Notice
While reasonable efforts were made to ensure that the information in this document was complete and accurate at the time of printing, Avaya Inc. can assume no liability for any errors. Changes and corrections to the information in this document may be incorporated in future releases.

Documentation disclaimer
Avaya Inc. is not responsible for any modifications, additions, or deletions to the original published version of this documentation unless such modifications, additions, or deletions were performed by Avaya.

Link disclaimer
Avaya Inc. is not responsible for the contents or reliability of any linked Web sites referenced elsewhere within this Documentation, and Avaya does not necessarily endorse the products, services, or information described or offered within them. We cannot guarantee that these links will work all of the time and we have no control over the availability of the linked pages.

License
USE OR INSTALLATION OF THE PRODUCT INDICATES THE END USER'S ACCEPTANCE OF THE TERMS SET FORTH HEREIN AND THE GENERAL LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE AT http://support.avaya.com/LicenseInfo/ ("GENERAL LICENSE TERMS"). IF YOU DO NOT WISH TO BE BOUND BY THESE TERMS, YOU MUST RETURN THE PRODUCT(S) TO THE POINT OF PURCHASE WITHIN TEN (10) DAYS OF DELIVERY FOR A REFUND OR CREDIT.

Avaya grants End User a license within the scope of the license types described below. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the Documentation or other materials available to End User. "Designated Processor" means a single stand-alone computing device. "Server" means a Designated Processor that hosts a software application to be accessed by multiple users. "Software" means the computer programs in object code, originally licensed by Avaya and ultimately utilized by End User, whether as stand-alone Products or pre-installed on Hardware. "Hardware" means the standard hardware Products, originally sold by Avaya and ultimately utilized by End User.

License type(s)
Concurrent User License (CU). End User may install and use the Software on multiple Designated Processors or one or more Servers, so long as only the licensed number of Units are accessing and using the Software at any given time. A "Unit" means the unit on which Avaya, at its sole discretion, bases the pricing of its licenses and can be, without limitation, an agent, port or user, an e-mail or voice mail account in the name of a person or corporate function (eg., webmaster or helpdesk), or a directory entry in the administrative database utilized by the Product that permits one user to interface with the Software. Units may be linked to a specific, identified Server.

Copyright
Except where expressly stated otherwise, the Product is protected by copyright and other laws respecting proprietary rights. Unauthorized reproduction, transfer, and or use can be a criminal, as well as a civil, offense under the applicable law.

Third-party components
Certain software programs or portions thereof included in the Product may contain software distributed under third party agreements (“Third Party Components”), which may contain terms that expand or limit rights to use certain portions of the Product (“Third Party Terms”). Information identifying Third Party Components and the Third Party Terms that apply to them is available on Avaya’s web site at: http://support.avaya.com/ThirdPartyLicense/.

**Avaya fraud intervention**

If you suspect that you are being victimized by toll fraud and you need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. Suspected security vulnerabilities with Avaya Products should be reported to Avaya by sending mail to: securityalerts@avaya.com.

For additional support telephone numbers, see the Avaya Web site: http://www.avaya.com/support.

**Trademarks**

Avaya is a trademark of Avaya Inc.

All non-Avaya trademarks are the property of their respective owners.

**Avaya support**

Avaya provides a telephone number for you to use to report problems or to ask questions about your contact center. The support telephone number is 1-800-242-2121 in the United States. For additional support telephone numbers, see the Avaya Web site: http://www.avaya.com/support.

For the most current versions of documentation, go to the Avaya support Web site: http://www.avaya.com/support.
Preface

Contents

This following few pages provides information that will help you use this document.

<table>
<thead>
<tr>
<th>Preface</th>
<th>iv</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related Documents</td>
<td>vi</td>
</tr>
<tr>
<td>Product Name Changes</td>
<td>vi</td>
</tr>
<tr>
<td>Knowledge Base</td>
<td>vi</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 1 System Requirements</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation Requirements</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 2 Introduction</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the Interaction Data Service?</td>
<td>11</td>
</tr>
<tr>
<td>Error Logging</td>
<td>13</td>
</tr>
<tr>
<td>Licensing</td>
<td>13</td>
</tr>
<tr>
<td>Setting Up Your Interaction Data Service</td>
<td>14</td>
</tr>
<tr>
<td>Install Interaction Data Service</td>
<td>15</td>
</tr>
<tr>
<td>Create/Upgrade Databases</td>
<td>23</td>
</tr>
<tr>
<td>Delete Record Types from Database</td>
<td>26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 3 Administration</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Interaction Data Server - Voice and Presence</td>
<td>29</td>
</tr>
<tr>
<td>Configure Interaction Data Server - Multimedia</td>
<td>35</td>
</tr>
<tr>
<td>Configure Interaction Data Server - View</td>
<td>40</td>
</tr>
<tr>
<td>Start Service</td>
<td>45</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 4 Developing Client Applications</th>
<th>47</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>48</td>
</tr>
<tr>
<td>Real Time Reporting Data</td>
<td>51</td>
</tr>
<tr>
<td>Device</td>
<td>52</td>
</tr>
<tr>
<td>Agent</td>
<td>54</td>
</tr>
<tr>
<td>VDN</td>
<td>57</td>
</tr>
<tr>
<td>Queue</td>
<td>59</td>
</tr>
</tbody>
</table>
Historical Reporting Data ................................................................. 61
Agent ......................................................................................... 61
Skill ......................................................................................... 61
Queue ......................................................................................... 61
Interaction .................................................................................. 62
Conversation & Customer ........................................................... 63
Miscellaneous Capabilities .......................................................... 64
Enumerations .............................................................................. 66

Chapter 5  Reporting via Avaya CMS  .................................................. 67
Overview ....................................................................................... 68
Install & Configure Multimedia CMS Plug-in .............................. 69
Run CCE Queue Data Report ....................................................... 73

Index ......................................................................................... 75
Related Documents

There are many documents that are helpful when setting up a CTI environment. The following are included on the Avaya Application Enablement Services CD-ROM:

- *Application Enablement Services: Administration and Maintenance Guide* (AES Admin Guide.pdf)
- *Application Enablement Services: Installation Guide Software-Only Offer* (AES Installation Guide SW only.pdf)
- *Application Enablement Services: Overview* (AES Overview.pdf)
- *Application Enablement Services: Document Roadmap* (AES Roadmap.pdf)
- *White paper on AE Services 3.0 Support for Communication Manager Enterprise Survivable Server (ESS) and Local Survivable Processor (LSP)* (AES Whitepaper on CM Survivability.pdf)
- *White-paper on Security in Application Enablement Services for Bundled and Software only solutions* (AES Whitepaper Security.pdf)
- *Application Enablement Services Readme* (AES_ReadMe_301.pdf)

For more documentation on Application Enablement Services, refer to Avaya's Support Center website http://www.avaya.com/support.

Product Name Changes

**Avaya Contact Center Express**

The Avaya Contact Center Express suite of CTI applications and development tools was previously known as Avaya Active Telephony.

Active Telephony was previously known as Avaya Active Enterprise.

**Avaya Telephony Services**

Avaya Application Enablement Services (AE Services) was previously known as Avaya Computer Telephony (Avaya CT) software.

Avaya CT was previously known as CentreVu Computer Telephony (CentreVu CT).

Knowledge Base

For information on any errors and updates relating to this document, visit the *Avaya Contact Center Express Knowledge Base* (http://support.avayacce.com).
C H A P T E R 1

System Requirements

In This Chapter

Installation Requirements...................................................7

Installation Requirements

The specifications listed here are the minimum recommended. As far as is known and unless otherwise stated, Contact Center Express and its components should be compatible with all higher specification hardware configurations and software versions that those listed here.

Note: The following are not yet supported

- Microsoft .Net Framework 3.0
- Microsoft .Net Framework 3.5
- SQL Server 2008
Desktop

For Contact Center Express Desktop, Contact Center Express Reporting, Contact Center Express Control Panel and Supervisor.

**Hardware:** 1.6 GHz Pentium, 512MB of RAM, 50MB of free hard disk space, DVD drive, graphics card capable of supporting 1024x768 resolution monitor, mouse or other Windows-compatible pointing device, and TCP/IP LAN connection.

**Software:** Either Windows Vista (Enterprise), Windows XP Professional 32-bit SP2/SP3, Citrix Presentation Server 3.0.2 or Windows Terminal Services 32-bit with Application Enablement Services (AE Services) client software release 4.2.1, Microsoft Internet Explorer 6.0 SP1 and Microsoft .Net Framework 2.0 or 2.0 SP1.

Server

For License Director, XML Server, Configuration Server, Application Management Director, Media Director, all Media Stores and Gateways, Call Routing Server and Virtual Agent.

**Hardware:** A 2.4 GHz Pentium with 2GB of RAM.

**Software:** Either Windows 2003 Server (Enterprise & Standard) 32-bit, Windows 2008 Server (Enterprise & Standard) 32-bit with Application Enablement Services (AE Services) client software release 4.2.1, Microsoft Internet Explorer 6.0 SP1 and Microsoft .Net Framework 2.0 or 2.0 SP1. VMWare and Virtual Server are also supported.

Interaction Data Service

The Interaction Data Service requires its own dedicated server with the same minimum requirements as the server specifications listed above - excluding Microsoft Internet Explorer which is not required.

Database

Interaction Data Server (ActiveInteractionData) and Configuration Server (ACS) databases may be installed on either Microsoft SQL Server 2008 or Microsoft SQL Server 2005. ASMSControl, ASMSData and ASContact Databases must be installed on either Microsoft SQL Server 2005 or Microsoft SQL Server 2005 Express. For best performance host Microsoft SQL Server on a dedicated machine and ensure that both the Interaction Data Server and ASContact database are run on identical operating systems.

Email Server


Developers

**Developing applications:** Windows XP Professional 32-bit SP2/SP3 with Microsoft Visual Studio 2005, Application Enablement Services (AE Services) client software release 4.2.1, Microsoft Internet Explorer 6.0 SP1 and Microsoft .Net Framework 2.0 or 2.0 SP1.
CHAPTER 2

Introduction

This chapter explains the function of the Interaction Data Server - Voice and Presence.

In This Chapter

What is the Interaction Data Service? ........................................ 11
Setting Up Your Interaction Data Service ............................ 14
Install Interaction Data Service ............................................. 15
Create/Upgrade Databases .................................................. 23
Delete Record Types from Database ................................. 26
What is the Interaction Data Service?

The Interaction Data Service is the suite name for three Contact Center Express products:

- Interaction Data Server - Voice and Presence
- Interaction Data Server - Multimedia, and
- Interaction Data Server - View

All three servers work together to gather, store and display statistical data on Contact Center Express voice and multimedia work items. All use a single database (ActiveInteractionData).

**Interaction Data Server - Voice and Presence.** This server monitors VDNs, splits/skills, trunk groups and agent extensions to gather detailed statistical information about all facets of a call. The server receives real-time information from your Avaya Communication Manager and, based on the regularity you specify, calculates statistics such as talk and wait-time averages. Data is stored in the ActiveInteractionData database.

**Interaction Data Server - Multimedia.** This server receives a stream of event information from all Contact Center Express media stores and Media Directors and aggregates this into real-time and historical information on all multimedia work items that flow through your call center. Data is stored in the ActiveInteractionData database.

**Interaction Data Server - View.** This server allows you to access the statistical data generated by Interaction Data Server - Voice and Presence and Interaction Data Server - Multimedia and to display that data on individual client PCs. Interaction Data Server - View is currently used by Contact Center Express's premier desktop application Contact Center Express Desktop, and it can also be used by developers to build their own client applications.

Note: Interaction Data Server - View replaces the current Interaction Data Client which will be depreciated from version 3.1 onward.
Contact Center Express also makes some of the multimedia reporting data held within the Interaction Data Service easily available to the Avaya Call Management System (Avaya CMS) package. For more information, see the chapter *Reporting via Avaya CMS* (on page 67).
Error Logging

Interaction Data Server - Voice and Presence, Interaction Data Server - Multimedia and Interaction Data Server - View log error information relating to their own operation to a series of log files.

A new log file is created for each day of the week. The name of the error log file clearly identifies the application and the day of the week, for example, MonAIDS.log, MonASIDSMultiMedia.log and MonASView.log.

Once a log file reaches the size limit specified in its configuration, it is archived and a new file is created. (Note: The archive only stores one log file. If the second error log reaches the specified maximum size, it overrides the previously archived log file.)

The types of errors logged by the error log are determined by the logging level retrieved from the application's configuration data. Levels of error logging are:

- **0.** No error logging takes place.
- **1.** Logs fatal, major, minor and trace information.
- **2.** Logs fatal and major errors.
- **4.** Logs fatal and major errors.
- **8.** Logs fatal errors only.

Each file records the selected logging level as well as the date, time, location and description of every error that occurs.

Error log files are automatically saved into the application's current working folder (the same folder as the application executable).

Diagnostic Testing Error Logging

There is one other error log level, which enables you to create log files that don't override each other every time the maximum log file size limit is reached. This logging level is designed for diagnostic purposes only and can be achieved by adding 128 to one of the logging level values mentioned above. For example, if you specify `Error Log Level=129`, new error log files will be continuously created for this application that contain fatal, major, minor and trace information.

Each new file has a unique name based on the date (year, month and day) and time (in hours, minutes and seconds) it was created. For example: 20061116155404ASMediaDirector.log.

**Warning:** To prevent a multitude of log files affecting your available disk space, this error logging level should only be used for short periods of time while carrying out diagnostic testing.

Licensing

To run Interaction Data Service, you do not need to purchase additional Contact Center Express licenses.
Setting Up Your Interaction Data Service

Avaya Communication Manager and Telephony Server

The ActiveInteractionData database depends on the Universal Call ID (UCID) to relate different tables in the database.

For the database to work correctly, you must configure each Avaya Communication Manager to have 'Create UCID' turned on, and also add a valid UCID network node ID.

Additionally, you need to configure the switch to pass the UCID to the Telephony Server ('Send UCID to ASAI?' is set to yes).

For more information, refer to the *Definity production documentation* (see "Related Documents" on page vi) (Installation/Administration).

➢ **To set up your Interaction Data Service environment:**

1. Configure your Avaya Communication Manager to add UCID to its event records and pass on the UCID to the Telephony Server.


3. Create the ActiveInteractionData database.


Install Interaction Data Service


The Interaction Data Service should run on a server operating system: Microsoft Windows 2003 Server (Enterprise & Standard) or 2000 Server SP4.

The server's database can run with Microsoft SQL Server 2005 or 2000 SP3a, or the truncated versions, SQL Server 2005 Express or MSDE 2000 SP3a. **Note:** The SQL Server must be configured for SQL Server and Windows (mixed mode authentication) and not Windows only authentication:

➢ **To install Interaction Data Server - Voice and Presence, Interaction Data Server - Multimedia, Interaction Data Server - View and related database scripts:**

1. Close any applications you have open.
2. Insert the Contact Center Express DVD into your computer’s DVD drive.
3 Select the language you would like to use for the installation from the drop-down list box.

4 Click Server.
5 Click *Interaction Data Service*. The Interaction Data Service setup screen appears.

6 Click **Next** to continue. The *License Agreement* screen appears.
7 Read through the license agreement and, if you are happy with the terms, click **Yes**.

![Interaction Data Server - Configure Installation Location](image1)

8 To select a different install location, click the **Browse** button. Otherwise, click **Next** to continue.

![Interaction Data Server - Configure Interaction Data Server](image2)

9 Use the following definitions to configure Interaction Data Server - Voice and Presence.

*Note: The information you enter in this screen is saved into the application's configuration .ini file. If you would rather enter the data directly into the .ini file at a later stage, click **Next** to by-pass this screen and continue the install.*

**Interaction Data Server ID.** A unique number that identifies the Interaction Data Server - Voice and Presence.

**Interaction Data Server Port.** The port number the Interaction Data Server - Voice and Presence uses to accept connections from clients and other servers. The default is 29090.
**Switch ID.** The ID number of the switch the Interaction Data Server - Voice and Presence is connected to. The number should be unique. The switch ID is particularly important in complex configurations, where there is a network of Interaction Data Server - Voice and Presence servers receiving information from multiple Avaya switches.

**Primary Link Name.** The name of the primary link this application will use to connect to the Avaya Telephony Server and switch.

**Primary User Name.** The user name required to gain access to the primary link.

**Primary User Password.** The password associated with above user name. By default, the Contact Center Express application will encrypt this data. For more information, see the *Contact Center Express Installation Guide* (Configuration Commands).

10 To configure switch information, click More >.

11 To specify which:
- VDNs you want the Interaction Data Server - Voice and Presence to monitor, type a VDN number in the VDN text box and click Add.

- split/skills you want the Interaction Data Server - Voice and Presence to monitor, type a split/skill number in the ACD Split/Skill text box and click Add.

- station extensions you want the Interaction Data Server - Voice and Presence to monitor, type an extension in the Station text box and click Add.

- trunk groups you want the Interaction Data Server - Voice and Presence to monitor, type a trunk group number in the Trunk Group text box and click Add.

12 To remove an item from any list, select it and click Delete.

13 Click the Exit button to return to the Configure Interaction Data Server - Voice and Presence screen.

14 Click Next to continue.

15 Use the following definitions to configure the Interaction Data Service.
**Database Server Name.** The name of the server on which the 'ActiveInteractionData' database is located. This is used by Interaction Data Server - Voice and Presence, Interaction Data Server - View and Interaction Data Server - Multimedia.

**Database Server Name.** The name or TCP/IP address of the server on which the Avaya CMS Informix database is located. This information is used by the CMS Plug-in for Interaction Data Server - Multimedia.

**Database User Name.** A valid user name for the CMS Informix database.

**Database User Password.** The password associated with the above user name.

16 Click **Next** to continue.

17 Click **Next** to install sample applications that demonstrate how to use Interaction Data Service.
18 Review your install settings and click **Next** to start the install.

19 Click **Finish**.

20 If you are asked to restart your machine (this will happen if some application components need updating or registering), click **Yes**.
Create/Upgrade Databases

Installing SQL Server 2005 Express

If you are installing SQL Server 2005 Express as part of your Contact Center Express installation, you must go through a few steps before using Contact Center Express's database script files. For information, see the Install SQL Server 2005 Express procedure in the Contact Center Express Installation Guide.

Running Contact Center Express database scripts

Contact Center Express provides a set of database script files (.sql extension) and batch files (.bat extension) for the creation and maintenance of all databases used by its applications:

- Interaction Data Service
- Configuration Server
- ASContact Database
- ASMediaStore

Please note that from release 4.0, these database scripts default Contact Center Express database collations to that of your SQL Server collation. Future database upgrades will rely on your Contact Center Express database collation remaining the same as the server collation; and that the server system databases also default to the server collation. Before proceeding, make sure your server collation is correct for your locale and likely usage.

You will find these scripts in a folder named SQL Script under the individual application folder structure. The folders reside on the distribution DVD and are also installed when you install these applications.

Run the scripts to either create or maintain the database by double clicking the Run AS Maintain Database.bat batch file in the relevant SQL Script folder.

If your SQL Server resides on:

- a different machine to the Contact Center Express application, you must run the Run AS Maintain Database.bat batch file from the DVD folder structure.
- the same machine as the Contact Center Express application, you can run the Run AS Maintain Database.bat batch file from either the DVD or the installed folder structure.
- the same machine as the Contact Center Express application, but the SQL Server is not the default instance (as for SQL Server 2005 Express) you must first modify the Run AS Maintain Database.bat batch file. The second to last line in this file reads:

  SET ASServerName=

  which designates the default SQL instance. Change this line to equal the server name \ specific instance. For example to run SQL Server 2005 Express on the machine shown in the screen shot below, this line becomes:

  SET ASServerName=MWERENX7000\SQLEXPRESS

If you are in any doubt as to the machine and SQL instance names, running SQL Server Management Studio Express will show you what SQL named instances are running on your machine.
This records information about your current folder and then runs a batch file which in turn runs a sql script (AS Maintain Database.sql) - both of which reside in the Utilities folder. This suite of batch and script files reads a control file (AS Script Master.txt) that directs, based on the current state of the database, the specific script files that will be run for the database.

With each release of Contact Center Express you must run Run AS Maintain Database.bat for the application databases you are already using to upgrade the database to the current version. You must also run this batch file for databases you would like to install for the first time (this will create the database).

After you run Run AS Maintain Database.bat, you can view the message log (ASDatabaseLog_YYYYMMDD_HHMMS.txt) which summarizes the processes that have taken place. Note that YYYYMMDD and HHMMSS will be replaced by the date and time you ran the job. This log file will appear in the folder from which you ran the batch file; or, if you ran it from the DVD, in the Temp folder of the Windows directory of your database server. The message shown by Run AS Maintain Database.bat shows the exact location of the log file.

Please note that the log file should end with the message “**Job SUCCEEDED to completely maintain ........ database ...”. If it ends with “**Job FAILED to completely maintain ...”, read the detail of the log file, make necessary adjustments and run Run AS Maintain Database.bat again. Running it additional times will not harm the database - it will do nothing if there is nothing to do.
The suite of scripts creates a table tblASSystem that holds the current database version and logs all attempts (suContact Center Expressssful or otherwise) to modify the database with the scripts.

**Media Store Conversion**

If you have 2.x or an earlier version of Contact Center Express media store databases, refer to Upgrade 2.x Media Store Databases to 3.0.1.

**Backup/Restore/Delete Databases**

If you are using SQL Server 2005, SQL Server 2000 or either of the truncated versions (SQL Sever 2005 Express or MSDE 2000) the simplest way to backup, restore and delete Contact Center Express databases is via Microsoft’s GUI management tool, SQL Server Management Studio Express.

Delete Record Types from Database

If required, you can directly execute stored procedures to delete various record types from the database.

The following stored procedures all delete categories of records. All can delete records older than a given number of minutes (@OlderThanMinutes) or between specified dates and times.

Probably it is easier to routinely use @OlderThanMinutes and as a guideline for setting this parameter:

<table>
<thead>
<tr>
<th>Period</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 day</td>
<td>1,440</td>
</tr>
<tr>
<td>1 week</td>
<td>10,080</td>
</tr>
<tr>
<td>30 days</td>
<td>43,200</td>
</tr>
</tbody>
</table>

There are presently no procedures to delete tblCallSummary records or tblErrorLog records.

**procDeleteEventDetail**

Input parameters:

- @OlderThanMinutes
- @EarlyDateTime
- @LateDateTime

procDeleteEventDetail deletes all tblEvent and child event data that is older than @OlderThanMinutes or between @EarlyDateTime and @LateDateTime.

If @OlderThanMinutes is passed, then values provided for @EarlyDateTime and @LateDateTime are ignored. If @EarlyDateTime is not passed, it defaults to the beginning of the 20th century, effectively deleting all event records up to @LateDateTime. This delete routine ensures that all event records for a particular call (based on UCID) are deleted, leaving no orphaned event records.

**procDeleteClientLog**

Input parameters:

- @ClientApplication
- @Client Key
- @OlderThanMinutes
- @EarlyDateTime
- @LateDateTime
proDeleteClientLog deletes all User (tblClientLog) data for the given @ClientApplication and @ClientKey combination that is older than @OlderThanMinutes or between @EarlyDateTime and @LateDateTime. If @ClientKey is not passed, then all records for the nominated @ClientApplication are deleted; otherwise only those records with the passed @ClientApplication and @ClientKey are deleted.

If @OlderThanMinutes is passed, then values provided for @EarlyDateTime and @LateDateTime are ignored. If @EarlyDateTime is not passed, it defaults to beginning of the 20th century, effectively deleting all tblClientLog records up to @LateDateTime.

**procDeleteStats**

Input parameters:
- @OlderThanMinutes
- @EarlyDateTime
- @LateDateTime

procDeleteStats deletes all statistics data that is older than @OlderThanMinutes or between @EarlyDateTime and @LateDateTime.

If @OlderThanMinutes is passed, then values provided for @EarlyDateTime and @LateDateTime are ignored. If @EarlyDateTime is not passed, it defaults to the beginning of the 20th century, effectively deleting all statistics records up to @LateDateTime.

**procGetCallSummary**

Input parameters:
- @CallFromNumber
- @EnteredCodeData
- @StartDateTime
- @EndDateTime

Returns all data from tblCallSummary with the given CallFromNumber (defaults to any), the given EnterCodeData (defaults to any) and between StartDateTime (defaults to 1970/01/01) and @EndDateTime (defaults to present date and time).
CHAPTER 3

Administration

In This Chapter

Configure Interaction Data Server - Voice and Presence...29
Configure Interaction Data Server - Multimedia.............35
Configure Interaction Data Server - View....................40
Start Service .........................................................45
Configure Interaction Data Server - Voice and Presence

1. Click the Start button on the Windows Taskbar and select Programs > Avaya Contact Center Express > Server > Interaction Data Server - Voice and Presence > Edit AIDServer.ini from the pop-up menu.

2. Use the following definitions to define your configuration data. Note: Do not change section names or parameter names.
[Error Logging]

**Error Log Level.** The value that determines what level of error detail will be saved in the error log: 0=No error logging takes place, 1=Logs fatal, major, minor and trace information, 2=Logs fatal and major errors, 4=Logs fatal errors only.

There is also another error log level, which enables you to create log files that don't override each other every time the maximum log file size limit is reached. This logging level is designed for diagnostic purposes only and can be achieved by adding 128 to one of the logging level values mentioned above. For example, if you specify Error Log Level=129, new error log files will be continuously created for this application that contain fatal, major, minor and trace information.

**Error Log File Path.** The directory path for saving error log files. By default, this parameter is left blank, which automatically sets the path to the application's current working folder (the same folder as the application executable).

**Error Log File Extension.** The extension of error log files for this application. Extension refers to part of the file name (usually the name of the application) and the file type extension (eg .log). The application will automatically precede the default extension with the day of the week (eg, Mon, Tue) when it creates its error logs.

**Maximum Error Log File Size KB.** The maximum amount of information, in kilobytes, that will be stored in an error log file before it is archived and a new file is created. The default is 1000. The minimum you can set this to is 100. Note: The archive will only store one log file. If a second error log reaches the specified maximum size, it overrides the previously archived file. If, however, the diagnostic testing error log level is selected in Error Log Level (this is achieved by adding 128 to any one of the other error log values), a new file with a new name is created every time the maximum log file size limit is reached.

[Settings]

**Interaction Data Server ID.** A unique number that identifies the Interaction Data Server - Voice and Presence.

**Interaction Data Server Name.** A user-specified, user-friendly name for the Interaction Data Server - Voice and Presence. Note: Do not include spaces.

**Interaction Data Server Port.** The port number the Interaction Data Server - Voice and Presence uses to accept connections from clients and other servers. The default is 29090.

**Statistical Data Cache Interval Seconds.** The regularity, in seconds, in which the server internally caches statistical data. Note: The suggested value is 1 or 2. Do not exceed 5.

[ASM Client To Be Loaded]

Reserved for future use.

[Switchx]
**Switch ID.** The ID number of the switch the Interaction Data Server - Voice and Presence is connected to. The number should be unique. The switch ID is particularly important in complex configurations, where there is a network of Interaction Data Server - Voice and Presence servers receiving information from multiple Avaya switches.

**Primary Link Name.** The name of the primary link this application will use to connect to the Avaya Telephony Server and switch.

**Primary User Name.** The user name required to gain access to the primary link.

**Primary User Password.** The password associated with above user name. By default, the Contact Center Express application will encrypt this data. For more information, see the *Contact Center Express Installation Guide* (Configuration Commands).

**Database Server Name.** The name of the server on which the database is located.

**Database Name.** The name of the database. This is automatically named ActiveInteractionData when the database script is run.

**Database User Name.** The user name required to gain access to the database. This is automatically named ActiveInteractionData when the database script is run.

**Database User Password.** The password associated with the above user name. When creating new databases, the default is: CCEUser0 before encryption. For encryption information, see the *Contact Center Express Installation Guide*.pdf.

**Log Event.** The value that determines if event logging is enabled or not. False=disabled, True=enabled.

**Log Interaction Summary.** The value that determines if interaction summary logging is enabled or not. False=disabled, True=enabled. Note: If event logging (**Log Event**) is enabled, interaction summary logging must also be enabled.

**Log Call Summary.** The value that determines if call summary logging is enabled or not. False=disabled, True=enabled. Note: If event logging (**Log Event**) is enabled, call summary logging must also be enabled.

**Log Statistics Data.** The value that determines if statistical data logging is enabled or not. False=disabled, True=enabled. Note: The setting for statistical data logging is not dependent on event logging (**Log Event**).

**Log User Data.** The value that determines if user data logging is enabled or not. False=disabled, True=enabled. Note: The setting for statistical data logging is not dependent on event logging (**Log Event**).

**Service Level Time Seconds.** The amount of time, in seconds, logged-in agents are expected to take to answer calls. **Service Level Time Seconds** is a switch-wide service level setting for VDNs or split/skills that do not have their own specific service levels. For example: **Service Level Time Seconds**=15

**VDN.** The VDNs to be monitored by the Interaction Data Server - Voice and Presence. Note: Separate VDNs with commas. Do not include spaces. Use a hyphen to specify a range. For example: VDN=4481-4485,4542,4819,4830
To assign a specific service level to a VDN, separate the VDN number and the service level by a full colon. For example: \texttt{VDN}=4481-4485,4542:20,4819:30,4830

In this example, VDNs 4481-4485 and 4830 use the service level value set for the switch (see \textit{Service Level Time Seconds}). VDN 4542 has a service level of 20 seconds and 4819 has a service level of 30 seconds.

\textbf{ACD Split}. The hunt group extension numbers that relate to your split/skills to be monitored by the Interaction Data Server - Voice and Presence. Note: Separate hunt group extension numbers with commas. Do not include spaces. Use a hyphen to specify a range. To assign a specific service level to a VDN, separate the VDN number and the service level by a full colon. For example: \texttt{ACD Split}=8001,8002,8003:25,8004:10,8005

In this example, hunt group extension numbers 8001, 8002 and 8005 use the service level value set for the switch (see \textit{Service Level Time Seconds}). Hunt group extension number 8003 has a service level of 25 seconds and 8004 has a service level of 10 seconds.

\textbf{Station}. The station extensions to be monitored by the Interaction Data Server - Voice and Presence. Note: Separate extensions with commas. Do not include spaces. Use a hyphen to specify a range. For example: \texttt{Station}=4401-4424,4637

\textbf{Trunk Group}. The trunk groups to be monitored by the Interaction Data Server - Voice and Presence. Note: Separate groups with commas. Do not include spaces. Use a hyphen to specify a range. 

\textbf{Shift Start Times}. The start times for shifts throughout the day. (Note: A shift can be any value between 1 and 24 hours.) At the end of each shift, statistical data will be written to the database and then reset for the next shift calculation. The times will be used in the order they are entered. If shift times are missing, a default of one shift will be assumed that covers a single day (i.e. that starts at 00:00 and ends at 23:59). Note: Separate start times with commas. For example: \texttt{Shift Start Times}=08:00, 14:00, 22:30, 03:00

This example will split the day into four shifts. Shift 1 will start at 08:00 and run until the start of shift 2 at 14:00. Shift 2 will run until the start of shift 3 at 22:30. Shift 3 will run until the start of shift 4 at 03:00 the next day. Shift 4 will run until shift 1 starts again at 08:00. Every day will have the same shift times.

\textbf{Statistic Interval Minutes}. The amount of time, in minutes, between statistic calculations. The interval can be any value between 1 and 60 minutes. Values outside this range will not be accepted. At the end of each interval, statistical data will be written to the database and then reset for the next interval calculation.

Where possible, select an interval that is a divisor of 60 minutes. Where a value is chosen that is not a divisor of 60 minutes, the result will be a smaller final period in the hour. For example, if 13 minutes is chosen, this will result in four intervals of 13 minutes and one final interval of 8 minutes.

If the value is missing or invalid, the interval is set to 5 minutes.

\textbf{Polling Agent Interval Seconds}. How often, in seconds, the application polls for agent mode status. To reduce the network traffic, increase this number. Note: To turn this feature off, use 0.

\textbf{Polling Trunk Group Interval Seconds}. How often, in seconds, the application polls for trunk group status. Note: To turn this feature off, use 0.
**Force Agent Polling.** If enabled (set to True), this parameter forces the application to poll the switch for agent state information. This is particularly useful if the switch is an earlier version of Definity (prior to release 10). If an earlier version of Definity mistakenly informs the application that it has Agent States functionality (only present with Release 10 and above), this parameter prevents the application from receiving no agent event information at all. (The Agent States feature allows Contact Center Express applications to automatically receive agent event information for all agents it is monitoring.)

Note: If you disable this parameter (set it to False) and **Enable Agent Events** is also disabled, Interaction Data Server - Voice and Presence will ignore your False value and automatically poll the switch using the polling interval defined by **Polling Agent Interval Seconds**.

**Enable Agent Events.** If enabled (set to True), this parameter allows the Interaction Data Server - Voice and Presence to automatically receive agent state change information (events) from the Definity. (Note: This assumes the Definity is Release 10 or above and has its Agent States functionality activated.)

If this parameter is disabled (set to False), Interaction Data Server - Voice and Presence will automatically poll the switch for agent state information using the polling interval defined by the **Polling Agent Interval Seconds**. Note: Interaction Data Server - Voice and Presence will poll the switch even if the **Force Agent Polling** parameter is disabled.

**Automatically Add Devices.** If enabled (set to True), this parameter forces Interaction Data Server - Voice and Presence to query the switch to determine if any new devices (stations, VDNs, skills etc) have been added to the switch that haven't been added to its own configuration. If it finds a new device on the switch that it isn't currently monitoring, Interaction Data Server - Voice and Presence will add it to its configuration.

**Validate Each Device Type On Startup.** If enabled (set to True), this parameter forces Interaction Data Server - Voice and Presence to query the switch to determine if the devices you have added to its configuration are listed under the correct device type (ie. station, VDN, split/skill, trunk group). If Interaction Data Server - Voice and Presence finds a device type is wrong, it will not monitor the device.

**[Other AIDServer x]**

**Interaction Data Server ID.** A unique number that identifies the Interaction Data Server - Voice and Presence.

**Interaction Data Server Name.** A user-specified, user-friendly name for the Interaction Data Server - Voice and Presence. Note: Do not include spaces.

**Interaction Data Server IP.** The IP address of the Interaction Data Server - Voice and Presence.

**Interaction Data Server Port.** The port number the Interaction Data Server - Voice and Presence uses to accept connections from clients and other servers. The default is 29090.

**Switch IDs.** The ID numbers of the switches the Interaction Data Server - Voice and Presence is connected to.
Note: If you want the Interaction Data Server - Voice and Presence to monitor information for more than one switch, you need to copy and paste the [Switchx] section and modify it accordingly. If you want the Interaction Data Server - Voice and Presence to accept connections from more than one other Interaction Data Server - Voice and Presence, you need to copy and paste the [Other Interaction Data Server x] section and modify it accordingly.

Never delete a default configuration parameter. If you don't need to enter a value, leave the line blank, for example, VDN=
Configure Interaction Data Server - Multimedia

1  Click the Start button on the Windows Taskbar and select Programs > Avaya Contact Center Express > Server > Interaction Data Server - Multimedia > Edit ASIDSMultimedia.ini from the pop-up menu.

2  Use the following definitions to define your configuration data. Note: Do not change section names or parameter names.
[Service Plug In Host]

Service Display Name. The text that displays for this service under the Name column of the Microsoft Windows Services screen. The default value is: AS Interaction Data Server - Multimedia

Service Command Line. Leave this parameter blank. By default, Service Host Plug-in will source its configuration data from the same working folder that contains the application executable.

Service Description. The text that displays for this service under the Description column of the Microsoft Windows Services screen. The default is: The IDS MultiMedia is a component working with a single focus on the Multi Media staff.

Service Startup State. A value that determines the state of the service when it is installed. 0=Disabled, 1=Manual, 2=Automatic.

[Error Logging]

Error Log Level. The value that determines what level of error detail will be saved in the error log: 0=No error logging takes place, 1=Logs fatal, major, minor and trace information, 2=Logs fatal, major and minor errors, 4=Logs fatal and major errors, 8=Logs fatal errors only.

There is also another error log level, which enables you to create log files that don't override each other every time the maximum log file size limit is reached. This logging level is designed for diagnostic purposes only and can be achieved by adding 128 to one of the logging level values mentioned above. For example, if you specify Error Log Level=129, new error log files will be continuously created for this application that contain fatal, major, minor and trace information.

Error Log File Path. The directory path for saving error log files. By default, this parameter is left blank, which automatically sets the path to the application's current working folder (the same folder as the application executable).

Error Log File Extension. The extension of error log files for this application. Extension refers to part of the file name (usually the name of the application) and the file type extension (eg .log). The application will automatically precede the default extension with the day of the week (eg, Mon, Tue) when it creates its error logs.

Maximum Error Log File Size KB. The maximum amount of information, in kilobytes, that will be stored in an error log file before it is archived and a new file is created. The default is 1000. The minimum you can set this to is 100. Note: The archive will only store one log file. If a second error log reaches the specified maximum size, it overrides the previously archived file. If, however, the diagnostic testing error log level is selected in Error Log Level (this is achieved by adding 128 to any one of the other error log values), a new file with a new name is created every time the maximum log file size limit is reached.

[Server Identifier]

Server Instance Friendly Name. Any name you want to display for this application within Contact Center Express Control Panel.

Server Instance ID. A unique identifier for the server application, which is created automatically when it runs for the first time.
**Server Instance Type.** An ID used by other applications to determine what type of component they are communicating with.

**[Application Management Service]**

**Multicast IP.** The IP address that will be used for multicasting between applications. When an application starts, it will join this multicast address and receive packet information from the Application Management Director. The default is 239.29.9.67.

**Multicast Port.** The port number that will be used for multicasting between applications. The default is 29075.

**Enable Multicast.** A value that determines if multicasting will be used to locate the Application Management Director or not. True=enabled, False=disabled. The default is True.

**Application Management Director URL List.** If multicasting is disabled, these URLs are used to find the Application Management Directors set up in your contact center. Items are separated by commas and follow the format: IP address:port number, IP address:port number etc.

**Management Object URL.** The URL used by Contact Center Express Control Panel to connect to the remoting management object. The URL must use the following format: channeltype://fullyqualifiedcomputername:port/uri. If the entry is empty, a default URL is automatically created:

**[Client Connections]**

**IP Address.** The local IP address for accepting incoming client connections.

**IP Port.** The local IP port for accepting incoming client connections.

**Allow Multicast.** A value that determines whether multicasting of the connection information is enabled or not. True=enabled, False=disabled.

**Enable Trace.** A value that determines whether information is logged to the log file. True=enabled, False=disabled.

**Broadcast Interval In Seconds.** How often, in seconds, connection information is multicasted.

**Multicast IP.** The IP address that will be used for multicasting the connection information. The default is 239.29.9.67.

**Multicast Port.** The port number that will be used for multicasting the connection information.

**Receive Buffer Length.** The buffer size for incoming messages. The default is 20000.

**[Plug In Assembly List]**

This section lists all loadable generic plug-ins. Each entry has the format "Friendly name=Plug-in section name". The plug-in section name points to (and is the same as) the section in the file that contains configuration data for that plug-in.

For example:
IDS MMAE Database Engine = IDS MMAE Database Engine
IDS MMAE Realtime Engine = IDS MMAE Realtime Engine
IDS MMAE CMS Engine = IDS MMAE CMS Engine

[IDS MMAE Database Engine]

**Plug-in ID.** A globally unique identifier (GUID) that identifies this Plug-in.

**Assembly File Name.** The name of the plug-in file to be loaded. If the plug-in is not located in the default file path (the same folder as the host application), also specify the file path. For this plug-in, use: Assembly File Name = ASIDSMMAEDatabasePlugin.dll

**Connection String.** An ADO connection string that will be used for database connectivity. If this parameter is empty, the Database Server Name, Database Name and Database User Name parameters are used to form the connection string. Encrypt this string if it contains a password by using the %%ENCRYPT command. For more information, see the Contact Center Express Installation Guide (Configuration Commands).

**Database Name.** The name of the database. This is automatically named ActiveInteractionData when the database script is run.

**Database User Name.** The user name required to gain access to the database. This is automatically named ActiveInteractionData when the database script is run.

**Database User Password.** The password associated with the above user name. When creating new databases, the default is: CCEUser0 before encryption. For encryption information, see the Contact Center Express Installation Guide.pdf.

**Receive Buffer Length.** The buffer size for incoming messages. The default is 20000.

**Enable Error Logging.** A setting that allows you to write plug-in specific error information to the application's error log files. True=enabled, False=disabled.

[IDS MMAE Realtime Engine]

**Plug-in ID.** A globally unique identifier (GUID) that identifies this Plug-in.

**Assembly File Name.** The name of the plug-in file to be loaded. If the plug-in is not located in the default file path (the same folder as the host application), also specify the file path. For this plug-in, use: ASIDSMMAERealtimePlugin.dll

**Receive Buffer Length.** The buffer size for incoming messages. The default is 20000.

**Statistic Interval.** The amount of time, in minutes, between statistic calculations. The interval can be any value between 1 and 60 minutes. Values outside this range will not be accepted. The default is 15. At the end of each interval, statistical data will be written to the database and then reset for the next interval calculation.

Where possible, select an interval that is a divisor of 60 minutes. Where a value is chosen that is not a divisor of 60 minutes, the result will be a smaller final period in the hour. For example, if 13 minutes is chosen, this will result in four intervals of 13 minutes and one final interval of 8 minutes.
If the value is missing or invalid, the interval is set to 5 minutes.

**Shift Start Times.** The start times for shifts throughout the day. (Note: A shift can be any value between 1 and 24 hours.) At the end of each shift, statistical data will be written to the database and then reset for the next shift calculation. The times will be used in the order they are entered. If shift times are missing, a default of one shift will be assumed that covers a single day (i.e. that starts at 00:00 and ends at 23:59). Note: Separate start times with commas.

For example: **Shift Start Times**=08:00, 14:00, 22:30, 03:00

This example will split the day into four shifts. Shift 1 will start at 08:00 and run until the start of shift 2 at 14:00. Shift 2 will run until the start of shift 3 at 22:30. Shift 3 will run until the start of shift 4 at 03:00 the next day. Shift 4 will run until shift 1 starts again at 08:00. Every day will have the same shift times.

**Enable Error Logging.** A setting that allows you to write plug-in specific error information to the application's error log files. True=enabled, False=disabled.

**[IDS MMAE CMS Engine]**

This section configures the Multimedia CMS Plug-in.

**Plug-in ID.** A globally unique identifier (GUID) that identifies this Plug-in.

**Server Instance Id.** The server instance ID of the media store referenced in the above GUID that generates data for CMS.

**Assembly File Name.** The name of the plug-in file to be loaded. If the plug-in is not located in the default file path (the same folder as the host application), also specify the file path. For this plug-in, use: ASIDSCMSPlugin.dll

**Database Server Name.** The name or TCP/IP address of the server on which the Avaya CMS Informix database is located.

**Database User Name.** A valid user name for the CMS Informix database.

**Database User Password.** The password associated with the above user name.

**IDS MMAE URL.** The URL for Interaction Data Server - Multimedia. The URL must use the following format: gtcp://localhost:29077/InteractionDataServiceMultimedia.rem.

**Enable Error Logging.** A setting that allows you to write plug-in specific error information to the application's error log files. True=enabled, False=disabled.

**Fire Test Mock Event.** Set to False. This is used for development only.
Configure Interaction Data Server - View

1. Click the Start button on the Windows Taskbar and select Programs > Avaya Contact Center Express > Server > Interaction Data Server - View > Edit ASIDSVIEW.ini from the pop-up menu.

2. Use the following definitions to define your configuration data. **Note:** Do not change section names or parameter names.
**[Service Plug In Host]**

**Service Display Name.** The text that displays for this service under the Name column of the Microsoft Windows Services screen. The default value is: AS Interaction Data Server - View

**Service Command Line.** Leave this parameter blank. By default, Service Host Plug-in will source its configuration data from the same working folder that contains the application executable.

**Service Description.** The text that displays for this service under the Description column of the Microsoft Windows Services screen. The default is: IDS View is a service for extracting the aggregated realtime and historical data from the IDS suite.

**Service Startup State.** A value that determines the state of the service when it is installed. 0=Disabled, 1=Manual, 2=Automatic.

**[Error Logging]**

**Error Log Level.** The value that determines what level of error detail will be saved in the error log: 0=No error logging takes place, 1=Logs fatal, major, minor and trace information, 2=Logs fatal, major and minor errors, 4=Logs fatal and major errors, 8=Logs fatal errors only.

There is also another error log level, which enables you to create log files that don't override each other every time the maximum log file size limit is reached. This logging level is designed for diagnostic purposes only and can be achieved by adding 128 to one of the logging level values mentioned above. For example, if you specify Error Log Level=129, new error log files will be continuously created for this application that contain fatal, major, minor and trace information.

**Error Log File Path.** The directory path for saving error log files. By default, this parameter is left blank, which automatically sets the path to the application's current working folder (the same folder as the application executable).

**Error Log File Extension.** The extension of error log files for this application. Extension refers to part of the file name (usually the name of the application) and the file type extension (eg .log). The application will automatically precede the default extension with the day of the week (eg, Mon, Tue) when it creates its error logs.

**Maximum Error Log File Size KB.** The maximum amount of information, in kilobytes, that will be stored in an error log file before it is archived and a new file is created. The default is 1000. The minimum you can set this to is 100. Note: The archive will only store one log file. If a second error log reaches the specified maximum size, it overrides the previously archived file. If, however, the diagnostic testing error log level is selected in Error Log Level (this is achieved by adding 128 to any one of the other error log values), a new file with a new name is created every time the maximum log file size limit is reached.

**[Server Identifier]**

**Server Instance Friendly Name.** Any name you want to display for this application within Contact Center Express Control Panel.

**Server Instance ID.** A unique identifier for the server application, which is created automatically when it runs for the first time.
**Server Instance Type.** An ID used by other applications to determine what type of component they are communicating with.

### Application Management Service

- **Multicast IP.** The IP address that will be used for multicasting between applications. When an application starts, it will join this multicast address and receive packet information from the Application Management Director. The default is 239.29.9.67.

- **Multicast Port.** The port number that will be used for multicasting between applications. The default is 29075.

- **Enable Multicast.** A value that determines if multicasting will be used to locate the Application Management Director or not. True=enabled, False=disabled. The default is True.

- **Application Management Director URL List.** If multicasting is disabled, these URLs are used to find the Application Management Directors set up in your contact center. Items are separated by commas and follow the format: IP address:port number, IP address:port number etc.

- **Management Object URL.** The URL used by Contact Center Express Control Panel to connect to the remoting management object. The URL must use the following format: channeltype://fullyqualifiedcomputername:port/uri. If the entry is empty, a default URL is automatically created: gtcp://fullyqualifiedlocalcomputername:29069/XMLServer.rem

### Client Connections

- **IP Address.** The local IP address for accepting incoming client connections.

- **IP Port.** The local IP port for accepting incoming client connections.

- **Allow Multicast.** A value that determines whether multicasting of the connection information is enabled or not. True=enabled, False=disabled.

- **Enable Trace.** A value that determines whether information is logged to the log file. True=enabled, False=disabled.

- **Broadcast Interval In Seconds.** How often, in seconds, connection information is multicasted.

- **Multicast IP.** The IP address that will be used for multicasting the connection information. The default is 239.29.9.67.

- **Multicast Port.** The port number that will be used for multicasting the connection information.

- **Receive Buffer Length.** The buffer size for incoming messages. The default is 20000.

### Plug In Assembly List

This section lists all loadable generic plug-ins. Each entry has the format "Friendly name=Plug-in section name". The plug-in section name points to (and is the same as) the section in the file that contains configuration data for that plug-in.

For example:
IDS View Engine = IDS View Engine

[IDS View Engine]

**Plug-in ID.** A globally unique identifier (GUID) that identifies this Plug-in.

**Assembly File Name.** The name of the plug-in file to be loaded. If the plug-in is not located in the default file path (the same folder as the host application), also specify the file path. For this plug-in, use: ASIDSViewPlugin.dll

**Enable Error Logging.** A setting that allows you to write plug-in specific error information to the application's error log files. True=enabled, False=disabled.

**Multicast IP.** The multicast IP address for sending out notifications. The default is 239.29.9.67.

**Multicast Port.** The multicast port address for sending out notifications. The default is 29084.

**IDS Voice and Presence IP.** The IP address of Interaction Data Server - Voice and Presence.

**IDS Voice and Presence Port.** The IP port of Interaction Data Server - Voice and Presence. The default is 29090.

**IDS MMAE URL.** The URL for connecting to Interaction Data Server - Multimedia. The URL must use the following format: g tcp://localhost:29077/InteractionDataServiceMultimedia.rem. If this parameter is empty, the next four parameters are used to form the URL.

**IDS MMAE Channel Type.** The type of channel to use to connect to Interaction Data Server - Multimedia. Valid values are 'ipc', 'tcp' and 'gtcp'. If the **IDS MMAE URL** parameter contains data, this entry is ignored. The default value is g tcp.

**IDS MMAE IP.** The IP address of Interaction Data Server - Multimedia. If the **IDS MMAE URL** parameter contains data, this entry is ignored.

**IDS MMAE Port.** The IP port of Interaction Data Server - Multimedia. If the **IDS MMAE URL** parameter contains data, this entry is ignored.

**IDS MMAE Remote Factory URI.** The URI (Uniform Resource Identifier) of the remote communication object factory on Interaction Data Server - Multimedia. In this case, the URI is the name of the object factory: InteractionDataServiceMultimedia.rem. If the **IDS MMAE URL** parameter contains data, this entry is ignored.

**Connection String.** An ADO connection string that will be used for database connectivity. If this parameter is empty, the Database Server Name, Database Name and Database User Name parameters are used to form the connection string. Encrypt this string if it contains a password by using the %%ENCRYPT command. For more information, see the Contact Center Express Installation Guide (Configuration Commands).

**Database Name.** The name of the database. This is automatically named ActiveInteractionData when the database script is run.

**Database User Name.** The user name required to gain access to the database. This is automatically named ActiveInteractionData when the database script is run.
**Database User Password.** The password associated with the above user name. When creating new databases, the default is: CCEUser0 before encryption. For encryption information, see the *Contact Center Express Installation Guide.pdf*.


**Maximum History Entry.** The maximum number of history entries that are returned by Interaction Data Server - View. The default is 100. If you enter 0, Interaction Data Server - View will return all the records retrieved from the database.
Start Service

➢ To start your Contact Center Express service:

1. Click the Start button on the Windows Taskbar and select Control Panel from the pop-up menu.
2. Click Administrative Tools.
3. Click Services.
4. Right-click an Contact Center Express service and select Start from the pop-up menu.
   - AS AOL-ICQ Instant Messenger Gateway
   - AS Application Management Director
   - AS Call Routing Server
   - AS Email Media Store
   - AS IDS MultiMedia
   - AS IDS View
   - AS IDS Voice
   - AS IVR Server
   - AS Media Proxy Service
   - AS MSN Messenger Gateway
   - AS Preview Contact Media Store
   - AS Short Message Service Gateway
   - AS Simple Messaging Media Store
   - AS Web Chat Gateway
   - AS XML Server

   In this example, AS IDS Voice is started:
CHAPTER 4

Developing Client Applications

In This Chapter

Overview .................................................................48
Real Time Reporting Data ........................................51
Historical Reporting Data ........................................61
Miscellaneous Capabilities ......................................64
Enumerations ..........................................................66
Overview

Interaction Data Server - View allows you to access the statistical data generated by Interaction Data Server - Voice and Presence and Interaction Data Server - Multimedia and to display that data on individual client PCs. Interaction Data Server - View is currently used by Contact Center Express's premier desktop application Contact Center Express Desktop, and it can also be used by developers to build their own client applications.

Note: Interaction Data Server - View replaces the current Interaction Data Client which will be depreciated from version 3.1 onward.

Client Connections

Clients will connect to Interaction Data Server - View using the remoting technology already deployed throughout Contact Center Express. Interaction Data Server - View defines a remoting object for this communication, called IDSViewClientWrapper, which has the methods and events to allow communication between the two points. This wrapper simplifies connection to Interaction Data Server - View Client, exposing only those methods and events relevant to a client connection. Note that the Interaction Data Server - View Client object also has methods and events for its communication with Interaction Data Server - View which should not be used by a client.

Transfer of data to the clients is a mixture of one-to-one communication and (where at all possible) multicast.

Method return codes

All methods from the client to Interaction Data Server - View return a long value error code. In the method description this is referred to as enIDSViewClientReturn. See below under enumerations for its values.

Preamble

Begin by creating a new IDSViewClientWrapper object. This creation allows the client to decide from which Interaction Data Server to obtain data and whether the status change messages can be received via multicast or sent to the client directly.

Syntax:

```java
public IDSViewClientWrapper(
    IDSClientInfo iDSViewClientInfo,
    String strMulticastIP,
    Int32 nMulticastPort,
    String iDSViewURL,
    Boolean bSendByMulticast
)
```
Where:

- iDSViewClientInfo contains the basic information for an Interaction Data Server - View Client – the ServerInstanceId and ServerInstanceType GUIDs
- strMulticastIP is the multicast IP address for receiving multicast notifications.
- nMulticastPort is the multicast port for receiving multicast notifications.
- iDSViewURL the URL of the Interaction Data Server - View remoting object.
- bSendByMulticast specifies whether the notification message will be sent through IP multicasting in which case the multicast parameters above must be passed.

**Events**

The following events become available and may be fired on creation or at other times during the client connection to IDSViewClientWrapper.

**OnConnected**

Fired, with no parameters, when Interaction Data Server - View Client successfully connects to Interaction Data Server - View.

**OnDisconnect**

Fired, with no parameters, when Interaction Data Server - View Client disconnects from Interaction Data Server - View.

**OnInternalServerError**

Fired when Interaction Data Server - View Client/Interaction Data Server - View comes across an unexpected error.

Event syntax:

```
OnInternalServerError (String errorMessage)
```

**OnIDSMultimediaStatusChanged**

Fired when Interaction Data Server - View connects or disconnects from the Interaction Data Server - Multimedia database.

Event syntax:

```
OnIDSMultimediaStatusChanged (enum enIDSMultimediaStatus)
```

Where Connected = 1, Disconnected = 2 indicates the current status of the connection.

**OnIDSVoiceStatusChanged**

Fired when Interaction Data Server - View connects or disconnects from the Interaction Data Server - Voice and Presence.

Event syntax:

```
OnIDSVoiceStatusChanged (enum enIDSVoiceStatus)
```
Where Connected = 1, Disconnected = 2 indicates the current status of the connection.

**RegisterIDSViewClient**

Use this method after creating an IDSViewClientWrapper object before requesting and receiving notifications.

Method syntax:
```
enIDSViewClientReturn RegisterIDSViewClient (Int32 nCheckRegInSecond)
```
Where nCheckRegInSecond is the interval in seconds after which the connection between Interaction Data Server - View and Interaction Data Server - View Client is checked. A recommended default value is 10 seconds.

**UnRegisterIDSViewClient**

Use this method to stop using a IDSViewClientWrapper object for requesting and receiving notifications. Note that this method clears all set notifications automatically.

There are no parameters for this method.

Method syntax:
```
enIDSViewClientReturn UnregisterIDSViewClient()
```
Real Time Reporting Data

This section describes the real-time reporting data available to clients. Some of this data will follow the request/response message flow and others will follow the asynchronous notification pattern via the multicast functionality.

Note: The following method and event descriptions have been split into logical functional groups (device, agent etc) however these methods and events are presented in a single object exposed to the client by IDSViewClientWrapper.
Device

Clients use the following methods to set or clear interest of notification in a specific device. When interest is requested then status updates will be sent to the client.

Interaction Data Server - Voice and Presence holds the last message about all devices. When a new client connects and adds the requirement for notification about a specific device, this stored message is forwarded immediately.

Methods

SetDeviceStatusNotification

The client calls this method to signal to IDSViewClientWrapper that it requires status notification on the device or devices listed. The client is able to specify multiple devices in this call.

Method syntax:

```java
enIDSViewClientReturn
SetDeviceStatusNotification(String DeviceList)
```

ClearDeviceStatusNotification

The client calls this method to signal to IDSViewClientWrapper to remove status notification on the device or devices listed. The client is able to specify multiple devices in this call.

Method syntax:

```java
enIDSViewClientReturn
ClearDeviceStatusNotification(String DeviceList)
```

GetDeviceStatus

A client can request status information on any group of devices to be sent immediately. This message, when received will result in IDSViewClientWrapper sending 1 or more DeviceStatusNotification events to the requesting client. In this scenario only the client that requested the information receives it and it is sent to the client via the direct socket link, not via the multicast facility.

Method syntax:

```java
enIDSViewClientReturn
GetDeviceStatus(String DeviceList)
```

Events

These events are fired to one or more clients.

OnDeviceStatusNotified

This message is sent to all interested parties, either directly or via the multicast mechanism when the status of a device changes. It is also sent as a response to the GetDeviceStatus method being received from a specific client.
Each message may contain 1 or more DeviceStatus sections that contain status information. Where a request is received for an unknown device then the status will contain the word “Unknown”.

Event syntax:

```csharp
OnDeviceStatusNotified(String AgentID,
                     String AgentName,
                     Int32 AgentState,
                     Int32 CallCount,
                     String Device,
                     String DeviceName,
                     Int32 FeatureState,
                     Int MessagingState,
                     String OSLoggedInUserID,
                     String ServerInstanceID,
                     String ServerInstanceType)
```

**Note:** FeatureState will be reserved for future use and will be set to 0 for all method calls. AgentState will follow the standard enumeration state.
Agent

IDSViewClientWrapper allows clients to notify interest in a specific agent ID Interaction Data Server - Voice and Presence holds information until the next polling period offers it to any clients that request information in the period between polling attempts.

Methods

SetRealtimeAgentNotification

The client calls this method to signal to IDSViewClientWrapper that it requires real-time notification of agent statistical data. Every period (eg 5 seconds) IDSViewClientWrapper provides current real-time statistics. The client may send 1 or more agent IDs within the single request.

Method syntax:

```java
enIDSViewClientReturn
SetRealtimeAgentNotification(String AgentIDList)
```

ClearRealtimeAgentNotification

The client calls this method to signal to IDSViewClientWrapper that it no longer requires real-time notification of statistics on 1 or more agents.

Method syntax:

```java
enIDSViewClientReturn
ClearRealtimeAgentNotification(String AgentIDList)
```

Events

OnRealtimeAgentStatusNotified

This message is sent to all interested parties, either directly or via the multicast mechanism when IDSViewClientWrapper gets an update of agent status.

Each message may contain 1 or more AgentStatus sections that contain status information. Where a request is received for an unknown device then the status will contain the word “Unknown”.

Event syntax:

```java
OnRealtimeAgentStatusNotified (String AgentID,
String AgentName,
Int32 AgentState,
Int32 AverageAfterCallWorkTime,
Int32 AverageAUXTime,
Int32 AverageAvailableTime,
Int32 AverageTalkTime,
...)
```
Int32 CallsPerHour,
String Device,
Int32 PendingWorkMode,
Int32 ReasonCode,
String ServerInstanceID,
String ServerInstanceType,
String SplitSkill,
Int32 TalkState,
Int32 WorkMode)

**OnAgentLoggedIn**

IDSViewClientWrapper sends this message to all connected clients when an agent logs in. This message will be forwarded immediately to all clients that have requested notification via the RealtimeAgentNotification method.

Event syntax:

```plaintext
OnAgentLoggedIn (String AgentID,
String AgentName,
String Device,
String ServerInstanceID,
String ServerInstanceType,
String SplitSkill)
```

**OnAgentLoggedOut**

IDSViewClientWrapper sends this message to all connected clients on agent log out.

Event syntax:

```plaintext
OnAgentLoggedOut (String AgentID,
String AgentName,
String Device,
Int32 ReasonCode,
String ServerInstanceID,
String ServerInstanceType,
String SplitSkill)
```
**OnAgentStateChanged**

IDSViewClientWrapper sends this message to all connected clients on an agent state changing.

Event syntax:

```java
OnAgentStateChanged (String AgentID,
String AgentName,
Int32 AgentState,
String Device,
Int32 MessagingState,
String OSLoggedInUserID,
Int32 PendingReasonCode,
Int32 PendingWorkMode,
Int32 ReasonCode,
String ServerInstanceID,
String ServerInstanceType,
String SplitSkill,
Int32 TalkState,
Int32 WorkMode)
```
VDN

IDSViewClientWrapper allows clients to notify interest in a specific VDN and sends reports periodically to clients.

IDSViewClientWrapper holds information until the next polling period and to send to any clients that request information in the period between polling attempts.

Methods

SetRealtimeVDNNotification

The client calls this method to signal to IDSViewClientWrapper that it requires real-time notification of VDN statistical data. Every period (e.g., 5 seconds) IDSViewClientWrapper returns current real-time statistics.

The client may send 1 or more VDNs within the single request.

Method syntax:

```
enIDSViewClientReturn SetRealtimeVDNNotification(String VDNList)
```

ClearRealtimeVDNNotification

Use this method to tell IDSViewClientWrapper to no longer provide real-time notification of statistics on 1 or more VDNs.

Method syntax:

```
enIDSViewClientReturn ClearRealtimeVDNNotification(String VDNList)
```

Events

OnRealtimeVDNStatusNotified

This message is sent periodically to all interested parties, either directly or via the multicast mechanism. This message will also be sent to all clients that request notification via the SetRealtimeVDNNotification method.

Each message may contain 1 or more VDNStatus sections that contain status information. Where a request is received for an unknown device then the status will contain the word “Unknown”.

Event syntax:

```
OnRealTimeVDNStatusNotified (Int32 AbandonCalls,
Int32 AverageAbandonTime,
Int32 AverageTalkTime,
Int32 AverageWaitTime,
Int32 CallsWaiting,
```
Int32 LongestCall,
String ServerInstanceID,
String ServerInstanceType,
String VDN,
String VDNNName)
Queue

IDSViewClientWrapper allows clients to notify interest in a specific queue and manages the periodic return of information to the client. IDSViewClientWrapper holds information until the next polling period to send to any clients that request information in the period between polling attempts.

Methods

**SetRealtimeQueueNotification**

The client calls this method of IDSViewClientWrapper to get real-time notification of queue statistical data. If the queue is a split skill device then every period (e.g., 5 seconds) IDSViewClientWrapper returns real-time statistics. Initially the determination of whether this is a split/skill device will be based on whether the device is a numeric value. If the queue is a multimedia device then IDSViewClientWrapper will forward queue information.

The client may send 1 or more queue IDs within the single request.

Method syntax:

```
enIDSViewClientReturn SetRealtimeQueueNotification(String QueueList)
```

**ClearRealtimeQueueNotification**

The client calls this method to signal to IDSViewClientWrapper that it no longer requires real-time notification of statistics on 1 or more queues.

Method syntax:

```
enIDSViewClientReturn ClearRealtimeVDNNotification(String QueueList)
```

Events

**OnRealtimeQueueStatusNotified**

This message is sent periodically to all interested parties, either directly or via the multicast mechanism. It is also sent to all clients that request notification via the SetRealtimeQueueNotification message.

Each message may contain 1 or more QueueStatus sections that contain status information. Where a request is received for an unknown queue the status will contain the word “Unknown”.

Event syntax:

```
OnRealtimeQueueStatusNotified (Int32 AgentsAvailable, Int32 AgentsStaffed, Int32 AverageAbandonTime, Int32 AverageHandleTime, Int32 AverageWaitTime,
```
Int32 InteractionsWaiting,
Int32 OldestInteraction,
String QueueID,
String QueueName,
String ServerInstanceFriendlyName
String ServerInstanceID,
String ServerInstanceType,
Int32 TotalAbandonedInteractions,
Int32 TotalInteractionsArrived,
Int32 totalInteractionsArrivedLastHour,
Int32 TotalInteractionsArrivedThisHour,
Int32 TotalInteractionsArrivedThisInterval,
Int32 TotalInteractionsInProgress,
Int32 TotalInteractionsSuspended,
Int32 TotalInteractionsSuspendedThisInterval)
Historical Reporting Data

Agent

Methods
These methods allow details of the agent’s activity to be returned to the requestor.

GetAgentHistory
This method returns the variable number of interactions that this agent has been a party to. The returned records list all the interaction sectors (a span from an interaction create through to close sequence) and the current state of each. The records include major Associated data (keyed to the conversation), the Server Instance ID and Type, the current interaction state and the associated agent ID (if any).

Message syntax:
[out] ReturnData

GetAgentHistory(String AgentID, String MediaTypes, string InteractionStates, int DaysOld)

Events
No events are defined at this time.

Skill

Methods
No methods are defined at this time.

Events
No events are defined at this time.

Queue

Methods
No methods are defined at this time.

Events
No events are defined at this time.
Interaction

Methods
These methods allow details of an interaction to be returned to the client. The data the client receives is read-only; there is no facility to modify the data in the database.

GetInteractionHistory
This method returns the variable number of state changes (segments) that have been recorded for all sectors of this Interaction to date.

Method syntax:
[out] ReturnData
GetInteractionHistory(
String InteractionID,
int DaysOld)

Events
No events are defined at this time.
Conversation & Customer

Methods

These methods allow details of a conversation, including all interactions, to be returned to the client. The data the client receives is read-only; there is no facility to modify the data.

GetConversationHistory

This method returns the variable number of interaction sectors (a span from an interaction create through to close sequence) that belong to a single conversation. The data includes major Associated data (keyed to the conversation), the Server Instance ID and Type, the current interaction state and the associated agent ID (if any). This data is read-only. There is no facility to change this data.

Message syntax:

[out] ReturnData

GetConversationHistory(
String ConversationID,
String CustomerID,
int DaysOld)

Events

No events are defined at this time.
**Miscellaneous Capabilities**

IDSViewClientWrapper offers the following capabilities that don’t easily fall into the statistics/reporting paradigm.

**Methods**

**SetClientIdentity**

This method allows the client application connected to IDSViewClientWrapper to identify itself and some details about the user logged into the machine. This information can be updated at any time by the client to reflect changes in the client application. The values sent with this method will replace those values currently held by IDSViewClientWrapper for this client.

Method syntax:

```java
enIDSViewClientReturn SetClientIdentity(String OSloggedInUserID, String AgentID, String StationID)
```

Where:

- **OSloggedInUserID** is the Windows user name that is currently logged into the client machine.
- **AgentID** is the agent ID (switch ACD) that is currently logged in at the client application.
- **StationID** is the station ID that is currently controlled by this client application.

**SetConversationIDForInteraction**

This method allows the client application to specify the conversation id for a specific interaction. This linkage will allow multiple interactions to be grouped together to form a conversation.

Method syntax:

```java
enIDSViewClientReturn SetConversationIDForInteraction(String InteractionID, String ConversationID)
```

**SetCustomerIDForInteraction**

This method allows the interaction to be associated with a specific customer. This method can be called at the client once the client has correctly identified the customer. This could be via an automatic mechanism (eg CLI lookup) or through manual searching. Linking interactions to customers allows extended information to be retained about an interaction and also allows the customer history through Contact Center Express to be retained.

Message syntax:

```java
enIDSViewClientReturn SetCustomerIDForInteraction(String InteractionID, String CustomerID)
```

**SetWorkCodeForInteraction**
Adding work codes to an interaction allows extended information about the reason for a call and its disposition to be recorded. These work codes can then be used for reporting purposes.

Note: For voice calls in Contact Center Express 3.0, the Interaction ID and the UCID must be considered the same.

Message syntax:

```java
```

**Events**

No events are defined at this time.
Enumerations

The following enumerations are defined for use in Interaction Data Server - View methods and events.

### enIDSVewClientReturn

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnknownError</td>
<td>Unknown error encountered</td>
<td>0</td>
</tr>
<tr>
<td>NoError</td>
<td>The method executed correctly</td>
<td>1</td>
</tr>
<tr>
<td>NewClient</td>
<td>Registration of new client succeeded</td>
<td>2</td>
</tr>
<tr>
<td>AlreadyRegistered</td>
<td>Client has already been registered</td>
<td>3</td>
</tr>
<tr>
<td>NotRegistered</td>
<td>Client has not been registered</td>
<td>4</td>
</tr>
<tr>
<td>Succeed</td>
<td>The method call completed successfully.</td>
<td>5</td>
</tr>
<tr>
<td>PartiallySucceed</td>
<td>Function has partially succeeded</td>
<td>6</td>
</tr>
<tr>
<td>EmptyInput</td>
<td>No parameters were provided.</td>
<td>7</td>
</tr>
<tr>
<td>InvalidInput</td>
<td>A specified parameter is either invalid or missing.</td>
<td>8</td>
</tr>
</tbody>
</table>
CHAPTER 5

Reporting via Avaya CMS

In This Chapter

Overview ............................................................................68
Install & Configure Multimedia CMS Plug-in ...............69
CCE Queue Data Table ......................................................70
Run CCE Queue Data Report ..........................................73
Overview

Contact Center Express 4.0 makes some of the multimedia reporting data held within the Interaction Data Service easily available to the Avaya Call Management System (Avaya CMS) package.

To enable an Avaya CMS user to view information about multimedia work items currently held within Contact Center Express media store queues, a plug-in named Multimedia CMS Plug-in extracts the data from Interaction Data Server - Multimedia and inserts it into a custom table in the CMS Informix database. This data is then displayed to the user via a custom report called CCEData.rep.

On start up, Multimedia CMS Plug-in deletes (if it exists) all Contact Center Express-related data from the CMS database and recreates the table and statistical queue data. Table rows are automatically updated or added on receipt of events.
Install & Configure Multimedia CMS Plug-in

The Multimedia CMS Plug-in is installed and made available with the standard Interaction Data Server - Multimedia install.

After configuring the Multimedia CMS Plugin (see the [IDS MMAE CMS Engine] section of parameters within the Interaction Data Server - Multimedia ini file (on page 35)), you must also configure the ODBC driver that talks to the CMS Informix database.

To configure the ODBC driver:

1. On the server running the CMS Plug-in, click the Start button on the Windows Taskbar and select Control Panel from the pop-up menu.
2. Click Administrative Tools.
3. Double-click Data Sources (ODBC) to open ODBC Data Source Administrator dialog box.
4. Click the File DSN tab and click the Add button.
5. Select OpenLink Generic Driver and click Next.
6. Enter the name 'CMS' as the file data source, click Next and then Finish.
7. In the OpenLink ODBC Login window that appears, on the Identity tab enter the CMS username and password (see your CMS administrator for this).
8. On the Database tab, select Informix 2000 as the domain; leave the Name and Server Name blank; Hostname is the CMS server TCP/IP address; Protocol is TCP/IP; Port is 5000; Row buffer size is 60 and check the No Login box.
9. Click Connect. When you are asked which table to connect to, enter 'root.c_cceqd' and select the Remember password option.
Multimedia CMS Plug-in deletes and creates a new custom table called `c_cceqd` (CCE Queue Data) each time it starts up. This table contains the following:

<table>
<thead>
<tr>
<th><code>root_cmstbls</code></th>
<th><code>colname</code></th>
<th><code>_colname</code></th>
<th><code>syn_type</code></th>
<th><code>disp_type</code></th>
<th><code>csa_type</code></th>
<th><code>tabtype</code></th>
<th><code>descr</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>e_cceqd</td>
<td>ACD</td>
<td>acd</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td></td>
<td>ACD number associated with the row of data.</td>
</tr>
<tr>
<td>e_cceqd</td>
<td>AGENTSAVAILABLE</td>
<td>agentsavailable</td>
<td>no_syn</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td>The number of agents available in this queue. NOT USED</td>
</tr>
<tr>
<td>e_cceqd</td>
<td>AGENTSSTAFFED</td>
<td>agentsstaffed</td>
<td>no_syn</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td>The number of agents staffed in this queue. NOT USED</td>
</tr>
<tr>
<td>e_cceqd</td>
<td>AVGABNTIME</td>
<td>avgabntime</td>
<td>no_syn</td>
<td>duration</td>
<td>N</td>
<td>h</td>
<td>The average time an interaction waited before it was abandoned.</td>
</tr>
<tr>
<td>e_cceqd</td>
<td>AVGHANDLETIME</td>
<td>avghandletime</td>
<td>no_syn</td>
<td>duration</td>
<td>N</td>
<td>h</td>
<td>The average time an interaction takes to be completed.</td>
</tr>
<tr>
<td>e_cceqd</td>
<td>AVGWAITTIME</td>
<td>avgwaittime</td>
<td>no_syn</td>
<td>duration</td>
<td>N</td>
<td>h</td>
<td>The average time an interaction waits for service in this queue. This time does not include time an interaction is suspended.</td>
</tr>
<tr>
<td>e_cceqd</td>
<td>INTWAITING</td>
<td>intwaiting</td>
<td>no_syn</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td>The number of interactions waiting to be processed by this queue. This value does not include interactions in progress or suspended.</td>
</tr>
<tr>
<td>e_cceqd</td>
<td>LASTUPDATETIME</td>
<td>lastupdatetime</td>
<td>no_syn</td>
<td>date</td>
<td>N</td>
<td>h</td>
<td>The time the row was last updated.</td>
</tr>
<tr>
<td>e_cceqd</td>
<td>OLDESTINT</td>
<td>oldestint</td>
<td>no_syn</td>
<td>duration</td>
<td>N</td>
<td>h</td>
<td>How long the oldest interaction has been waiting for service. This will not include interactions that have been delivered to agents and then suspended for future processing.</td>
</tr>
<tr>
<td>e_cceqd</td>
<td>OWNERSID</td>
<td>ownersid</td>
<td>no_syn</td>
<td>text</td>
<td>N</td>
<td>h</td>
<td>The server instance ID of the Multimedia CMS Plug-in that generated data for this row.</td>
</tr>
<tr>
<td>tabname</td>
<td>colname</td>
<td>l_colname</td>
<td>syn_type</td>
<td>disp_type</td>
<td>csa_type</td>
<td>tabtype</td>
<td>descr</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------</td>
<td>-----------</td>
<td>----------</td>
<td>-----------</td>
<td>----------</td>
<td>---------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>c_cceqd</td>
<td>QUEUEID</td>
<td>queueid</td>
<td>no_syn</td>
<td>text</td>
<td>N</td>
<td>h</td>
<td>The identifier of the queue this information represents.</td>
</tr>
<tr>
<td>c_cceqd</td>
<td>QUEUENAME</td>
<td>queuename</td>
<td>no_syn</td>
<td>text</td>
<td>N</td>
<td>h</td>
<td>The queue name, eg email address, SMS number.</td>
</tr>
<tr>
<td>c_cceqd</td>
<td>SID</td>
<td>sid</td>
<td>no_syn</td>
<td>text</td>
<td>N</td>
<td>h</td>
<td>The server instance ID of the media store that generated this row of queue information.</td>
</tr>
<tr>
<td>c_cceqd</td>
<td>SIDFRIENDLYNAME</td>
<td>sidfriendlyname</td>
<td>no_syn</td>
<td>text</td>
<td>N</td>
<td>h</td>
<td>The friendly (display) name for the server that generated this row of queue information.</td>
</tr>
<tr>
<td>c_cceqd</td>
<td>SIT</td>
<td>sit</td>
<td>no_syn</td>
<td>text</td>
<td>N</td>
<td>h</td>
<td>The server instance type of the server that generated this row of queue information.</td>
</tr>
<tr>
<td>c_cceqd</td>
<td>TOTABNINT</td>
<td>totalnint</td>
<td>no_syn</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td>The number of interactions that have been abandoned before being delivered to an agent for processing. This figure will only be valid for session-based interactions (eg, web chat, MSN Messenger).</td>
</tr>
<tr>
<td>c_cceqd</td>
<td>TOTINTARRIVED</td>
<td>totintarrived</td>
<td>no_syn</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td>The total number of interactions received.</td>
</tr>
<tr>
<td>c_cceqd</td>
<td>TOTINTINPROGRESS</td>
<td>totintinprogress</td>
<td>no_syn</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td>The total number of interactions that have been delivered to agents for processing. This count will include interactions that have yet to be accepted by the agent.</td>
</tr>
<tr>
<td>c_cceqd</td>
<td>TOTINTINTERVAL</td>
<td>totintinterval</td>
<td>no_syn</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td>The total number of interactions received this interval.</td>
</tr>
<tr>
<td>c_cceqd</td>
<td>TOTINTLASTHOUR</td>
<td>totintlasthour</td>
<td>no_syn</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td>The total number of interactions that were received in the last hour.</td>
</tr>
<tr>
<td>tabname</td>
<td>colname</td>
<td>l_colname</td>
<td>syn_type</td>
<td>disp_type</td>
<td>csa_type</td>
<td>tabtype</td>
<td>descr</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------</td>
<td>-----------</td>
<td>----------</td>
<td>-----------</td>
<td>----------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>c_ceeqd</td>
<td>TOTINTSUSPENDED</td>
<td>totintsuspended</td>
<td>no_syn</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td>The total number of interactions that are currently suspended for this queue.</td>
</tr>
<tr>
<td>c_ceeqd</td>
<td>TOTINTSUSTHISINTV</td>
<td>totintsusthisintv</td>
<td>no_syn</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td>The total number of interactions suspended this interval.</td>
</tr>
<tr>
<td>c_ceeqd</td>
<td>TOTINTTHISHOUR</td>
<td>totinthishour</td>
<td>no_syn</td>
<td>number</td>
<td>N</td>
<td>h</td>
<td>The total number of new interactions that have been received this hour.</td>
</tr>
</tbody>
</table>
Run CCE Queue Data Report

On the Contact Center Express DVD (see Server\Interaction Data Server\CMS Report), there is the report called CCEData.rep. It is similar to the VDN real-time report found in the base CMS package.

To use this report, it must be copied to the directory on the CMS box that contains CMS reports.

1. Open Avaya CMS Supervisor.
2. Select **Commands > Reports** from the menu bar. The **Select a Report** dialog box appears:

3. Click the **Historical** tab.
4. Click ‘Designer’ within the **Category** list box.
5. Click the **Copy** button at the bottom of the dialog box. The **Copy Report** dialog box appears:

6. Select **From a PC File to the CMS Server**.
7. Click **OK**.
8. Browse the Contact Center Express DVD (Server\Interaction Data Service\CMS Report) and select ‘CCEData.rep’ file.
A message appears confirming the report has been successfully loaded.

9 To run the report, double-click CCE QueueData.
Index

**A**
- Administration • 28
- Agent • 54, 61

**C**
- CCE Queue Data Table • 70
- Configure Interaction Data Server - Multimedia • 35, 69
- Configure Interaction Data Server - View • 40
- Configure Interaction Data Server - Voice and Presence • 29
- Conversation & Customer • 63
- Create/Upgrade Databases • 23

**D**
- Delete Record Types from Database • 26
- Developing Client Applications • 47
- Device • 52

**E**
- Enumerations • 66
- Error Logging • 13

**H**
- Historical Reporting Data • 61

**I**
- Install & Configure Multimedia CMS Plug-in • 69
- Install Interaction Data Service • 15
- Installation Requirements • 7
- Interaction • 62
- Introduction • 10

**K**
- Knowledge Base • vi

**L**
- Licensing • 13

**M**
- Miscellaneous Capabilities • 64

**O**
- Overview • 48, 68

**P**
- Preface • iv
- Product Name Changes • vi

**Q**
- Queue • 59, 61

**R**
- Real Time Reporting Data • 51
- Related Documents • vi, 14
- Reporting via Avaya CMS • 12, 67
- Run CCE Queue Data Report • 73

**S**
- Setting Up Your Interaction Data Service • 14
- Skill • 61
- Start Service • 45
- System Requirements • 7

**V**
- VDN • 57

**W**
- What is the Interaction Data Service? • 11