



## **Reporting Guide for Cisco Unified Customer Voice Portal**

Release 8.5(1)

*July 2011*

**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: 408 527-0833



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

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

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

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

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

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at <http://www.cisco.com/go/trademarks>

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0711R)

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

Copyright 2011 Cisco Systems, Inc. All rights reserved.

---

## Table of Contents

Preface .....	1
Purpose .....	1
Audience .....	1
Organization .....	1
Related Documentation .....	2
Conventions.....	3
Obtaining Documentation and Submitting a Service Request.....	3
Documentation Feedback.....	4
1. Introduction to the Reporting Server.....	5
The Reporting Service.....	5
Reporting Architecture.....	6
Deployment Options and Sizing.....	7
Installation and Upgrade.....	7
Configuration.....	8
Database Maintenance.....	8
2. Using the Unified Intelligence Center Reporting Application.....	9
Install and Configure Unified IC.....	10
Acquire the License File from the License Website.....	11
Sign in to the Administration Console.....	12
Upload the License.....	12
Create Unified CVP Reporting Users.....	13
Creating Super Users.....	13
Configuring the Active Directory Server for LDAP Users.....	14
Sign in to the Unified IC Reporting Interface.....	14
Add User Roles for Unified CVP Reporting Users.....	15
Create a Data Source for Unified CVP Report Data.....	15
Obtain the Import Unified CVP Report Templates.....	17
Import Unified CVP Report Templates into Unified IC and Set the Data Source.....	18
Working with the Imported Unified CVP Report Templates.....	19
Unified IC Reporting: Basic Concepts.....	19
Working with Reports.....	21
Running Reports.....	21
Filtering Reports.....	22
The Report Viewer.....	26
Editing Reports.....	27
Changing and Moving Report Fields .....	29
Exporting Reports.....	31
Unified IC Reporting: Unified IC User Roles.....	32
Other Roles.....	33
3. Unified CVP Templates for Unified Intelligence Center.....	35
Application Summary Reports (15, Daily, and Weekly).....	35
Call Report.....	38
Call Detail Report.....	41
Call Traffic Reports (15, Daily, and Weekly).....	47
Call Traffic Charts.....	49
Callback Reports (Current and Historical).....	50
Trunk Group Utilization Report.....	51

4. Database Schema.....	53
About the Database Schema.....	53
Data Model Diagram.....	54
Keys.....	56
The Unified CVP Reporting Data Model.....	57
DateTime Columns.....	57
Informix Dates and Times.....	57
SIP Calls.....	59
Trunk Utilization.....	60
Unified CVP Database Tables.....	61
Call Tables.....	61
Call Table.....	61
CallEvent Table.....	63
CallICMInfo Table.....	64
OutgoingECCVariable Table.....	66
VXML Tables.....	67
VXMLCustomContent Table.....	68
VXMLElement Table.....	68
VXMLElementDetail Table.....	69
VXMLElementFlag Table.....	70
VXMLError Table.....	71
VXMLHotEvent Table.....	71
VXMLHotLink Table.....	72
VXMLSession Table.....	73
VXMLSessionVariable Table.....	74
VXMLVoiceInteractDetail Table.....	75
Summary / Aggregate Tables.....	75
ApplicationSummary_15 Table.....	77
ApplicationSummary_Daily Table.....	78
ApplicationSummary_Weekly Table.....	78
Call_15 Table.....	79
Call_Daily Table.....	79
Call_Weekly Table.....	80
Usage Table.....	81
Lookup and Reference Tables.....	81
ActionTypeRef Table.....	82
CallTypeRef Table.....	82
CauseRef Table.....	83
Device Table.....	86
ElementtypeRef Table.....	86
EventTypeRef Table.....	87
QueueRef Table.....	88
Resource Table.....	88
ResultRef Table.....	89
SubSystemTypeRef Table.....	89
TransferTypeRef Table.....	90
UserInputModeRef Table.....	90
VarDataTypeRef Table.....	90
VoiceActionTypeRef Table.....	91
Courtesy CallBack Tables.....	91
CallBack Table.....	92

CallBackEvent Table.....	94
CallBackQueue Table.....	94
5. Managing the Database.....	97
Passwords.....	97
Database Users.....	98
Instance Owner.....	98
Database Administrator.....	98
Application User.....	98
CVP Reporting Users.....	99
Data Categories and Data Retention.....	99
Default Data Retention for Data Categories.....	99
Database Purge.....	100
Scheduling Purges.....	100
Nightly Purge and Midday Purge.....	101
Emergency Purge.....	101
Best Practices for Purge.....	102
Database Backup.....	102
Database Compression.....	104
Compression Ratio.....	105
Retries: Backup and Purge.....	105
Database Recovery.....	106
Failure and Restoration.....	106
6. Reporting Best Practices.....	109
Allow Only Reporting Users to Query the Database.....	109
Assuring Accurate Time Stamps for Reporting and Logging.....	110
CPU Intensive Reports.....	110
Database Backup and Recovery.....	110
Database Retention Settings.....	110
Data Security.....	110
Database Sizing Issues.....	111
Filtering Data to be Stored in the Database.....	111
Inclusive and Exclusive VXML Filters for Reporting.....	111
Procedure.....	111
Rules for VXML Inclusive and Exclusive Filters.....	112
Examples of VXML Filter Wildcard Matching.....	113
Examples of Inclusive and Exclusive VXML Filters for Reporting.....	113
Informix, Operating System Time, and Local Time.....	114
Joining Unified CVP and SQL Server Data.....	115
Password Policy.....	115
Purge and Backup Database Maintenance Tasks.....	115
Reporting Isolation Level.....	115
Synchronizing Timestamps.....	115
Writing Efficient SQL Queries when Creating Reports.....	116
Zero Duration Calls and Writing Reports.....	117
Index .....	119

---

## List of Figures

Figure 1: CVP Architecture.....	6
Figure 2: Unified IC Cluster.....	10
Figure 3: License Management .....	12
Figure 4: User List Page.....	15
Figure 5: Data Sources page.....	16
Figure 6: Filters.....	22
Figure 7: Application Summary 15 Report, No Advanced Filter.....	24
Figure 8: Setting an Advanced Filter.....	24
Figure 9: Application Summary 15 Report, Filtered.....	24
Figure 10: Filter Tab for Stored Procedure Reports 1 of 2.....	25
Figure 11: Filter Tab for Stored Procedure Reports 2 of 3.....	25
Figure 12: Filter Tab for Stored Procedure Reports 3 of 3.....	26
Figure 13: Report Viewer.....	26
Figure 14: Report Editor.....	28
Figure 15: Report Export.....	32
Figure 16: Application Summary 15.....	37
Figure 17: Application Summary Daily.....	37
Figure 18: Application Summary Weekly.....	37
Figure 19: Basic Filters with Absolute Dates.....	38
Figure 20: Call Report 1 of 2.....	40
Figure 21: Call Report 2 of 2.....	40
Figure 22: Call Detail Report 1 of 6.....	45
Figure 23: Call Detail Report 2 of 6.....	45
Figure 24: Call Detail Report 3 of 6.....	46
Figure 25: Call Detail Report 4 of 6.....	46
Figure 26: Call Detail Report 5 of 6.....	46
Figure 27: Call Detail Report 6 of 6.....	46
Figure 28: Call Traffic 15 Report.....	48
Figure 29: Call Traffic Daily Report.....	49
Figure 30: Call Traffic Weekly Report.....	49
Figure 31: Selecting a Chart View.....	49
Figure 32: Call Traffic Daily Chart.....	50
Figure 33: Callback (Pending) Report.....	51

---

Figure 34: Callback Historical Report.....	51
Figure 35: Trunk Group Utilization Report.....	52
Figure 36: Call Flow.....	54
Figure 37: Call Tables.....	55
Figure 38: Callback Tables.....	55
Figure 39: Summary Tables.....	56
Figure 40: Trunk Group Utilization Tables.....	56







## 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 IC) reporting application.
- The templates that are installed by Unified CVP for import into Unified IC.
- The reporting database schema.
- Reporting best practices.

### Audience

This guide assists Call Center managers, Unified CVP system managers, Cisco Unified Contact Center Enterprise system managers, VoIP technical experts, and IVR application developers.

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

### Organization

This guide is divided into the following chapters:

## Related Documentation

Chapter	Description
<a href="#">Chapter 1, Introduction to the Reporting Server (page 5)</a>	Provides an introductory discussion of the reporting server and points to documents that detail its sizing, installation, and configuration.
<a href="#">Chapter 2, Using the Unified Intelligence Center Reporting Application (page 9)</a>	Explains how to work with Unified CVP templates in the Unified IC reporting product.
<a href="#">Chapter 3, Unified CVP Templates for Unified Intelligence Center (page 35)</a>	Presents the Unified CVP templates that can be imported into—and generated from—Unified IC and how they are populated from the reporting database.
<a href="#">Chapter 4, Database Schema (page 53)</a>	Documents the reporting database schema.
<a href="#">Chapter 5, Managing the Database (page 97)</a>	Discusses concepts to keep in mind while managing the database, such as data retention, backup, and purge.
<a href="#">Chapter 6, "Reporting Best Practices" (page 109)</a>	Provides a list of best practices.

**Troubleshooting** - Point your browser to [www.docwiki.cisco.com](http://docwiki.cisco.com/wiki/Main_Page) ([http://docwiki.cisco.com/wiki/Main\\_Page](http://docwiki.cisco.com/wiki/Main_Page)) to read troubleshooting information for Unified CVP and for 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 IC tips, click the icon for Cisco Products. Then look under the "U"s.

## Related Documentation

Planning your Unified CVP solution is an important part of the process in setting up Unified CVP. Cisco recommends that you read the [Cisco Unified Customer Voice Portal \(CVP\) Solution Reference Network Design Guide \(SRND\)](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_implementation_design_guides_list.html) ([http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products\\_implementation\\_design\\_guides\\_list.html](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_implementation_design_guides_list.html)) before configuring your Unified CVP solution.

**Cisco.com** - You can find the complete documentation set for Unified CVP on [Cisco.com](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/tsd_products_support_series_home.html) ([http://www.cisco.com/en/US/products/sw/custcosw/ps1006/tsd\\_products\\_support\\_series\\_home.html](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/tsd_products_support_series_home.html)).

The documentation set includes the:

- Installation and Upgrade Guide
- Configuration and Administration Guide
- Planning Guide

The documentation set for Cisco Unified Intelligence Center is [at this location](http://www.cisco.com/en/US/products/ps9755/tsd_products_support_series_home.html) ([http://www.cisco.com/en/US/products/ps9755/tsd\\_products\\_support\\_series\\_home.html](http://www.cisco.com/en/US/products/ps9755/tsd_products_support_series_home.html)).

## Conventions

This manual uses the following conventions:

Convention	Description
<b>boldface font</b>	<p>Boldface font is used to indicate commands, such as user entries, keys, buttons, and folder and submenu names. For example:</p> <ul style="list-style-type: none"> <li>Choose <b>Edit &gt; Find</b>.</li> <li>Click <b>Finish</b>.</li> </ul>
<i>italic font</i>	<p>Italic font is used to indicate the following:</p> <ul style="list-style-type: none"> <li>To introduce a new term. Example: A <i>skill group</i> is a collection of agents who share similar skills.</li> <li>For emphasis. Example: <i>Do not</i> use the numerical naming convention.</li> <li>A syntax value that the user must replace. Example: IF (<i>condition, true-value, false-value</i>)</li> <li>A book title. Example: See the <i>Cisco CRS Installation Guide</i>.</li> </ul>
<b>window font</b>	<p>Window font, such as Courier, is used for the following:</p> <ul style="list-style-type: none"> <li>Text as it appears in code or that the window displays. Example: <code>&lt;html&gt;&lt;title&gt;Cisco Systems, Inc. &lt;/title&gt;&lt;/html&gt;</code></li> </ul>
<b>&lt; &gt;</b>	<p>Angle brackets are used to indicate the following:</p> <ul style="list-style-type: none"> <li>For arguments where the context does not allow italic, such as ASCII output.</li> <li>A character string that the user enters but that does not appear on the window such as a password.</li> </ul>

## Obtaining Documentation and Submitting a Service Request

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 the *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 email to the following address:

[mailto:ccbu\\_docfeedback@cisco.com](mailto:ccbu_docfeedback@cisco.com)

We appreciate your comments.



# Chapter 1

## Introduction to the Reporting Server

---

This chapter contains the following topics:

- [The Reporting Service, page 5](#)
- [Reporting Architecture, page 6](#)
- [Deployment Options and Sizing, page 7](#)
- [Installation and Upgrade, page 7](#)
- [Configuration, page 8](#)
- [Database Maintenance, page 8](#)

### The Reporting Service

Reporting is an optional component for Unified 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 IVR Service, the SIP Service (if used), and the 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 call activity summary information, which assists call center managers in reviewing and managing daily operations. It can also include operational detail data for various IVR applications. The Unified CVP report templates that display this data are documented in Chapter 3.

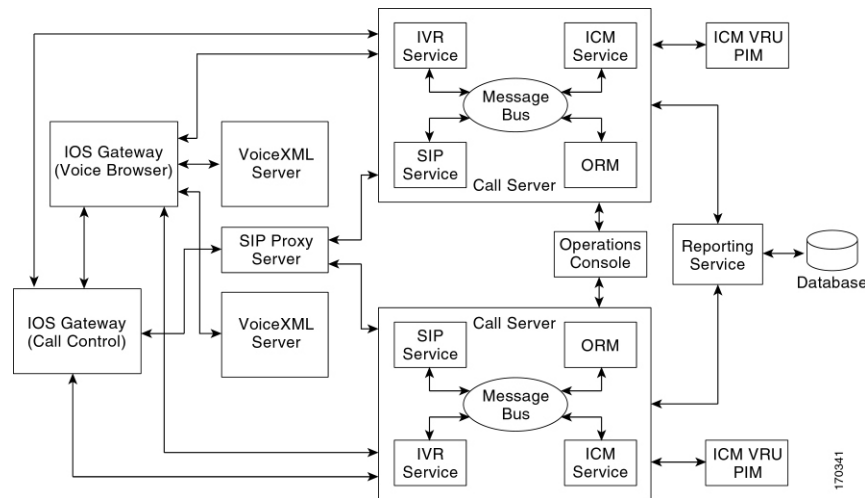
The Unified CVP reporting data model is documented in [Chapter 4 \(page 53\)](#).

## 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. 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 resides, for example, the IVR Service, the SIP Service, and the ICM Service. 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 JVM with the message bus system.

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

The reporting service then parses those messages, converts them into batches of applicable SQL statements, and executes them into a SQL database using the Java Database Connectivity (JDBC) 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, querying statistics, and so forth. As the diagram 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 when the database comes back on-line. The amount of time that messages can be buffered depends on the system throughput. See [Failure and Restoration \(page 99\)](#).

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 reporting application. You can import these into Unified IC and run them from the Unified IC interface. See Chapter 2.

## 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 \(CVP\) Solution Reference Network Design Guide \(SRND\)](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_implementation_design_guides_list.html) ([http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products\\_implementation\\_design\\_guides\\_list.html](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_implementation_design_guides_list.html))

## 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](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/prod_installation_guides_list.html) ([http://www.cisco.com/en/US/products/sw/custcosw/ps1006/prod\\_installation\\_guides\\_list.html](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/prod_installation_guides_list.html)).

Topics in the Installation and Upgrade guide include:

- Installing the reporting component
- Specifying the reporting password
- Running the reporting batch file after the installation
- 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
- Backup and purge of the reporting database

## Configuration

You can find explanations and procedures regarding the configuration and maintenance of the Unified CVP Reporting server in the *Operations Console Online Help for Cisco Unified Customer Voice Portal* and in the [Configuration and Administration Guide for Cisco Unified Customer Voice Portal](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_installation_and_configuration_guides_list.html) ([http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products\\_installation\\_and\\_configuration\\_guides\\_list.html](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_installation_and_configuration_guides_list.html))

Topics in the operations console help and in the Configuration and Administration 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 VXML Server for Reporting (adding and editing)
- 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.

Refer to [Chapter 5 \(page 97\)](#) for high-level information on database maintenance. Refer to the [Configuration and Administration Guide for Cisco Unified Customer Voice Portal](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_installation_and_configuration_guides_list.html) ([http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products\\_installation\\_and\\_configuration\\_guides\\_list.html](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_installation_and_configuration_guides_list.html)) for details on database operations.





## Chapter 2

# Using the Unified Intelligence Center Reporting Application

---

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

These templates are documented in Chapter 3.

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 IC reporting application.

Use the Cisco Unified Intelligence Center (Unified IC) 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 IC, you can also modify the SQL query for the report definition and design custom reports and report definitions using Save As.

This section explains how to set up the Unified IC application to run Unified CVP report templates. It includes an overview of working with reports in Unified IC. You can find full details on installing, configuring, and operating Unified IC in the [Cisco Unified Intelligence Center user documentation section of Cisco.com](http://www.cisco.com/en/US/products/ps9755/products_user_guide_list.html) ([http://www.cisco.com/en/US/products/ps9755/products\\_user\\_guide\\_list.html](http://www.cisco.com/en/US/products/ps9755/products_user_guide_list.html)).

This chapter contains the following topics:

- [Install and Configure Unified IC, page 10](#)
- [Acquire the License File from the License Website, page 11](#)
- [Sign in to the Administration Console, page 12](#)
- [Upload the License, page 12](#)
- [Create Unified CVP Reporting Users, page 13](#)

## Install and Configure Unified IC

- [Sign in to the Unified IC Reporting Interface, page 14](#)
- [Add User Roles for Unified CVP Reporting Users, page 15](#)
- [Create a Data Source for Unified CVP Report Data, page 15](#)
- [Obtain the Import Unified CVP Report Templates, page 17](#)
- [Import Unified CVP Report Templates into Unified IC and Set the Data Source, page 18](#)
- [Working with the Imported Unified CVP Report Templates, page 19](#)
- [Unified IC Reporting: Basic Concepts, page 19](#)
- [Working with Reports, page 21](#)
- [Unified IC Reporting: Unified IC User Roles, page 32](#)

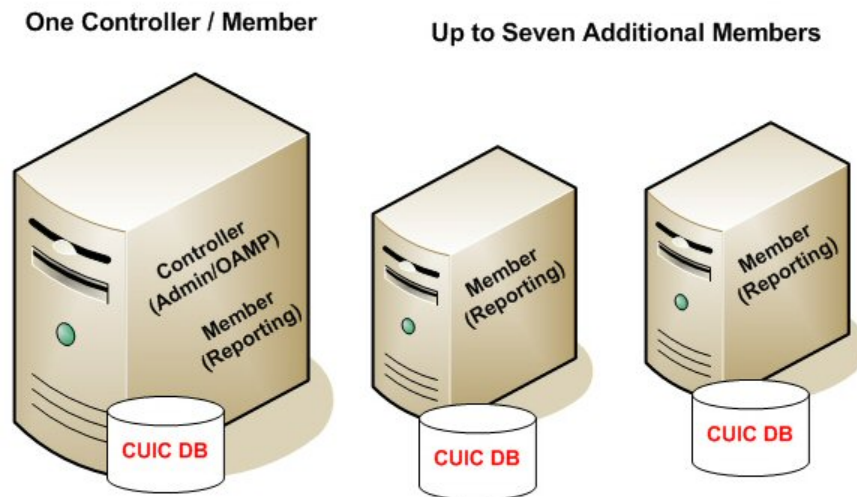
## Install and Configure Unified IC

**Who can do this:** any user in your organization.

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

Unified IC can be installed as a standalone server or as a cluster of a maximum of eight server nodes. There is 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 only.

Figure 2: Unified IC Cluster



There are two Unified IC web interfaces:

- The **Administration** application

This is the Operations Administration Maintenance Provisioning (OAMP) interface that provides application-level configuration for Unified IC. This interface is offered on the Controller node only and is used by Unified IC *Super Users*.

- The Unified IC **Reporting** application - This 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](http://www.cisco.com/en/US/products/ps9755/tsd_products_support_series_home.html) ([http://www.cisco.com/en/US/products/ps9755/tsd\\_products\\_support\\_series\\_home.html](http://www.cisco.com/en/US/products/ps9755/tsd_products_support_series_home.html))

Next step: [Acquire the License File from the License Website \(page 11\)](#)

## Acquire the License File from the License Website

**Who can do this:** any user in your organization.

After installing the Unified IC Controller node, and before you install any Member nodes, you must contact Cisco Systems, Inc. to request a Unified IC license.

---

**Step 1** To acquire the license file, go to the Cisco Product License Registration website at this URL: <https://tools.cisco.com/SWIFT/Licensing/PrivateRegistrationServlet>

**Step 2** If you do not have a PAK, click the [available licenses link](https://tools.cisco.com/SWIFT/Licensing/PrivateRegistrationServlet?DemoKeys=Y) (<https://tools.cisco.com/SWIFT/Licensing/PrivateRegistrationServlet?DemoKeys=Y>).

**Step 3** Scroll to Voice Products and click **Cisco Unified Intelligence Center - v8.0**.

**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 do not have time to write it down, you can find it through the Command Line Interface, once the install is complete, as follows:

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 pages.

You will receive an email from Cisco Systems 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 \(page 34\)](#) can access it.

**Note:** As a best practice, 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.

---

There are four license types: Demo, Lab/Trial, Standard, and Premium. See the *User Guide for Cisco Unified Intelligence Center Reporting Application* for an explanation of each license type.

Next step: [Sign in to the Administration Console \(page 12\)](#)

## Sign in to the Administration Console

**Who can do this:** The [System Application User \(page 34\)](#) who is the default [Super User \(page 33\)](#).

To upload the license, you must sign in to the Unified IC Administration Console. This is the OAMP interface for Unified IC. 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 \(page 34\)](#) during the installation. This user is the initial Super User for Unified IC Administration.

To sign in:

1. Enter this URL: **http://<HOST ADDRESS>/oamp** where HOST ADDRESS is the IP Address or Hostname of your Controller node.
2. Enter the System Application User ID and password that you defined during installation.

Next step: [Upload the License \(page 12\)](#).

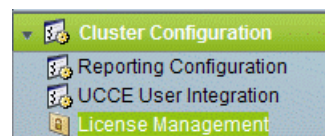
## Upload the License

**Who can do this:** The [System Application User \(page 34\)](#) who is the default [Super User \(page 33\)](#).

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

- 
- Step 1** Navigate to **Cluster Configuration > License Management** in the Administration application.

*Figure 3: License Management*



This opens the License File Management page.

- Step 2** Click **Upload license file (Browse)**.
- Step 3** Navigate to the location where the \*.lic file was saved.
- Step 4** Click **Apply License** to load the license.

A message displays 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 will occur within a minute.

---

Next step: [Creating Unified CVP Reporting Users \(page 13\)](#).

## Create Unified CVP Reporting Users

### Who can do this:

- Initially, the [System Application User \(page 34\)](#) who is the default [Super User \(page 33\)](#).
- Eventually, any [Super User \(page 33\)](#).

Unified CVP reporting users can sign in to Unified IC only if they exist in the Administration console as Super Users or if Active Directory has been configured in the Unified IC Administration console for their domain.

- Super Users who are added are considered to be IMS users. See [Creating Super Users](#).
- Users authenticated through Active Directory are considered to be LDAP users. See [Configuring the Active Directory Server for LDAP Users](#).

Both IMS users and LDAP users can log in to Unified IC Reporting and are restricted to the limited Login User role until the Unified IC Reporting security administrator gives them additional roles and tags them as Active users.

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

## Creating Super Users

To create Super Users, navigate to **Admin User Management drawer > Admin User Management** to open the Users page. Then click **Add New** to add and configure a new user or click an existing User Name to edit the configuration for that user.

This page has three tabs - **General**, **Credentials**, and **Policy**.

For information on completing these tabs, refer to the *Administration Console User Guide for Cisco Unified Intelligence Center* or to the Administration console online help.

## Configuring the Active Directory Server for LDAP Users

To configure the Active Directory tab in the Administration console so that Unified CVP reporting users can log in to the Unified IC reporting application with the User Name and password that is defined in their domain.

### Who can do this:

- Initially, the [System Application User \(page 34\)](#) who is the default Super User.
- Any [Super User \(page 33\)](#).

- 
- |               |                                                                                                                                               |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | In the Administration application, navigate to <b>Cluster Configuration &gt; Reporting Configuration</b> and select the Active Directory tab. |
| <b>Step 2</b> | Complete all fields on this page, referring to the online help for guidance.                                                                  |
| <b>Step 3</b> | Click <b>Test Connection</b> .                                                                                                                |
| <b>Step 4</b> | When the connection is confirmed, click <b>Save</b> .                                                                                         |
- 

Next step: [Sign in to the Unified IC Reporting Interface \(page 14\)](#)

## Sign in to the Unified IC Reporting Interface

### Who can do this:

- Initially, the [System Application User \(page 34\)](#) who is the default Super User.
- Any Unified CVP User created in the Administration Console as an IMS Super User or an LDAP user. See [Creating Unified CVP Reporting Users](#).

To sign in to the Unified IC Reporting interface:

1. From the home page of the Administration interface, click the Control Center drawer.
2. Locate and double-click the name of the member node you want to access. This opens the Cisco Unified Intelligence Center login page for that member.
3. Enter your User ID and password.

The Overview page displays.

Next step: [Add User Roles for Unified CVP Reporting Users \(page 15\)](#).

## Add User Roles for Unified CVP Reporting Users

### Who can do this:

- Initially, the [System Application User \(page 34\)](#) who has full permissions in Unified IC Reporting
- Eventually, any [Security Administrator \(page 32\)](#).

Once Unified CVP users log in to Unified IC, they are added to the Unified IC database (CUIC) and appear on the User List.

They are initially defined as Login Users: the lowest level User Role of Unified IC. A Unified IC Security Admin User must access the User List page to check User is Active and to grant additional User Roles to them.

Figure 4: User List Page

Overview Security User List

User List > Create

Required fields

General Information Parent Groups

User Name CVP User

Alias

☒ User is active

First Name Sam

Last Name Lee

Organization My Company

Email slee@mycompany.com

Phone 555-123-4455

Description CVP Reporting user (At most 255 characters)

Time Zone US/Hawaii

Roles

☒ Login User ☐ System Configuration Administrator ☐ Security Administrator ☒ Dashboard

☒ Report Designer ☐ Report Definition Designer ☐ Value List Collection Designer

Next step: [Create a Data Source for Unified CVP Report Data \(page 15\)](#).

## Create a Data Source for Unified CVP Report Data

### Who can do this:

- Initially, the [System Application User \(page 34\)](#) who has full permissions in Unified IC Reporting
- Eventually, any [System Configuration Administrator \(page 32\)](#).



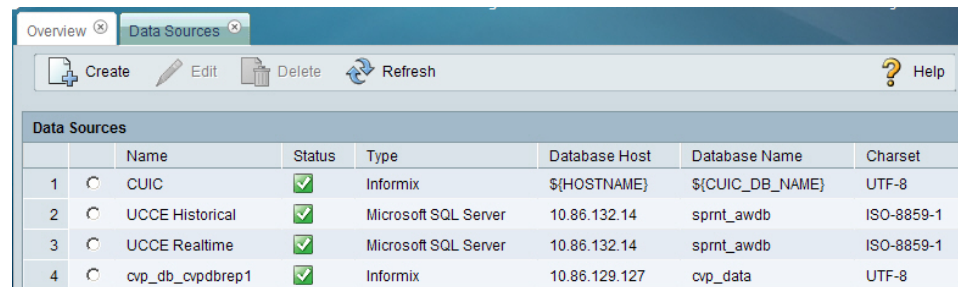
## Add User Roles for Unified CVP Reporting Users

Similar to creating an ODBC connection, this task is necessary to access the Unified CVP reporting data.

In Unified IC, this task must be performed by an user with the System Configuration Administrator User Role.

**Step 1** Select the Data Sources drawer to open the Data Sources page.

Figure 5: Data Sources page



	Name	Status	Type	Database Host	Database Name	Charset
1	CUIC	✓	Informix	\${HOSTNAME}	\${CUIC_DB_NAME}	UTF-8
2	UCCE Historical	✓	Microsoft SQL Server	10.86.132.14	sprnt_awdb	ISO-8859-1
3	UCCE Realtime	✓	Microsoft SQL Server	10.86.132.14	sprnt_awdb	ISO-8859-1
4	cvp_db_cvpdbrep1	✓	Informix	10.86.129.127	cvp_data	UTF-8

This page shows three Data Sources that are installed by Unified IC. They are the CUIC data source, the UCCE Historical Data Source, and the UCCE Realtime data source.

- The CUIC data source represents the database on the node that stores records for reports, dashboards, and users maintained on that node. This data is replicated across all nodes in the cluster.
- The two UCCE data sources represent the databases that populated the stock Unified CCE reports.

Step 2 explains how to add a data sources for Unified CVP.

**Step 2** Click **Create** to open an Add Data Source window.

**Step 3** 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. As a Best Practice, to benefit those users, give a new Data Source a meaningful name.
Description	Enter a description for this data source.
Type	Options are Informix and SQL Server.  Select Informix.  <b>Type is disabled in Edit mode.</b>



Field	Value
Database Host	Enter the IP address or DNS name for the server and append the port for the Informix database.
Port	Enter the port. Typically, the port is 1526.
Database name	Enter the name of the reporting database on the Unified CVP reporting server.  The database name can be <b>cvp_db</b> or <b>callback</b> . Use <b>callback</b> for the stock callback reports.
Instance	Specify the instance name of the database you are connecting to.
Timezone	Select the correct timezone for the data stored in the database. In locations that change from Eastern Standard Time to Daylight Savings Time, this timezone is updated automatically.
Database User ID	Enter the User ID for the Reporting User who is configured in the Operations Console to access the Unified CVP reporting database.
Password and Confirm Password	Enter and confirm the password for the database user.
Charset	Select UTF-8 for the Unified CVP Informix database.  If this is not set correctly, Unified IC will be unable to connect.
Default Permissions	View or edit the permissions for this datasource for My Group and for the All Users group. See .

**Step 4** Click **Test Connection**.

If the status is not Online, review the error message to determine the cause and edit the data source accordingly. Repeat Step 3 until the pages indicates that the data source is Online.

**Step 5** Click **Save** to close the Add Data Source window.

You return to the Data Sources window. The new data source appears on the list.

---

Next step: [Obtain the Import Unified CVP Report Templates \(page 17\)](#).

## Obtain the Import Unified CVP Report Templates

**Who can do this:** any user in your organization.

The Unified CVP reporting template .xml files are installed with Unified CVP. You need to locate them and copy them to a Unified IC client workstation.

## Add User Roles for Unified CVP Reporting Users

- 
- Step 1** In the Unified CVP server, locate the Unified CVP template files. These are XML files that reside on the reporting server in %CVP\_HOME%\CVP\_Reporting\_Templates. You can also find them on the Installation DVD in \Downloads and Samples\Reporting Templates.
- Step 2** Select the files and copy them to the client computer from which you will launch the Unified IC Reporting web application.
- 

Next step: [Import Unified CVP Report Templates into Unified IC and Set the Data Source \(page 18\)](#).

## Import Unified CVP Report Templates into Unified IC and Set the Data Source

**Who can do this:**

- Initially, the [System Application User \(page 34\)](#) who has full permissions in Unified IC Reporting
- Eventually, any [Report Designer \(page 33\)](#).

Before reporting users can run the Unified CVP report templates in the Unified IC 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.

---

- Step 1** Launch the Unified IC web application using the URL `http://<HOST ADDRESS>/cuic`, where HOST ADDRESS is the IP Address or Hostname of your Member node.
- This opens the Log In window.
- Step 2** Enter your User Name and Password.
- This opens the Overview page.
- Step 3** Click the **Reports** drawer to open the Available Reports page.
- Step 4** Right-click the topmost Reports folder and select **Create New Sub-Category**.
- Step 5** Name the new sub-category to identify it 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 XML template report templatesfiles.
- Step 8** Select a report.
- This populates the File Name with the full path for the report.
- Step 9** From the Data Source To Use drop down, select the Data Source you created to access the Unified CVP reporting database.

- Step 10** From the Data Source To Use drop down, select the Unified CVP sub-category folder you created in Step 5.
- Step 11** Click **Import**.
- Step 12** Repeat Steps 6- 11 for each Unified CVP report template.

## Working with the Imported Unified CVP Report Templates

Unified IC has the concept of "stock" report templates and "custom" report templates.

Stock report templates are the UCCE templates that are installed with Unified IC. 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.

Although Unified IC does not protect custom templates from modification, it is a best practice to retain the imported Unified CVP templates and to create Save As versions of the Unified CVP reports and Unified CVP report definitions.

## Unified IC Reporting: Basic Concepts

Term	Explanation
Dashboards	Dashboards are web pages that 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.
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 IC.
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 - this includes the SQL query, the fields, the filters, the formulas, the refresh rate, and the key criteria field for the report.

Term	Explanation
	<p>Unified IC separates Reports from Report Definitions.</p> <p>Only users who have a Premium license can view, create, or edit Report Definitions.</p>
Report Templates	Report Templates are well-formed XML files based on Report Definitions.
Report Views	<p>A report view is a layout presentation for the data that is retrieved for the report. Unified IC supports three types of views:</p> <ul style="list-style-type: none"> <li>• Grid Views</li> <li>• Chart Views</li> <li>• Gauge Views</li> </ul> <p>All reports have a grid view. The Unified CVP Call Traffic reports also have a chart view.</p> <p>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.</p> <p>You cannot delete all views. Every report must have at least one view.</p>
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.
User Groups	<p>User Groups are constructs that allow security administrators to partition Unified IC functionality.</p> <p>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.</p>
User Permissions	<p>Users have permissions associated with the groups in which they are members, and each member of a group has specific permissions in that group.</p> <p>There are three levels of permissions, represented by three check boxes in the User Information page: READ, EXECUTE, and WRITE.</p>

Term	Explanation
User Roles	User Roles confer the actions and capabilities that a user has in Unified IC. There are seven User Roles, and each user can have multiple roles.

## Working with Reports

Who can do this: [Report Designers \(page 33\)](#)

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 IC templates. Expand that category and right-click a report to open a menu of functions you can perform.

Based on permissions, a reporting user 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 IC online help.

## Running Reports

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 [Filtering Reports \(page 22\)](#).

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

## Filtering Reports

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 6: 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** dropdown, 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 AM of the first date in your range through 11:59 PM 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 AM of the first date in your range through 11:59 PM 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 will return in a reasonable time. Datetime 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**.

## Advanced Filters Tab

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).

---

**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 entering 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 7: 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	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	561	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	431	0:00:00
3/7/10 12:00 AM	Perf_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 8: Setting an Advanced Filter

Basic Filters

Advanced Filters

Name	Display Name	Description
appname	Application Name	
elementname	Element Name	
elementtype	Element Type	
exitstate	Exit State	
result	Result	
sourceappname	Source Application Name	

Edit

**Application Name (appname)**

☐ Any value
   
☒ Filter according to the following criteria:

Operator: Equal to

Value: Nuance21

Here is an Application\_Summary\_15 report for the same date range that is filtered to display only the Nuance21 application:

Figure 9: 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 CallGuid, ANI, DNIS, Query Type, Call Type, and the number of rows you want returned.

Figure 10: Filter Tab for Stored Procedure Reports 1 of 2

**Start Date and Time (@param1)**

Date:

Time:  :  :

---

**End Date and Time (@param2)**

Date:

Time:  :  :

---

**CallGuid (@param3)**

Value (STRING):

---

**ANI (@param4)**

Value (STRING):

Figure 11: Filter Tab for Stored Procedure Reports 2 of 3

**DNIS (@param5)**

Value (STRING):

---

**Query Type (@param6)**

Search in available:

Search in selected:

Choose Collection:

Available:

- INCOMING
- OUTGOING
- FAILED
- ABANDONED
- ALL
- INCOMINGOUTGOING

Selected:

Figure 12: Filter Tab for Stored Procedure Reports 3 of 3

**Call Type (@param7)**

Search in available:

Search in selected:

Choose Collection:

Available:   
 H323  
 SIP

Selected:

Refresh List

**Row Count (@param8)**

Value (DECIMAL):

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

**Note:** There is no validation, and no error displays, if you enter a Start Date Time that is subsequent to the End Date Time.

## The 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 13: Report Viewer

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

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

The menubar across the top of the Report Viewer has these selections:

<b>Save</b>	Saves the report.
<b>Save As</b>	Opens the SaveAs dialog box.
<b>Edit</b>	Launches a page where you can edit the view. <ul style="list-style-type: none"> <li>For grid views, Edit opens the Grid Editor.</li> </ul>

	<ul style="list-style-type: none"> <li>• For gauge views, Edit opens the Gauge Editor.</li> <li>• For chart views, Edit opens the Chart Editor.</li> </ul>
<b>Print</b>	Prints the report.
<b>Filter</b>	Opens the filter page so that you can change the filter values (such as date/time and values) for the report. See Filtering Reports.
<b>SQL</b>	Opens a window with a read-only display of the SQL query on which the report is based.
<b>Refresh</b>	<p>Sends a request to the server to refresh the report dataset.</p> <p><b>Note:</b> If the report view is a grid, and if you have sorted the grid, Refresh resets the view and cancels the sort.</p>
<b>Pop Out</b>	<p>Opens the report in a new, separate browser display window. The popout has no Unified IC edit or toolbar functions.</p> <p>Click <b>x</b> to close the popout.</p>
<b>Export</b> (Grids only)	Launches the Export page, where you can export the report grid to Microsoft Excel.
<b>View Dropdown</b>	If there is more than one view associated with this report template, use the dropdown 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.
<b>Help</b>	Opens a dropdown where you can select help for Unified IC reporting or for the fields in the report template.

## Editing Reports

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 14: Report Editor

Report Description: Call Traffic at 15 Minute Intervals

Report Definition: Call Traffic 15

Default View: Stacked Columns

Online Help:

☐ Bypass Filter Dialog

Permissions:

My Group	All Users
<input checked="" type="checkbox"/> Read	<input checked="" type="checkbox"/> Read
<input checked="" type="checkbox"/> Execute	<input checked="" type="checkbox"/> Execute
<input checked="" type="checkbox"/> Write	<input checked="" type="checkbox"/> Write

Buttons: Edit Default Filter, Edit Views, Save, Save As, Refresh

Table 1: Fields on the Report Editor page

Field	Explanation
<b>Report Description</b>	This field displays a description for the report.
<b>Report Definition</b>	This field displays Report Definition for the report.
<b>Default View</b>	<p>From the dropdown, select the default view that is to display when users run the report.</p> <p><b>Note:</b> 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.</p>
<b>Online Help</b>	Shows the path for the online help topic for this report.
<b>Bypass Filter Dialog</b>	<p>Check this box so that the report runs directly and users have no option to filter the report.</p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>• Even if the report has run directly, you can click the Filter icon in the Report Viewer to refilter and rerun the report.</li> <li>• 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.</li> </ul>
<b>Permissions</b>	Use these boxes to view or change user permissions for My Group (the default group) and for the All Users group.

**Actions on the Report Editor page:**

- **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.
- **Run** - runs the report.
- **Save**
- **Save As** - creates a cloned copy of the report.
- **Refresh** updates the page to show changes another user has made.

## Changing and Moving 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.

To:	Do This:
View the properties of a Current (visible) Field	Right-click a field and select <b>Properties</b> to open a page where you can set Field Properties.
Manage thresholds for a Current (visible)field	Right-click a field and select <b>Thresholds</b> .  <b>Note:</b> Thresholds are not implemented for Unified CVP templates.

To:	Do This:
Move a Current (visible) field so it is no longer visible in a report.	<p>Right-click a field and select <b>Remove Selected</b>. Click <b>Yes</b> at the confirmation message.</p> <p>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.</p> <p>If you remove all fields from the Current panel, the generated report will show no data.</p> <p><b>Note:</b> 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.</p>
Move a field from the Available panel to the Current panel.	<p>Click (highlight) GridHeaders at the top of the Current panel.</p> <p>Select the field you want to move in the Available panel.</p> <p>Click &gt; to move the field over. It 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.</p> <p><b>Note:</b> 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 <i>subsystemname</i> from Available, and <i>Subsystem</i> is already part of Current), the <i>Subsystem</i> field in Current becomes highlighted. You cannot have the same field more than once in the Current panel.</p>
Reorder Current Fields (Up and Down)	Use the up and down arrows to the right of the Current field order in the grid panel.

**Actions on this page:**

- **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.
- **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.

## Exporting Reports

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 exported:

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

### What is not exported:

- Report Filters
- Collections

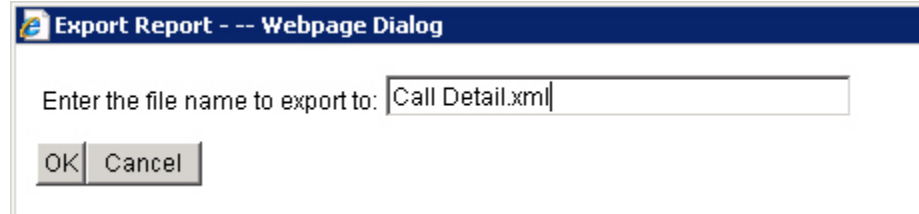
To export a report:

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.

2. Click **OK** to open the Windows file download dialog box.
3. Click **Save** and navigate to the location where you want to save the report.

Figure 15: Report Export



## Unified IC Reporting: Unified IC User Roles

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

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.

### These are the User Roles for Unified IC:

- **Login User**

By default, everyone who can sign in to Unified IC is a Login User. Login Users have that role and only that role until the SecurityAdministrator 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 IC
- 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.

- **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 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

- **Value List Collection Designer**

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

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

## Other Roles

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

- **Super Users**

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

The initial and default Super User is the [System Application User \(page 34\)](#) who is configured during installation of the Controller..

The initial Super User (the SystemApplication User) can sign in to the Unified IC Reporting console and has *all* User Roles and *full permissions* for all drawers in Unified IC Reporting. Those credentials cannot be removed from the initial Super User.

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

- **The System Application User**

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

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

- **The 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).

- **UCCE Supervisors**

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



# Chapter 3

## Unified CVP Templates for Unified Intelligence Center

---

This chapter presents the Cisco-designed reports for Unified CVP. These template are available to you once you import them and set their data source as described in Chapter 2.

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 \(15, Daily, and Weekly\), page 35](#)
- [Call Report, page 38](#)
- [Call Detail Report, page 41](#)
- [Call Traffic Reports \(15, Daily, and Weekly\), page 47](#)
- [Callback Reports \(Current and Historical\), page 50](#)
- [Trunk Group Utilization Report, page 51](#)

### Application Summary Reports (15, Daily, and Weekly)

There are 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 \(page 75\)](#)

Fields in this report are populated from these tables:

- [Application Summary 15 table \(page 77\)](#)

- [ApplicationSummary\\_Daily](#) (page 78)
- [ApplicationSummary\\_Weekly](#) (page 78)
- [ResultRef](#) (page 89)
- [ElementTypeRef](#) (page 86)

**Fields in this report:**

- **Date and Time**

The date and time when the period of data collection represent in this report ~~no~~ 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 SourceAppName 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 ExitStatefield of the appropriateApplication 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.

## Sample Report Application Summary 15 Report

Figure 16: Application Summary 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	561	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	431	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	NumberSelectAnalysis	Decision	number_dtmf	Normal	6070	0:00:00

## Sample Report Application Summary Daily Report

Figure 17: 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	18079	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	18864	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	2180	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	18075	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	19723	0:00:00

## Sample Report Application Summary Weekly Report

Figure 18: 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	24069	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	2628	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	24606	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	26443	0:00:00
3/7/10 12:00 AM	Perf_31Msgs	null	Subdialog Return_02	Subdialog_Return	none	Hang Up	56	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 19: Basic Filters with Absolute Dates

**Basic Filters** | Advanced Filters

**Call Started (startdatetime)**

☐ Relative Date Range ☒ Absolute Date Range

From (mmVddVyyyy): 3/17/2010 To (mmVddVyyyy): 3/18/2010

☒ Only show results that are within a specific time period

From: 6:00 AM To: 6:01 AM

☐ Only show results that are on certain days of the week

Fields in this report are populated from these tables:

- [Call \(page 61\)](#)
- [CallTypeRef \(page 82\)](#)
- [SubsystemTypeRef \(page 89\)](#)

### Fields in this report:

- **Call Guid**

The global unique id of a call. From Call.CallGuid.

- **ANI**

The Automatic Number Identification 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 sent by the telephony provider. From Call.DNIS.

- **UID**

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

- **UUI**

The User-To-User Information 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.SubsystemTypeID.

See [SubsystemTypeRef \(page 89\)](#) for the list of subsystem types.

- **Call Type**

The type of call, where Call.CallTypeID = CallTypeRef.CallTypeID. See [CallTypeRef \(page 82\)](#) 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 will have a UTC startdatetime ( + 7 hours), and the dbdatetime for this call will be 4 PM (Boston time)

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

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). Today, the timezone offset is 240 minutes. Last week (before the datetime rolled), the offset was 300 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.

## Call Report

- **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 VRU leg. The VRU leg is the leg that talks to the VoiceXML Gateway(s).

## Sample Call Report

Figure 20: Call Report 1 of 2

First 500 Calls From Starting Time									
Call GUID	ANI	DNIS	UID	UUI	IIDigits	Sub System Type	Call Type	Call Started	Call Ended
281FE49630C111DF908F00146944B76A	194194194	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:52 AM
2849652930C111DF909000146944B76A	194194194	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:52 AM
2852C4B830C111DF909C00146944B8CC	193193193	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:52 AM
2871E57330C111DF909D00146944B8CC	193193193	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:52 AM
2876914630C111DF909100146944B76A	194194194	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:52 AM
2894B12230C111DF909E00146944B8CC	193193193	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:00 AM	3/17/10 6:00:53 AM
28B6DEA430C111DF909F00146944B8CC	193193193	6002510001	null	NA	NA	VXML	VXML	3/17/10 6:00:01	3/17/10 6:00:53

Figure 21: Call Report 2 of 2

Time Persisted to Database	Time Zone Offset from UTC (Minutes)	Applications Visited	Errors	On Hold	Opted Out	Timed Out	Transferred
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 \(page 61\)](#)
- [CallEvent \(page 63\)](#)
- [CallICMInfo \(page 64\)](#)
- [VXMLElement \(page 68\)](#)
- [VXMLSession \(page 73\)](#)
- [CallTypeRef \(page 82\)](#)
- [CauseRef \(page 83\)](#)
- [EventTypeRef \(page 87\)](#)
- [ResultRef \(page 89\)](#)
- [SubsystemTypeRef \(page 89\)](#)
- [TransferTypeRef \(page 90\)](#)

### 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).Today, the timezone offset is 240 minutes. Last week (before the datetime rolled), the offset was 300 minutes.

- **Call Guid**

The global unique id of a call. From Call.CallGuid.

- **Call Type**

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

- **ANI**

The Automatic Number Identification 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 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 of the originating caller, sent by telephony provider. From Call.UID.

- **UUI**

The User-To-User Information 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 application svisited 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. The VRU leg is the leg that talks to the VoiceXML Gateway(s). 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 Unified CVP Service used for the call, such as "2" for VXML. From SubsystemTypeRef.SubsystemTypeID.

- **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 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.

- **RouterCallKeySequenceNumber**

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**

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 | Call Type = SIP | Rows = 100.

Figure 22: Call Detail Report 1 of 6

callstartdate	startdatetime	enddatetime	localtimezoneoffset	callguid	calltype	ANI	Call dnis	iidigits
3/26/10	3/27/10 3:52:29 AM	3/27/10 3:53 AM	null	000C0E9F388B11DF9D2FC5AA41DD481D	VXML	203203203	6001430001	NA
3/26/10	3/27/10 3:52:29 AM	3/27/10 3:53 AM	null	000C0E9F388B11DF9D2FC5AA41DD481D	VXML	203203203	6001430001	NA
3/26/10	3/27/10 3:52:29 AM	3/27/10 3:53 AM	-240	000C0E9F388B11DF9D2FC5AA41DD481D	VXML	203203203	6001430001	NA
3/26/10	3/27/10 3:45:20 AM	3/27/10 3:46 AM	null	00169AC9388A11DF8330A657EF609B3C	VXML	202202202	6001430001	NA
3/26/10	3/27/10 3:45:20 AM	3/27/10 3:46 AM	null	00169AC9388A11DF8330A657EF609B3C	VXML	202202202	6001430001	NA
3/26/10	3/27/10 3:45:20 AM	3/27/10 3:46 AM	-240	00169AC9388A11DF8330A657EF609B3C	VXML	202202202	6001430001	NA
3/26/10	3/27/10 3:52:29 AM	3/27/10 3:53 AM	null	0020335A388B11DF9D2FC5AA41DD481D	VXML	203203203	6001430001	NA

Figure 23: 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 3:45:20 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 3:45:20 AM	null
null	NA	1	null	0	0	0	0	3/27/10 3:52:29 AM	New Call

## Call Detail Report

Figure 24: Call Detail Report 3 of 6

Call Event								
subsystemname	subsystemtypeid	calllegid	cause	result	mediafilename	transferlabel	transfertype	
cvpsld143.SYS_VXML7	2	000C0E9F388B11DF8D2FC5AA41DD481D	None	null	NA	null	null	
cvpsld143.SYS_VXML7	2	000C0E9F388B11DF8D2FC5AA41DD481D	Normal Completion	null	NA	null	null	
cvpsld143.SYS_VXML7	2	null	null	Normal	null	null	null	
cvpsld143.SYS_VXML7	2	00169AC9388A11DF8330A657EF609B3C	None	null	NA	null	null	
cvpsld143.SYS_VXML7	2	00169AC9388A11DF8330A657EF609B3C	Normal Completion	null	NA	null	null	
cvpsld143.SYS_VXML7	2	null	null	Normal	null	null	null	
cvpsld143.SYS_VXML7	2	0020335A388B11DF8D34C5AA41DD481D	None	null	NA	null	null	

Figure 25: Call Detail Report 4 of 6

ICM								
messagebusname	routercallkey	routercallkeyday	routercallkeysequencenumber	appname	enterdatetime	exitdatetime	duration	
cvpsld143.MsgBus009	null	null	null	null	null	null	null	
cvpsld143.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	
cvpsld143.MsgBus009	null	null	null	null	null	null	null	
cvpsld143.MsgBus009	null	null	null	null	null	null	null	
null	null	null	null	Perf_65Msgs_Req	3/27/10 3:45:20 AM	3/27/10 3:45:20 AM	63	
cvpsld143.MsgBus009	null	null	null	null	null	null	null	

Figure 26: 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	1007661269661949570	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	1007381269661520130	10.30.30.143.1269661520187.1182.Perf_65Msgs_Req	null	Normal Completion	
null	null	null	null	null	null	null	

Figure 27: Call Detail Report 6 of 6

vxmlcause	vxmlenddatetime	vxmlleventype	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)

There are three CallTraffic 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 callservers/vxml servers.

The three reports display the same data and aggregate the data for different time periods. See [Summary / Aggregate Tables \(page 75\)](#)

Fields in this report are populated from these tables:

- [Call\\_15 \(page 79\)](#)
- [Call\\_Daily \(page 79\)](#)
- [Call\\_Weekly \(page 80\)](#)
- [CallTypeRef \(page 82\)](#)
- [SubsystemTypeRef \(page 89\)](#)

### Fields in this report:

- **Date and Time**

The date and time of the call, from the start of the increment. From the dbatetimfield 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.SubsystemTypeID.

See [SubsystemTypeRef \(page 89\)](#) for the list of subsystem types.

- **Call Type**

The type of call, where the Call Summary Table CallTypeID = CallTypeRef.CallTypeID. See [CallTypeRef \(page 82\)](#) for the list of call types.

- **Number of Calls**

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

- **Average Call Length**

**Call Traffic Reports (15, Daily, and Weekly)**

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. The report will show 0. 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).

### Sample Call Traffic 15 Report for "Last Week"

Figure 28: Call Traffic 15 Report

Call Traffic at 15 Minute Intervals										
Date and Time	Sub System 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	6681	0:00:50.856	5587	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	7601	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	4822	0	0	0	0	0

### Sample Call Traffic Daily Report for "Last Week"



Figure 29: 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

## Sample Call Traffic Weekly Report for "Last Week"

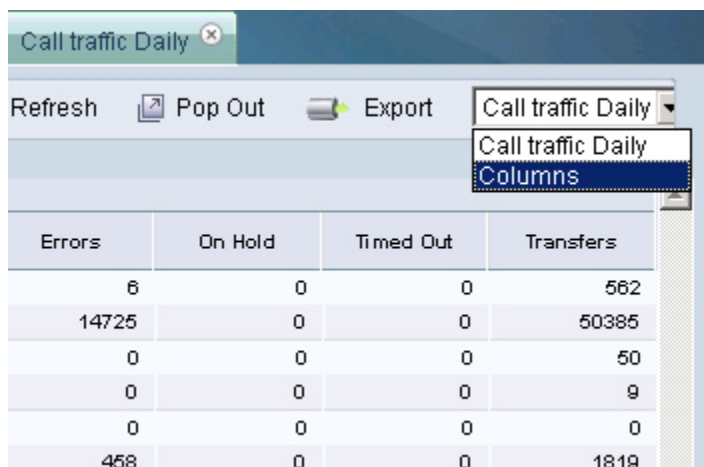
Figure 30: Call Traffic Weekly Report

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 dropdown and select Columns.

Figure 31: Selecting a Chart View



The screenshot shows a web application window titled 'Call traffic Daily'. It has a toolbar with 'Refresh', 'Pop Out', and 'Export' buttons. A dropdown menu is open, showing 'Call traffic Daily' and 'Columns' (which is highlighted). Below the menu is a table with four columns: 'Errors', 'On Hold', 'Timed Out', and 'Transfers'. The table contains six rows of data.

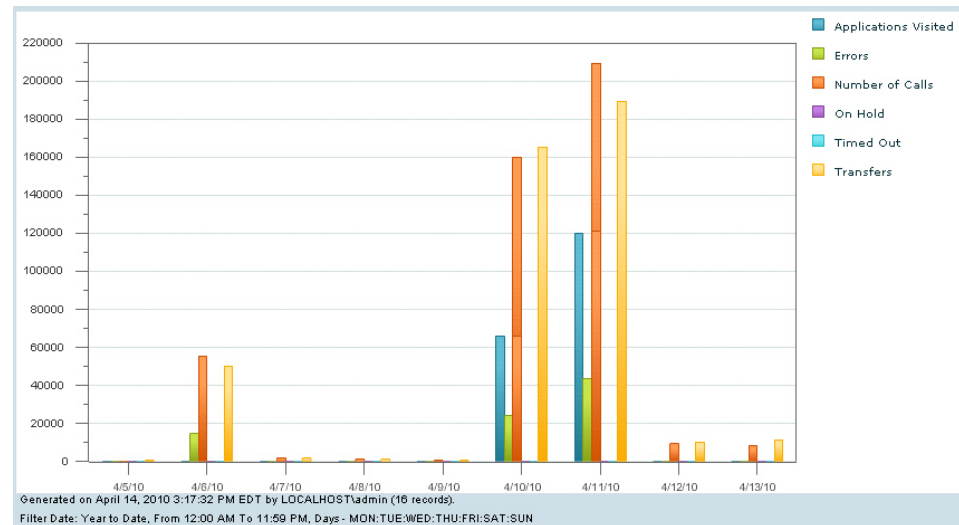
Errors	On Hold	Timed Out	Transfers
6	0	0	562
14725	0	0	50385
0	0	0	50
0	0	0	9
0	0	0	0
458	0	0	1819

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 is a Call Traffic chart from Call Traffic Daily data:

## Callback Reports (Current and Historical)

Figure 32: Call Traffic Daily Chart



## Callback Reports (Current and Historical)

There are two callback reports: Callback Current and Callback Historical.

They both pull data from the [CallBack \(page 92\)](#) table.

**Note:** The callback 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 Callback Pending report takes data from calls that have occurred in the past fifteen minutes. The Callback Historical report shows calls that occurred prior to fifteen minutes ago.

This report is populated by data from the [CallBack \(page 92\)](#) 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 \(page 87\)](#) table for the list of values.

- **Cause**

The cause of the event, from Callback.CauseID, where Callback.CauseID = CauseRef.causeid. See the [CauseRef \(page 83\)](#) 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`.

## Sample Pending Callback Report

Figure 33: Callback (Pending) Report

Pending Callbacks				
Date and Time	Event Type	Cause	Gateway	Average Number of Attempts
Feb 11, 2010 1:15 PM	Callback In Progress	None	gateway_3	0
Feb 12, 2010 1:15 PM	Callback In Progress	None	10.86.129.176	0
Feb 15, 2010 1:45 PM	Callback In Progress	None	gateway_4	0
Feb 15, 2010 1:45 PM	Callback In Progress	None	gateway_3	0
Feb 15, 2010 2:00 PM	Callback In Progress	None	gateway_4	0
Feb 16, 2010 11:15 AM	Callback In Progress	None	10.86.129.176	0
Feb 18, 2010 3:00 PM	Callback In Progress	None	10.86.129.176	0
Feb 19, 2010 4:00 PM	Callback Pending	None	gateway_1	0
Feb 22, 2010 6:45 AM	Callback In Progress	None	gateway_4	0
Feb 22, 2010 6:45 AM	Callback In Progress	None	gateway_3	0
Feb 22, 2010 7:00 AM	Callback In Progress	None	gateway_3	0
Feb 22, 2010 7:00 AM	Callback In Progress	None	gateway_4	0
Feb 24, 2010 9:00 AM	Callback In Progress	None	gateway_4	0
Feb 24, 2010 9:00 AM	Callback In Progress	None	gateway_3	0
Feb 24, 2010 9:15 AM	Callback In Progress	Busy	gateway_0	3
Feb 24, 2010 12:00 PM	Callback In Progress	None	10.86.129.176	0

## Sample Historical Callback Report

Figure 34: Callback Historical Report

Historical Callbacks				
Datetime	Event Type	Cause	Gateway	Average Number of Attempts
Feb 8, 2010 5:45 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 tabular display).

This report is populated by data from the [Usage Table \(page 81\)](#), the [Resource Table \(page 88\)](#), and the [Device Table \(page 86\)](#).

**Fields in this report are:**

- **Date and Time**

## Trunk Group Utilization Report

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 \(page 88\)](#) 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.

### Sample Report:

Figure 35: 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	69	
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	69	
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	69	
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	69	



# Chapter 4

## Database Schema

---

This chapter contains the following topics:

- [About the Database Schema, page 53](#)
- [Data Model Diagram, page 54](#)
- [The Unified CVP Reporting Data Model, page 57](#)
- [Unified CVP Database Tables, page 61](#)
- [Call Tables, page 61](#)
- [VXML Tables, page 67](#)
- [Summary / Aggregate Tables, page 75](#)
- [Lookup and Reference Tables, page 81](#)
- [Courtesy CallBack Tables, page 91](#)

### About the Database Schema

The Unified 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 IC) as their reporting platform must add the Informix database as a data source in Unified IC.

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:

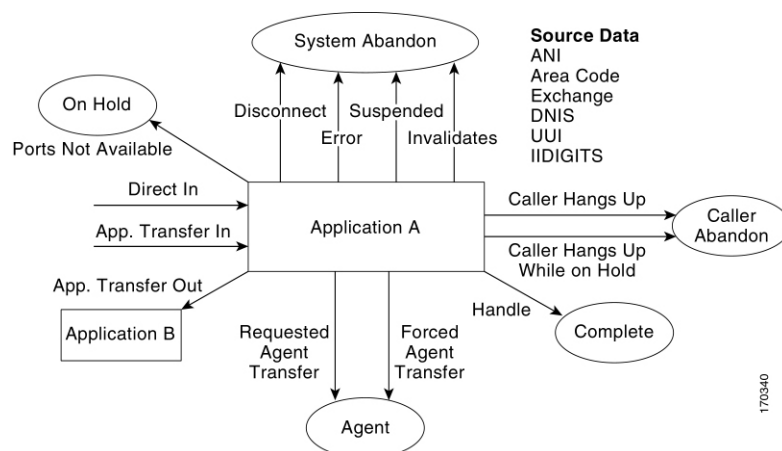
- Establish database connectivity with Unified IC and to import and run the Unified CVP templates with Unified IC.
- Establish database connectivity with other commercial off-the-shelf reporting and analytics engines and then build custom reports against the Unified CVP database schema.

## Data Model Diagram

**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 36: Call Flow



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

## Data Model Diagram

The following entity-relationship diagram depicts the Unified CVP database schema.

[illegible]

The diagram illustrates the following tables and their attributes:

- CallBackQueue**
  - SurrogateID: integer (FK)
  - CallStartDate: date (FK)
  - QueueID: smallint (FK)
  - QueueStatus: smallint
  - EnterDateTime: datetime year to second
  - LeaveDateTime: datetime year to second
  - CVPEstimatedWaitTime: smallint
  - ICMEstimatedWaitTime: smallint
  - ValidationStatus: smallint
  - DBDateTime: datetime year to fraction(3)
- CallBack**
  - SurrogateID: serial
  - CallStartDate: date
  - CallGUID: char(32)
  - OldGUID: char(32)
  - ANI: varchar(32)
  - CallBackType: char(1)
  - Gateway: varchar(40)
  - RecordingURL: nvarchar(255)
  - ScheduledCallBackDateTime: datetime year to second
  - ScheduledCallBackDN: varchar(32)
  - DBDateTime: datetime year to fraction(3)
  - Location: varchar(32)
  - NbrAttempts: smallint
  - EventTypeId: integer (FK)
  - CauseId: integer (FK)
- QueueRef**
  - QueueID: smallint
  - QueueName: char(40)
- CauseRef**
  - CauseId: integer
  - Cause: nvarchar(96)
- EventTypeRef**
  - EventTypeId: integer
  - EventType: nvarchar(96)
- CallBackEvent**
  - SurrogateID: integer (FK)
  - CallStartDate: date (FK)
  - EventDateTime: datetime year to fraction(3)
  - CauseId: integer (FK)
  - DBDateTime: datetime year to fraction(3)
  - EventTypeId: integer (FK)

**Relationships:**

- CallBackQueue** to **CallBack**: One-to-many relationship. CallBackQueue has a mandatory one (indicated by a vertical bar and crow's foot notation) and CallBack has a mandatory many (indicated by a vertical bar and crow's foot notation).
- CallBackQueue** to **QueueRef**: One-to-one relationship. CallBackQueue has a mandatory one and QueueRef has a mandatory one.
- CallBack** to **CallBackEvent**: One-to-many relationship. CallBack has a mandatory one and CallBackEvent has a mandatory many.
- CallBack** to **CauseRef**: One-to-many relationship. CallBack has a mandatory one and CauseRef has a mandatory many.
- CallBack** to **EventTypeRef**: One-to-many relationship. CallBack has a mandatory one and EventTypeRef has a mandatory many.

## Data Model Diagram

Figure 39: Summary Tables

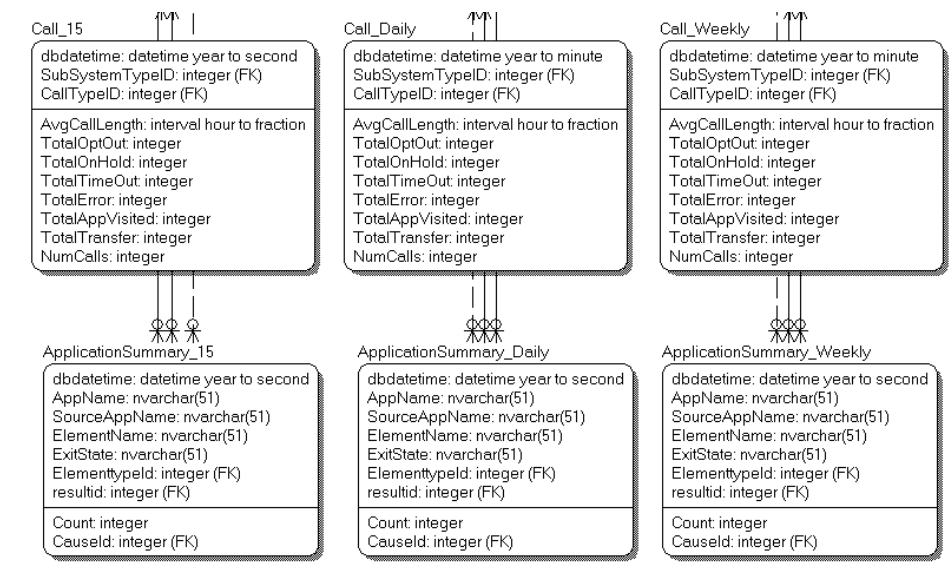
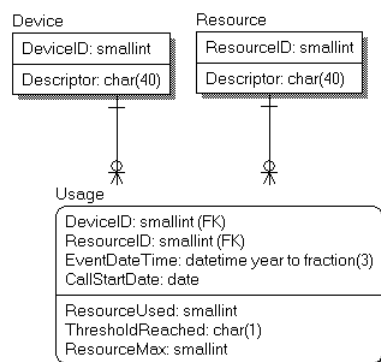


Figure 40: Trunk Group Utilization Tables



## Keys

The documentation for the reporting schema lists fields as *PK*, *FK*, *AK*, 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) will have a supporting index. Foreign Keys which refer to lookup tables are not supported by an index.



## The Unified CVP Reporting Data Model

Topics:

- [DateTime Columns \(page 57\)](#)
- [Informix Dates and Times \(page 57\)](#)
- [SIP Calls \(page 59\)](#)
- [Trunk Utilization \(page 60\)](#)

### DateTime Columns

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

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

This is date 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 \(page 61\)](#) 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 CVP Reporting Database supports three concepts of time.

#### 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= '05/31/2009';
```

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:** Weekdays 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 the last 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 or WW. 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)*

## 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 above 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](http://www-01.ibm.com/software/data/informix/pubs/library/ids_100.html) ([http://www-01.ibm.com/software/data/informix/pubs/library/ids\\_100.html](http://www-01.ibm.com/software/data/informix/pubs/library/ids_100.html)).

### See Also

Informix, Operating System Time, and Local Time

## SIP Calls

These are recorded in the [Call table \(page 61\)](#) along with VXML calls.

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

Events for these calls (such as start and end) are recorded in the [CallEvent table \(page 63\)](#).

## Sample Query and SIP Calls

Detail 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 that 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 \(page 81\)](#)) with three dimensions, Resource, Device, and Time.

Since 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*.

## Unified CVP Database Tables

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

Tables are categorized as follows:

- [Call Tables \(page 61\)](#)
- [VXML Tables \(page 67\)](#)
- [Summary / Aggregate Tables \(page 75\)](#)
- [Lookup / Reference Tables \(page 81\)](#)
- [Callback Tables \(page 91\)](#)

## Call Tables

These tables are:

- [Call \(page 61\)](#)
- [CallEvent \(page 63\)](#)
- [CallICMInfo \(page 64\)](#)

## 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.

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

## Call Tables

Field	Type	Null	Index	Description
CallStartDate	date	No	PK (Composite CallGUID, CallStartDate)	Date of the call, for data pugging purposes.
StartDateTime	datetime YEAR to FRACTION (3)	No	Yes 8.0	EventDateTime for the date and time a call was made.  <i>Uses UTC in Rel. 8.0.</i>
EndDateTime	datetime YEAR to FRACTION(3)	Yes	Yes 8.0	EventDateTime for the date and time a call ended with hang-up or disconnect.  <i>Uses UTC in Rel. 8.0.</i>
ANI	varchar(32)	Yes	No	ANI of the caller sent by telephony provider
DNIS	varchar(32)	No	No	DNIS of a call sent by telephony provider
UUI	varchar(100)	Yes	No	The UUI of the originating caller sent by telephony provider
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 opt 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 8.0	The date and time of the database operation (when the record was inserted).
CallTypeID	smallint  <i>Formerly char(1)</i>	No	Non-Indexed FK	The type of call. See <a href="#">CallTypeRef table (page 82)</a> .

Field	Type	Null	Index	Description
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.  <i>Replaces LocalTimeZone.</i>

## 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 \(page 73\)](#).

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)	Date of the call, for data purging purposes
CallLegID	varchar(255,43)  <i>Formerly Varchar(43)</i>	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 8.0	Date and time of the event  <i>Uses UTC in Rel. 8.0.</i>
EventTypeID	int	No	Non-Indexed FK	The mechanism used to generate the call event. See <a href="#">EventTypeRef (page 87)</a> .
CauseID	int	No	Non-Indexed FK	The reason that the call event was generated. See <a href="#">CauseRef (page 83)</a> .
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes (8.0)	The date and time of the database operation (when the record was inserted).

## Call Tables

Field	Type	Null	Index	Description
TransferTypeID	integer	No	Non-Indexed FK	A unique id of the transfer type. See <a href="#">TransferTypeRef (page 90)</a> .  <i>New in Rel. 8.0.</i>
SubSystemTypeID	int	No	Non-Indexed FK	The type of the Service. See <a href="#">SubsystemTypeRef (page 89)</a> .
SubSystemName	varchar(41)	No	No	The name of the Service the event originated from
MediaFileName	nvarchar(255)	Yes	No	<i>This is always null in Release 8.0.</i>
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  <i>New in Rel. 8.0.</i>
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 TCD records are cut, the RouterCallKeySequenceNumber will increment in 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.

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)	Date of the call, for data purging purposes



Field	Type	Null	Index	Description
RouterCallKey	Integer	No	AK (Composite index RouterCallKey, RouterCallKey Day)	<p>ICM Router CallKey - single value per call.</p> <p>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.</p> <p>If the call is a consult or conference, then Unified CVP will see two different callguids for the same call in its database.</p> <ul style="list-style-type: none"> <li>The first callguid is the incoming callguid when the call is established.</li> <li>The second callguid is for an agent originated/consult call.</li> </ul> <p>The routercallkey and routercallkeyday 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.</p>
RouterCallKeyDay	Integer	No	AK(Composite index RouterCallKey, RouterCallKey Day)	<p>ICM RouterCallKeyDay</p> <p>Typically this number changes on the switch and VRU leg of a call.</p> <p>You will see 0 for the switch leg of the call and 1 for the VRU leg of the call.</p> <p><b>Note:</b></p> <p>VRU leg is not captured in Release 8.0.) we do not capture the vru leg in cvp 8.0)</p> <p>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 Unified Communications Manager. This</p>

## Call Tables

Field	Type	Null	Index	Description
				comes back via jtapi and is on a separate PG. As the new call shows on a separate PG, Unified ICM cuts a new TCDrecord when the call ends. The routercallkeyseq number increments on that switch leg.
RouterCallKeySequenceNumber	int	Yes	Yes	ICM RouterCallKeySequenceNumber
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes (8.0)	Date and time of the event
DBDateTime	datetime YEAR to FRACTION(3)	No	Yes (8.0)	The date and time of the database operation (when the record was inserted).

## 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.

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)	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
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	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

Field	Type	Null	Index	Description
				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).

## VXML Tables

Tables:

- [VXMLCustomContent](#) (page 68)
- [VXMLElement](#) (page 68)
- [VXMLElementDetail](#) (page 69)
- [VXMLElementFlag](#) (page 70)
- [VXMLError](#) (page 71)
- [VXMLHotEvent](#) (page 71)
- [VXMLHotLink](#) (page 72)
- [VXMLSession](#) (page 73)
- [VXMLSessionVariable](#) (page 74)
- [VXMLVoiceInteractDetail](#) (page 75)

The data for VXML treatment is much richer than what 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](#) (page 61) and are linked to the [VXMLSession table](#) (page 73) 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](#) (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
```

## VXML Tables

```

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 VoiceXML 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.

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)	Date of the call, for data purging purposes
VarName	nvarchar(51)	No	No	The name of the custom event variable
VarValue	nvarchar(255)	Yes	No	The value of the custom event variable
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	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 VoiceXML 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.

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)	Date of the call, for data pugging 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
ElementTypeID	int	No	Non-Indexed FK	The type of element
EnterDateTime	datetime YEAR to FRACTION(3)	No	Yes	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.

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

## VXML Tables

Field	Type	Null	Index	Description
CallStartDate	date	No	Yes (Composite index ElementID, CallStartDate)	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  <i>Uses UTC in Rel. 8.0.</i>
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.

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)	Date of the call, for data purging purposes
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the flag activated  <i>Uses UTC in Rel. 8.0.</i>
Name	nvarchar(51)	No	No	The flag name
PreviousElementName	nvarchar(51)	Yes	No	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).

## VXML\_Error 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.

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)	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  <i>Uses UTC in Rel. 8.0.</i>
Description	nvarchar(255)	No	No	The detailed error 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).

## VXML\_HotEvent 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.

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

## VXML Tables

Field	Type	Null	Index	Description
CallStartDate	date	No	Yes (Composite index ElementID, CallStartDate)	Date of the call, for data purging purposes
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when hot event occurred  <i>Uses UTC in Rel. 8.0.</i>
Name	nvarchar(51)	No	No	The name of the hot event
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 and/or 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.

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),	Date of the call, for data purging purposes
EventDateTime	datetime YEAR to FRACTION(3)	No	Yes	The date and time when the hot link activated  <i>Uses UTC in Rel. 8.0.</i>
Name	nvarchar(51)	No	No	The name of the hot link
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



Field	Type	Null	Index	Description
				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 \(page 63\)](#).

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)	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	Date and time when session starts
AppName	nvarchar(51)	No	Yes	The name of the VXML application
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

## VXML Tables

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).
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.  <i>Replaces LocalTimeZone.</i>

## 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.

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)	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	Date and time when the session variable changed  <i>Uses UTC in Rel. 8.0.</i>

Field	Type	Null	Index	Description
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).
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 will have one record for each Voice Interaction with the caller.

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);	Date of the call, for data purging purposes
ElapsedTimeMillisecond	int	No	No	Time since the last interaction
VoiceActionTypeID	int	No	Non-Indexed FK	Type of interaction
Value	nvarchar(255)	Yes	No	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

Tables:

- [ApplicationSummary\\_15 \(page 77\)](#)

- [ApplicationSummary\\_Daily](#) (page 78)
- [ApplicationSummary\\_Weekly](#) (page 78)
- [Call\\_15](#) (page 79)
- [Call\\_Daily](#) (page 79)
- [Call\\_Weekly](#) (page 80)

Version 8.0 of the 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.

In this release of the CVP Reporting Server, 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 periods, the system can process can at most summarize 96 days' worth of data.

- \_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.

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

**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. There is no guarantee as to the latency of source data. 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

A 15-minute 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
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

## Summary / Aggregate Tables

## ApplicationSummary\_Daily Table

A daily 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
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
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

Field	Type	Null	Index	Description
Avg_elapsed	int	yes	No	The average elapsed time for this element

## Call\_15 Table

A 15-minute summary of call activity by SubSystemType and CallType.

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	Average call length for this period
TotalOptOut	integer	Yes	No	Total number of Opt Outs in this period
TotalOnHold	integer	Yes	No	Total number of Holds in this period
TotalTimeOut	integer	Yes	No	Total number of Time Outs in this period
TotalError	integer	Yes	No	Total number of errors in this period
TotalAppVisited	integer	Yes	No	Total number of applications visited in this period
TotalTransfer	integer	Yes	No	Total number of transfers in this period
NumCalls	integer	Yes	No	Total number of calls in this period

## Call\_Daily Table

A daily summary of call activity by SubSystemType and CallType

Field	Type	Null	Index	Description
dbdatetime	datetime year to minute	Yes	PK	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	Average call length for this period

## Summary / Aggregate Tables

Field	Type	Null	Index	Description
TotalOptOut	integer	Yes	No	Total number of Opt Outs in this period
TotalOnHold	integer	Yes	No	Total number of Holds in this period
TotalTimeOut	integer	Yes	No	Total number of Time Outs in this period
TotalError	integer	Yes	No	Total number of errors in this period
TotalAppVisited	integer	Yes	No	Total number of applications visited in this period
TotalTransfer	integer	Yes	No	Total number of transfers in this period
NumCalls	integer	Yes	No	Total number of calls in this period

## Call\_Weekly Table

A weekly summary of call activity by SubSystemType and CallType

Field	Type	Null	Index	Description
dbdatetime	datetime year to minute	Yes	PK	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	Average call length for this period
TotalOptOut	integer	Yes	No	Total number of Opt Outs in this period
TotalOnHold	integer	Yes	No	Total number of Holds in this period
TotalTimeOut	integer	Yes	No	Total number of Time Outs in this period
TotalError	integer	Yes	No	Total number of errors in this period
TotalAppVisited	integer	Yes	No	Total number of applications visited in this period
TotalTransfer	integer	Yes	No	Total number of transfers in this period
NumCalls	integer	Yes	No	Total number of calls in this period



## 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
CallStartDate	date	No	PK	The date of this measurement for purge purposes
ResourceUsed	smallint	Yes	No	The amount of resource 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

## Lookup and Reference Tables

These tables are:

- [ActionTypeRef \(page 82\)](#)
- [CallTypeRef \(page 82\)](#)
- [CauseRef \(page 83\)](#)
- [Device \(page 86\)](#)
- [ElementTypeRef \(page 86\)](#)
- [EventTypeRef \(page 87\)](#)
- [OutgoingECCVariable \(page 66\)](#)
- [QueueRef \(page 88\)](#)
- [Resource \(page 88\)](#)
- [ResultRef \(page 89\)](#)
- [SubsystemTypeRef \(page 89\)](#)
- [TransferTypeRef \(page 90\)](#)

## Lookup and Reference Tables

- [Usage \(page 81\)](#)
- [UserInputModeRef \(page 90\)](#)
- [VarDataTypeRef \(page 90\)](#)
- [VoiceActionTypeRef \(page 91\)](#)

## ActionTypeRef Table

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

Field	Type	Null	Index	Description
ActionTypeID	int	No	PK	The unique id of an action type
ActionType <i>Formerly Name</i>	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.

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 is the table to map a CauseID to the text value for the cause.

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 <i>Formerly Name</i>

### 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"

---

Lookup and Reference Tables

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, "Gen 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

## Lookup and Reference Tables

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. Presently, this is an IP Address.

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 is the table to map an ElementTypeID to a text value for the VXML element type.

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 <i>Formerly Name</i>

**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, "ReqICMLLabel"
- 13, "General"

## 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 <i>Formerly Name</i>

### 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"

## Lookup and Reference Tables

- 11, "Run Script"
- 12, "Agent Recording"
- 13, "ICM Recording"
- 14, "Agent Video"
- 15, "ICM Video"
- 16, "Bridge Transfer"
- 17, "Blind Transfer"
- 18, "ReqICMLLabel"
- 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"

## QueueRef Table

QueueRef is a callback lookup table. This table resolves QueueID to a text value for the queue in which a callback is waiting.

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

The QueueName stores whatever you decide to call the queues.

## Resource Table

The resource that will be measured. At this time this will include MEMory, CPU, DSO and DSP.



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

## ResultRef Table

This is the table to map a ResultID to a text value for a result.

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 <i>Formerly Name</i>

### Table Values (ID, Result):

- 1, "Normal"
- 2, "Invalidated"
- 3, "HotEvent"
- 4, "HotLink"
- 5, "Hang Up"
- 6, "Error"
- 7, "Transfer"

## SubSystemTypeRef Table

This is the table to map a SubSystemTypeID to a Unified CVP Service type.

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"

## Lookup and Reference Tables

- 3, "OAMP" [Operate, Administer, Maintain, Provision = Operations Console]
- 4, "CONTROLLER"
- 5, "RPT"
- 6, "ICM"
- 7, "ORM" [OAMP Resource Manager = element collocated 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. Transfer Types.

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

## UserInputModeRef Table

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

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  <i>Formerly Name</i>

**Table Values (ID, Name):**

- 1, "DTMF"
- 2, "Voice"
- 3, "DTMF Voice"

## VarDataTypeRef Table

This is the table to map a VarDataTypeID to the data type of a variable.

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 <i>Formerly Name</i>

**Table Values (ID, Name):**

0, "String"

1, "Int"

2, "Float"

3, "Boolean"

## VoiceActionTypeRef Table

This is the table to map 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 <i>Formerly Name</i>

**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

Tables:

## Courtesy CallBack Tables

- [CallBack \(page 92\)](#)
- [CallBackEvent \(page 94\)](#)
- [CallBackQueue \(page 94\)](#)

These tables support Courtesy Callback functionality.

Since this data is of an 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 \(page 92\)](#).

A row is placed into the [CallBackQueue table \(page 94\)](#) for the call to manage timing and sequencing of calls.

Events that occur during the callback are registered in the [CallBackEvent table \(page 94\)](#). 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\_Current and Callback\_Historical. The two tables are identical; every 30 minutes, data for completed calls is pulled from Callback\_Current and moved to Callback\_Historical.

There is one row in this table for each callback that is made.

Field	Type	Null	Index	Description
SurrogateID	serial	No	PK	A Unique generated key to replace the CallGuid

Field	Type	Null	Index	Description
				throughout the Callback schema.
CallStartDate	date	No	PK	Date of the callback for data purging purposes
OldGuid	char(32)	Yes	No	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	Number to which the callback will be placed
CallBackType	char(1)	Yes	No	P = preemptive.  S = scheduled
Gateway	varchar(40)	Yes	No	Identifier for the gateway. Can be an IP address or other string identifier.
RecordingURL	nvarchar(255)	Yes	No	URL that points to a wav file of the recorded name of the caller
ScheduledCallBack DateTime	datetime year to second	Yes	No	Datetime (including timezone) for a scheduled callback. Not included for preemptive callbacks
ScheduledCallBackDN	varchar(32)	Yes	No	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	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
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. See Database Purge.

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	Date of the callback for data purging purposes
EventDateTime	datetime year to fraction(3)	No	No	Date and time of the event
CauseId	integer	Yes	Yes	See Values below. <a href="#">CauseRef (page 83)</a>
DBDateTime	datetime year to fraction(3)	No	No	The date and time of the database operation.
EventTypeId	integer	No	FK	The unique id of a event type. See <a href="#">ElementTypeRef (page 86)</a> .

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 untill a slot becomes available.

Field	Type	Null	Index	Description
SurrogateID	integer (serial)	No	PK, FK	
CallStartDate	date	No	PK	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
EnterDateTime	datetime year to second	No	No	Datetime entered queue
LeaveDateTime	datetime year to second	Yes	No	Datetime left queue
CVPEstimatedWaitTime	smallint	No	No	CVP-calculated estimated wait time (in seconds) Since enterdatetime. This will be generated during the insert, it will not be maintained.
ICMEstimatedWaitTime	smallint	No	No	Unified ICM-calculated estimated wait time (in seconds) Since enterdatetime. This will be generated during the insert, it will not be maintained.
ValidationStatus	smallint	No	No	The bitmask result obtained from the Validation method.  See sample code below.
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.

---

**Courtesy CallBack Tables**

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	EWI
00000000	01000000	PROBE_FAILED_NO_RESPONSE
00000000	10000000	PROBE_FAILED_NO_CONFIG
00000001	00000000	EXCEED_CAPACITY_GW
00000010	00000000	EXCEED_CAPACITY_QUEUE





# Chapter 5

## Managing the Database

---

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 Informix user.

This chapter contains the following topics:

- [Passwords, page 97](#)
- [Database Users, page 98](#)
- [Data Categories and Data Retention, page 99](#)
- [Database Purge, page 100](#)
- [Database Backup, page 102](#)
- [Database Compression, page 104](#)
- [Retries: Backup and Purge, page 105](#)
- [Database Recovery, page 106](#)
- [Failure and Restoration, page 106](#)

### 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),

then 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 Informix, Database Administrator, and [Database Users \(page 98\)](#) passwords before the passwords expire to avoid the possibility of data loss and/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 four categories of database users. The four categories are instance owner (Informix), database administrator, application user, and reporting user.

### Instance Owner

During the Unified CVP installation, a user named **informix** is created. This user owns the Informix instance and can run the IDS service. This account should never be used to access the Unified CVP database.

### Database Administrator

The **Informix user** 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 Unified IC data source.

## CVP Reporting Users

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.

Once 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 (CVP) Solution Reference Network Design Guide (SRND)*, have increased 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 under 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 occurs by the first purge to run after a date boundary has been crossed.

If the nightly purge is scheduled to run at 1:00 AM, it will typically perform this task. If the nightly purge is scheduled to run at 11:00 PM, then the next day will not occur until 11:00 AM 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.

## Scheduling Purges

Database purge is run from the Operations Console as follows:

1. Select **Device Management > CVP Reporting Server**.
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**.
3. At the Edit Reporting Server Configuration window, select the Database Administration menu in the toolbar, then select **Data Delete**.
4. On the Reporting Server - Data Delete page, change the data retention time for each category of data.
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.

6. Enter your Informix Password and click **Save & Deploy**.

Refer to the [Operations Console Online Help for Cisco Unified Customer Voice Portal](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_user_guide_list.html) ([http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products\\_user\\_guide\\_list.html](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 Purge and Midday Purge

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 below), in addition to other tasks like 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 logfile 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 from 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 'emergency' purge old data to create space for new data. Emergency purge is, therefore, a critical safety mechanism for Unified CVP.

If used space has grown past the system's high water mark, the user will be notified by an SNMP trap message and data will be purged. The SNMP notification will alert the user to the loss of data and request that they shrink their 'retention days' data settings.

Users should reduce the number of actual days of data retained until emergency purge is no longer required. Additionally, users can reduce the data generated by means of 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 there is limited space 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 can only operate 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 3 hours 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 done. The same memory limitations as described in the section "[Failure and Restoration](#)" (page 99) apply.
- **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 there are no status messages.

## Database Backup

Unified CVP allows users to turn the scheduling of data backups on or off, as well as 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 responsibility of the user. Data loss may occur if the backing up of files is not managed appropriately by the user.**

If Unified CVP backup scheduling is turned on, the backup frequency is once per day. Backups should be scheduled to run no sooner than 3 hours after the scheduled purge job.

Additionally, users can run a backup on demand—as long as another backup, or a purge, is not already running. All 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 will be available at any time on the local machine. Having two files is critical. If the system were to fail while writing a backup, and the failure was such that a restore is necessary the older backup file would be 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).

Unified CVP names backup files as follows. 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. Cisco Systems strongly recommends that customers 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.

**Note:** 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 will lock the file and prevent another backup from running.

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

There are two reasons why you would want to restore a backup image. The first would be in the event that older data on a backup image needs to be recovered. The second reason would be the case of a machine that has been rebuilt after a hardware failure, where you would want to recover as much data as possible.

**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 CVP is as follows:

- Step 1: Stop the CallServer process (Reporting Server).
- Step 2: Execute the script: `%CVP_Home%\bin\cvprestore.bat`
- Step 3: Restart the CallServer process.

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

For information on configuring backups, see the [Operations Console Online Help for Cisco Unified Customer Voice Portal](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_user_guide_list.html) ([http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products\\_user\\_guide\\_list.html](http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_user_guide_list.html)).

## Database Compression

In previous releases, selecting **Medium** for your Unified CVP database deployment (50 GB database size) required you to use a MCS 7845 H1/H2 with four 146.8 GB 15K SCSI discs for database storage at RAID level 10. The minimum free disk space requirement for a medium-size (50 GB) database was 250 GB.

Selecting **Large** for your CVP database (100 GB database size) deployment required you to use an MCS 7845 H2 with six 146.8 GB SAS discs at RAID level 10. The minimum free disk space requirement for a large-size database (100 GB) was 375 GB.

The rationale for using such large disc space was that in previous releases, the Unified CVP reporting server stored one live and two backup copies of the CVP Informix database on disc. You were responsible for moving/compressing these backups. If the backup database size was not managed, the potential existed for lost data due to the nightly purge process set up during reporting server configuration.

CVP 8.5(1) uses the gzip compression package (installed with the reporting server) to compress database backups. No other compression packages are supported. As the database is backed up, the data is streamed from the backup utility through gzip, resulting in a smaller backup file. All database backup files are compressed and stored on the reporting server in the `cvp_db_backup` folder.

Database backups can be scheduled or invoked on demand through the Unified CVP OAMP.

**Note:** Database compression uses system resources during the compression time. As a result, the reporting server might not be able to process high volume call data and to perform heavy database operations while the database backup and compression are taking place. Perform the backup and compression during off peak hours to minimize the impact to performance. Avoid



report generation during the database backup and compression as this could tax the system resources, which may lead to loss of call data.

Gzip supports compression rates of 25% - 30% on each of the database backup files. This allows the database to increase in size. The compressed backup is decompressed during a restore.

## Compression Ratio

The compression package targets a 25%-30% compression rate for database back compression. This allows a 100 GB backup database file to compresses to 30 GB, freeing up an additional 140 GB of disk space (2 backups x 70 GB) to store additional call data. As a result, the Reporting Database can double in size.

Selecting **Medium** for your database size now installs at 100 GB.

Selecting **Large** installs at 200 GB.

### Note:

- The **Small** selection is unchanged (2 GB) and is supported for lab deployments or demo purposes only.
- Unified CVP does not support manual upgrades of a Medium Reporting Server to a Large Reporting server. A customer who wishes to do this must uninstall the medium database and reinstall a large database. This will result in data loss, as you cannot restore a medium database on a large database. If you try to restore a medium database to a large database, the database size will revert to medium.

## Retries: Backup and Purge

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 **pu**ge), 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 over 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:** Prior to and 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 described above.
- 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 described above 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. They will be buffered in memory for as much as possible, but at some point they may be dropped either partially or fully.





# Chapter 6

## Reporting Best Practices

---

**Note:** You can find troubleshooting information for Unified CVP Reporting on the [Troubleshooting Doc Wiki site](http://docwiki.cisco.com/wiki/Main_Page) ([http://docwiki.cisco.com/wiki/Main\\_Page](http://docwiki.cisco.com/wiki/Main_Page))

This chapter contains the following topics:

- [Allow Only Reporting Users to Query the Database, page 109](#)
- [Assuring Accurate Time Stamps for Reporting and Logging, page 110](#)
- [CPU Intensive Reports, page 110](#)
- [Database Backup and Recovery, page 110](#)
- [Database Retention Settings, page 110](#)
- [Data Security, page 110](#)
- [Database Sizing Issues, page 111](#)
- [Filtering Data to be Stored in the Database, page 111](#)
- [Inclusive and Exclusive VXML Filters for Reporting, page 111](#)
- [Informix, Operating System Time, and Local Time, page 114](#)
- [Joining Unified CVP and SQL Server Data, page 115](#)
- [Password Policy, page 115](#)
- [Purge and Backup Database Maintenance Tasks, page 115](#)
- [Reporting Isolation Level, page 115](#)
- [Synchronizing Timestamps, page 115](#)
- [Writing Efficient SQL Queries when Creating Reports, page 116](#)
- [Zero Duration Calls and Writing Reports, page 117](#)

### Allow Only Reporting Users to Query the Database

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

## Assuring Accurate Time Stamps for Reporting and Logging

Unified CVP components do not themselves synchronize machine times. However, customers must provide a cross-component time synchronization mechanism, such as 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 (CVP) Release 8.x Solution Reference Network Design (SRND)* document.

## Database Backup and Recovery

Issues to keep in mind are:

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

These issues are discussed in ["Database Backup" \(page 102\)](#) and ["Database Recovery" \(page 106\)](#).

## 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 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, tag these two variables as not persistent

in Unified ICME 7.1(1). 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, would cut 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 number of credit card number, can be set not to be logged, providing a security layer in case logs are compromised.

## Database Sizing Issues

See the discussion in "[Data Retention](#)" (page 99). Also see the *Planning Guide for Cisco Unified Customer Voice Portal* and the *Cisco Unified Customer Voice Portal (CVP) Release 7.x Solution Reference Network Design (SRND)* document.

## Filtering Data to be Stored in the Database

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 [Operations Console Online Help for Cisco Unified Customer Voice Portal](#) ([http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products\\_user\\_guide\\_list.html](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.

## Procedure

To configure inclusive and exclusive filters for a Reporting Server:

- 
- 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.

---

### See Also

[Rules for VXML Inclusive and Exclusive Filters \(page 112\)](#)

[Examples of VXML Filter Wildcard Matching \(page 113\)](#)

[Examples of Inclusive and Exclusive VXML Filters for Reporting \(page 113\)](#)

## Rules for VXML Inclusive and Exclusive Filters

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 single 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.

## Examples of VXML Filter Wildcard Matching

The table below provides examples of VXML filter wildcard matching.

**Table 2: 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 <b>var</b> .
MyApplication.*.*.3	Matches all fields in MyApplication that end with <b>3</b> .
MyApplication.*.*.SESSION:Company	Matches the Company session variable in MyApplication.

## Examples of Inclusive and Exclusive VXML Filters for Reporting

The table below 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 3: 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

## Informix, Operating System Time, and Local Time

Inclusive Filter	Exclusive Filter	Data the VXML Server Feeds To the Reporting Server
<b>Application1.*.*.*</b>	<b>*.*.Element1.*;</b> <b>*.*.Element2.*;</b> <b>*.*.*.Field1</b>	All Application1 data, except Element1, Element2, and Field1
<b>Application1.*.*.*</b>	<b>*.voice.*.*</b> which matches Element3 and Element4	All Application1 data, except Element3 and Element4
<b>*.*.Element1.*;</b> <b>*.*.Element2.*;</b> <b>*.*.*.Field1</b>	<b>Application1.*.*.*</b>	No data for Application1. Other Data for other applications, such as Application2, which contain Element1, Element2 and Field1, will be fed.
<b>*.voice.*.*</b> which matches Element1, Element2, Element3, and Element4	<b>*.*.Element3.*;</b> <b>*.*.Element4.*</b>	Only Element1 and Element2 and all applications.
<b>*.voice.*.*</b> which matches Element1 and Element2	<b>*.*.*.Field1</b>	Element1 and Element2, except for Field1, if it exists in those elements
<b>*.*.Element1.*</b>	None	Element1
<b>*.*.Element1.*</b>	<b>*.*.*.Field1</b>	Element1, except for Field1 if it exists in Element1
<b>*.*.*.Field1</b>	<b>*.*.Element3.*;</b> <b>*.*.Element4.*</b>	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 OSs are set to the same time zone. This helps avoid confusion.

## Joining Unified CVP and SQL Server Data

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

## Password Policy

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.

### See Also

[Passwords \(page 97\)](#)

## 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 ScheduledTasks 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 ScheduledTasks to ensure the Last RunTime was as expected and there are no status messages.

## Reporting Isolation Level

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

## Synchronizing Timestamps

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:

Examples: **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.

- Do not ever set isolation level to repeatable read.
- 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, `EndTime` is set to the same value as `StartTime`. 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.



# Index

---

- ActionTypeRef table....[82](#)
- Active Directory
  - and supervisors....[14](#)
  - configuring....[14](#)
- adding
  - data sources....[15](#)
  - reporting users....[13](#)
  - super users....[13](#)
- Administration UI
  - logging in....[12](#)
  - upload license....[12](#)
- advanced filters....[23](#)
- analytics engine....[53](#)
- ApplicationSummary\_15 table....[77](#)
- ApplicationSummary\_Daily table....[78](#)
- ApplicationSummary\_Weekly table....[78](#)
- Application Summary reports....[35](#)
- application user....[98](#)
- Available Fields....[29](#)
  - and footers....[29](#)
  - and report summaries....[29](#)
- backups....[8](#), [97](#), [102](#), [110](#), [115](#)
  - retries....[105](#)
- basic report filters....[22](#)
- best practices
  - license....[11](#)
  - running reports....[110](#)
  - SQL queries....[116](#)
- bypass filter....[28](#)
- Call\_15 table....[79](#)
- Call\_Daily table....[79](#)
- Call\_Weekly table....[80](#)
- CallBackContext table....[92](#)
- CallBackEvent table....[92](#), [94](#)
- CallBackQueue table....[92](#), [94](#)
- Callback reports....[50](#)
- Callback table....[92](#)
- CallBack table....[92](#)
- Call category....[99](#)
- CallCMInfo Table....[64](#)
- Call Detail report....[41](#), [47](#)
- Call Event category....[99](#)
- CallEvent table....[59](#), [63](#), [64](#)
- CallICMInfo table....[64](#), [66](#)
- Call report....[38](#)
- Call Server....[6](#)
- CallStartDate....[57](#)
- Call table....[59](#), [61](#), [67](#)
- Call Traffic charts....[49](#)
- CallTypeRef table....[82](#)
- CauseRef table....[83](#)
- Chart Editor....[26](#)
- Cisco Unified Voice Operating System....[10](#)
- collections....[20](#)
- controller node....[10](#)
- courtesy callback....[92](#)
- Current fields....[29](#)
- cvp\_dbuser....[98](#)
- dashboard designer user....[33](#)
- dashboards....[19](#)
- data
  - filtering....[111](#)
  - retention....[110](#), [111](#)
  - security....[110](#)
- database....[5](#)
  - backups....[110](#)
  - multiple....[7](#)

- recovery....[110](#)
- sizing....[110](#), [111](#)
- database administrator....[98](#)
- database backup (see backups)....[102](#)
- database connection failure....[106](#)
- database purge (see purging)....[99](#)
- database recovery....[106](#)
- databases
  - joining....[115](#)
- database tables
  - ActionTypeRef....[82](#)
  - ApplicationSummary\_15 table....[77](#)
  - ApplicationSummary\_Daily table....[78](#)
  - ApplicationSummary\_Weekly table....[78](#)
  - Call....[61](#)
  - Call\_15....[79](#)
  - Call\_Daily....[79](#)
  - Call\_Weekly....[80](#)
  - CallBack....[92](#)
  - CallBackEvent....[94](#)
  - CallBackQueue....[94](#)
  - CallEvent....[63](#)
  - CallICMInfo....[64](#)
  - CallTypeRef....[82](#)
  - CauseRef....[83](#)
  - Device....[86](#)
  - ElementTypeRef....[86](#)
  - EventTypeRef table....[87](#)
  - keys....[56](#)
  - OutgoingECCVariable....[66](#)
  - QueueRef....[88](#)
  - Resource....[88](#)
  - Result Ref....[89](#)
  - SubSystemTypeRef....[89](#)
  - TransferTypeRef....[90](#)
  - Usage....[81](#)
  - UserInputModeRef table....[90](#)
  - VarDataTypeRef table....[90](#)
  - VoiceActionTypeRef table....[91](#)
  - VXMLCustomContent table....[68](#)
  - VXMLElement....[68](#)
  - VXMLElementDetail....[69](#)
  - VXMLElementFlag....[70](#)
  - VXMLError table....[71](#)
  - VXMLHotEvent table....[71](#)
  - VXMLHotLink....[72](#)
  - VXMLSession table....[73](#)
  - VXMLSessionVariable....[74](#)
  - VXMLVoiceInteractDetail table....[75](#)
- database users (see users)....[98](#)
- data categories....[99](#)
- data retention....[99](#)
- data sources....[15](#), [19](#), [53](#)
  - adding....[15](#)
- DBDateTime....[57](#)
- Device table....[86](#)
- ECC variable....[66](#), [67](#)
- ECC variables....[110](#)
- editing reports....[27](#)
- ElementTypeRef table....[86](#)
- emergency purge (see purging)....[101](#)
- Entity-Relationship diagram....[54](#)
- ER diagram....[54](#)
- EventDateTime....[57](#)
- EventTypeRef table....[87](#)
- exporting
  - reports to Excel....[27](#)
  - report xml....[31](#)
- failure
  - database connection....[106](#)



- Reporting Server....[106](#)
- filtering data....[111](#)
- filtering reports
  - advanced filters tab....[23](#)
  - basic filters tab....[22](#)
  - bypass filter....[28](#)
  - stored procedures....[24](#)
- filters for reporting
  - VXML Server....[111](#)
- footers
  - and Availalbe Fields....[29](#)
- foreign keys....[56](#)
- Gauge Editor....[26](#)
- Grid Editor....[26](#)
- grouping report fields....[30](#)
- IBM Informix Dynamic Server database (see Informix)..[5](#)
- ICM....[66](#)
- ICM Service....[6](#)
- IDS (see Informix)....[5](#)
- IMS users....[33](#)
- informix....[98](#)
- Informix database (see database)....[5](#)
- Informix date and time....[57](#), [114](#)
- Informix user....[97](#), [98](#)
- instance owner....[98](#)
- isolation level....[115](#)
- IVR Service....[6](#)
- license
  - best practices....[11](#)
  - obtaining....[11](#)
  - replication....[12](#)
  - uploading from Administration UI....[12](#)
- login
  - for supervisors....[14](#)
  - to the Administration UI....[12](#)
  - to Unified IC Reporting....[14](#)
  - login user....[32](#)
- member nodes....[10](#)
- MS SQL queries
  - in Report Viewer....[27](#)
- Operations Console....[97](#)
- OutgoingECCVariable table....[66](#), [67](#)
- passwords....[98](#), [115](#)
- popouts....[27](#)
- primary keys....[56](#)
- printing
  - reports....[27](#)
- purge....[115](#)
- purging....[8](#), [97](#), [99](#)
  - emergency....[101](#)
  - retries....[105](#)
- queries
  - writing....[116](#)
- QueueRef table....[88](#)
- recovery....[106](#)
- reordering
  - report fields....[30](#)
- replication
  - of license....[12](#)
- report definitions....[9](#), [19](#), [28](#)
- report designer user....[33](#)
- report fields
  - Available....[29](#)
  - Current....[29](#)
  - grouping....[30](#)
  - removing....[30](#), [31](#)
  - reordering....[30](#)
  - thresholds....[29](#)
- reporting engine....[53](#)
- reporting server....[5](#)

- Reporting Server....[7](#)
  - failure....[106](#)
- Reporting Service....[6](#)
- reporting users....[13](#), [99](#)
- reports....[19](#)
  - CPU-intensive....[110](#)
  - default view....[28](#)
  - editing....[27](#)
  - exporting to Excel....[27](#)
  - exporting xml file....[31](#)
  - filtering....[22](#)
  - generating....[109](#)
  - printing....[27](#)
  - removing and renaming fields....[29](#)
  - repeatable read....[115](#)
  - running....[21](#)
  - saving....[26](#)
  - see also report templates....[21](#)
  - viewing SQL....[27](#)
- report templates....[9](#), [20](#)
  - Application Summary reports....[35](#)
  - Callback reports....[50](#)
  - Call Detail report....[41](#)
  - Call report....[38](#)
  - Call Traffic reports....[47](#)
  - importing....[18](#)
  - locating....[17](#)
  - Trunk Group Utilization report....[51](#)
- report viewer....[26](#)
- Report Viewer
  - popouts....[27](#)
  - views....[27](#)
- Report Views (see also Views Editor)....[20](#)
- Resource table....[88](#)
- Result Ref table....[89](#)
- retries
  - backup and purge....[105](#)
- saving
  - reports....[26](#)
- schema (see database schema)....[5](#)
- security....[109](#), [110](#)
- security administrator user....[32](#)
- SIP calls....[59](#)
- SIP Service....[6](#)
- sizing
  - reporting....[7](#)
- SQL Server
  - joining databases....[115](#)
  - queries....[116](#)
- stored procedures....[24](#)
- SubSystemTypeRef table....[89](#)
- super users....[13](#), [33](#)
  - adding....[13](#)
- synchronized clocks....[115](#)
- system administration user....[34](#)
- system application user....[34](#)
- system configuration administrator user....[32](#)
- thresholds....[20](#), [29](#)
- time
  - time stamps....[110](#)
- timestamps
  - synchronized....[115](#)
- timezones
  - and data sources....[17](#)
- TransferTypeRef table....[90](#)
- Trunk Group Utilization report....[51](#)
- trunk utilization....[60](#)
- UCCE reporting supervisors....[34](#)
- Unified CVP database....[53](#), [61](#)
  - datetime columns....[57](#)

- Informix dates and times....[57](#)
- Unified CVP reporting users
  - authentication for Unified IC....[14](#)
- Unified CVP report templates
  - Application Summary reports....[35](#)
  - Callback reports....[50](#)
  - Call Detail report....[41](#)
  - Call report....[38](#)
  - Call Traffic reports....[47](#)
  - importing....[18](#)
  - obtaining....[17](#)
  - Trunk Group Utilization report....[51](#)
- Unified IC
  - administration (OAMP) application....[10](#)
  - controlller node....[10](#)
  - data sources....[15](#)
  - license....[11](#)
  - member nodes....[10](#)
  - reporting application....[10](#)
  - super users....[13](#)
  - user list....[15](#)
  - user roles....[15](#), [32](#)
- Unified IC Reporting
  - logging in....[14](#)
- Unified IC reporting users....[15](#), [18](#)
  - user roles....[15](#)
- Usage table....[60](#), [81](#)
- User Groups....[20](#)
- UserInputModeRef table....[90](#)
- User Permissions....[20](#)
- User Roles....[21](#)
- users....[98](#)
  - application user....[98](#)
  - Informix user....[98](#)
  - instance owner....[98](#)
  - reporting user....[99](#)
- UTC....[114](#)
- value list collection designer user....[33](#)
- value lists....[20](#)
- VarDataTypeRef table....[90](#)
- viewing
  - report SQL....[27](#)
- views....[20](#)
- views on Report Viewer....[27](#)
- VoiceActionTypeRef table....[91](#)
- VoiceML Element Detail category....[99](#)
- VoiceXML ECC Variable category....[99](#)
- VoiceXML Element category....[99](#)
- VoiceXML filters
  - example inclusive and exclusive....[113](#)
  - example wildcards....[113](#)
  - rules....[112](#)
- VoiceXML Interact Detail category....[99](#)
- VoiceXML Session category....[99](#)
- VoiceXML Session Variable category....[99](#)
- VOS....[10](#)
- VXML calls....[67](#)
- VXMLCustomContent table....[68](#)
- VXMLElementDetail table....[69](#)
- VXMLElementFlag table....[67](#), [70](#)
- VXMLElement table....[68](#)
- VXMLError table....[71](#)
- VXMLHotEvent table....[71](#), [72](#)
- VXMLHotLink table....[72](#), [73](#)
- VXML Server....[6](#)
  - example filter wildcards....[113](#)
  - example reporting filters....[113](#)
  - filters for reporting....[111](#)
  - rules for reporting filters....[112](#)
- VXMLSession table....[67](#), [73](#)

VXMLSessionVariable table....[74](#)

VXMLVoiceInteractDetail table....[75](#)

zero duration calls....[117](#)