, Unified Intelligence Center searches for that user in the LDAP (Lightweight Directory Access Protocol) using the login attribute specified in the LDAP configuration. After the user is found, the full DNS of the user is extracted and used for authenticating the user. The login attribute specified in the LDAP configuration will be the property against which LDAP search is issued to find the matching username. If you do not know which attribute to use, use *sAMAccountName*, which is the default Microsoft username attribute. Different organizations settle on different LDAP attributes to identify the user name across the organization, depending on the tools used to administer LDAP within their organizations. This attribute allows you to customize the login depending on the attribute used. Even a custom attribute can be specified using this dialog. *sAMAccountName* indicates the user attribute to search the user for is the *userPrincipalName*. *sAMAccountName* contains just the short user name. For example, jDoe for the user John Doe. *userPrincipalName* indicates the user attribute to search the user for is the *userPrincipalName*. This attribute contains user name in the email format, in the form user@compay.com. Therefore this entire string becomes the user name and not just user. Therefore when this attribute is selected this entire form of username has to be typed in as the username in the login box. **Custom User Attribute** allows you to specify the attribute used for searching the user in LDAP. **Note** Custom User attributes are not validated and are used as is. Ensure that the correct case and attribute name are used. Contact your Active Directory Administrator for the correct attribute to use.

Field	Description
UserName Identifiers	Users are stored in Unified Intelligence Center in the format <UserName Identifier>\<username> The UserName Identifiers are used to identify the different kinds of users within Unified Intelligence Center. For example, local, LDAP, user-synced user, users from different LDAP domains and so on. The username identifier has to be first declared for use in this page before it can be used. When LDAP is configured at least one identifier must be configured and set as default so that LDAP users can be identified in the system. When *userPrincipalName* are used as the LDAP attribute for searching users in the domain, valid formats for username has to be supplied in the form of *@company.com*. Unlike *sAMAccountName* any identifier cannot be configured. Only existing identifiers as configured in the LDAP Active Directory *userPrincipalName* attribute should be configured here. Users are created as company\user. UserSychronization brings in users in format <syncdomain>\username and collections will have users in the same format. It is therefore required that these users login to Unified Intelligence Center using the syncdomain\user syntax. To enable please add syncdomain or @syncdomain.com (if you are using *userPrincipalName*) to the list of valid identifiers. The maximum allowed length of a UserName identifier is 128 characters.
set Default. (UserName Identifier)	Default identifiers allows users to login without typing the full domain identifier (<domain>\user) or the *userPrincipalName* suffixes to usernames (user <@company.com>) on the Login page. It can be set by choosing one of the Identifiers from the list box and by clicking the **Set Default** button. Users who need to use any other identifier can still login by typing their full identifier in the login box. For example, domain2\user or netbiosname\user, provided those identifiers have already been configured.
Test Connection button	Click to test the connection to the primary and secondary LDAP servers and display the connection status.

- **Save** saves the configuration information you entered for the active directory. Clicking **Save** *does not validate the configuration*.
- **Refresh** rolls back all changes since the last save and reloads the values set during the last save.

The UserName Identifier list box is pre-populated with the UserName Identifiers after upgrade to 9.0 release from 8.x releases based on the list of user names stored in the Unified Intelligence Center database. The most frequently occurring identifier in the list of user name is auto-selected as the default.

Note You cannot save LDAP configuration unless you choose a default Identifier from the UserName Identifiers list box and clicking the Set Default button.

Sign In to Cisco Unified Intelligence Center Reporting Interface

Who can sign in to the Unified Intelligence Center reporting interface:

- Initially, the System Application User who is the default Superuser.
- Eventually, any Unified CVP user who was created in the Administration Console as an IMS superuser or an LDAP user.

Perform the following procedure to sign in to the Unified Intelligence Center reporting interface.

Procedure

Step 1 Sign in to the Cisco Unified Intelligence Center Administration Console (`http://{hostname}/oamp`).

Step 2 Navigate to **Control Center > Device Control**.

Step 3 Click on the name of the Member node you want to access. This opens the Cisco Unified Intelligence Center login page for that member.

Step 4 Enter your user ID and password. The Overview page appears.

What to Do Next

Cisco Unified Customer Voice Protocol Reporting User Role Additions

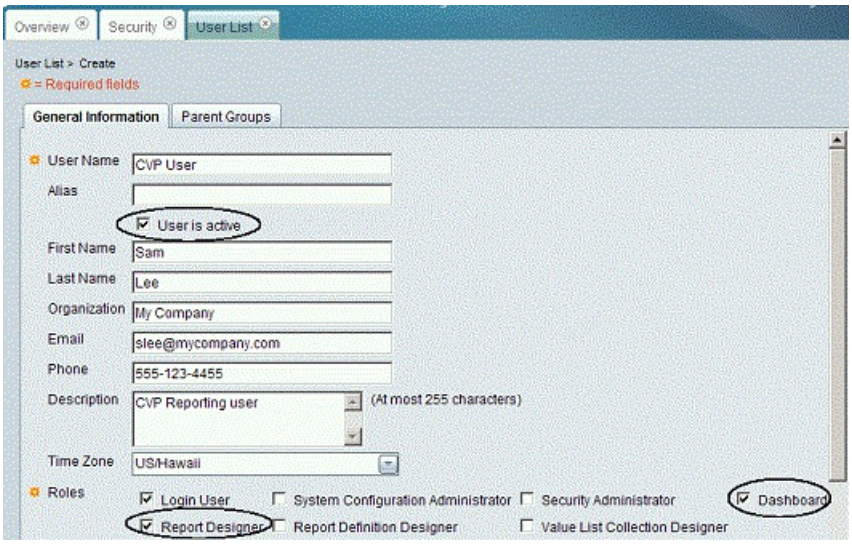
Who can add Unified CVP user roles:

- Initially, the system application user (see System Application User, on page 32) who has full permissions in Unified Intelligence Center reporting
- Eventually, any security administrator (see Security Administrator, on page 31).

Once Unified CVP users log in to Unified Intelligence Center, they are added to the Unified Intelligence Center database and appear on the user list.

New users are initially defined as Login Users: the lowest level user role of Unified Intelligence Center. A Unified Intelligence Center Security Admin user must access the User List page to check a **User is Active** check box and to grant additional user roles to the user.

Figure 4: User List Page



What To Do Next

Create Data Source for Cisco Unified CVP Report Data, on page 14.

Create Data Source for Cisco Unified CVP Report Data

Similar to creating an Open Database Connectivity (ODBC) connection, this task is necessary to access the Unified CVP reporting data.

In Unified Intelligence Center, the user must perform this task with the System Configuration Administrator User Role.

Perform the following procedure to create a data source.

Procedure

Step 1 Log in to the Unified Intelligence Center at `https://<hostname of CUIC Publisher>:8444/cuic`.

Step 2 Select the **Data Sources** drawer to open the Data Sources page.

Step 3 Click **Create** to open an Add Data Source window.

Step 4 Complete fields on this page as follows:

Field	Value
Name	Enter the name of this data source. Report Designers and Report Definition Designers do not have access to the Data Sources page but can see the list of Data Sources when they create custom reports. To benefit those users, give a new Data Source a meaningful name.
Description	Enter a description for this data source.
Type	Choose **Informix**. **Note** Type is disabled in Edit mode.
Database Host	Enter the IP address or Domain Name System (DNS) name for the Unified CVP Reporting server.
Port	Enter the port number. Typically, the port is 1526.
Database name	Enter the name of the reporting database on the Unified CVP reporting server.
Instance	Specify the instance name of the desired database. By default, this is `cvp`.
Timezone	Choose the correct time zone for the data stored in the database. In locations that change from Standard Time to Daylight Savings Time, this time zone is updated automatically.
Database User ID	Enter the user ID of the Reporting User who is configured in the Operations Console to access the Unified CVP reporting database. (The cvp_dbuser account is created automatically during Unified CVP Reporting server installation.)
Password and Confirm Password	Enter and confirm the password for the database user.
Charset	Choose UTF-8. **Note** If this field is not set correctly, the Unified Intelligence Center cannot connect.
Default Permissions	View or edit the permissions for this datasource for My Group and for the All Users group.

Step 5 Click **Test Connection**.

If the status is not Online, review the error message to determine the cause and edit the data source accordingly.

Step 6 Click **Save** to close the Add Data Source window.
The new data source appears on the Data Sources list.

Obtain Cisco Unified Customer Voice Portal Report Templates

Who can obtain import Unified CVP report templates: any user in your organization.

To import Unified CVP report templates complete the following:

Procedure

Step 1 On the Unified CVP Reporting Server, click **Start**.
Step 2 In the search box, type **%CVP_HOME%\CVP_Reporting_Templates** and press **Enter**.
Step 3 Choose the files and copy them to the client computer from which you will launch the Unified IC Reporting web application.

What to Do Next

Import Unified CVP Report Templates and Set Data Source, on page 16

Import Unified CVP Report Templates and Set Data Source

Who can do this:

- Initially, the System Application User who has full permissions in Unified Intelligence Center Reporting.
- Eventually, any Report Designer who has full permissions.

Before reporting users can run the Unified CVP report templates in the Unified Intelligence Center reporting application, a Unified IC reporting user with permission to do so must import them into Unified IC and associate them with the Unified CVP Data Source.

Procedure

Step 1 Launch the Unified Intelligence Center web application using the `URL http://<HOST ADDRESS>:8444/cuic`
Step 2 Enter your User Name and Password.

This opens the Overview page.

Step 3 Click**Reports**.

Step 4 Right-click the top Reports folder and select **Create Sub-Category**.

Step 5 Name the new sub-category as a container for Unified CVP reports. Click **OK**.

Step 6 Click **Import Reports**.

Step 7 Browse to the location where you copied the Unified CVP Reporting templates files.

Step 8 Select a report.
This populates the File Name with the full path for the report.

Step 9 Click **Import**.

Step 10 From the **Data source for Report Definition** and **Data source for value List** drop down lists, Choose the Data source you created to access the Unified CVP Reporting database.

Step 11 **Save to** the Unified CVP sub-category folder you created in Step 5.

Step 12 Click **Import**.

Step 13 Repeat for the callback templates.

Imported Cisco Unified Customer Voice Protocol Report Templates

Unified Intelligence Center has stock report templates and custom report templates.

Stock report templates are the UCCE templates that are installed with Unified Intelligence Center. These templates are populated from the Unified ICM/CC database. They are protected from modification and must be cloned using “Save As”.

Custom report templates are templates that are imported. Unified CVP templates, therefore, fall in the custom template category.

Unified Intelligence Center does not protect custom templates from modification.

Best Practice: retain the imported Unified CVP templates and create Save As versions of the Unified CVP reports and Unified CVP report definitions.

Basic Cisco Unified Intelligence Center Reporting Concepts

The following table provides information about basic Unified Intelligence Center reporting concepts.

Table 2: Basic Unified Intelligence Center Reporting Concepts

Term	Explanation
Dashboards	Dashboards are web pages can display reports, report lists, scheduled reports, notes, and web-based elements--such as URLs and widgets--that are relevant to specific workflows and responsibilities.

Term	Explanation
Data Sources	A data source is a connection to a database from which reports are populated. Each data source has a configuration page with the IP address, username, password, and database type for a database used by Unified Intelligence Center.
Reports	Reports contain data sets that are extracted by database queries and that can be displayed in various views--as grids, as charts, and as gauges. All reports are based on Report Definitions.
Report Definitions	A Report Definition defines the interface for a report. Each Report Definition contains the dataset that is obtained for a report including the SQL query, the fields, the filters, the formulas, the refresh rate, and the key criteria field for the report. Unified Intelligence Center separates Reports from Report Definitions. Only users who have a Premium license can view, create, and edit Report Definitions.
Report Templates	Report Templates are well-formed XML files based on Report Definitions.
Report Views	A report view is a layout presentation for the data that is retrieved for the report. Unified Intelligence Center supports three types of views: • Grid Views • Chart Views • Gauge Views All reports have a grid view. The Unified CVP Call Traffic reports also have a chart view. You can create many views for a report, can define the default view for a report, and can change a report view once the report is generated. You cannot delete all views. Every report must have at least one view.
Value Lists	Value lists contain all reportable items of the same type, for example, all agents or all skill groups.
Collections	Collections are subsets of value lists that can be used to control the amount of data that users can select to populate a report.
Thresholds	You can set a threshold for a field in a report grid to configure that field to display in a distinctive format.

Term	Explanation
User Groups	User Groups are constructs that allow security administrators to partition Unified Intelligence Center functionality. Creating User Groups expedites the provisioning process when multiple users need the same access to dashboards and reports, or when users require distinct permissions and features based on regional or organizational requirements.
User Permissions	Users have permissions associated with the groups in which they are members, and each member of a group has specific permissions in that group. There are three levels of permissions, represented by three check boxes in the User Information page: READ, EXECUTE, and WRITE.
User Roles	User Roles confer the actions and capabilities that a user has in Unified Intelligence Center. There are seven User Roles, and each user can have multiple roles.

Report Functions

Who can manage reports: Report Designer, on page 31

Users with the Report Designer user role manage reports from the Available Reports page, which opens when you click the **Reports** drawer.

This page displays a category for the imported Unified Intelligence Center templates. Expand that category and right-click a report to open a menu of functions you can perform.

Based on permissions, a Report Designer can perform some or all of these functions:

- Run
- Schedule
- Edit
- Rename
- Edit Views
- Export
- Delete

These functions are explained in the *User Guide for Cisco Unified Intelligence Center Reporting Application* and in the Unified Intelligence Center online help.

Run Report

To run a report, double-click the report name or right-click the report name and select **Run**.

If the Report Designer has selected to bypass the filter dialog on the Report Editor page, the report opens immediately.

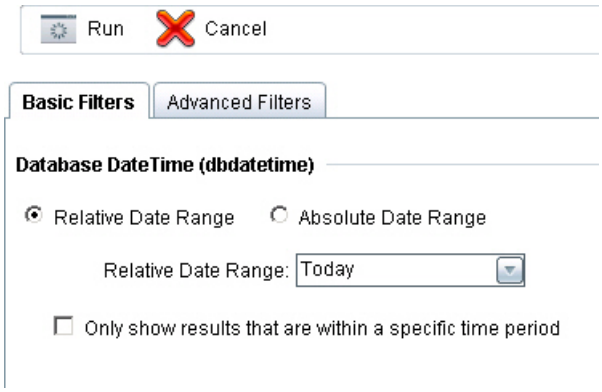
If the report designer did not select to bypass the filter dialog, selecting a report to view opens the Filters page for that report. See Basic and Advanced Report Filters, on page 20.

Once you select to run and filter a report, the report displays in the Report Viewer.

Basic and Advanced Report Filters

In most cases, when you run a report, a filter page opens where you select start and end dates to filter the report by a date range. For reports based on simple queries, this page has two tabs: Basic Filters and Advanced Filters. For reports based on stored procedures or anonymous blocks, the filter is a single page.

Figure 5: Filters



Note If the Report Designer has defined a Default Filter for the report and selected Bypass Filter on the Report Editor page, the report runs directly, and filters page does not display.

Basic Filters for Simple Query Reports

Use this page to filter the report to display data for a specific data or date range.

Relative Date Range:

- From the **Relative Date Range** drop-down, select from Today, Yesterday, This Week, Last Week, This Month, Last Month, Year to Date, or Last Year.
- Check **Only show results that are within a specific time period** to enter a starting and ending time range. If you do not check this box, the report shows all values from 12:00 a.m. of the first date in your range through 11:59 p.m. of the last date in the range.

Absolute Date Range:

- In the **From** and **To** fields, click the calendar icons to select a starting and ending date range.
- Check **Only show results that are within a specific time period** to enter a starting and ending time range. If you do not check this box, the report shows all values from 12:00 a.m. of the first date in your range through 11:59 p.m. of the last date in the range.
- Check **Only show results that are on certain days of the week** to check/uncheck days. By default, all days of the week are checked.

It is possible to capture gigabytes of Unified CVP data in a single day. Select filtering values that target time ranges and subsets of data that return in a reasonable time. Date Time columns are crucial selections. Filtering to return large quantities of data may exceed the capacity of the reporting server database.

You can select the date range and click **Run** or further filter the report on the Advanced Filters tab and then click **Run**.

Refine Database Query Report Results Using Advanced Filter

For reports defined as Database Queries, this second tab on the Filter page allows you to further refine the results in a report.

This tab is a list of *all* the fields that have Available in Filter checked in the Edit Field Properties tab of the Report Definition. It shows the field name, display name, and description.

Select one or more of these fields and then click **Edit** to indicate *any value* or a filtered value.

Filter criteria depend on the field type (Date, Decimal, Value List, String, or Boolean).

To use the advance filter complete the following:

Procedure

Step 1 Select one or more of the fields in the Advanced Filter tab, and then click **Edit** to indicate *any value* or a filtered value.
Filter criteria depend on the field type (Date, Decimal, Value List, String, or Boolean).

- For type **DateTime**, click **Edit** to specify any value or to filter by selecting either Relative Date Range or Absolute Date Range.

  For both Relative and Absolute date ranges, you can further indicate a specific time period and certain days of the week.
- For type **Decimal**, click **Edit** to specify any value or to select an Operator from Equal To, Not Equal To, Less Than, Less Than or Equal To, or Greater Than and then enter a value, for example, Operator = Greater Than and Value = 16.5.
- For type **String**, click **Edit** to specify any value or to filter by selecting an Operator from Equal To, Not Equal To, or Matches Pattern and then entering a value for the string, for example, Operator = Matches Pattern and Value = Team Green.
- For type **Boolean**, click **Edit** to specify any value or to filter by selecting an Operator and then selecting True or False.

Step 2 Run the report.

Here is an Application Summary [15] report that is not filtered for Application Name.

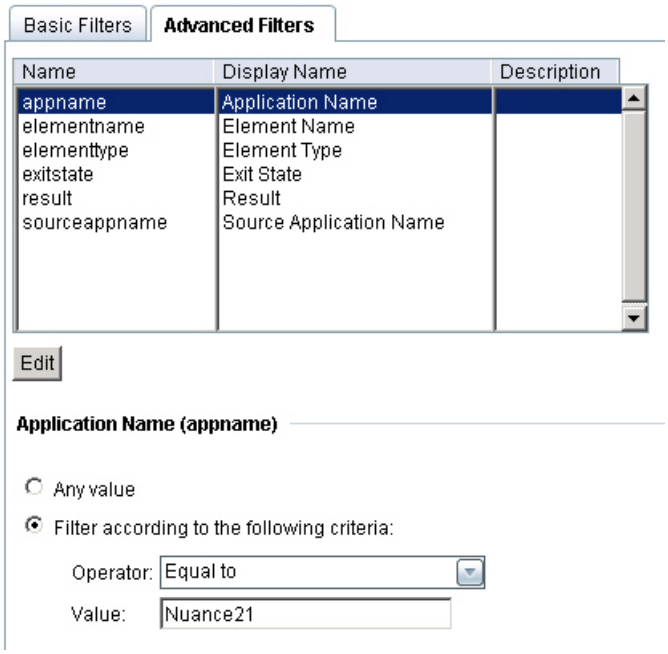
Figure 6: Application Summary [15] Report, No Advanced Filter

Application Summary Report (15)

Database DateTime	Application Name	Source Application Name	Element Name	Element Type	Exit State	Result	count	avg_elapsed
3/10/10 10:00 PM	Nuance21	null	ElementSelectForm	Voice	none	Hang Up	1	0:00:09
3/10/10 10:00 PM	Nuance21	null	Grammar Extension Setter_01	Action	done	Normal	1	0:00:00
3/10/10 10:00 PM	Nuance21	null	Subdialog Start_01	Subdialog_Start	done	Normal	1	0:00:00
3/10/10 10:00 PM	Nuance21	null	end	End	done	Normal	1	0:00:00
3/10/10 10:00 PM	Nuance21	null	start	Start	next	Normal	1	0:00:00
3/7/10 12:00 AM	Perl_31Msgs	null	Audio_Exit	Voice	done	Normal	6072	0:00:00
3/7/10 12:00 AM	Perl_31Msgs	null	Audio_Exit	Voice	done	Normal	431	0:00:00
3/7/10 12:00 AM	Perl_31Msgs	null	DigitSelectionNumber	Voice	done	Normal	413	0:00:07
3/7/10 12:00 AM	Perl_31Msgs	null	DigitSelectionNumber	Voice	done	Normal	6038	0:00:07
3/7/10 12:00 AM	Perl_31Msgs	null	ElementSelectAnalysis	Decision	number	Normal	507	0:00:00
3/7/10 12:00 AM	Perl_31Msgs	null	ElementSelectAnalysis	Decision	number	Normal	6083	0:00:00
3/7/10 12:00 AM	Perl_31Msgs	null	ElementSelectForm	Voice	done	Normal	6074	0:00:08
3/7/10 12:00 AM	Perl_31Msgs	null	ElementSelectForm	Voice	done	Normal	507	0:00:06
3/7/10 12:00 AM	Perl_31Msgs	null	Grammar Extension Setter_01	Action	done	Normal	551	0:00:00
3/7/10 12:00 AM	Perl_31Msgs	null	Grammar Extension Setter_01	Action	done	Normal	6110	0:00:00
3/7/10 12:00 AM	Perl_31Msgs	null	NumberSelectAnalysis	Decision	number_dtmf	Normal	421	0:00:00
3/7/10 12:00 AM	Perl_31Msgs	null	NumberSelectAnalysis	Decision	number_dtmf	Normal	6070	0:00:00

Here is an Advanced filter to restrict the content to the Nuance21 application name:

Figure 7: Setting an Advanced Filter



Here is an Application Summary [15] report for the same date range that is filtered to display only the Nuance21 application:

Figure 8: Application Summary [15] Report, Filtered

Application Summary Daily

Database DateTime	Application Name	Source Application Name	Element Name	Element Type	Exit State	Result	count	avg_elapsed
3/10/10	Nuance21	null	ElementSelectForm	Voice	none	Hang Up	1	0:00:09
3/10/10	Nuance21	null	Grammar Extension Setter_01	Action	done	Normal	1	0:00:00
3/10/10	Nuance21	null	Subdialog Start_01	Subdialog_Start	done	Normal	1	0:00:00
3/10/10	Nuance21	null	end	End	done	Normal	1	0:00:00
3/10/10	Nuance21	null	start	Start	next	Normal	1	0:00:00
							5	

Filter Page for Stored Procedure and Anonymous Block Reports

The filter page for a report that is based on a Stored Procedure, as is the case for the Call Detail report, shows all parameters defined in the Stored Procedure.

The Call Detail report is the only report installed with Unified CVP that uses a Stored Procedure. No installed Unified CVP reports use Anonymous Blocks.

For the Call Detail report, you are required to enter Start and End dates and to select a Call Type. You may further constrain the data that appears by filtering for ANI, DNIS, Query Type.

Figure 9: Filter Tab for Stored Procedure Reports 1 of 3

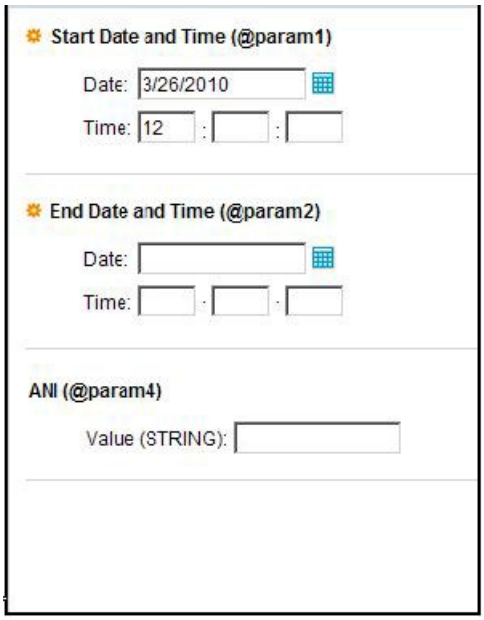
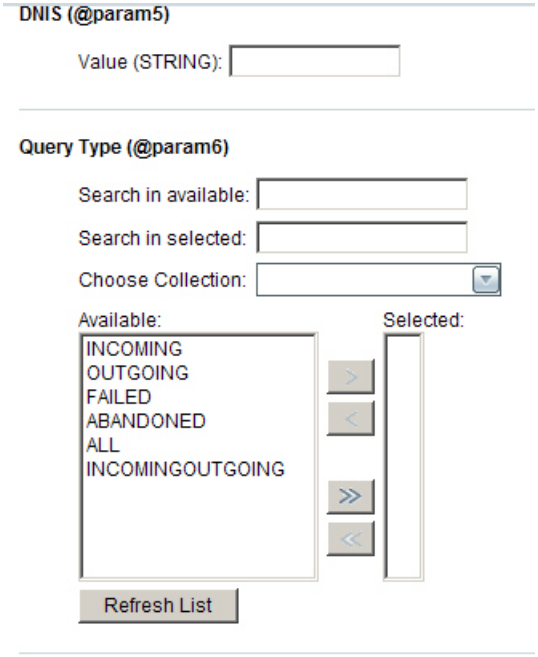


Figure 10: Filter Tab for Stored Procedure Reports 2 of 3



Select the parameter by which you want to filter the report and click **Run**.

Note Entering a Start Date Time that is subsequent to the End Date Time yields no validation and no error displays.

Report Viewer

A generated report displays in a Report Viewer page. This page is a container for the generated report. Its content varies, based on which view (data presentation) of a report is displayed--a grid, a chart, or a gauge. You can change the report view on this page.

Figure 11: Report Viewer Page

If the report view is a grid, you can review the field definitions for its template in the help topic for that template.

The menu bar across the top of the Report Viewer has these selections:

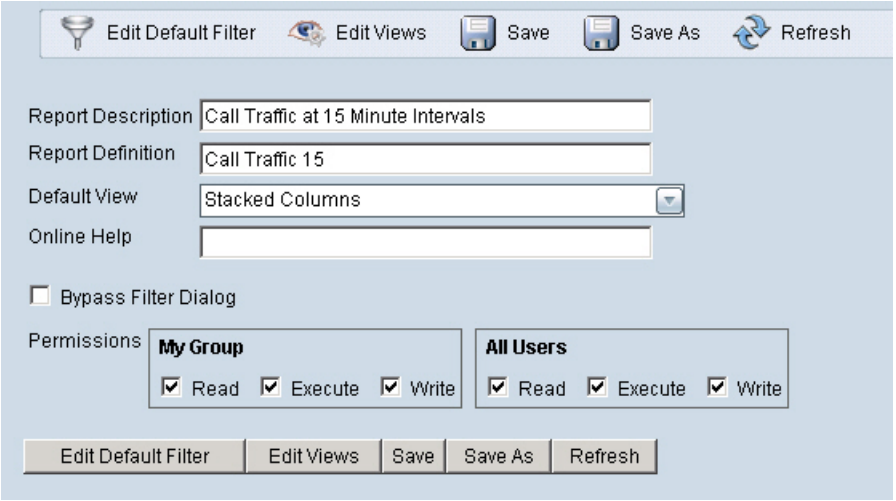
Menu item	Explanation
Save	Saves the report.
Save As	Opens the Save As dialog box.
Edit	Launches a page where you can edit the view. • For grid views, edit opens the Grid Editor. • For gauge views, edit opens the Gauge Editor. • For chart views, edit opens the Chart Editor.
Print	Prints the report.
Filter	Opens the filter page so that you can change the filter values (such as date/time and values) for the report.
SQL	Opens a window with a read-only display of the SQL query on which the report is based.
Refresh	Sends a request to the server to refresh the report dataset. **Note** If the report view is a grid, and if you have sorted the grid, Refresh resets the view and cancels the sort.

Menu item	Explanation
Pop Out	Opens the report in a new, separate browser display window. The popout has no Unified Intelligence Center edit or toolbar functions. Click **x** to close the popout.
Export (Grids only)	Launches the Export page, where you can export the report grid to Microsoft Excel.
View Dropdown	If more than one view is associated with this report template, use the drop-down to select the view you want to display. For example, if a report has both a grid and a chart view, select the view you want to refresh the report so that the data displays in that view.
Help	Opens a drop-down where you can select help for Unified Intelligence Center reporting or for the fields in the report template.

Edit Report

Right-click any report and select **Edit** to open the Report Editor page. Use this page to review or edit the information for a report.

Figure 12: Report Editor Page



The following table describes the fields of the Report Editor page.

Table 3: Fields on Report Editor Page

Field	Explanation
Report Description	This field displays a description for the report.
Report Definition	This field displays the Report Definition for the report.

Field	Explanation
Default View	From the drop-down, select the default view that is to display when users run the report. **Note** Once the report has generated, users can change the view. For example, if the default view is a grid, and a gauge has been developed for the report, users can change the generated report to show the gauge view.
Online Help	Shows the path for the online help topic for this report.
Bypass Filter Dialog	Check this box so that the report runs directly and users have no option to filter the report. **Note** • Even if the report has run directly, you can click the Filter icon in the Report Viewer to refilter and rerun the report. • As a best practice, do not check Bypass Filter until you have defined a Default Filter. Bypassing with no default filter set will run the report for all dates and times and for all data.
Permissions	Use these boxes to view or change user permissions for My Group (the default group) and for the All Users group.

The following table describes the function of the menu items of the Report Editor page.

Table 4: Report Editor Page Menu Items

Menu item	Function
Edit Default Filter	Opens the filter page for the report, where you can review the basic and advanced filters that are defined in the Report Definition.
Edit Views	Opens the Views Editor
Save	Saves the report.
Save As	Creates a cloned copy of the report.
Refresh	Updates the page to show changes another user has made.

Change and Move Report Fields

To see the fields that display in a report grid, right-click the report, and select **Edit Views** to open the Available Views for that report.

Select the grid view and then click **Edit**.

This opens a page that shows the name and description for the report and the font size for the report fields.

Panels on this page:

Available Fields. This panel shows all fields that are collected from the database, that are available to be used in the report, and that are checked as Allow to show if Invisible in the Report Definition Fields tab.

The fields in the Available Fields panel include the Current fields (fields that appear in the current grid view) as well as all other fields that are eligible to be used in the report view.

Current field order in the grid:

This panel shows all fields that are currently used in the grid view of the report, the order in which they appear, and the headers (if any) under which they are grouped.

Fields in this list appear as columns in the report. You can reorder and rename these fields. You can also remove them so they are no longer visible in the report.

The following table describes how to manage the fields in the Available Fields panel.

To:	Do This:
View the properties of a Current (visible) Field	Right-click a field and select **Properties** to open a page where you can set Field Properties.
Manage thresholds for a Current (visible) field	Right-click a field and select Thresholds. **Note** Thresholds are not implemented for Unified CVP templates.
Move a Current (visible) field so it is no longer visible in a report.	Right-click a field and select **Remove Selected**. Click **Yes** at the confirmation message. To add the field back, select it in the Available Fields panel and move it back to the Current field order in the grid panel. If you remove all fields from the Current panel, the generated report will show no data. **Note** To regain the default Current panel, you can re-import the report, or you can move fields from the Available panel back into the Current panel.

To:	Do This:
Move a field from the Available panel to the Current panel.	Select GridHeaders at the top of the Current panel. Select the field you want to move in the Available panel. Click > to move the field over. The field is placed at the bottom of the Current panel. Locate it; then use the up arrow to move it to the position in which you want it to display in the grid. **Note** The Available panel shows the default (database) name. The Current panel shows the display name. If you move an Available field that is already present in the Current panel (for example, if you move *subsystemname* from Available, and *Subsystem* is already part of Current), the *Subsystem* field in Current becomes highlighted. You cannot have the same field more than once in the Current panel.
Reorder Current Fields (Up and Down)	Use the up and down arrows to the right of the Current field order in the grid panel.

The following table describes the function of the menu items of the Report page.

Table 5: Report Page Menu Items

Menu item	Function
Grouping	Opens the Grouping dialog box
Save	Saves your changes to the grid view and closes the page
Save As	Opens a dialog box where you can enter a name for the changes you have made to the grid view
Save As	Creates a cloned copy of the report
Cancel	Cancels entries you have made and closes the page
Add Header	Opens the Create Super Headers dialog box
Remove Selected	Removes a field from the Current panel so that it no longer appears in the report. The field remains in the Available panel
Help	Opens the online help topic for this page

Export Report

A Report Designer with WRITE permission can export a custom report for troubleshooting or so that it can be archived or imported to another server in XML format.

What is included when a report is exported:

- Report Definition
- Value Lists
- Views--including all custom grids, charts, and gauges
- Values defined for the report in Report Editor (default view, online help, etc)
- Thresholds
- Permissions

What is not exported:

- The Report Filters
- The Collections

To export a report:

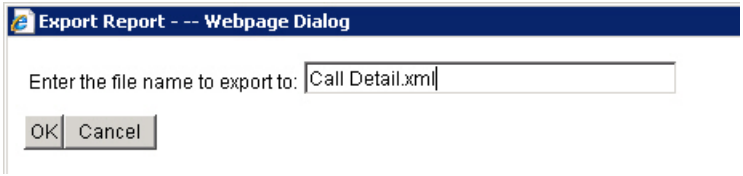
Procedure

Step 1 Right-click a report and select **Export** to open the Export Report dialog box, which gives you the option to rename the report. You can change the name but do not change the file extension.

Step 2 Click **OK** to open the Windows file download dialog box.

Step 3 Click **Save** and navigate to the location where you want to save the report.

Figure 13: Export Report



Cisco Unified Intelligence Center Reporting User Roles

There are seven User Roles, and a user can be assigned to one, any, or all of them. The roles are:

- Login User
- System Configuration Administrator

- Security Administrator
- Dashboard Designer
- Value List Collection Designer
- Report Designer
- Report Definition Designer

Depending on the size, staff, geographical distribution, and security practices of your call center, you might prefer to assign multiple user roles to a few people or to distribute user roles to many people.

Login User

By default, everyone who can sign in to Unified Intelligence Center is a Login User. Login Users have that role and only that role until the Security Administrator assigns additional roles or deactivates (removes) the Login User role.

A Security Administrator or System Application user can remove the login role from any user.

An active login user can:

- Log in to Unified Intelligence Center
- Open the Security drawer, access the User List, and edit his own User Information page; for example, to change his alias or phone number.

System Configuration Administrator

This user has all the rights of an active Login User and also:

- Has full access to the Data Sources drawer and its functions.
- Has full access to the Scheduler drawer and its functions.

Security Administrator

This user has all the rights of an active Login User and also has full access to the Security drawer and its functions.

Dashboard Designer

This user has all the rights of an active Login User and also has full access to the Dashboard drawer.

Value List Collection Designer

This user has all the rights of an active Login User and also:

- Has full access to the Value Lists drawer.
- Has View (Read) access to the Data Sources drawer.

Report Designer

This user has all the rights of an active Login User and also:

- Has full access to the Reports drawer.

- Has View (Read) access to the Data Sources and Value Lists drawers.
- Can access the Scheduler drawer to work with the user's own reports.

Report Definition Designer

This user has all the rights of an active Login User and also:

- Has full access to the Report Definition drawer.
- Has View (Read) access to the Data Sources and Value Lists drawers.

Other Cisco Unified Intelligence Center User Roles

In addition to the seven designated Unified Intelligence Center User Roles, the following individuals have access to Unified Intelligence Center as follows:

Superuser

This user role is defined in the Administration console. It is the only user role for Administration.

The initial and default superuser is the System Application User who is configured during installation of the Controller.

The initial superuser (the System Application User) can sign in to the Unified Intelligence Center Reporting console and has all User Roles and full permissions for all drawers in Unified Intelligence Center Reporting. Those credentials cannot be removed from the initial superuser.

Additional superusers who are added in the Administration Console can also sign in to Unified Intelligence Center Reporting and are considered to be IMS users. They have the limited Login User role only until the Unified Intelligence Center Reporting security administrator gives them additional roles and flags them as Active users.

System Application User

The user who is configured as the Application User during installation of the Controller node:

- Is the initial superuser in the Administration console.
- Can create additional superusers in the Administration application.
- Can sign in to Unified Intelligence Center and has full rights to all functions in Unified IC.
- Can create additional Security Administrator users in the Unified Intelligence Center reporting application.
- Cannot have any role taken away from him.
- Cannot take any role away from himself.

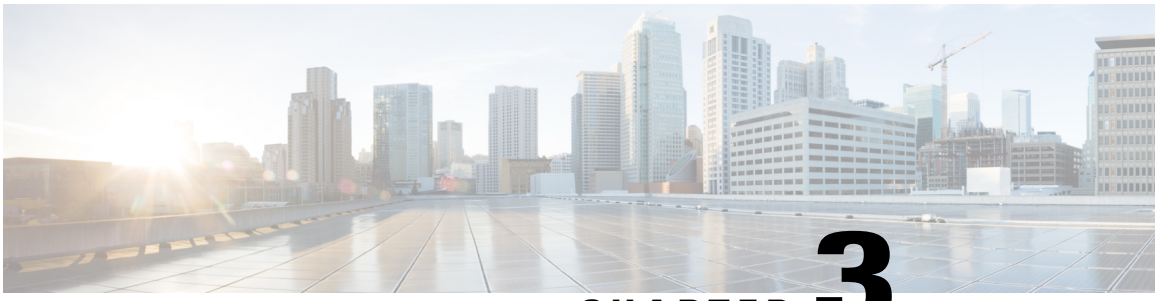
System Administration User

The user who is configured as the Administration User during installation of the Controller node:

- Can access the Command Line Interface on the Controller server.
- Can log in to the Cisco Unified Communications Solutions Tools (Unified Serviceability, Unified OS Administration, and Disaster Recovery System).

Unified CCE Supervisors

Unified CCE Reporting Supervisors are imported when Unified CCE User Integration is enabled. They become Unified Intelligence Center users and are automatically given these roles: Login User, Report Designer, and Dashboard Designer.



CHAPTER 3

Cisco Unified Customer Voice Portal Templates for Cisco Unified Intelligence Center

This chapter presents the Cisco-designed reports for Cisco Unified Customer Voice Portal (Unified CVP). These templates are available to you once you import them and set their data source.

To become familiar with developing additional custom reports, run one of the stock reports, perform a Save As operation, and modify the Save As report.

This chapter contains the following topics:

Application Summary Reports (15, Daily, and Weekly)

Unified CVP has three Application Summary Reports: Application Summary 15, Application Summary Daily, and Application Summary Weekly.

These reports are useful for Dominant Path Analysis. They display which elements of the voice applications are being executed and the number of times they are executed.

The three reports display the same data but aggregate the data for different time periods. See Summary / Aggregate Tables, on page 77

Fields in this report are populated from these tables:

- ApplicationSummary_15 Table, on page 79
- ApplicationSummary_Daily Table, on page 80
- ApplicationSummary_Weekly Table, on page 80

- ResultRef Table, on page 94
- ElementtypeRef Table, on page 90

Fields in this report:

- **Date and Time**

  The date and time when the period of data collection began. From the dbdatetime field of the appropriate Application Summary table - 15, Daily, or Weekly.

- **Application Name**

  The name of the voice application. From the AppName field of the appropriate Application Summary table - 15, Daily, or Weekly.

- **Source Application Name**

  The name of the source application that transferred the element. This field typically shows *null*. From The SourceAppName field of the appropriate Application Summary table - 15, Daily, or Weekly.

- **Element Name**

  The name of the element. From the ElementName field of of the appropriate Application Summary table - 15, Daily, or Weekly.

- **Element Type**

  The element type. From the ElementTypeID field of the appropriate Application Summary table - 15, Daily, or Weekly.

- **Exit State**

  The exit state of the element. From the ExitState field of the appropriate Application Summary table - 15, Daily, or Weekly.

- **Result**

  Indicates how an element ended. From result, where ResultID = ResultRef.resulttypeID.

- **Count**

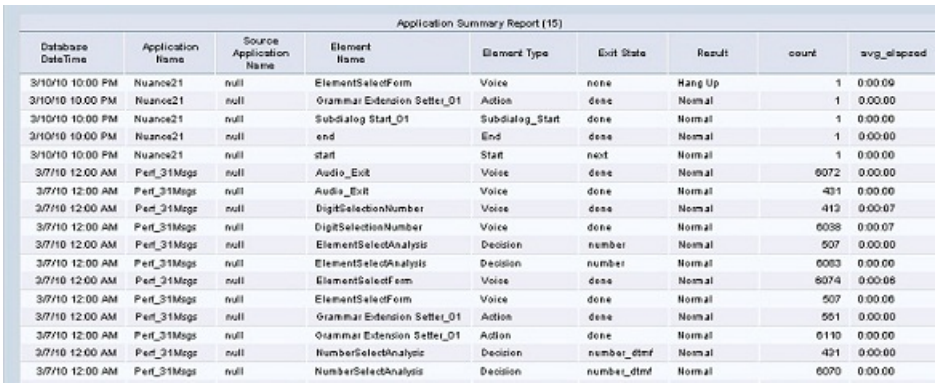
  The number of calls that executed a particular application. From the Count field of the appropriate Application Summary table - 15, Daily, or Weekly.

- **Average Elapsed Time**

  The average number of seconds a call was in a particular element. Calculated by subtracting the start time from the end time over all the elements within the grouping.

The following figure displays a sample Application Summary Report [15].

Figure 14: Application Summary Report [15]

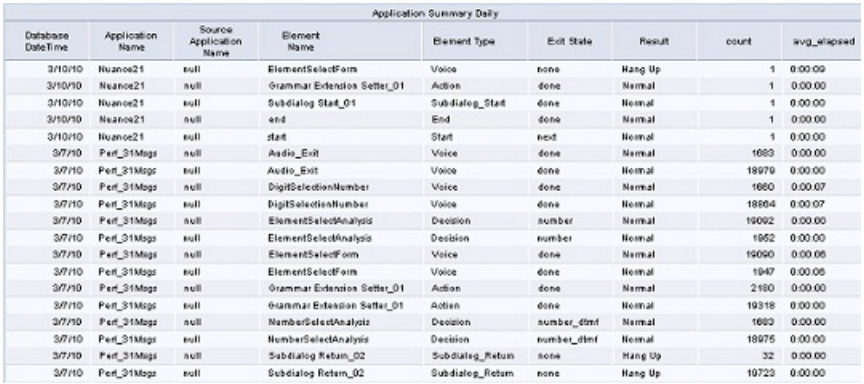


Application Summary Report (15)

Database DateTime	Application Name	Source Application Name	Element Name	Element Type	Exit State	Result	count	avg_elapsed
3/10/10 10:00 PM	Nuance21	null	ElementSelectForm	Voice	none	Hang Up	1	0:00:09
3/10/10 10:00 PM	Nuance21	null	Grammar Extension Setter_01	Action	done	Normal	1	0:00:00
3/10/10 10:00 PM	Nuance21	null	Subdialog Start_01	Subdialog_Start	done	Normal	1	0:00:00
3/10/10 10:00 PM	Nuance21	null	end	End	done	Normal	1	0:00:00
3/10/10 10:00 PM	Nuance21	null	start	Start	next	Normal	1	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	Audio_Exit	Voice	done	Normal	6072	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	Audio_Exit	Voice	done	Normal	431	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	DigitSelectionNumber	Voice	done	Normal	413	0:00:07
3/7/10 12:00 AM	Perf_31Msgs	null	DigitSelectionNumber	Voice	done	Normal	6038	0:00:07
3/7/10 12:00 AM	Perf_31Msgs	null	ElementSelectAnalysis	Decision	number	Normal	507	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	ElementSelectAnalysis	Decision	number	Normal	6083	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	ElementSelectForm	Voice	done	Normal	6074	0:00:06
3/7/10 12:00 AM	Perf_31Msgs	null	ElementSelectForm	Voice	done	Normal	507	0:00:06
3/7/10 12:00 AM	Perf_31Msgs	null	Grammar Extension Setter_01	Action	done	Normal	551	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	Grammar Extension Setter_01	Action	done	Normal	6110	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	NumberSelectAnalysis	Decision	number_dtmf	Normal	421	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	NumberSelectAnalysis	Decision	number_dtmf	Normal	6070	0:00:00

The following figure displays a sample Application Summary Daily report.

Figure 15: Application Summary Daily

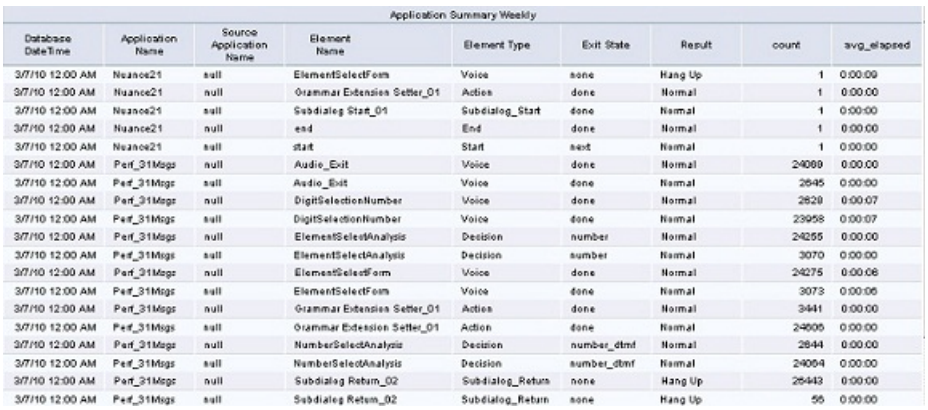


Application Summary Daily

Database DateTime	Application Name	Source Application Name	Element Name	Element Type	Exit State	Result	count	avg_elapsed
3/10/10	Nuance21	null	ElementSelectForm	Voice	none	Hang Up	1	0:00:09
3/10/10	Nuance21	null	Grammar Extension Setter_01	Action	done	Normal	1	0:00:00
3/10/10	Nuance21	null	Subdialog Start_01	Subdialog_Start	done	Normal	1	0:00:00
3/10/10	Nuance21	null	end	End	done	Normal	1	0:00:00
3/10/10	Nuance21	null	start	Start	next	Normal	1	0:00:00
3/7/10	Perf_31Msgs	null	Audio_Exit	Voice	done	Normal	1683	0:00:00
3/7/10	Perf_31Msgs	null	Audio_Exit	Voice	done	Normal	18979	0:00:00
3/7/10	Perf_31Msgs	null	DigitSelectionNumber	Voice	done	Normal	1660	0:00:07
3/7/10	Perf_31Msgs	null	DigitSelectionNumber	Voice	done	Normal	18804	0:00:07
3/7/10	Perf_31Msgs	null	ElementSelectAnalysis	Decision	number	Normal	19092	0:00:00
3/7/10	Perf_31Msgs	null	ElementSelectAnalysis	Decision	number	Normal	1952	0:00:00
3/7/10	Perf_31Msgs	null	ElementSelectForm	Voice	done	Normal	19090	0:00:06
3/7/10	Perf_31Msgs	null	ElementSelectForm	Voice	done	Normal	1947	0:00:06
3/7/10	Perf_31Msgs	null	Grammar Extension Setter_01	Action	done	Normal	2160	0:00:00
3/7/10	Perf_31Msgs	null	Grammar Extension Setter_01	Action	done	Normal	19318	0:00:00
3/7/10	Perf_31Msgs	null	NumberSelectAnalysis	Decision	number_dtmf	Normal	1683	0:00:00
3/7/10	Perf_31Msgs	null	NumberSelectAnalysis	Decision	number_dtmf	Normal	18975	0:00:00
3/7/10	Perf_31Msgs	null	Subdialog Return_02	Subdialog_Return	none	Hang Up	32	0:00:00
3/7/10	Perf_31Msgs	null	Subdialog Return_02	Subdialog_Return	none	Hang Up	10723	0:00:00

The following figure displays a sample Application Summary Weekly report.

Figure 16: Application Summary Weekly



Application Summary Weekly

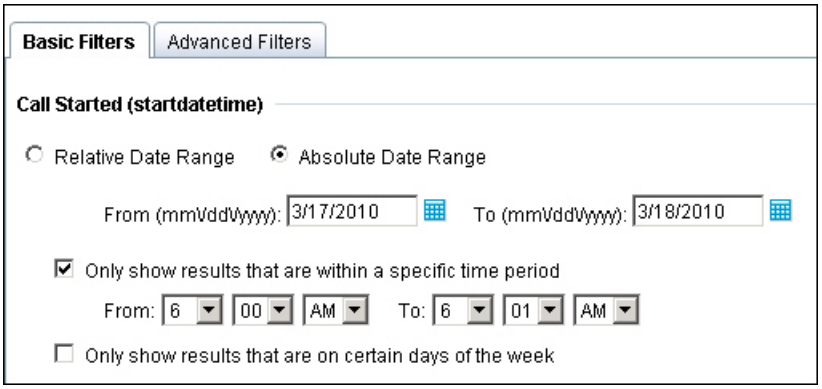
Database DateTime	Application Name	Source Application Name	Element Name	Element Type	Exit State	Result	count	avg_elapsed
3/7/10 12:00 AM	Nuance21	null	ElementSelectForm	Voice	none	Hang Up	1	0:00:09
3/7/10 12:00 AM	Nuance21	null	Grammar Extension Setter_01	Action	done	Normal	1	0:00:00
3/7/10 12:00 AM	Nuance21	null	Subdialog Start_01	Subdialog_Start	done	Normal	1	0:00:00
3/7/10 12:00 AM	Nuance21	null	end	End	done	Normal	1	0:00:00
3/7/10 12:00 AM	Nuance21	null	start	Start	next	Normal	1	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	Audio_Exit	Voice	done	Normal	24089	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	Audio_Exit	Voice	done	Normal	2645	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	DigitSelectionNumber	Voice	done	Normal	2629	0:00:07
3/7/10 12:00 AM	Perf_31Msgs	null	DigitSelectionNumber	Voice	done	Normal	23958	0:00:07
3/7/10 12:00 AM	Perf_31Msgs	null	ElementSelectAnalysis	Decision	number	Normal	24255	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	ElementSelectAnalysis	Decision	number	Normal	3070	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	ElementSelectForm	Voice	done	Normal	24275	0:00:06
3/7/10 12:00 AM	Perf_31Msgs	null	ElementSelectForm	Voice	done	Normal	3073	0:00:06
3/7/10 12:00 AM	Perf_31Msgs	null	Grammar Extension Setter_01	Action	done	Normal	3441	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	Grammar Extension Setter_01	Action	done	Normal	24605	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	NumberSelectAnalysis	Decision	number_dtmf	Normal	2644	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	NumberSelectAnalysis	Decision	number_dtmf	Normal	24064	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	Subdialog Return_02	Subdialog_Return	none	Hang Up	25443	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	Subdialog Return_02	Subdialog_Return	none	Hang Up	55	0:00:00

Call Report

The Call report shows the first 500 calls, starting at the date/time you select in the Basic Filter.

For this report, you can take advantage of the Absolute Date feature of the Basic Filter and filter to a narrow range, for example, to a one-minute time interval:

Figure 17: Basic Filters with Absolute Dates



Fields in this report are populated from these tables:

- Call Table
- CallTypeRef Table
- SubSystemTypeRef Table

Fields in this report:

- **Call Guid**

  The Global Unique ID (GUID) of a call. From Call.CallGuid.

- **ANI**

  The Automatic Number Identification (ANI) of the caller sent by the telephony provider.

  ANI does not need to be configured, and can therefore be a null field. From Call.ANI.

- **DNIS**

  The Dialed Number Identification Service (DNIS) sent by the telephony provider. From Call.DNIS.

- **UID**

  The external User Identifier (UID) of the originating caller, sent by telephony provider. From Call.UID.

- **UUI**

  The User-To-User Information (UUI) of the originating caller, sent by telephony provider. From Call.UUI.

- **IIDigits**

  The ANI II Digits of the originating caller, sent by the telephony provider. From Call.IIDigits.

- **Sub Subsystem Type**

The type of Unified CVP Service used for the call, such as SIP, IVR,VXML, where Call.SubSystemTypeID = SubsystemTypeRef.SubsystemTypeRefID.

See SubSystemTypeRef Table for the list of sybsystem types.

- **Call Type**

  The type of call, where Call.CallTypeID = CallTypeRef.CallTypeID. See CallTypeRef Table for the list of call types.

- **Call Started**

  The time when the call started. From Call.StartDateTime.

- **Call Ended**

  The time when the call ended. From Call.EndDateTime.

- **Time Persisted to Database**

  This is the timestamp when data is written to the database. If the reporting server is in Boston, and the Callserver is in San Diego, then calls placed in San Diego at 1 PM have a Coordinated Universal Time (UTC) startdatetime (+ 7 hours), and the dbdatetime for this call is 4 PM (Boston time).

- **Time Zone Offset from UTC (Minutes)**

  The offset in minutes of the local timezone from the UTC timezone. From Call.LocalTimeZoneOffset.

  If the reporting server is in Boston, and the Callserver is in San Diego, then calls placed in San Diego at 1 PM have a UTC startdatetime (+ 7 hours), and the dbdatetime for this call is 4 PM (Boston time). The timezone offset is 240 minutes.

- **Applications Visited**

  The number of applications visited during the life of the call. From Call.NumAppVisited.

- **Errors**

  The number of errors that occurred during the call. From Call.NumError.

- **On Hold**

  The number of times the call was on hold due to unavailable ports. From Call.NumOnHold.

- **Opted Out**

  This will always show 0. From Call.NumOptOut, deprecated field.

- **Timed Out**

  The number of times the call timed out because it exceeded a processing time. From Call.NumTimedOut.

- **Transferred**

  The number of times the call was transferred out to an agent or to a Voice Response Unit (VRU) leg. The VRU leg is the leg that talks to VoiceXML Gateways.

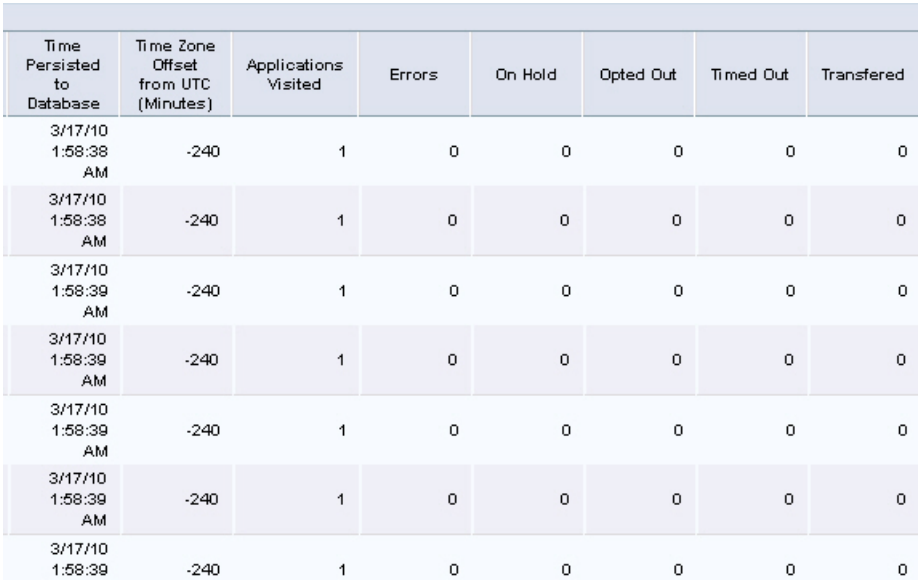
The following figures display a Sample Call report.

Figure 18: Call Report 1 of 2

First 500 Calls From Starting Time

CallGuid	ANI	DNIS	UID	UUI	IIDigits	Sub System Type	Call Type	Call Started	Call Ended
291FE40830C111DF008F0014694 4B76A	194194194	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:52 AM
2840652930C111DF009000146944B76A	194194194	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:52 AM
2952C48830C111DF009C0014694 4B8CC	193193193	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:52 AM
2871E57330C111DF009D0014694 4B8CC	193193193	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:52 AM
2876914030C111DF009100146944B76A	194194194	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:52 AM
2894B12230C111DF009E0014694 4B8CC	193193193	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:53 AM
2886DEA430C111DF00 9F0014694 4B8CC	193193193	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:01	3/17/10 6:00:53

Figure 19: Call Report 2 of 2



Time Persisted to Database	Time Zone Offset from UTC (Minutes)	Applications Visited	Errors	On Hold	Opted Out	Timed Out	Transfered
3/17/10 1:58:38 AM	-240	1	0	0	0	0	0
3/17/10 1:58:38 AM	-240	1	0	0	0	0	0
3/17/10 1:58:39 AM	-240	1	0	0	0	0	0
3/17/10 1:58:39 AM	-240	1	0	0	0	0	0
3/17/10 1:58:39 AM	-240	1	0	0	0	0	0
3/17/10 1:58:39 AM	-240	1	0	0	0	0	0
3/17/10 1:58:39	-240	1	0	0	0	0	0

Call Detail Report

This report displays robust detail for calls according to the filters you set. This report is based on a Stored Procedure and uses the *Filter Page for Stored Procedure and Anonymous Block Reports*.

Note that this is a wide report. You might need to set the printer to Landscape mode, use legal-sized paper, or Export to Excel.

Fields in this report include data from these tables:

- Call Table
- CallEvent Table

- CallICMInfo Table
- VXMLElement Table
- VXMLSession Table
- CallTypeRef Table
- CauseRef Table
- EventTypeRef Table
- ResultRef Table
- SubSystemTypeRef Table
- TransferTypeRef Table

Fields in this report are:

- **Call Start Date**

  The time when the call started. From Call.StartDateTime.

- **End Date and Time**

  The time when the call ended. From Call.EndDateTime.

- **Local Timezone Offset**

  The offset in minutes of the local timezone from UTC timezone. From Call.LocalTimeZoneOffset.

  If the reporting server is in Boston, and the Callserver is in San Diego, then calls placed in San Diego at 1 PM will have a UTC startdatetime (+ 7 hours), and the dbdatetime for this call will be 4 PM (Boston time). The timezone offset is 240 minutes.

- **Call Guid**

  The Global Unique ID (GUID) of a call. From Call.CallGuid.

- **Call Type**

  The type of call, where Call.CallTypeID = CallTypeRef.CallTypeID.

- **ANI**

  The Automatic Number Identification (ANI) of the caller sent by the telephony provider. From Call.CallGuid.

  ANI does not need to be configured, and can therefore be a null field. From Call.ANI.

- **DNIS**

  The Dialed Number Identification Service (DNIS) sent by the telephony provider. From Call.DNIS.

- **IIDigits**

  The ANI II Digits of the originating caller, sent by the telephony provider. From Call.IIDigits.

- **UID**

  The external User Identifier (UID) of the originating caller, sent by telephony provider. From Call.UID.

- **UUI**

  The User-To-User Information (UUI) of the originating caller, sent by telephony provider. From Call.UUI.

- **Number of Applications Visited**

  The number of applications visited during the life of the call. From Call.NumAppVisited.

- **Number of Interactions**

  The number of applications visited during the life of the call. From VXMLElement.NumberofInteractions.

- **Number of Errors**

  The number of errors that occurred during the call. From Call.NumError.

- **Number of Times On Hold**

  The number of times the call was on hold due to unavailable ports. From Call.NumOnHold.

- **Number of Timeouts**

  The number of times the call timed out because it exceeded a processing time. From Call.NumTimedOut.

- **Total Number of Transfers**

  The number of times the call was transferred out to an agent or to a VRU leg. (*VRU* leg is the leg that talks to VoiceXML Gateways.) From Call.TotalTransfer.

- **Event Date and Time**

  The date and time of the event. From CallEvent.EventDateTime.

- **Event Type**

  The name of the event type, from Callback.eventtypeID, where Callback.eventtypeID = EventTypeRef.eventtypeid.

- **Subsystem**

  The Subsystem Name. From SubsystemTypeRef.Subsystem.

- **Subsystem Type ID**

  The ID of the Unified CVP Service used for the call, such as "2" for VXML. From SubsystemTypeRef.SubsystemTypeRefID.

- **CallLegId**

  This is an ID assigned by the service. From CallEvent.CallLegID.

- **Cause**

  The reason that the call event was generated. From CallEvent.CauseID.

- **Result**

  Indicates how an element ended. From result, where ResultID = ResultRef.resulttypeID.

- **Media Filename**

  This field is always *null* in this release. From CallEvent.MediaFileName.

- **Transfer Label**

  The destination to which Unified CVP transfers the call. From CallEvent.TransferLabel.

- **Transfer Type**

  The unique id of the transfer type. From CallEvent.TransferTypeID.

- **Messagebus**

The name of the Call Server (its message adapter name) with which the Call Event is associated. From CallEvent.MessageBusName.

- **RouterCallKey**

  The Unified ICM router call key. From CallICMInfo.RouterCallKey.

- **RouterCallKeyDay**

  The Unified ICM router call key day. From CallICMInfo.RouterCallKeyDay.

- **RouterCallKeySeqenceNumber**

  The Unified ICM router call key sequence number. From CallICMInfo.RouterCallKeySequenceNumber.

- **Application Name**

  The name of the VXML application. From VXMLSession.AppName.

- **Enter Date and Time**

  The date and time when the element was entered. From VXMLElement.EnterDateTime.

- **Exit Date and Time**

  The date and time when the element was exited. From VXML.ExitDateTime.

- **Duration**

  The length of the session. From VXMLSession.Duration.

- **Element Name**

  The name of the element. From VXMLElement.ElementName.

- **Element Type**

  The type of element. From VXMLElement.ElementTypeID.

- **Exit State**

  The exit state of the element. From VXMLElement.ExitState.

- **Session ID**

  The unique id of a VXML application session. From VXMLSession.SessionID.

- **Session Name**

  The name of the session assigned by VXML Server. From VXMLSession.SessionName.

- **Source Application Name**

  The name of the application that transferred to this one. From VXMLSession.SourceAppName.

- **VXML Cause**

  The reason that the application visit ended. From VXMLSession.CauseID.

- **VXML End Date and Time**

  The end date and time of the session. From VXMLSession.EndDateTime.

- **VXML Event Type**

  The mechanism used to end the application visit. From VXMLSession.EventTypeID.

- **VXML Start Date and Time**

Date and time when session began. From VXMLSession.StartDateTime.

Sample Call Detail Report for filters Query Type = All.

Figure 20: Call Detail Report 1 of 6

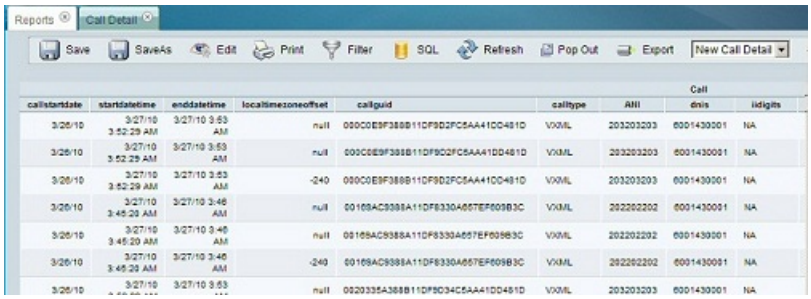


Figure 21: Call Detail Report 2 of 6

uid	uui	numappvisited	numberofinteractions	numerror	numonhold	numtimeout	totaltransfer	eventdatetime	eventtype
null	NA	1	null	0	0	0	0	3/27/10 3:52:29 AM	New Call
null	NA	1	null	0	0	0	0	3/27/10 3:53:32 AM	Hang Up
null	NA	1	null	0	0	0	0	3/27/10 3:52:29 AM	null
null	NA	1	null	0	0	0	0	3/27/10 3:46:20 AM	New Call
null	NA	1	null	0	0	0	0	3/27/10 3:46:23 AM	Hang Up
null	NA	1	null	0	0	0	0	3/27/10 3:46:20 AM	null
null	NA	1	null	0	0	0	0	3/27/10 3:45:59 AM	New Call

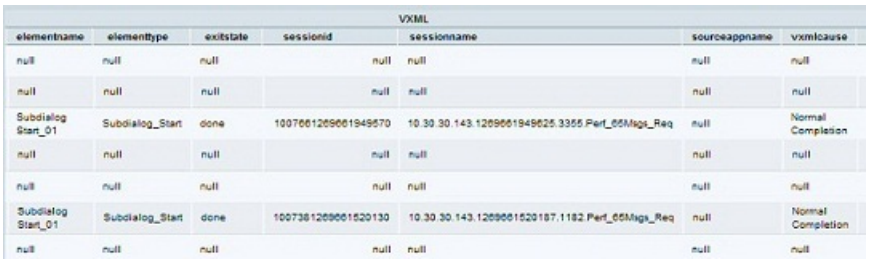
Figure 22: Call Detail Report 3 of 6

Call Event							
subsystemname	subsystemtypeid	callegid	cause	result	mediafilename	transferlabel	transfertype
ovpcald143.SYS_VXML7	2	000C0E9F388B11DF9D2FC5AA41DD481D	None	null	NA	null	null
ovpcald143.SYS_VXML7	2	000C0E9F388B11DF9D2FC5AA41DD481D	Normal Completion	null	NA	null	null
ovpcald143.SYS_VXML7	2	null	null	Normal	null	null	null
ovpcald143.SYS_VXML7	2	00169AC9388A11DF8330A657EF609B3C	None	null	NA	null	null
ovpcald143.SYS_VXML7	2	00169AC9388A11DF8330A657EF609B3C	Normal Completion	null	NA	null	null
ovpcald143.SYS_VXML7	2	null	null	Normal	null	null	null
ovpcald143.SYS_VXML7	2	0020335A388B11DF9D34C5AA41DD481D	None	null	NA	null	null

Figure 23: Call Detail Report 4 of 6

	ICM						
messagebusname	routercallkey	routercallkeyday	routercallkeysequencenumber	appname	enterdatetime	exitdatetime	duration
ovpcald143.MsgBus009	null	null	null	null	null	null	null
ovpcald143.MsgBus009	null	null	null	null	null	null	null
null	null	null	null	Perf_65Msgs_Req	3/27/10 3:52:29 AM	3/27/10 3:52:29 AM	63
ovpcald143.MsgBus009	null	null	null	null	null	null	null
ovpcald143.MsgBus009	null	null	null	null	null	null	null
null	null	null	null	Perf_65Msgs_Req	3/27/10 3:46:20 AM	3/27/10 3:46:20 AM	63
ovpcald143.MsgBus009	null	null	null	null	null	null	null

Figure 24: Call Detail Report 5 of 6



VXML

elementname	elementtype	exitstate	sessionid	sessionname	sourceappname	vxmlcause
null	null	null	null	null	null	null
null	null	null	null	null	null	null
Subdialog Start_01	Subdialog_Start	done	10076612696619495 70	10.30.30.143.1269661949625.3355.Perf_65Msgs_Req	null	Normal Completion
null	null	null	null	null	null	null
null	null	null	null	null	null	null
Subdialog Start_01	Subdialog_Start	done	10073612696615201 30	10.30.30.143.1269661520187.1182.Perf_65Msgs_Req	null	Normal Completion
null	null	null	null	null	null	null

Figure 25: Call Detail Report 6 of 6

vxmlcause	vxmlenddatetime	vxmleventtype	vxmlstartdatetime
null	null	null	null
null	null	null	null
Normal Completion	3/27/10 3:53:32 AM	Hang Up	3/27/10 3:52:29 AM
null	null	null	null
null	null	null	null
Normal Completion	3/27/10 3:46:23 AM	Hang Up	3/27/10 3:45:20 AM
null	null	null	null

Call Traffic Reports (15, Daily, and Weekly)

Unified CVP has three Call Traffic Reports: Call Traffic 15, Call Traffic Daily, and Call Traffic Weekly.

These reports indicate the CallServer/VXML Server load during the course of the day. Call center administrators can monitor peak call volume times and monitor load levels on various call servers and VXML servers.

The three reports display the same data and aggregate the data for different time periods. See Summary / Aggregate Tables, on page 77

Fields in this report are populated from these tables:

- Call_15 Table, on page 82
- Call_Daily Table, on page 83
- Call_Weekly Table, on page 84
- CallTypeRef Table, on page 87
- SubSystemTypeRef Table, on page 94

Fields in this report:

- **Date and Time**

  The date and time of the call, from the start of the increment. From the datetime field of the appropriate Call Traffic table (15, Daily, or Weekly).

- **SubSystem Type**

The type of Unified CVP Service used for the call, such as SIP, IVR,VXML, where the Call Summary table SubSystemTypeID = SubsystemTypeRef.SubsystemTypeRefID.

See SubSystemTypeRef Table, on page 94 for the list of subsystem types.

- **Call Type**

  The type of call, where the Call Summary Table CallTypeID = CallTypeRef.CallTypeID. See CallTypeRef Table, on page 87 for the list of call types.

- **Number of Calls**

  Total number of calls in this period. From the NumCalls field of the appropriate Call Traffic table (15, Daily, or Weekly).

- **Average Call Length**

  The average call length. From the AvgCallLength field of the appropriate Call Traffic table (15, Daily, or Weekly).

- **Applications Visited**

  The total number of applications visited in this period. From the TotalAppVisited field of the appropriate Call Traffic table (15, Daily, or Weekly).

- **Errors**

  The total number of errors in this period. From the TotalError field of the appropriate Call Traffic table (15, Daily, or Weekly).

- **On Hold**

  The total number of times that calls were placed on hold in this period. From the TotalOnHold field of the appropriate Call Traffic table (15, Daily, or Weekly).

- **Opted Out**

  This field is deprecated. From the TotalOptOut field of the appropriate Call Traffic table (15, Daily, or Weekly).

- **Timed Out**

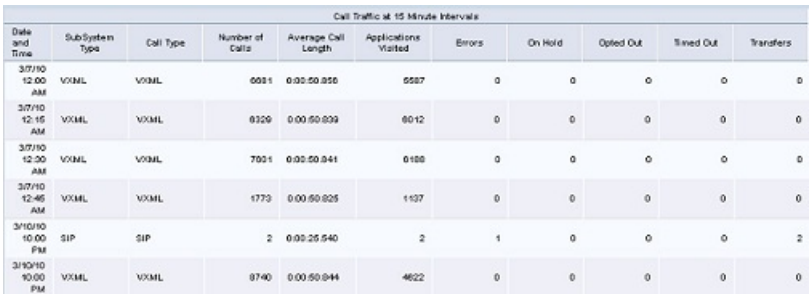
  The total number of calls that timed out in this period. From the TotalTimeOut field of the appropriate Call Traffic table (15, Daily, or Weekly).

- **Transfers**

  The total number of calls were transferred in this period. From the TotalTransfer field of the appropriate Call Traffic table (15, Daily, or Weekly).

The following figure shows a Call traffic at 15 Minute Intervals report.

Figure 26: Call Traffic at 15 Minute Intervals Report



Call Traffic at 15 Minute Intervals

Date and Time	SubSystem Type	Call Type	Number of Calls	Average Call Length	Applications Visited	Errors	On Hold	Opted Out	Timed Out	Transfers
3/7/10 12:00 AM	VXML	VXML	6631	0:00:50.856	6587	0	0	0	0	0
3/7/10 12:15 AM	VXML	VXML	6329	0:00:50.839	6012	0	0	0	0	0
3/7/10 12:30 AM	VXML	VXML	7031	0:00:50.841	6188	0	0	0	0	0
3/7/10 12:45 AM	VXML	VXML	1773	0:00:50.825	1137	0	0	0	0	0
3/10/10 10:00 PM	SIP	SIP	2	0:00:25.540	2	1	0	0	0	2
3/10/10 10:00 PM	VXML	VXML	8740	0:00:50.844	4622	0	0	0	0	0

The following figure shows a Call Traffic Daily report.

Figure 27: Call Traffic Daily Report

Call Traffic Daily

Date	SubSystem Type	Call Type	Number of Calls	Average Call Length	Applications Visited	Errors	On Hold	Opted Out	Timed Out	Transfers
3/7/10	VXML	VXML	22384	0:00:50.843	18924	0	0	0	0	0
3/10/10	VXML	VXML	8740	0:00:50.844	4622	0	0	0	0	0
3/10/10	SIP	SIP	2	0:00:25.540	2	1	0	0	0	2
			31126		23548	1	0	0	0	2

The following figure shows a Call Traffic at Weekly Intervals report.

Figure 28: Call Traffic at Weekly Intervals Report

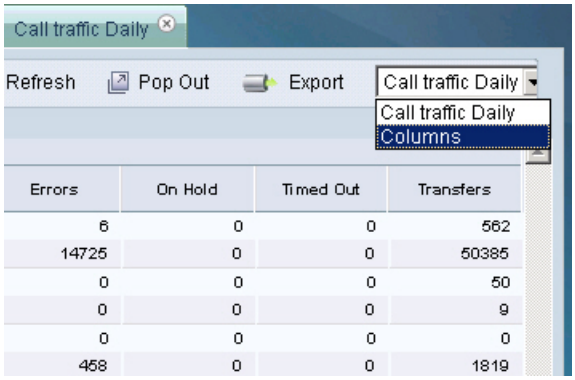
Call Traffic at Weekly Intervals

Week Beginning Date	SubSystem Type	Call Type	Number of Calls	Average Call Length	Applications Visited	Errors	On Hold	Opted Out	Timed Out	Transfers
3/7/10	VXML	VXML	31124	0:00:50.843	23546	0	0	0	0	0
3/7/10	SIP	SIP	2	0:00:25.540	2	1	0	0	0	2
			31126		23548	1	0	0	0	2

Call Traffic Charts

All three Call Traffic reports have chart views. To display the report data in the chart view, click the View drop-down and select Columns.

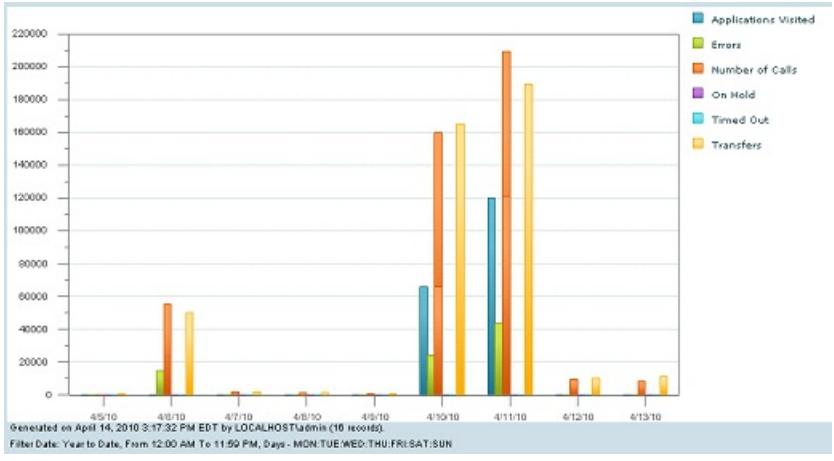
Figure 29: Selecting a Chart View



Call Traffic charts are columns that show the total Applications Visited, Errors, Number of Calls, On Hold calls, Timed Out calls, and Transfers for the 15 minute, Daily, or Weekly time interval.

This figure shows a Call Traffic Daily chart from Call Traffic Daily data:

Figure 30: Call Traffic Daily Chart



Current and Historical Callback Reports

Unified CVP has two callback reports: Pending Callbacks and Historical Callbacks.

They both pull data from the CallBack Table.

Note The Callbacks reports must use the `callback` database as the data source or an error occurs when you attempt to access the report: *"Import could not be completed: Query validation failed against the selected data source."*

The Pending Callbacks report takes data from calls that have occurred in the past fifteen minutes. The Historical Callbacks report shows calls that occurred prior to fifteen minutes ago.

This report is populated by data from the CallBack Table.

Fields in this report are:

- **Date and Time**

  The date and time for the callback call, from Callback.eventdatetime, from Callback.dbdatetime,'15'

- **Event Type**

  The name of the event type, from Callback.eventtypeID, where Callback.eventtypeID = EventTypeRef.eventtypeid. See the EventTypeRef Table for the list of values.

- **Cause**

  The cause of the event, from Callback.CauseID, where Callback.CauseID = CauseRef.causeid. See the CauseRef Table for the list of values.

- **Gateway**

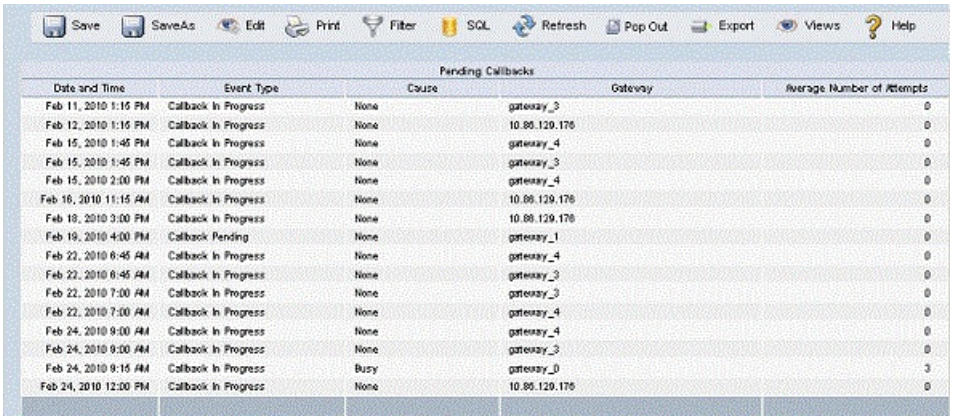
  The identifier for the gateway. This can be an IP address or a string identifier. From Callback.Gateway.

- **Average Number of Attempts**

  The number of attempts that were made to call back. From Callback.NbrAttempts.

The following figure shows a sample of a Pending Callbacks report.

Figure 31: Pending Callbacks Report



The following figure shows a sample of a Historical Callbacks report.

Figure 32: Historical Callbacks Report

Historical Callbacks

Datetime	Event Type	Cause	Gateway	Average Number of Attempt
Feb 8, 2010 6:45 PM	Callback Complete	Caller Canceled	10.86.129.176	0
Feb 8, 2010 6:00 PM	Callback Complete	Connected	10.86.129.176	0
Feb 8, 2010 6:00 PM	Callback Complete	Caller Canceled	10.86.129.176	0
Feb 8, 2010 6:15 PM	Callback Complete	Connected	10.86.129.176	0
Feb 9, 2010 11:30 AM	Callback Complete	No Response	10.86.129.176	0
Feb 9, 2010 12:30 PM	Callback Complete	Connected	10.86.129.176	0
Feb 9, 2010 1:00 PM	Callback Complete	Caller Canceled	10.86.129.176	0
Feb 9, 2010 1:45 PM	Callback Complete	Caller Canceled	10.86.129.176	0
Feb 9, 2010 1:45 PM	Callback Complete	No Response	10.86.129.176	0
Feb 9, 2010 2:30 PM	Callback Complete	Connected	10.86.129.176	0
Feb 10, 2010 12:00 PM	Callback Complete	Caller Canceled	10.86.129.176	0
Feb 10, 2010 1:45 PM	Callback Complete	Connected	10.86.129.176	0
Feb 11, 2010 9:00 AM	Callback Complete	Caller Canceled	10.86.129.176	0
Feb 11, 2010 11:45 AM	Callback Complete	Connected	10.86.129.176	0
Feb 11, 2010 12:30 PM	Callback Complete	Caller Canceled	10.86.129.176	0
Feb 11, 2010 12:45 PM	Callback Complete	Connected	10.86.129.176	0
Feb 11, 2010 1:15 PM	Callback Complete	Connected	gateway_1	0

Trunk Group Utilization Report

This report shows a summary of trunk group utilization. It is a grid (a tablular display).

This report is populated by data from the Usage Table, the Resource Table, and the Device Table.

Fields in this report are:

- **Date and Time**

  The date for the event. From Usage.eventdatetime.

- **Device**

  The IP address of the device. From Usage.DeviceID, where Usage.DeviceID = Device.DeviceID.

- **Resource**

  The unique identifier of the resource being measured. From Usage.ResourceID, where Usage.ResourceID = Resource.ResourceID. See the Resource Table for the resource descriptors.

- **Average Resource Used**

  The average resource used, from Usage.ResourceUsed.

- **Max Threshold Reached**

  Was the maximum threshold reached (Yes | No), from Usage.ThresholdReached.

- **Max Resource Used**

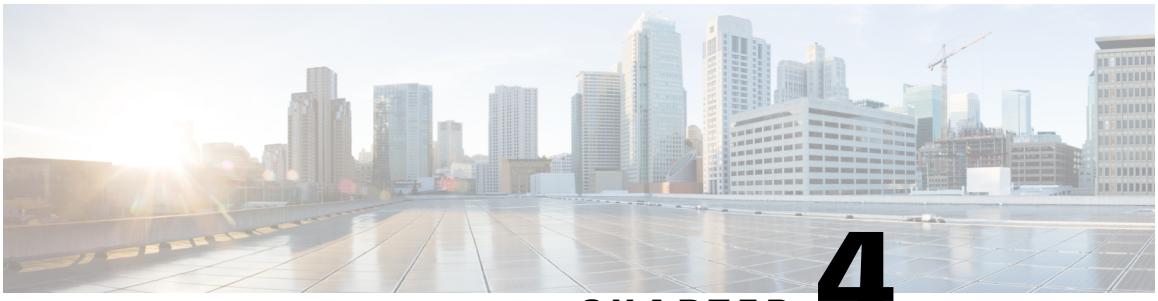
  The maximum resource used, from Usage.ResourceMax.

The following figure shows a sample of a Trunk Group Utilization report.

Figure 33: Trunk Group Utilization Report

Trunk Group Utilization

Date and Time	Device	Resource	Average Resource Used	Max Threshold Reached	Max Resource Used
Feb 3, 2010 2:45 PM	10.86.129.44	CPU	1	0	100
Feb 3, 2010 2:45 PM	10.86.129.44	DSO	0	0	59
Feb 3, 2010 2:45 PM	10.86.129.44	DSP	0	0	384
Feb 3, 2010 2:45 PM	10.86.129.44	MEM	17	0	100
Feb 3, 2010 2:45 PM	10.86.129.44	SYSTEM	0	0	0
Feb 3, 2010 3:00 PM	10.86.129.44	CPU	1	0	100
Feb 3, 2010 3:00 PM	10.86.129.44	DSO	0	0	59
Feb 3, 2010 3:00 PM	10.86.129.44	DSP	0	0	384
Feb 3, 2010 3:00 PM	10.86.129.44	MEM	17	0	100
Feb 3, 2010 3:00 PM	10.86.129.44	SYSTEM	0	0	0
Feb 3, 2010 3:15 PM	10.86.129.44	CPU	1	0	100
Feb 3, 2010 3:15 PM	10.86.129.44	DSO	0	0	59
Feb 3, 2010 3:15 PM	10.86.129.44	DSP	0	0	384
Feb 3, 2010 3:15 PM	10.86.129.44	MEM	17	0	100
Feb 3, 2010 3:15 PM	10.86.129.44	SYSTEM	0	0	0
Feb 3, 2010 3:30 PM	10.86.129.44	CPU	1	0	100
Feb 3, 2010 3:30 PM	10.86.129.44	DSO	0	0	59



CHAPTER 4

Database Schema

About Database Schema

The Cisco Unified Customer Voice Portal (United CVP) reporting server hosts an IBM Informix Dynamic Server (IDS) database, which stores reporting data in a defined database schema. Customers who choose to deploy Cisco Unified Intelligence Center (Unified Intelligence Center) as their reporting platform must add the Informix database as a data source in Unified Intelligence Center.

The schema is fully published so that customers can develop custom reports. Customers may not, however, extend the schema for their own purposes.

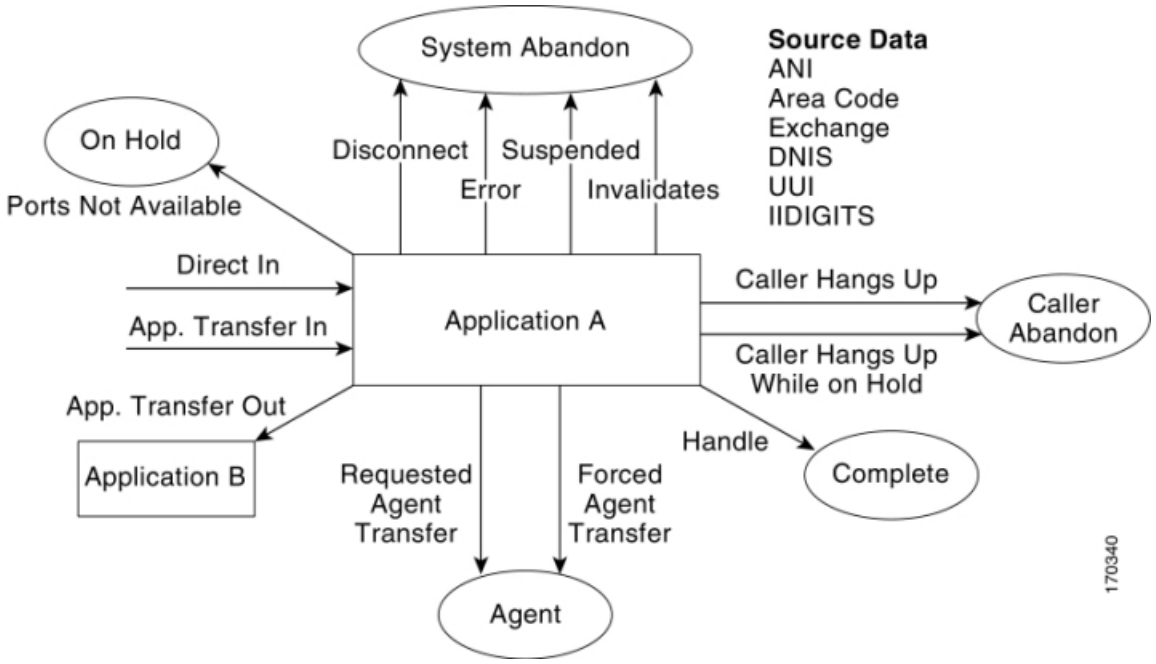
The schema provides Unified CVP customers with the ability to:

- Establish database connectivity with Unified Intelligence Center and to import and run the Unified CVP templates with Unified Intelligence Center.
- Establish database connectivity with other commercial off-the-shelf reporting and analytics engines and then build custom reports against the Unified CVP database schema.

Note Your support provider cannot assist you with custom reports or with commercial (non-Cisco) reporting products.

The following diagram indicates a common set of incoming and outgoing entry and exit states for a call to a self-service application.

Figure 34: Call Flow



Note When basic video is transferred to an audio-only agent, the call remains classified as basic video accepted.

Data Model Diagram

The following entity-relationship diagrams depict the Unified CVP database schema.

Figure 35: Call Tables

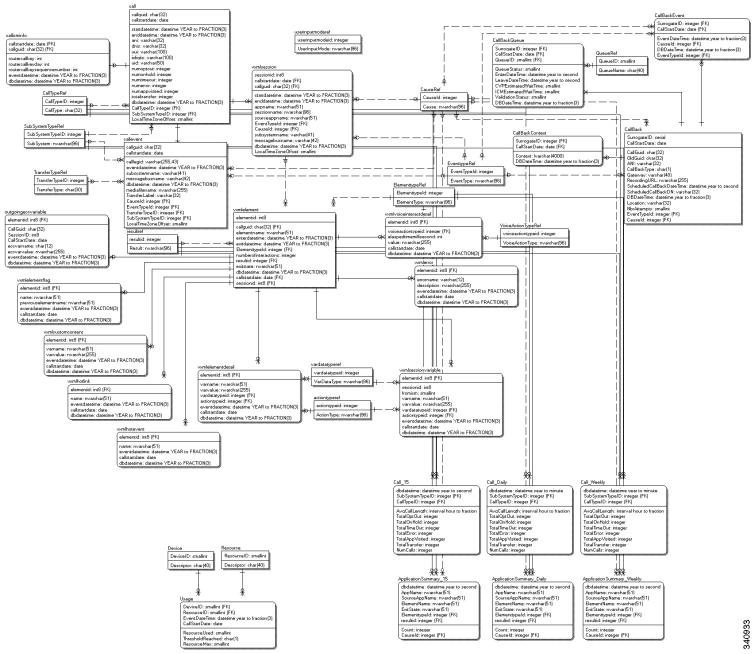


Figure 36: Callback Tables

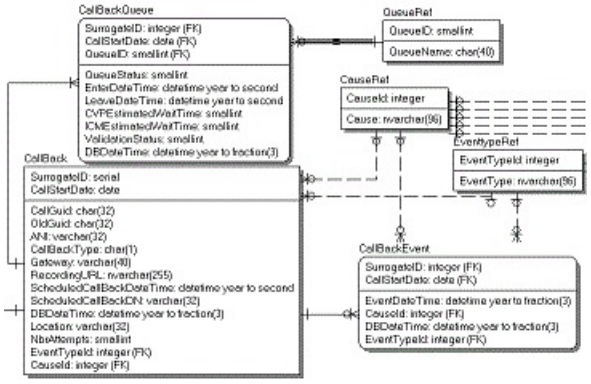


Figure 37: Summary Tables

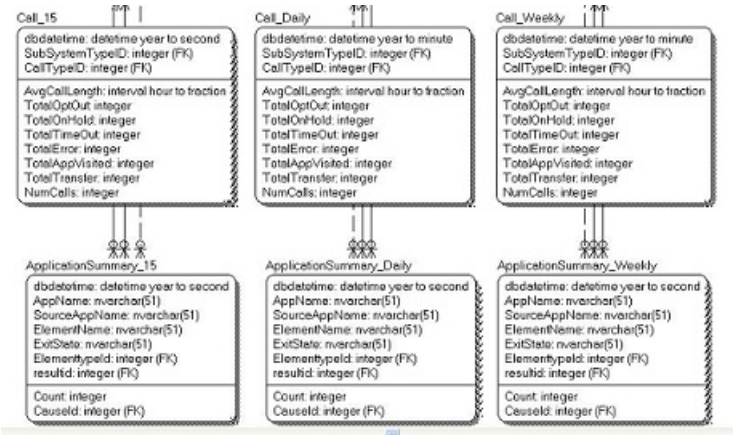
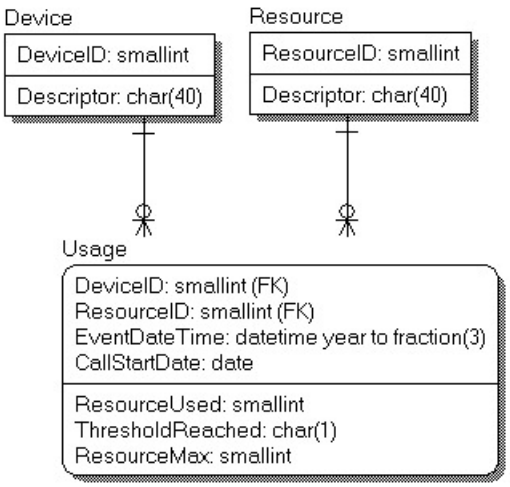


Figure 38: Trunk Group Utilization Tables



Keys

The documentation for the reporting schema lists fields as PK, FK, A, or No.

Fields are designated in this document as Primary Key (PK), Foreign Key (FK), or Alternate Key (AK) for informational purposes only. For performance reasons, the schema does not enforce Primary, Foreign, or Alternate keys. When the Index column for a field shows FK or AK, it means that a field can be looked up to return a text explanation of the code in another table.

Primary and Alternate Keys are in fact supported by an index. Major Foreign Keys (CallGUID, SessionID, ElementID) have a supporting index. Foreign Keys which refer to lookup tables are not supported by an index.

Unified Customer Voice Portal Reporting Data Model

The following section provides information on the following topics:

DateTime Columns

Most major tables have three columns to assist in managing the reporting server itself.

- **CallStartDate** - This column is used for partitioning and purging data.

  This is the date and time the call started and is meant to ensure that detail data, which may cross a date boundary, are tied to the original call and can all be stored and removed together.

- **EventDateTime** - This is the date and time that the recorded event transpired. This is recorded in UTC time.

  The Call Table table has two EventDateTime fields, recorded as *StartDateTime* and *EndDateTime.*

- **DBDateTime** - This is the date and time that the recorded event was written to the database. It is meant to contrast with the EventDateTime. A marked difference between these values indicates a delay in the data arriving at the reporting server. This delay should either be allowed for or investigated.

Informix Dates and Times

The Informix engine that hosts the Unified CVP reporting database supports three concepts of time:

- Dates
- DateTimes
- Intervals

Dates

A date (for example CallStartDate) has no time element to it. It is specified between single quotes as 'MM/DD/YYYY'.

```
SELECT count(*)
  FROM Call
 WHERE CallStartDate='05/31/2012';
```

This date format can be modified to suit the locale with the DBDATE environment variable: in this case DBDATE=MDY4/, or Month/Day/Year(4) with a forward slash separator. These can be arranged in any order (DMY4-, or DMY2/ or Y4MD/) by modifying the DBDate enumeration variable.

Date also supports: key words such as 'TODAY' and date arithmetic.

For example, this returns a count of calls received yesterday:

```
SELECT count(*)
  FROM Call
 WHERE CallStartDate=TODAY-1;
```

Functions such as YEAR(), MONTH() and WEEKDAY().

```
SELECT count(*)
  From Call
 WHERE WEEKDAY(CallStartDate)=1
```

Note Days of the week are numbered from 0 through 6 where 0 is Sunday and 6 is Saturday.

DateTimes

DateTimes include a time component and use the ANSI standard: 'YYYY-MM-DD HH:MM:SS.FFF' where FFF are fractions of seconds. For example, this returns a count of calls received in a given 48 hours:

```
SELECT count(*)
  FROM Call
 WHERE Call.StartDateTime between '2009-05-01 00:00:00' AND '2009-05-3
       23:59:59';
```

These support the same YEAR(), MONTH() and WEEKDAY() functions as the Date datatype. The Current date and time is specified as 'CURRENT YEAR TO SECOND' and also supports date arithmetic.

```
SELECT count(*)
  FROM Call
 WHERE Call.StartDateTime > CURRENT YEAR TO SECOND - 2 UNITS DAY;
```

Unified CVP DateTimes are all recorded as UTC time, with the exception of *dbdatetime* which is recorded as a local time. Localtimezoneoffset is a column in the Call table that contains the number of minutes offset from UTC to derive the Local Time. This can be used as an interval. (In the example below, localtimezoneoffset is -240 minutes).

```
select first 10 enddatetime, enddatetime + localtimezoneoffset units
minute as LocalTime from call;

enddatetime             localtime

2010-02-09 15:03:54.453 2010-02-09 11:03:54.453
2010-02-09 15:03:54.453 2010-02-09 11:03:54.453
2010-02-09 15:03:54.469 2010-02-09 11:03:54.469
2010-02-09 15:01:23.125 2010-02-09 11:01:23.125
2010-02-09 15:03:54.469 2010-02-09 11:03:54.469
2010-02-09 15:01:23.141 2010-02-09 11:01:23.141
2010-02-09 15:03:54.500 2010-02-09 11:03:54.500
2010-02-09 15:01:23.156 2010-02-09 11:01:23.156
2010-02-09 15:01:23.156 2010-02-09 11:01:23.156
2010-02-09 15:01:23.156 2010-02-09 11:01:23.156
```

An aggregation function lastperiod(datetime, *Period*) is supported. *Period* can be: 15, 30, 60, DD, WW, or MM. This will convert the datetime into the date and time at which the current period started. Hence:

Lastperiod(2009-10-14 12:46:56,*15*) returns *2009-10-14 12:45:00*

Lastperiod(2009-10-14 12:46:56, *30*) returns *2009-10-14 12:30:00*

Lastperiod(2009-10-14 12:46:56, *60*) returns *2009-10-14 12:00:00*

Lastperiod(2009-10-14 12:46:56, *DD*) returns *2009-10-14 00:00:00*

Lastperiod(2009-10-14 12:46:56, *WW*) returns *2009-10-11 00:00:00 (Sunday)*

Lastperiod(2009-10-14 12:46:56, *MM*) returns *2009-10-1 00:00:00 (1st day of the month)*

Intervals

An Interval is a span of time and can be specified as *n UNITS period* where *period* can be:

- YEAR
- MONTH
- DAY
- HOUR
- MINUTE
- SECOND

A database query with an interval must be sent in the preceding format. When returned from the database, the interval will look like a datetime (YYYY-MM-DD HH:MM:SS.FFF). The components that are returned depend on the interval definition. It is unlikely that a DAY component will be returned from Unified CVP intervals; instead, expect a format like HH:MM:SS.FFF.

For a full discussion of Informix, refer to the Informix Guide to SQL: Reference Manual.

SIP Calls

SIP calls are recorded in the Call Table along with VXML calls.

They can be distinguished from VXML calls with the CallTypeID column. (Contains "4". Refer to the CallTypeRef Table, on page 87, where 4 is a SIP call.)

Events for these calls (such as start and end) are recorded in the CallEvent Table.

Sample Query and SIP Calls

Details for a SIP call could be retrieved using the following query:

```
SELECT Call.*, CallEvent.*
  FROM Call, CallEvent
 WHERE Call.CallGUID=CallEvent.CallGuid
   AND Call.CallGuid='CallGuid';
```

where CallGuid is replaced by the value of the CallGuid for which information is desired.

Trunk Utilization

Trunk utilization is a record of state messages from various devices linked to the reporting server and their current status. The frequency in which these messages are written is controlled by the IOS Gateway (Gateway Utilization). This data captures a point-in-time over time. It is laid out in a fact table (Usage Table) with three dimensions - Resource, Device, and Time.

Because time is not likely to be consistent across all devices, the Usage table has not been codified as an official dimension table, but rather as a date and time. Queries for usage should aggregate from this table.

Sample Queries, Trunk Utilization

Query for average CPU across all devices for the month of May:

```
SELECT avg(ResourceUsed)
  FROM Usage, Resource
 WHERE Resource.ResourceID=Usage.ResourceID
   AND Resource= 'CPU'
   AND Usage.EventDateTime between '2009-05-01 00:00:00' AND '2009-05-31
23:59:59';
```

Note that BETWEEN is inclusive. This query can also be written as:

```
AND Usage.EventDateTime >=  '2009-05-01 00:00:00' AND Usage.EventDateTime <=
  '2009-05-31
23:59:59';
```

Query for a list of devices and a count of the number of times they exceeded a threshold during the month of May:

```
SELECT Device, Resource, count(*)
  FROM Device, Resource, Usage
 WHERE Resource.ResourceID=Usage.ResourceID
   AND Device.DeviceID=Usage.DeviceID
   AND Usage.ThresholdReached= 'Y'
   AND month(Usage.EventDateTime) = 5
GROUP BY Device, Resource;
```

Note the use of the Month() function in *AND month (Usage.EventDateTime) = 5* .

Cisco Unified Customer Voice Portal Database Tables

This section lists the Unified CVP tables that hold reporting data.

Tables are categorized as follows:

Call Tables

The following Call tables are described in this section:

Call Table

This table is the primary record of a call and contains the basic metrics for each call. It contains one record per call.

Any drill into a specific call should start here to obtain the proper CallGUID.

On occasion, messages are dropped, even for an otherwise successful call. In such cases, EndDateTime is set to the same value as StartDateTime. Thus, if a call appears to be of 0 duration, report writers will know to exclude such a call from consideration in cases where it would otherwise skew metrics.

Table 6: Call Table

Field	Type	Null	Index	Description
CallGUID	char(32) for new installations char(35) for upgrades	No	PK (Composite CallGUID,CallStartDate)	The global unique id of a call.
CallStartDate	date	No	PK (Composite CallGUID,CallStartDate)	The date of the call, for data purging purposes.
StartDateTime	datetime YEAR to FRACTION(3)	No	Yes	EventDateTime for the date and time a call was made.
EndDateTime	datetime YEAR to FRACTION(3)	Yes	Yes	EventDateTime for the date and time a call ended with hang-up or disconnect.
ANI	varchar(32)	Yes	No	The ANI of the caller sent by telephony provider
DNIS	varchar(32)	No	No	The DNIS of a call sent by telephony provider.
UUI	varchar(100)	Yes	No	The UUI of the originating caller sent by telephony provider.

Field	Type	Null	Index	Description
IIdigits	varchar(100)	Yes	No	The IIDIGITS of the originating caller sent by telephony provider
UID	varchar(50)	Yes	No	The external UID of the caller if the call is associated with a user.
Numoptout (deprecated)	int	No	No	The number of times that the call is opted out to an agent.
NumTimeOut	int	No	No	The number of times the call timed out.
NumError	int	No	No	The number of errors that occurred during the call.
NumOnHold	int	No	No	The number on hold within a VXML application.
NumAppVisited	int	No	No	The number of applications visited during the life of the call.
TotalTransfer	int	No	No	The total number of times the call is transferred out. A transfer includes transfers to agents as well as a transfer to the VRU leg.
DBDateTime	datetime YEAR to FRACTION (3)	No	Yes	The date and time of the database operation (when the record was inserted).
CallTypeID	smallint *Formerly char(1)*	No	Non-Indexed FK	The type of call. See CallTypeRef Table.
SubSystemTypeID	int	No	Non-Indexed KX	The type of Unified CVP Service, such as SIP, IVR, VXML.
LocalTime ZoneOffset	smallint		No	The offset in minutes of the local timezone from UTC timezone. *Replaces LocalTimeZone.*

CallEvent Table

This table tracks each event that occurs within a call.

This table is populated for SIP calls. VXML calls will be recorded in the analogous VXMLSession Table.

Table 7: CallEvent Table

Field	Type	Null	Index	Description
CallGUID	char(32) for new installations char(35) for upgrades	No	Indexed FK (Composite CallGUID, CallStartDate)	The global unique id of a call
CallStartDate	date	No	FK (Composite CallGUID, CallStartDate)	The date of the call, for data purging purposes
CallLegID	varchar(255,43) *Formerly Varchar(43)*	Yes	No	A call id assigned by a Service
MessageBusName	varchar(42)	No	No	The name of the Call Server (its message adapter name) with which the event is associated
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the event
EventTypeID	int	No	Non-Indexed FK	The mechanism used to generate the call event. See EventTypeRef Table.
CauseID	int	No	Non-Indexed FK	The reason that the call event was generated. See CauseRef Table.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted).
TransferTypeID	integer	No	Non-Indexed FK	A unique id of the transfer type. See TransferTypeRef Table.
SubSystemTypeID	int	No	Non-Indexed FK	The type of the Service. See SubSystemTypeRef Table.

Field	Type	Null	Index	Description
SubSystemName	varchar(41)	No	No	The name of the Service the event originated from
MediaFileName	nvarchar(255)	Yes	No	*This is always null.*
TransferLabel	varchar(32)	Yes	No	This is the destination to which CVP transfers the call. The label is received from ICM via the TEMPORARY_CONNECT or CONNECT message
LocalTimeZoneOffset	smallint	NULL	No	The offset in minutes of this the local timezone from UTC timezone.

CallICMInfo Table

This table contains information to associate a Unified CVP call to ICM. It stores the ICM Call RouteCallKey, RouterCallKeyDay and RouterCallSequenceNumber for a call.

The CallICMInfo table is populated when the call is on the switch leg. This table is populated by SIP or VXML subsystems.

Note Currently the system does not capture the VRU leg of the call; thus if you have a Capture element and multiple Termination Call Detail (TCD) records are cut, the RouterCallKeySequenceNumber will increment in Historical Data Server (HDS) but will not be captured in the Unified CVP database. This is a known limitation.

Refer to the *CVP Administration and Configuration Guide* for further explanation about using the ReqICMLabel element to pass data to a Unified ICME script.

Table 8: CallICMInfo Table

Field	Type	Null	Index	Description
CallGUID	char(32) for new installations char(35) for upgrades	No	Indexed FK (Composite CallGUID, CallStartDate)	The global unique id of a call
CallStartDate	date	No	FK (Composite CallGUID, CallStartDate)	The date of the call, for data purging purposes

Field	Type	Null	Index	Description
RouterCallKey	Integer	No	AK (Composite index RouterCallKey, RouterCallKey Day)	ICM Router CallKey - single value per call. This value does not increment if the call is transferred from switch leg to VRU leg or if the call is transferred to an agent. If the call is a consult or conference, then Unified CVP will see two different callguids for the same call in its database. • The first callguid is the incoming callguid when the call is established. • The second callguid is for an agent originated/consult call. The RouterCallKey and RouterCallKey Day will act as a binder/glue between the two callguids for that single call as these values will not change between the two legs of the call.

Field	Type	Null	Index	Description
RouterCallKeyDay	Integer	No	AK(Composite index RouterCallKey, RouterCallKey Day)	ICM RouterCallKeyDay Typically this number changes on the switch and VRU leg of a call. You will see 0 for the switch leg of the call and 1 for the VRU leg of the call. This number usually does not change for basic CVP calls, but will increment if customers are using the capture node in their ICM script or when there is a transfer to an agent on the switch leg. In this scenario, Unified CVP sends a new call to Cisco Unified Communications Manager (Unified CM). This comes back via JTAPI and is on a separate Peripheral Gateway (PG). As the new call shows on a separate PG, Unified ICM cuts a new TCD record when the call ends. The RouterCallKeySequenceNumber increments on that switch leg.
RouterCallKey SequenceNumber	int	Yes	Yes	ICM RouterCallKeySequenceNumber.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the event.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted).

VXML Tables

The following VXML tables are described in this section:

- VXMLElementDetail Table, on page 69
- VXMLElementFlag Table, on page 70
- VXMLError Table, on page 71
- VXMLHotEvent Table, on page 72
- VXMLHotLink Table, on page 73
- VXMLSession Table, on page 74
- VXMLSessionVariable Table, on page 75
- VXMLVoiceInteractDetail Table, on page 77

The data for VXML treatment is much richer than that which is available for SIP calls. Events can be captured from VXML for anything that occurs inside of the VXML script. These calls start at the Call Table, on page 61 and are linked to the VXMLSession Table, on page 74 using the CallGUID column.

The VXMLSession is made up of a series of elements that are visited within the context of an application. Each element may have multiple ancillary attributes such as flags that can be set in an element. Values for these flags may be found in the VXMLElementFlag Table, on page 70 and are linked to using the ElementID.

VXMLElementFlags information for a call can be retrieved using the following query:

```
SELECT VXMLElementFlag.Name
  FROM Call, VXMLSession, VXMLElement, VXMLElementFlag
 WHERE Call.CallGuid= CallGuid
AND Call.CallGuid=VXMLSession.CallGuid
AND VXMLSession.SessionID=VXMLElement.SessionID
AND VXMLElement.ElementID=VXMLElementFlag.ElementID;
```

where CallGuid is replaced by the value of the CallGuid for which information is desired.

VXMLCustomContent Table

This table contains one record for each VXML custom event. This event occurs if a custom component programmatically calls the AddToLog method of the Session API. The event will also occur when an element whose configuration contains entries in the Add To Log table in the General tab is run.

Table 9: VXMLCustomContent Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique ID of a visited element.
CallStartDate	date	No	FK (Composite index ElementID, CallStartDate)	The date of the call, for data purging purposes.
VarName	nvarchar(51)	No	No	The name of the custom event variable.

Field	Type	Null	Index	Description
VarValue	nvarchar(255)	Yes	No	The value of the custom event variable.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the variable is changed.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

VXMLElement Table

This table contains one record for each VXML script element visited by a call. For example, if the same element is visited twice in an application script during a call, there will be two separate element records.

Table 10: VXMLElement Table

Field	Type	Null	Index	Description
ElementID	int8	No	PK (Composite ElementID, CallStartDate)	The unique id of a visited element.
CallStartDate	date	No	PK (Composite ElementID, CallStartDate)	The date of the call, for data purging purposes.
SessionID	int8	No	Indexed FK	The unique id of a VXML application session.
CallGUID	char(32) for new installations char(35) for upgrades	No	FK	The global unique id of a call.
ElementName	nvarchar(51)	No	No	The name of an element.

Field	Type	Null	Index	Description
ElementTypeID	int	No	Non-Indexed FK	The type of element.
EnterDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the element was entered.
ExitDateTime	datetime YEAR to FRACTION(3)	Yes	No	Date and time when the element was exited.
ExitState	nvarchar(51)	Yes	No	The exit state of the element.
NumberOfInteractions	int	Yes	No	The number of interactions while the user visited this element.
ResultID	int	Yes	Non-Indexed FK	Indicates how an element ended.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

VXMLElementDetail Table

This table contains one detail record for each script element variable. VarValue holds the String value of the variable and VarDataTypeID specifies the data type of the variable to which the String value can be converted.

Table 11: VXMLElementDetail Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique id of an element.

Field	Type	Null	Index	Description
CallStartDate	date	No	Yes (Composite index ElementID, CallStartDate)	The date of the call, for data purging purposes.
VarName	nvarchar(51)	No	No	The name of the element variable.
VarValue	nvarchar(255)	Yes	No	The String value of the element variable.
VarDataTypeID	int	No	Non-Indexed FK	The data type of the element variable, such as String, Integer, Boolean.
ActionTypeID	int	No	Non-Indexed FK	The type of action for an element that changes data.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	Date and time when the variable was changed.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted).

VXMLElementFlag Table

This table contains one record for each element in which a flag was activated. The Name field holds the name of the flag.

Table 12: VXMLElementFlag Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique id for the element in which the flag activated.
CallStartDate	date	No	Yes (Composite index ElementID, CallStartDate)	The date of the call, for data purging purposes.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the flag activated.

Field	Type	Null	Index	Description
Name	nvarchar(51)	No	No	The flag name.
PreviousElementName	nvarchar(51)	Yes	No	The name of the previous application element.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted).

VXMLError Table

This table contains VXML errors that occurred during the life of the VXML application session. The table contains one record for each element in which an error occurred. The ErrorName field holds the name of the error.

Table 13: VXMLError Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique id for the element in which the error occurs.
CallStartDate	date	No	FK (Composite index ElementID, CallStartDate)	The date of the call, for data purging purposes.
ErrorName	varchar(12)	No	No	Name of an error.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the error occurred.
Description	nvarchar(255)	No	No	The detailed error message.

Field	Type	Null	Index	Description
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

VXMLHotEvent Table

HotEvent is a global event that when caught, executes developer-specified actions. This table contains information (HotEvent name, HotEvent DateTime and the ElementID) about the HotEvent occurred in an element.

Table 14: VXMLHotEvent Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique id for the element in which the HotEvent occurred.
CallStartDate	date	No	Yes (Composite index ElementID, CallStartDate)	The date of the call, for data purging purposes.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when HotEvent occurred.
Name	nvarchar(51)	No	No	The name of the HotEvent.

Field	Type	Null	Index	Description
DBDateTime	datetime YEAR to FRACTION(3)	No	No	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

VXMLHotLink Table

Hotlink is a globally accessible utterance key press that immediately brings the call to a specific part of the call flow or throws an event. This table contains information (HotLink name, HotLink DateTime and the ElementID) about the HotLink that occurred in an element.

Table 15: VXMLHotLink Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique id for the element in which the hotlink activated.
CallStartDate	date	No	FK (Composite index ElementID, CallStartDate),	The unique id for the element in which the HotLink activated.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date of the call, for data purging purposes
Name	nvarchar(51)	No	No	The name of the HotLink.

Field	Type	Null	Index	Description
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

VXMLSession Table

This table contains one record for each application visited by a VXML call. For example, if a call has transferred from one application to another one, the call with the same CallGUID will have two session records.

SIP calls are recorded in the CallEvent Table, on page 63.

Table 16: VXMLSession Table

Field	Type	Null	Index	Description
SessionID	int8	No	PK (Composite SessionID, CallStartDate)	The unique ID of a VXML application session.
CallStartDate	date	No	PK (second field in PK and Composite indexes)	The date of the call, for data purging purposes.
SessionName	nvarchar(96)	No	No	The name of the session assigned by VXML Server.
CallGUID	char(32) for new installations char(35) for upgrades	No	Indexed FK (Composite index CallGUID, CallStartDate);	The global unique id of a call.
StartDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when session starts.
AppName	nvarchar(51)	No	Yes	The name of the VXML application.

Field	Type	Null	Index	Description
EventTypeID	int	No	Non-Indexed FK	The mechanism used to end the application visit.
CauseID	int	No	Non-Indexed FK	The reason that the application visit ended.
EndDateTime	datetime YEAR to FRACTION(3)	Yes	No	The end date and time of the session.
SourceAppName	nvarchar(51)	Yes	No	The name of the application that transferred to this one.
SubSystemName	varchar(41)	No	No	The name of the VXML Service.
MessageBusName	varchar(42)	No	No	The name of the message bus that delivers the VXML data feed message.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).
Duration	int	Null	No	The length of the session.
LocalTimeZoneOffset	smallint	No	No	The offset in minutes of this the local timezone from UTC timezone. *Replaces LocalTimeZone.*

VXMLSessionVariable Table

This table contains one record for each instance of a session variable. For example, if the same session variable was modified once in an application script during a call, there will be two separate records, one for its initial value when it was created and another for the updated value.

Table 17: VXMLSessionVariable Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The identifier of the element in which the session variable changes.
SessionID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique ID of an IVR application session.
CallStartDate	date	No	Yes (second field in Composite indexes)	The date of the call, for data purging purposes.
VarName	nvarchar(51)	No	No	The name of the session variable that was exited.
VarValue	nvarchar(255)	Yes	No	The value of the session variable.
ActionTypeID	int	No	Non-Indexed FK	The type of action for a session variable that changes data.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the session variable changed.
VarDataTypeID	int	No	Non-Indexed FK	The data type of the session variable, such as Integer, String, Boolean.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

Field	Type	Null	Index	Description
FromICM	Boolean	No	No	Indicates whether this session variable change originated from Unified ICME; Informix stores these values as 1 or 0, but represents these values as “t” or “f”

VXMLVoiceInteractDetail Table

This table has one record for each Voice Interaction with the caller.

Table 18: VXMLVoiceInteractDetail Table

Field	Type	Null	Index	Description
ElementID	int8	No	Indexed FK (Composite index ElementID, CallStartDate)	The unique ID of a visited element.
CallStartDate	date	No	FK (Composite index ElementID, CallStartDate);	The date of the call, for data purging purposes.
ElapsedTimeMillisecond	int	No	No	The time since the last interaction.
VoiceActionTypeID	int	No	Non-Indexed FK	The type of interaction.
Value	nvarchar(255)	Yes	No	The value of interaction.
DBDateTime	datetime YEAR to FRACTION(3)	No	No	The date and time of the database operation (when the record was inserted).

Summary / Aggregate Tables

The Summary / Aggregate tables are described in the following section:

- ApplicationSummary_15 Table
- ApplicationSummary_Daily Table
- ApplicationSummary_Weekly Table

- ApplicationSummary_Monthly Table, on page 81
- Call_15 Table
- Call_Daily Table
- Call_Weekly Table

Unified CVP reporting server includes a summary process that aggregates data from the Call and VXMLElement tables into new summary tables. These six tables hold summary data on Call metrics and on elements visited in Unified CVP applications.

These metrics include:

- The datetime of the beginning of the summary period.
- The average call length for the calls in this period
- Various totals, including the total number of opt outs, timeouts, and on holds for the calls in this period; the total number of transfers; and the total number of applications visited for the calls in this period

Summary tables use a star schema. Each summary table has a collection of non-numeric attributes and one or more numeric attributes that can be aggregated according to their type. Adding or removing an attribute from a query in a report definition allows a drill up or drill down into the data presented.

For example: the Application Summary tables have the following non-numeric attributes: Dbdatetime | Appname | Sourceappname | Elementname | Elementtypeid | Resultid | Causeid | Exitstate.

The numeric data available to report on those dimensions are: Avg_elapsed and Count.

`Select Appname, avg(avg_elapsed), sum(count)` will yield the average elapsed time and number of occurrences for an application. Adding ElementName to the Select clause (`Select Appname, ElementName, avg(avg_elapsed), sum(count)`) will further elaborate on where time was spent within the application. This can be further qualified by checking for specific Results, Causes, or Exit states.

These summary tables are not pure fact tables in cases where the dimensions are not always ID columns which refer to dimension or lookup tables.

In an upgrade situation, the summary process will start aggregation at the earliest data date within the Call and VXMLelement tables. At most, once every 15 minutes, the summary process will aggregate one day's worth of data from historical records to avoid overtaxing the system by attempting to process too much data.

This means that in a single 24-hour period, the system can summarize 96 days of data at most.

- _Daily tables will be populated one day behind the _15 minute tables.
- _Weekly tables will be populated from _Daily tables once those have been fully populated for the week in question.
- _Monthly tables will be populated from _Weekly tables once those have been fully populated for the month in question.

Retention for summary tables is hardcoded to 60 days for 15 minute summaries, 18 months for daily summaries, 10 years for weekly data, and 40 years for monthly aggregation.

Note

- Take into consideration that it can take some time to collect aggregate-level data from the reporting server.
- Summary tables are built in 15-minute increments using the local time of the reporting server. Latency of source data is not guaranteed. In the event of a failover situation, data may arrive hours after it was initially created. For this reason all summary time periods reflect the time that the source data arrived at the database, which will generally be close to the time that it was created.

ApplicationSummary_15 Table

The ApplicationSummary_15 table is a 15-minute summary of Application/element data, useful for Dominant Path analysis.

Table 19: ApplicationSummary_15 Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to second	Yes	PK	The start of the time period for this row.
AppName	nvarchar(51)	Yes	PK	The name of the VXML application.
SourceAppName	nvarchar(51)	Yes	PK	The name of the application that transferred to this one.
ElementName	nvarchar(51)	No	PK	The name of the element.
ExitState	nvarchar(51)	Yes	PK	The exit state of the element.
ElementTypeID	integer	Yes	PK; FK	The unique ID of an element type.
ResultID	integer	Yes	PK, FK	The unique ID of a result
Count	integer	Yes	No	The number of occurrences for this time period.
CauseId	int	Yes	FK	The unique ID of a cause.
Avg_elapsed	int	yes	No	The average elapsed time for this element.

ApplicationSummary_Daily Table

The ApplicationSummary_Daily table provides a daily summary of Application/element data, useful for Dominant Path analysis.

Table 20: ApplicationSummary_Daily Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to second	Yes	PK	The start of the time period for this row.
AppName	nvarchar(51)	Yes	PK	The name of the VXML application.
SourceAppName	nvarchar(51)	Yes	PK	The name of the application that transferred to this one.
ElementName	nvarchar(51)	Yes	PK	The name of the element.
ExitState	nvarchar(51)	Yes	PK	The exit state of the element.
ElementTypeID	integer	Yes	PK, FK	The unique ID of an element type.
ResultID	integer	Yes	PK, FK	The unique ID of a result.
Count	integer	Yes	No	The number of occurrences for this time period.
CauseId	int	Yes	FK	The unique ID of a cause.
Avg_elapsed	int	yes	No	The average elapsed time for this element.

ApplicationSummary_Weekly Table

A weekly summary of Application/element data, useful for Dominant Path analysis.

Field	Type	Null	Index	Description
dbdatetime	datetime year to second	Yes	PK	The start of the time period for this row

Field	Type	Null	Index	Description
AppName	nvarchar(51)	Yes	PK	The name of the VXML application
SourceAppName	nvarchar(51)	Yes	PK	The name of the application that transferred to this one
ElementName	nvarchar(51)	Yes	PK	The name of the element
ExitState	nvarchar(51)	Yes	PK	The exit state of the element
ElementTypeID	integer	Yes	PK, FK	The unique id of an element type
ResultID	integer	Yes	PK, FK	The unique id of a result
Count	integer	Yes	No	The number of occurrences for this time period
CauseId	int	Yes	FK	The unique id of a cause
Avg_elapsed	int	yes	No	The average elapsed time for this element

ApplicationSummary_Monthly Table

The ApplicationSummary_Monthly table displays a monthly summary of Application/element data, useful for Dominant Path analysis.

Table 21: ApplicationSummary_Monthly Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to second	Yes	PK	The start of the time period for this row.
AppName	nvarchar(51)	Yes	PK	The name of the VXML application.
SourceAppName	nvarchar(51)	Yes	PK	The name of the application that transferred to this one.
ElementName	nvarchar(51)	Yes	PK	The name of the element

Field	Type	Null	Index	Description
ExitState	nvarchar(51)	Yes	PK	The exit state of the element.
ElementTypeID	integer	Yes	PK, FK	The unique ID of an element type.
ResultID	integer	Yes	PK, FK	The unique ID of a result.
Count	integer	Yes	No	The number of occurrences for this time period.
CauseId	int	Yes	FK	The unique id of a cause.
Avg_elapsed	int	yes	No	The average elapsed time for this element.

Call_15 Table

The Call_15 table dispalys a15-minute summary of call activity by SubSystemType and CallType.

Table 22: Call_15 Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to second	Yes	PK	time in 15-minute increments.
SubSystemTypeID	integer	Yes	PK, FK	The unique ID of a Service type.
CallTypeID	smallint	No	PK, FK	The unique ID of a Call type.
AvgCallLength	interval HOUR (3) to FRACTION (3)	Yes	No	The average call length for this period.
TotalOptOut	integer	Yes	No	The total number of Opt Outs in this period.
TotalOnHold	integer	Yes	No	The total number of Holds in this period.
TotalTimeOut	integer	Yes	No	The total number of Time Outs in this period.

Field	Type	Null	Index	Description
TotalError	integer	Yes	No	The total number of errors in this period.
TotalAppVisited	integer	Yes	No	The total number of applications visited in this period.
TotalTransfer	integer	Yes	No	The total number of transfers in this period.
NumCalls	integer	Yes	No	The total number of calls in this period.

Call_Daily Table

The Call_Daily table displays a daily summary of call activity by SubSystemType and CallType

Table 23: Call_Daily Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to minute	Yes	PK	The time in daily increments.
SubSystemTypeID	integer	Yes	PK, FK	The unique ID of a Service type.
CallTypeID	smallint	Yes	PK, FK	The unique ID of a Call type.
AvgCallLength	interval HOUR (3) to FRACTION (3)	Yes	No	The average call length for this period.
TotalOptOut	integer	Yes	No	The total number of Opt Outs in this period.
TotalOnHold	integer	Yes	No	The total number of Holds in this period.
TotalTimeOut	integer	Yes	No	The total number of Time Outs in this period.
TotalError	integer	Yes	No	The total number of errors in this period.

Field	Type	Null	Index	Description
TotalAppVisited	integer	Yes	No	The total number of applications visited in this period.
TotalTransfer	integer	Yes	No	The total number of transfers in this period.
NumCalls	integer	Yes	No	The total number of calls in this period.

Call_Weekly Table

The Call_Weekly table displays a weekly summary of call activity by SubSystemType and CallType.

Table 24: Call_Weekly Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to minute	Yes	PK	The time in weekly increments.
SubSystemTypeID	integer	Yes	PK, FK	The unique ID of a Service type.
CallTypeID	small int	Yes	PK, FK	The unique ID of a Call type.
AvgCallLength	interval HOUR (3) to FRACTION (3)	Yes	No	The average call length for this period.
TotalOptOut	integer	Yes	No	The total number of Opt Outs in this period.
TotalOnHold	integer	Yes	No	The total number of Holds in this period.
TotalTimeOut	integer	Yes	No	The total number of Time Outs in this period.
TotalError	integer	Yes	No	The total number of errors in this period.
TotalAppVisited	integer	Yes	No	The total number of applications visited in this period.

Field	Type	Null	Index	Description
TotalTransfer	integer	Yes	No	The total number of transfers in this period.
NumCalls	integer	Yes	No	The total number of calls in this period.

Call_Monthly Table

The Call_Monthly table displays a monthly summary of call activity by SubSystemType and CallType.

Table 25: Call_Monthly Table

Field	Type	Null	Index	Description
dbdatetime	datetime year to minute	Yes	PK	The time in weekly increments.
SubSystemTypeID	integer	Yes	PK, FK	The unique ID of a Service type.
CallTypeID	small int	Yes	PK, FK	The unique ID of a Call type.
AvgCallLength	interval HOUR (3) to FRACTION (3)	Yes	No	The average call length for this period.
TotalOptOut	integer	Yes	No	The total number of Opt Outs in this period.
TotalOnHold	integer	Yes	No	The total number of Holds in this period.
TotalTimeOut	integer	Yes	No	The total number of Time Outs in this period.
TotalError	integer	Yes	No	The total number of errors in this period.
TotalAppVisited	integer	Yes	No	The total number of applications visited in this period.
TotalTransfer	integer	Yes	No	The total number of transfers in this period.

Field	Type	Null	Index	Description
NumCalls	integer	Yes	No	The total number of calls in this period.

Lookup and Reference Tables

The Lookup and Reference tables are discussed in the following sections:

ActionTypeRef Table

This is a reference table that resolves an ActionTypeID to the text value for an element that changes data.

Table 26: ActionTypeRef Table

Field	Type	Null	Index	Description
ActionTypeID	int	No	PK	The unique id of an action type.
ActionType *Formerly Name*	nvarchar(96)	No	No	The name of the action type.

Table Values (ID, Action Name):

1, "Initialize"

2, "Update"

3, "Return"

CallTypeRef Table

This is a reference table that resolves CallTypeID to a text value.

Table 27: CallTypeRef Table

Field	Type	Null	Index	Description
CallTypeID	small int	No	PK	The unique ID of a call type reference.
CallType	char(32)	No	No	The Call Type.

Table Values (ID, CallType):

1, "Legacy Audio"

2, "Legacy Video"

4, "SIP"

5, "VRU"

6, "VXML"

7, "Basic Video"

8, "Full Video"

CauseRef Table

This table maps a CauseID to the text value for the cause.

Table 28: CauseRef Table

Field	Type	Null	Index	Description
CauseID	int	No	PK	The unique ID of a call event cause.
Cause	nvarchar(96)	No	No	The cause of the event. *Formerly Name*

Table Values (ID, Cause):

0 = "None"

1, "Normal Completion"

2, "Call Abandon"

3, "Call Transferred"

4, "New Transaction"

5, "Busy"

6, "No Answer"

7, "Maintenance"

8, "Net Congestion"

9, "Net Not Obtainable"

10, "Reorder Tone"

11, "Resources Not Available"

12, "Trunks Busy"

13, "Called Party Disconnected"

14, "Max Ports"

15, "Suspended"

16, "Time Out"

17, "Invalidated"

18, "Error"

19, "Video Answered"

20, "Post Call Answer"

21, "Invalid"

22, "Failure"

23, "Audio Recording Start"

24, "Audio Recording Stop"

25, "No Response"

26, "Invalid Number"

27, "Connected"

28, "Caller Canceled"

29, "Whisper Start"

30, "Whisper Done"

31, "Whisper Setup Failed"

32, "Abandon In Whisper"

33, "Whisper Media Error"

1001, "Hang Up"

1002, "Network"

1003, "System"

1004, "Script Type"

1005, "Unknown UApp"

1006, "Script Name"

1007, "Config Param"

1008, "Misconfig Ecc"

1009, "Media File"

1010, "Semantic"

1011, "VXML Format"

1012, "VXML Element"

1013, "Variable Data"

1014, "No Var Data"

1015, "Format"

1016, "Entry Invalid"

1017, "No Entry"

1018, "Media Resource Video" [Unable to perform video-related request due to resource limitations]

1019, "Recording Failed"

1020, "Data Range"

1021, "Timed Out"

1022, "Called Hung Up" [Agent, VRU, or other endpoint hung up on caller; that is, the caller did not hang up first]

1023, "No Answer"

1024, "Busy"

1025, "Transfer"

1026, "Invalid Extn"

1027, "Hang Up Forced"

1028, "After Trans Estab"

1030, "Unsupported Language"

1031, "Media Resource ASR

1032, "Media Resource TTS"

1033, "General ASR TTS"

1034, "Unknown Error

1035, "Missing Configuration"

1036, "Duplicate Record"

1037, "Not in Queue"

1039, "Unknown Callguid"

1040, "CVP System Unavailable"

1041, "CVP App Error"

1042, "CVP App Hang Up"

1043, "Error CVP App Suspended"

1044, "Error CVP No Session Error"

1045, "Error CVP Bad Fetch"

1046, "No Streaming Media Resource TTS"

Device Table

The device for which this resource is measured. This is an IP Address.

Table 29: Device Table

Field	Type	Null	Index	Description
DeviceID	smallint	No	PK	Unique identifier of this device.
Descriptor	char(40)	Yes	No	The IP address of this device.

ElementtypeRef Table

This table maps an ElementTypeID to a text value for the VXML element type.

Table 30: ElementtypeRef Table

Field	Type	Null	Index	Description
ElementTypeID	int	No	PK	The unique id of an element type.
ElementType	nvarchar(96)	No	Yes	The name of the element type . *Formerly Name*

Table Values (ID, ElementType):

0, Start"

1, "End"

2, "Subdialog_Start"

3, "Subdialog_Return"

4, "Decision"

5, "Action"

6, "Custom"

7, "HotLink"

8, "HotEvent"

9, "ElementFlag"

10, "Voice"

11, "VXMLInsert"

12, "ReqICMLabel"

13, "Genera"l

EventTypeRef Table

This is the table to map an EventID to the text value for its name (event type).

Field	Type	Null	Index	Description
EventTypeID	int	No	PK	The unique id of a call event type
EventType	nvarchar(96)	No	No	The name of the event type *Formerly Name*

Table Values (ID, EventType):

0, "New Call"

1, "Connect Failure"

2, "Busy"

3, "No Answer"

4, "Answer"

5, "Abandon"

6, "Disconnect"

7, "Hang Up"

8, "App Transfer"

9, "App Session Complete"

10, "Call Transfer"

11, "Run Script"

12, "Agent Recording"

13, "ICM Recording"

14, "Agent Video"

15, "ICM Video"

16, "Bridge Transfer"

17, "Blind Transfer"

18, "ReqICMLabel"

19, "Audio Recording"

20, "Callback Canceled"

21, "Callback Pending"

22, "Callback In Progress"

23, "Callback Tentative"

24, "Callback Complete"

25, "Callback Recover"

26, "Callback Created"

29, "Max allowed callbacks to this ANI exceeded"

OutgoingECCVariable Table

This table stores the ECC Variables that are returned from Unified CVP to an ICM script.

At present, this table is populated by the courtesy callback studio application element and when the ReqICMLabel element is used in a Call Studio script. Refer to the *CVP Administration and Configuration Guide* for further explanation about using the ReqICMLabel element to pass data to a Unified ICME script.

Table 31: OutgoingECCVariable Table

Field	Type	Null	Index	Description
CallGUID	char(32) for new installations char(35) for upgrades	No	Indexed FK (Composite index CallGUID, CallStartDate)	The global unique id of a call.
CallStartDate	date	No	FK (Composite CallGUID, CallStartDate)	The date of the call, for data purging purposes.
SessionID	int8	No	Yes	The identifier of the session in which the ECC variable changes.
ElementID	int8	No	Yes	The identifier of the element in which the ECC variable changes.

Field	Type	Null	Index	Description
ECCVarName	char(12)	No	No	The name of session variable that was exited.
ECCVarValue	nvarchar(255)	No	No	The value of session variable.
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the ECC variable changed.
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time of the database operation (when the record was inserted). This is useful for debugging purposes to determine lags between when the event occurred versus when it was written to the database (for example, a long lag may indicate problems with the reporting server).

QueueRef Table

QueueRef is a callback lookup table. This table maps QueueID to a text value for the queue in which a callback is waiting. The QueueName stores whatever you decide to call the queues.

Table 32: QueueRef Table

Field	Type	Null	Index	Description
QueueID	smallint	No	PK	The unique ID of a queue.
QueueName	char(40)	Yes	No	The name of the queue.

Resource Table

The resources that are measured include Memory, CPU, DSO and DSP.

Table 33: Resource Table

Field	Type	Null	Index	Description
ResourceID	smallint	No	PK	Unique Identifier.

Field	Type	Null	Index	Description
Descriptor	char(40)	Yes	No	The name of the resource we are measuring (CPU, Memory, DSP, DS0, System).

ResultRef Table

This table maps a ResultID to a text value for a result.

Table 34: ResultRef Table

Field	Type	Null	Index	Description
ResultID	int	No	PK	The unique ID of a result.
Result	nvarchar(96)	No	No	The name of the element result. *Formerly Name*

Table Values (ID, Result):

1, "Normal"

2, "Invalidated"

3, "HotEvent"

4, "HotLink"

5, "Hang Up"

6, "Error"

7, "Transfer"

SubSystemTypeRef Table

This table maps a SubSystemTypeID to a Unified CVP Service type.

Table 35: SubSystemTypeRef Table

Field	Type	Null	Index	Description
SubSystemTypeID	int	No	PK	The unique ID of a Service type.
SubSystem	nvarchar(96)	No	No	The name of the Service type.

Table Values (ID, Name):

0, "SIP"

1, "IVR"

2, "VXML"

3, "OAMP" [Operate, Administer, Maintain, Provision = Operations Console]

4, "Controller"

5, "RPT"

6, "ICM"

7, "ORM" [Element with Unified CVP components that allows the Operations Console to manage the components]

8, "System"

TransferTypeRef Table

This is a reference table to resolve TransferTypeID to a text value.

Table 36: TransferTypeRef Table

Field	Type	Null	Index	Description
TransferTypeID	integer	No	PK	A unique ID of the transfer type.
TransferType	varchar(30)	Yes	No	The type of transfer performed.

Usage Table

This is a fact table of device/resource measurements.

Field	Type	Null	Index	Description
DeviceID	smallint	No	FK	Unique identifier of this device
ResourceID	smallint	No	FK	Unique Identifier
EventDateTime	datetime year to fraction(3)	No	PK	Date and time of this measurement

Field	Type	Null	Index	Description
CallStartDate	date	No	PK	The date of this measurement for purge purposes
ResourceUsed	smallint	Yes	No	The amount of resouce used
ThresholdReached	char(1)	Yes	No	True/False. Was the maximum threshold for this resource reached?
ResourceMax	smallint	Yes	No	The amount of this resource available on this device

UserInputModeRef Table

This table maps a UserInputModeID to the name of the user input mode.

Table 37: UserInputModeRef Table

Field	Type	Null	Index	Description
UserInputModeID	int	No	PK	The unique ID of a user input mode.
UserInputMode	nvarchar(96)	No	No	The name of the user input mode. *Formerly Name*

Table Values (ID, Name):

1, "DTMF"

2, "Voice"

3, "DTMF Voice"

VarDataTypeRef Table

This table maps a VarDataTypeID to the data type of a variable.

Table 38: VarDataTypeRef Table

Field	Type	Null	Index	Description
VarDataTypeID	int	No	PK	The unique ID of a variable data type.
VarDataType	nvarchar(96)	No	No	The name of the variable data type. *Formerly Name*

Table Values (ID, Name):

0, "String"

1, "Int"

2, "Float"

3, "Boolean"

VoiceActionTypeRef Table

This table maps a VoiceActionTypeID to a text value.

Field	Type	Null	Index	Description
VoiceActionTypeID	int	No	PK	The unique ID of a VoiceActionTypeRef.
VoiceActionType	nvarchar(96)	No	No	The name of the call state. *Formerly Name*

Table Values (ID, Name):

1, "No Match"

2, "No Input"

3, "Audio Group"

4, "Input Mode"

5, "Utterance"

6, "Interpretation"

7, "Confidence"

Courtesy CallBack Tables

The following sections describe the Courtesy CallBack tables:

- CallBack Table
- CallBackEvent Table
- CallBackQueue Table

These tables support Courtesy Callback functionality.

Since this data is of an online-transaction-processing (OLTP) nature, it is retained in its own database, the callback database. When the caller registers a request for a callback, that request is stored in the CallBack Table.

A row is placed into the CallBackQueue Table for the call to manage timing and sequencing of calls.

Events that occur during the callback are registered in the CallBackQueue Table. This information can be retrieved using the following query:

```
SELECT Callback.*, CallbackContext.*, CallBackEvent.*,
       CallBackQueue.*
  FROM Callback, CallbackContext, CallBackEvent,
       CallBackQueue
 WHERE CallBack.CallGuid= CallGuid
   AND CallBack.SurrogateID=CallBackEvent.SurrogateID
   AND CallBack.SurrogateID=CallBackContext.SurrogateID
   AND CallBack.SurrogateID=CallBackQueue.SurrogateID;
```

Where CallGuid is replaced by the value of the CallGuid for which information is desired.

Query for number of callbacks currently pending:

```
SELECT count(*)
  FROM CallBack, EventTypeRef
 WHERE CallBack.EventTypeID=EventTypeRef.EventTypeID
   AND EventType in (Callback Pending);
```

Query for a list of failed callbacks with telephone number and failure reason code:

```
SELECT CallGuid, ANI, NbrAttempts, Cause
  FROM CallBack, CauseRef, EventTypeRef
 WHERE CallBack.CauseID=CauseRef.CauseID
   AND CallBack.EventTypeID=EventTypeRef.EventTypeID
   AND EventType in (Callback Canceled);
```

CallBack Table

The callback table is a view of two tables: Callback_Pending and Callback_Historical. The two tables are identical; every 30 minutes, data for completed calls is pulled from Callback_Pending and moved to Callback_Historical.

One row is generated in this table for each callback that is made.

Table 39: CallBack Table

Field	Type	Null	Index	Description
SurrogateID	serial	No	PK	A unique generated key to replace the CallGuid throughout the Callback schema.

Field	Type	Null	Index	Description
CallStartDate	date	No	PK	The date of the callback for data purging purposes.
OldGuid	char(32)	Yes	No	The Guid for the original scheduled callback. Used by the DB servlet to retrieve information about the old scheduled callback in order to create a new record for the pre-emptive callback.
ANI	varchar(32)	Yes	No	The number to which the callback will be placed.
CallBackType	char(1)	Yes	No	P = preemptive. S = scheduled
Gateway	varchar(40)	Yes	No	The identifier for the gateway. Can be an IP address or other string identifier.
RecordingURL	nvarchar(255)	Yes	No	The URL that points to a wav file of the recorded name of the caller.
ScheduledCallBack DateTime	datetime year to second	Yes	No	The Datetime (including timezone) for a scheduled callback. Not included for preemptive callbacks.
ScheduledCallBackDN	varchar(32)	Yes	No	The DN to which the scheduled callback will be placed. This will invoke an ICM script with this DN.
DBDateTime	datetime year to fraction(3)	Yes	No	The date and time of the database operation.
Location	varchar(32)	Yes	No	The location name assigned to a set of gateways. Used in scheduled callback to select applicable egress gateways for the callback.
NbrAttempts	smallint	Yes	No	The number of attempts made to call back.

Field	Type	Null	Index	Description
EventTypeId	integer	Yes	FK	The unique ID of a event type.
CauseId	integer	Yes	FK	The unique ID of a cause.

CallBackEvent Table

This table holds a record of each callback event that occurs for the call.

This table holds seven days worth of data. Purge daily.

Table 40: CallBackEvent Table

Field	Type	Null	Index	Description
SurrogateID	integer	No	PK, FK	A unique generated key to replace the CallGuid throughout the Callback schema.
CallStartDate	date	No	PK	The date of the callback for data purging purposes
EventDateTime	datetime year to fraction(3)	No	No	The date and time of the event
CauseId	integer	Yes	Yes	See Values below. CauseRef Table
DBDateTime	datetime year to fraction(3)	No	No	The date and time of the database operation.
EventTypeId	integer	No	FK	The unique ID of an event type. See ElementtypeRef Table.

Table Values (ID, CauseID)

0, "busy"

1, "noanswer"

2, "noresponse"

3, "invalid_number"

4, "connected"

5, "caller_canceled"

6, "trunksbusy"

7, "error"

CallBackQueue Table

This table holds data for the queue in which the call sits until its scheduled time or until a slot becomes available.

Table 41: CallBackQueue Table

Field	Type	Null	Index	Description
SurrogateID	integer (serial)	No	PK, FK	
CallStartDate	date	No	PK	The date of the callback for data purging purposes.
CallGuid	char(32) for new installations char(35) for upgrades	Yes	No	Unique ID for the call.
QueueID	smallint	No	FK	
QueueStatus	smallint	No	No	Status in queue: 0 = in_queue, 1 = not_in_queue 2 = Zombie
EnterDateTime	datetime year to second	No	No	The Datetime entered queue.
LeaveDateTime	datetime year to second	Yes	No	The Datetime left queue.
CVPEstimatedWaitTime	smallint	No	No	The CVP-calculated estimated wait time (in seconds) Since enterdatetime. This is generated during the insert, it will not be maintained.
ICMEstimatedWaitTime	smallint	No	No	Unified ICM-calculated estimated wait time (in seconds) Since enterdatetime. This is generated during the insert, it will not be maintained.

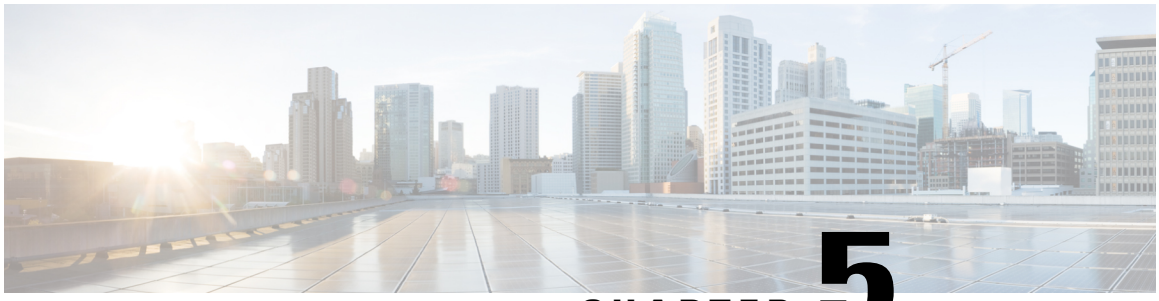
Field	Type	Null	Index	Description
ValidationStatus	smallint	No	No	The bitmask result obtained from the Validation method. See sample code that follows this table.
DBDateTime	datetime year to fraction(3)	No	No	The date and time of the database operation.

Validation Method sample code.

This is an example of a bitmask result obtained from the Validation method:

- TOD, Time of Day Error, meaning the callback was scheduled for a time of day when the queue is not open.
- EWT, Estimated Wait Time, indicates if the agent wait time for an agent is long enough to warrant a callback.

```
00000000 00000001 OK
00000000 00000010 ICM_NO_SCHEDULED_ALLOWED
00000000 00000100 ICM_NO_PREEMPTIVE_ALLOWED
00000000 00001000 NOT_IN_QUEUE
00000000 00010000 TOD
00000000 00100000 EWT
00000000 01000000 PROBE_FAILED_NO_RESPONSE
00000000 10000000 PROBE_FAILED_NO_CONFIG
00000001 00000000 EXCEED_CAPACITY_GW
00000010 00000000 EXCEED_CAPACITY_QUEUE
```



CHAPTER 5

Database Management

Cisco Unified Customer Voice Portal (Unified CVP) provides access to database maintenance tasks such as database backups and data purges through the Operations Console.

Although the Reporting Service does not directly perform database administrative and maintenance activities such as backups or purges, familiarize yourself with the database management concepts discussed in this chapter.

Note Operations such as purge and changing user passwords are executed through OAMP by the Administrator cvp_dbadmin.

Passwords

Passwords on the reporting server *must* be created and updated as part of the Unified CVP installation or by means of the Operations Console. Do not use any other means to create or update passwords.

Passwords for the Unified CVP Reporting Server are kept encrypted locally on the OAMP server and are set on the reporting server. If these passwords are not in synch (are not identical), the Operations Console will be unable to communicate with the reporting server, and restricted operations such as the purge will fail.

If you implement a password expiration policy, remember to use the Operations Console to change the Database Administrator, and Database Users passwords before the passwords expire to avoid the possibility of data loss or downtime.

Note Using the Operations Console to change passwords and to renew passwords before they expire ensures that all dependencies are synchronized.

Caution Changing passwords outside of the Operations Console can result in a failed connection between the reporting server and the database.

Reporting passwords are subject to both the Unified CVP password policy *and* the password policy enforced by the operating system of the computer on which the reporting server resides and must meet the requirements of the more restrictive policy.

Database Users

Unified CVP defines three categories of database users: database administrator, application user, and reporting user.

Database Administrator

The `cvp_dbadmin` creates, updates, and owns the database.

This user can create and delete reporting users and perform database administrative activities, such as purge and backup.

This account should not be used to run the database or to run reports against the system.

Note If this administrator's password expires, then data insertion and purge will fail, which could result in data loss.

Application User

The Unified CVP JDBC uses `cvp_dbuser` to access the Informix database. This user has the rights to connect, insert, update, and delete records in the Unified CVP database. If this user's password expires, then data insertion and purge will fail, which could result in data loss.

The User ID and password for the application user is required to access the Cisco Unified Intelligence Center data source.

Reporting User

The Unified CVP OAMP has a UI page for creating Reporting users who have read-only database access to the Unified CVP Informix reporting database.

After the Active Directory configuration for these users is enabled in the Unified IC Administration Console, they can log in to Unified IC reporting with their AD credentials.

They have the basic "Login User" user role only, until the Unified IC Security Administrator assigns additional roles and privileges to them.

Data Categories and Data Retention

Using the Operations Console, users are able to select the time of day to run database purge and to set the number of days of data to be retained by data category. During schema creation, default data retention values are specified for each data category. Note that a high-level category, such as Call, cannot have a lower retention time than a dependent category, such as Call Event.

Increased database space availability, as documented in the *Cisco Unified Customer Voice Portal Design Guide* , increase the data retention times.

Default Data Retention for Data Categories

The following data categories exist for Unified CVP. Note that a high level category, such as Call, cannot have a lower retention time than a dependent category, such as CallEvent. For each category, the default data retention times, in days, is given within parentheses.

level 1: `Call` (30)

level 2: `-Call Event` (30)

level 2: `-VoiceXML Session` (30)

level 3: `--VoiceXML Element` (15)

level 4: `---VoiceXML ECC Variable` (15)

level 4: `---VoiceXML Interact Detail` (15)

level 4: `---VoiceXML Session Variable` (15)

level 4: `---VoiceXML Element Detail` (15)

Database Purge

This section explains how to schedule purges, the difference between midday and nightly purges, and how to run emergency purges.

To allow for rapid space management, all Unified CVP Reporting Server data is kept in date-specific fragments. On a daily basis, new day fragments are created for incoming data. This allows the Unified CVP Reporting Server to quickly drop old data by dropping the disk fragment in which that data resides.

This means that new fragments must be created on a regular basis to ensure that they can be rapidly disposed of when their retention period expires. All of this is handled by the purge.

In a quiet environment, the purge can execute in less than one second. In a busy environment, the purge might take considerably longer (15-20 minutes) while the purge navigates around running processes.

To allow for situations where the purge may be unable to execute, space is allocated two days before it is needed. This allocation is triggered by the first purge to run after a date boundary has been crossed.

If the nightly purge is scheduled to run at 1:00 a.m., it will typically perform this task. If the nightly purge is scheduled to run at 11:00 p.m., then the next day will not occur until 11:00 a.m. the next morning, which may not be an optimal time to execute the process.

For this reason, schedule the nightly purge to occur after midnight.

Schedule Purges

To run database purge from the Operations Console:

Procedure

Step 1 Select **Device Management** > **CVP Reporting Server**.

Step 2 Select a reporting server by clicking on the link in its name field or by clicking the radio button preceding it and then clicking **Edit**.

Step 3 At the Edit Reporting Server Configuration window, select the Database Administration menu in the toolbar, then select **Data Delete**.

Step 4 On the Reporting Server - Data Delete page, change the data retention time for each category of data.

Step 5 Select the hours and minutes to run the purge each day. This defines the time for the primary (nightly) purge and sets the Midday purge to run 12 hours later.

Step 6 Click **Save & Deploy**.

What to Do Next

See the *Administration Guide for Cisco Unified Customer Voice Portal* at http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_user_guide_list.html for information on categories of reporting data that can be purged and the default number of days to retain data before purging.

Nightly and Midday Purges

When you schedule a purge from the Operations Console, two jobs are scheduled on the reporting server - the nightly purge and the midday purge:

- The **nightly purge** job runs at the time (hour and minute) that you define in the Operations Console. The nightly purge performs a purge if necessary (as required by a data retention value, or for an emergency purge—see the following section on emergency purges), in addition to other tasks updating the database statistics.

  If a purge *is* required and performed, the statistics are updated after the purge.

  In addition, on Sundays, the nightly purge also copies the Informix log file to a backup folder, creates a new log file and deletes the old.

  The nightly purge should be scheduled after midnight.

- The **midday purge** is automatically scheduled to run 12 hours after the nightly purge. So, for example, if you schedule a purge at 2 a.m., then the nightly purge is run at 2 a.m. and the midday purge at 2 p.m.

Midday purge also serves as a backup for the nightly purge. If the nightly purge fails to allocate new fragments for new data, this will be taken care of by the midday purge. A midday purge is not system intensive in the same way that the nightly purge is.

In the event that data volume spikes during the day and an emergency purge is required, it will be handled at midday.

Emergency Purge

If the number of days of data that you chose to retain cannot be contained within the database, then the database will initiate emergency purge of the old data to create space for new data. Emergency purge is, a critical safety mechanism for Unified CVP.

If used space has exceeded the systems threshold, the user is notified by an Simple Network Management Protocol (SNMP) trap message after the emergency purge is complete. The SNMP notification will alert the user of the loss of data and request that they reduce their retention days data settings.

Users must reduce the number of days of data retained until emergency purge is not required. Additionally, users can reduce the data generated by using data filters (for VXML Server application detail data filtering).

Best Practices for Purge

- **Data Granularity** - The CVP Reporting database houses records of calls handled by Cisco Unified Voice call servers. The amount of data captured for each call is managed by filters specified in the Operations Console. The granularity of data captured depends on these settings.
- **Data Retention** - As limited space is available to capture this data, the purge mechanism uses retention settings to govern how long data is retained.

  If there is insufficient space to retain data for the desired time frame, the oldest data is purged in one-day increments until there is sufficient space for the reporting server to remain operational.

  If more data is captured on a daily basis than can be stored, the purge mechanism will be unable to remove data because it operates only on a daily basis. If this is the case, consider installing a larger reporting server.
- **Database backup and purge** cannot run at the same time. Purge should be scheduled at least 30 minutes before a backup. These jobs, as well as on-demand backup, should be run at low call and reporting volume times. From the perspective of Unified CVP, database backups are optional, data purges are mandatory. However, from the perspective of the user, database backups should *not* be considered optional.
- **Reporting Server** - During a database purge operation, the reporting server disconnects from the database (though for no more than 10 minutes) and starts buffering messages in memory until the purge is finished. The same memory limitations apply as described in the section Reporting User.
- **Reporting users** may be disconnected from the database if they are holding locks that contend with purge. Notify reporting users not to run reports at this time.
- **Upgrades** - Turn off scheduled purge before performing an upgrade.
- **Windows Scheduled Tasks**- The database backup and purge maintenance tasks are created as Windows Scheduled Tasks, and can be viewed in the Scheduled Tasks window. (**Start** > **Programs** > **Accessories**

> **System Tools** > **Scheduled Tasks**.) Periodically, you should check the Scheduled Tasks to ensure the Last Run Time was as expected and no status messages exist.

Database Backup

Unified CVP lets users turn the scheduling of data backups on or off, and to run backups on demand. Backups are made to the reporting servers local file system. By default, scheduled backups are turned off.

Caution Unified CVP backup scheduling is an optional feature. Backup is the user's responsibility. Data loss may occur if the backing up of files is not managed properly by the user.

If Unified CVP backup scheduling is turned on, the backup occurs once per day. Backups must be scheduled to run no sooner than 30 minutes after the scheduled purge job.

Users can run a backup on demand—as long as another backup, or a purge, is not already running. Database backups are performed and stored on the local machine. Due to space limitations, a maximum of two backups and a minimum of one backup are available on the local machine. Retaining two backup files is critical. If the system fails while writing a backup, and a restore is necessary, the older backup file is required for restore.

It is a best practice to:

- Keep a given backup for at least two weeks.
- Check the integrity of the backup periodically.
- Run a backup before an upgrade.

Unified CVP uses the Informix backup utility ontape (for both backup and restore).

When a new backup launches—either scheduled, or on demand from the Operations Console—the new file is named `cvp_backup_data.gz`. The Unified CVP backup script copies the previous `cvp_backup_data.gz` backup file and renames it to `cvp_backup_data.old.gz`. This always leaves two backup files on the local system and makes it easy for Unified CVP administrators to script copy jobs to move the files. The backup script ensures that two backups cannot be launched at the same time.

Note The backup script also ensures that a backup cannot be launched if a purge is underway, and vice versa.

Storing a backup on the local machine does not protect against failure of that machine or loss of a site.

Important Cisco Systems strongly recommends that you manually or automatically create a job to copy the `cvp_backup_data.old.gz` file to a separate machine, preferably at a separate location. Again, the user is responsible for managing backup data.

Warning Only the `cvp_backup_data.old.gz` file can be copied. The `cvp_backup_data.gz` file cannot be copied. Attempting to copy the `cvp_backup_data.gz` file locks the file and prevent another backup from running.

Database backup updates the log file when the backup finishes. If the database server goes down during backup, the backup file gets corrupted.

To check if Informix is up and running and to validate the backup, execute the `cvpverifybackup.bat` file located at `%CVP_Home%\bin\cvpverifybackup.bat`.

While executing the script, you are prompted with following message:
Please put in Phys Tape 1. Type <return> or 0 to end: Press 0 and press Enter

Note Based on the size of the database, the prompt keeps changing as Tape 2, Tape 3 and so on.

This process execution takes a long time (based on the database size) to validate and the results are displayed on the console.

In Cisco Unified CVP, there is a supported script to perform a database restore.

Restoring a backup image is required when older data on a backup image needs to be recovered. It is also required when a machine is rebuilt after a hardware failure and you need to recover data.

Note Although it is possible to restore a backup image from one reporting server to another, such a restoration is not supported with the CVP restore process.

The restore process in Unified CVP is as follows:

- Stop the CallServer process (Reporting Server).
- Execute the script: `%CVP_Home%\bin\cvprestore.bat`.
- Restart the CallServer process.

Caution Using a third-party backup utility to back up the Informix database is ineffective and may be dangerous to the integrity of the reporting database. The only effective way to perform a reporting database backup is with the backup process provided by the OAMP interface.

For information on configuring backups, see the *Administration Guide for Cisco Unified Customer Voice Portal*.

Backup and Purge Retries

Occasionally, a backup or purge cannot run when scheduled. For example, if an on-demand backup is running when a purge is scheduled to run, the purge will be prevented from running.

Retries of scheduled backups or purges are performed according to the following rules.

Note There are no retries for an on-demand backup.

- A scheduled backup retries every 10 minutes, for up to 4 hours.
- A purge retries every 10 minutes, for up to 6 hours.
- At the end of 4 hours (for a backup) or 6 hours (for a purge), if the operation has not succeeded, retries stop and an SNMP alert is sent.

- If both a backup and a purge are retrying simultaneously, there is no guarantee as to which operation will run first.
- If a lock (the mechanism preventing a backup or purge from running) is more than 12 hours old, the system clears it.

Database Recovery

Unified CVP database recovery returns the database to the state of the most recent complete backup. For example, if the user schedules a backup at 01:00 and restores the database at 23:00, the same day, the restored database is in the state it was in at 01:00.

During a database restore, the database will go offline for the duration of the restore operation.

Note Data loss occurs if the reporting server is turned off and the message bus exceeds its temporary persistence capabilities.

Caution Before following a database restore, the following steps must be performed:

1. Before the restore, disable scheduled tasks (backup, purge).
2. After the restore, re-enable scheduled tasks.

Failure and Restoration

If the reporting server fails, messages destined for the reporting server are buffered by the Call Server, in memory, up to 200,000 messages. After that limit is reached, all new message detail information is dropped.

If the database connection fails, the reporting server sends out an SNMP alert and starts persisting messages to a file, up to a user-specified limit. During this time the reporting server stays *In Service*. When 75% of the specified limit is reached, a warning is written to the log file. Once 100% of the limit is reached, an SNMP alert is sent out and the reporting server goes into *Partial Service*—any new messages may be dropped.

When the database connection comes back up, the reporting server goes into recovery mode and changes its state to Partial Service if it is not in that state already. It then starts reading messages from the file and committing them to the database. Depending on the size of the file, it may take a long time (sometimes hours) to commit all of the data to the database. Any new messages that come in during recovery will be buffered in memory. There is, however, a limit to the number of messages that the reporting server can buffer. This is true regardless of the mode or state it is in. When the number of buffered messages reaches 100,000, an SNMP alert is sent out to warn the user. At 200,000 another SNMP alert is sent out and all new messages detail information is dropped—keeping only basic data like call, call event, and session information. Also at 200,000, the reporting server changes its state to *Partial Service*, if it is not already in that state. After the total number of buffered messages reaches 300,000, another SNMP alert is sent out and all new messages are dropped from that point forward.

When the number of messages in memory drops back below 50,000, an SNMP alert is sent out stating that the queue size is back to normal, and the reporting server's state goes back to *In Service*.

If, on startup, a persistent file exists, the reporting server stays in *Partial Service* and goes into recovery mode as previously described.

During a database purge operation, the reporting server disconnects from the database and starts buffering messages in memory until the purge is done. The same memory limitations as previously described apply in this case as well.

Caution **When the reporting server is in *Partial Service*, there are no guarantees that new messages will be kept and committed to the database. As many as possible will be buffered in memory, but at some point they may be dropped either partially or fully.**

CHAPTER 6

Reporting Best Practices

Note You can find troubleshooting information for Cisco Unified Customer Voice Portal (Unified CVP) Reporting on the Cisco Troubleshooting Doc Wiki site

The chapter contains the following topics:

Reporting Server Instance

The Reporting Server instance(Informix dB) must be named as **cvp**, and must not be renamed.

Allow Only Reporting Users to Query Database

In the interests of security, allow only reporting users to generate reports.

Accurate Time Stamps for Reporting and Logging

Cisco Unified Customer Voice Portal (Unified CVP) components do not themselves synchronize machine times. However, customers must provide a cross-component time synchronization mechanism, such as Network Time Protocol (NTP), to ensure accurate time stamps for reporting and logging.

CPU-Intensive Reports

Do not run CPU-intensive reports off the database while the database is receiving data.

Note Reports become more CPU intensive as the complexity associated with producing the report from the information available in the database increases. There is no sharp dividing line between intensive and non-intensive reports. The system performance must remain within the guidelines defined in the *Cisco Unified Customer Voice Portal Design Guide*.

Database Backup and Recovery

Issues to keep in mind are:

- Managing your backup strategy
- Turning off the reporting server when doing database recovery

These issues are discussed in Database Backup, on page 108 and Database Recovery, on page 110.

Database Retention Settings

Ensure that the database is sized conservatively so that it never needs to emergency purge.

Data Security

Ensure data is secure by the following practices:

- Unified CVP offers administrators the ability to choose not to persist sensitive ECC data in the database. Users define ECC variables in Cisco Unified Intelligent Contact Manager Enterprise (Unified ICME) and by default they are not persisted in the Unified CVP database.

  The Caller_input and FromExtVXML ECC variables are subject to many application-dependent uses. For security purposes, flag these two variables as not persistent in Unified ICME. If there is anything in them that must be stored, the routing script can copy the data to an applicable variable for storage in the database.

- Users can reduce the data generated by means of data filters (for VXML Server application detail data filtering. Either adding more exclusive filters, or using fewer inclusive filters, cuts down on the amount of data stored.
- Users can turn off logging of sensitive data containing caller's responses on a per-element basis. The caller's input, such as set of digits representing Social Security numbers or credit card numbers, can be set not to be logged, providing a security layer in case logs are compromised.

Database Sizing Issues

See the discussion in Data Categories and Data Retention, on page 105. Also see the *Planning Guide for Cisco Unified Customer Voice Portal* and the *Cisco Unified Customer Voice Portal Design Guide*.

Report Data Filtering Before Database Storage

Users can reduce the data generated by means of data filters. Either adding more exclusive filters, or using fewer inclusive filters, can cut down on the amount of data stored.

For information on filtering reporting data, see the http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_user_guide_list.html.

Inclusive and Exclusive VXML Filters for Reporting

You use Inclusive and Exclusive VXML filters to control the data that the VXML Server feeds to the Reporting Server. Data feed control is crucial for:

- Saving space in the reporting database.
- Preserving messaging communication bandwidth.

Inclusive and Exclusive Filter Configuration

To configure inclusive and exclusive filters for a Reporting Server:

Procedure

Step 1 Choose **Device Management** > **VXML Server**.

The Find, Add, Delete, Edit VXML Servers window opens.

Step 2 You can search for a VXML Server by using the procedure in the Finding a VXML Server topic.

Step 3 From the list of matching records, choose the VXML Server that you want to edit.

Step 4 Click **Edit**.
The VXML Server Configuration window opens to the General Tab.

Step 5 Select the **Configuration Tab**, then configure VXML Server properties.

Step 6 In the **VXML Applications Details: Filters** pane, enter an inclusive filter that defines the VXML elements to include in data sent to the Reporting Server.

Step 7 Optionally, enter an exclusive filter that excludes some of the data specified by the inclusive filter.

Step 8 When you finish configuring filters, click **Save** to save the settings in the Operations Console database or click **Save & Deploy** to save and apply the changes to the VXML Server.

Step 9 Shut down and then start the VXML Server and the primary and backup Call Servers.

Related Topics

VXML Inclusive and Exclusive Filter Rules, on page 116
VXML Filter Wildcard Matching Example, on page 117
Passwords, on page 103

VXML Inclusive and Exclusive Filter Rules

Inclusive and exclusive filters operate using the following rules:

- Filters are case sensitive.
- By default, all items but the Start, End, Subdialog_Start and Subdialog_End elements are filtered from reporting data unless they are added to an Inclusive Filter. The Subdialog_Start and Subdialog_End elements are never filtered from reporting data unless Reporting is disabled on the VXML Server.
- The Exclusive Filter takes precedence over the Inclusive Filter. For example, if an application name is in the Exclusive Filter, then all of the items of that applications are excluded from reporting data even if a particular field or element is listed in the Inclusive filter.
- The syntax for Inclusive/Exclusive filters is:

  `Appname.ElementType.ElementName.FieldName`

  or

  `AppName.*.*.SESSION:Varname`

  Note This syntax is used to indicate session variables.

- A semicolon (;) should be used to separate each item in a filter. For example, `ElementA ; ElementB` is valid.
- A wildcard (*) can be specified anywhere within the application name, element type, element name, or field name.

- Element types, element names, and field names can contain alphanumeric characters, underscores, and a space character.
- An application name can contain alphanumeric characters and underscores, but the space character is not allowed. For example, `A_aa.B_bb.*C_cc_DD.E_ee_F*` is valid.

VXML Filter Wildcard Matching Example

The following table provides examples of VXML filter wildcard matching.

Table 42: Examples of VXML Filter Wildcard Matching

Filter	What It Matches
MyApplication.voice.*.*	Matches all voice elements in MyApplication
.voice..*	Matches all Voice elements in all applications.
MyApplication.*.*.var*	Matches all fields in MyApplication that start with with the string `var`.
MyApplication.*.*.*3	Matches all fields in MyApplication that end with `3`.
MyApplication.*.*.SESSION:Company	Matches the Company session variable in MyApplication.

Inclusive and Exclusive VXML Filters for Reporting Example

The following table provides examples of some different combinations of Inclusive and Exclusive filters and the resulting data that the VXML Server feeds to the Reporting Server.

Table 43: Examples of Inclusive and Exclusive VXML Filters for Reporting

Inclusive Filter	Exclusive Filter	Data the VXML Server Feeds To the Reporting Server
`Application1.*.*.*`	None	All Application1 data
`Application1.*.*.*`	`*.*.Element1.*;` `*.*.Element2.*`	All Application1 data, except Element1 and Element2

`Application1.*.*.*`	`*.*.Element1.*;`
	`*.*.Element2.*;`
	`*.*.*.Field1`
		All Application1 data, except Element1, Element2, and Field1

Inclusive Filter	Exclusive Filter	Data the VXML Server Feeds To the Reporting Server
`Application1.*.*.*`	`*.voice.*.*` which matches Element3 and Element4	All Application1 data, except Element3 and Element4

`*.*.Element1.*;`
`*.*.Element2.*;`
`*.*.*.Field1`
	`Application1.*.*.*`	No data for Application1. Other data for other applications, such as Application2, which contain Element1, Element2 and Field1, will be fed.

`*.voice.*.*` which matches Element1, Element2, Element3, and Element4	`*.*.Element3.*;` `*.*.Element4.*`	Only Element1 and Element2 and all applications.
`*.voice.*.*` which matches Element1 and Element2	`*.*.*.Field1`	Element1 and Element2, except for Field1, if it exists in those elements
`*.*.Element1.*`	None	Element1
`*.*.Element1.*`	`*.*.*.Field1`	Element1, except for Field1 if it exists in Element1
`*.*.*.Field1`	`*.*.Element3.*;` `*.*.Element4.*`	Field1 in any elements except Element3 and Element4

A good strategy for using filters is to create an Inclusive filter that includes the data you want to save in the Reporting database and then create an Exclusive filter to *exclude* portions of the data, for example, sensitive security information such as Social Security Numbers. For example, you would:

- First, create an inclusive filter to include all information:

  `MyApp.voice.*.*`

- Then, create an exclusive filter to remove credit card and social security numbers information:

  `MyApp.voice.*.CreditCard; MyApp.voice.*.SSN`

Informix, Operating System Time, and Local Time

Informix displays a datetime that corresponds to the same time zone as the Informix server operating system's time zone, represented in Universal Time Coordinated (UTC).

- If you wish a datetime to be displayed for a time zone other than that of the Informix server operating system, you must use reporting tools or SQL tools (for example, Java).
- If, in your system, you are using more than one Informix server for Unified CVP reporting, it is best if all such server operating systems are set to the same time zone. This helps avoid confusion.

Cisco Unified Customer Voice Portal and SQL Server Data Joining

To join data from a SQL server database and an Informix Database you must use a reporting tool that supports the ability to join data from two heterogeneous databases.

Reporting Password Policy Adherence

Reporting passwords are subject to both the Unified CVP password policy *and* the password policy enforced by the operating system of the computer on which the reporting server resides. For each aspect of the password, the Reporting password *must* meet the requirement of the more restrictive policy.

Related Topics

Passwords, on page 103

Purge and Backup Database Maintenance Tasks

The database backup and purge maintenance tasks are created as Windows Scheduled Tasks, and can be viewed in the Scheduled Tasks window (**Start** > **Programs** > **Accessories** > **System Tools** > **Scheduled Tasks**). These jobs log in as SYSTEM.

If the CVPDBNightlyPurge and CVPDBMidDayPurge tasks do not run, then the database will not be purged and will eventually become full, resulting in data loss.

If the CVPDBBackup task does not run the database will not be backed up.

Periodically, you should check the Scheduled Tasks to ensure the Last Run Time was as expected and there are no status messages.

Reporting Isolation Level

Reporting clients should never run with an isolation level of repeatable read because this could hold locks and prevent updates to the data.

Timestamp Synchronization

Call Servers, VXML Servers, and reporting servers must have their clocks synchronized to assure accurate timestamps in both the database and log files.

Since Unified CVP components do not themselves synchronize machine times, you must deploy a cross-component time synchronization mechanism, such as NTP.

Writing Efficient SQL Queries When Creating Reports

Keep these guidelines in mind:

- When writing SQL, developers must organize their WHERE clauses and put the most important join first. The most important join is the one that will reduce the size of the dataset to the least amount of rows.
- Write reports so that every field in the WHERE and ORDER BY clauses uses an indexed field.
- A subset of the data that satisfies any given query can protect the user and the database from generating massive data results. Including the word FIRST in Select statements will return only the amount of data requested. For example, `SELECT FIRST 1000 * FROM Call`.
- The second column in a composite index should never be used in a JOIN statement without the first column.
- Engineers writing database code should treat database, table, and column names as case sensitive--even though the current database is case insensitive--to ensure that the application is portable.
- Many operations hold database locks; therefore, reports should use a wait time of 30 seconds, if possible.
- It is possible to capture gigabytes of Unified CVP data in a single day. Any query against the database should target time ranges and subsets of data that will return in a reasonable time. Datetime columns are crucial selections. Sorting or grouping large quantities of data may exceed the capacity of the reporting server database as delivered.
- All sessions that connect to the reporting database should initiate with two statements:

  ```
  SET ISOLATION DIRTY READ; SET LOCK MORE TO WAIT 30.
  ```

  This prevents reporting queries from interfering with CallServer message persistence and improves the performance of reporting queries.

- 

  Warning Do not ever set isolation level to repeatable read.

- 

  Warning Do not ever write a SQL statement that selects into temp without specifying the 'no log' option.

- The internal ID generator limits the amount of total VXML subsystems to 8,000 per deployment.

Zero Duration Calls and Writing Reports

On occasion, messages are dropped. even for an otherwise successful call. In such cases, EndDateTime is set to the same value as StartDateTime. Thus, if a call appears to be of zero duration, report writers will know to exclude such a call from consideration in cases where it would otherwise skew metrics.

INDEX

A

B

C

D

E

F

G

I

L

M

O

P

Q

R

S

T

U

V

Z