Avaya™ Call Management System (CMS)

Open Database Connectivity

585-780-701
Issue 1.0
May 2002
COMPAS ID 89705

Notice

Every effort was made to ensure that the information in this document was complete and accurate at the time of printing. However, information is subject to change.

Preventing Toll Fraud

"Toll fraud" is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or working on your company's behalf). Be aware that there may be a risk of toll fraud associated with your system and that, if toll fraud occurs, it can result in substantial additional charges for your telecommunications services.

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. For additional support telephone numbers, see the Avaya web site:

http://www.avaya.com

Select **Support**, then select **Escalation Lists US and International**. This web site includes telephone numbers for escalation within the United States. For escalation telephone numbers outside the United States, select **Global Escalation List**.

Providing Telecommunications Security

Telecommunications security (of voice, data, and/or video communications) is the prevention of any type of intrusion to (that is, either unauthorized or malicious access to or use of) your company's telecommunications equipment by some party.

Your company's "telecommunications equipment" includes both this Avaya product and any other voice/data/video equipment that could be accessed via this Avaya product (that is, "networked equipment").

An "outside party" is anyone who is not a corporate employee, agent, subcontractor, or working on your company's behalf. Whereas, a "malicious party" is anyone (including someone who may be otherwise authorized) who accesses your telecommunications equipment with either malicious or mischievous intent.

Such intrusions may be either to/through synchronous (time-multiplexed and/or circuit-based) or asynchronous (character-, message-, or packet-based) equipment or interfaces for reasons of:

- Utilization (of capabilities special to the accessed equipment)
- Theft (such as, of intellectual property, financial assets, or toll-facility access)
- Eavesdropping (privacy invasions to humans)
- Mischief (troubling, but apparently innocuous, tampering)
- Harm (such as harmful tampering, data loss or alteration, regardless of motive or intent)

Be aware that there may be a risk of unauthorized intrusions associated with your system and/or its networked equipment. Also realize that, if such an intrusion should occur, it could result in a variety of losses to your company (including but not limited to, human/data privacy, intellectual property, material assets, financial resources, labor costs, and/or legal costs).

Your Responsibility for Your Company's Telecommunications Security

The final responsibility for securing both this system and its networked equipment rests with you - an Avaya customer's system administrator, your telecommunications peers, and your managers. Base the fulfillment of your responsibility on acquired knowledge and resources from a variety of sources including but not limited to:

- Installation documents
- System administration documents
- Security documents
- Hardware-/software-based security tools
- Shared information between you and your peers
- Telecommunications security experts

To prevent intrusions to your telecommunications equipment, you and your peers should carefully program and configure:

- your Avaya-provided telecommunications systems and their interfaces
- your Avaya-provided software applications, as well as their underlying hardware/software platforms and interfaces
- any other equipment networked to your Avaya products.

Trademarks

The following trademarks are mentioned in this document:

- *MultiVantage and DEFINITY* are registered trademarks of Avaya Inc.
- *Enterprise, Solaris, SPARCserver, Network Terminal Server, Sun, SunSwift, Solstice, DiskSuite,* and *Ultra* are trademarks or registered trademarks of Sun Microsystems, Inc.
- *INFORMIX* is a registered trademark of Informix Software, Inc.
- *Multiport* is a registered trademark of Aurora Technologies, Inc.
- *Windows* is a registered trademark of Microsoft, Inc.

All other product names mentioned herein are the trademarks of their respective owners.

Ordering Information

Call: Avaya Publications Center
Voice +1 800 457 1235
Fax +1 800 457 1764
International Voice +1 410 568 3680
International Fax+1 410 891 0207

Write: Globalware Solutions
200 Ward Hill Avenue
Haverhill, MA 01835 USA
Attention: Avaya Account Manager

E-mail:totalware@gwsmail.com

Order: Document No. 585-780-701, Issue 1.0
May 2002

You can be placed on a Standing Order list for this and other documents you may need. Standing Order will enable you to automatically receive updated versions of individual documents or document sets, billed to account information that you provide. For more information on Standing Orders, or to be put on a list to receive future issues of this document, please contact the Avaya Publications Center.

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 and Canada. For additional support telephone numbers, see the Avaya web site:

http://www.avaya.com

Select **Support**, then select **Escalation Lists US and International**. This web site includes telephone numbers for escalation within the United States. For escalation telephone numbers outside the United States, select **Global Escalation List**.

Acknowledgment

This document was written by the CRM Development group.

Avaya Call Management System
Open Database Connectivity

Contents

Chapter 2: ODBC Driver and installation

Preface

Overview

Open Database Connectivity (ODBC) is an optional Avaya Call Management System (CMS) feature that allows you to access data in the CMS database for use in other software applications such as spreadsheet programs. With ODBC, you can access the CMS data directly from your application, without needing to understand database connectivity or format.

This feature is especially useful for call centers with multiple sites. ODBC allows access to data at multiple sites for reports. ODBC uses Structured Query Language (SQL) to access data, eliminating the need to access the database in the language in which it was written.

The ODBC feature is a client/server feature; that is, the clients must be attached to a network that is fully functional to access the server. The “clients” are the computers that are accessing data through ODBC; the “server” is the CMS machine where the CMS database is located. R3V6 through R3V11 ODBC, work only in English.

Organization of document

- Chapter 1 — Open Database Connectivity

  Presents an overview of the ODBC feature, its interaction with other CMS features, and its performance impact. Also included is an overview of ODBC functionality and its interaction with the CMS database.

- Chapter 2 — ODBC Driver and installation

  Presents an overview of the off-the-shelf driver that will be used for the ODBC feature and its function in relation to CMS data. The majority of this chapter covers basic installation, configuration, and initialization procedures for the driver on the server and the clients.

- Chapter 3 — Database tables

  Describes the CMS open database tables and the items in the database tables.

Audience

This document is written for CMS users who need to understand open access to database items. It is also written to help users decide which database items to use in custom reports, spreadsheets, and other user applications with external data access. The database items can be exported through ODBC for use in a variety of software programs. For data extraction purposes, users will need to have a basic understanding Structured Query Language (SQL) before using this document. Users should also have a basic understanding of database logic and purpose.

Trademarks

The following trademarks are mentioned in this document:

- *Avaya* is a registered trademark of Avaya Inc.
- *Enterprise, Solaris, SPARCserver, Network Terminal Server, Sun, SunSwift, Solstice, DiskSuite,* and *Ultra* are trademarks or registered trademarks of Sun Microsystems, Inc.
- *INFORMIX* is a registered trademark of Informix Software, Inc.
- *DEFINITY* is a registered trademark of Avaya.
- *Windows, Access*, *Microsoft*, *MS*, *MS-DOS*, *Windows*, *Windows 95, Windows 98, Windows NT,* and *Windows 2000* are trademarks or registered trademarks of Microsoft, Inc.
- *OpenLink* is a trademark of OpenLink Software
- *Crystal Reports* is a trademark of SeaGate Software
- *Intel* is a registered trademark of Intel Corp.

All other product names mentioned herein are the trademarks of their respective owners.

Conventions used in this document

The following conventions are used throughout this document:

CMS	The acronym CMS is used throughout this document instead of the full name *Avaya Call Management System.*
Related document callouts	Includes the document number (if it is an Avaya publication) and the book title in italic. For example, see the *Avaya CMS R3V11 Administration*, 585-210-910 document for more information.
`Courier` font	Represents Informix ® table names, or system output
`Bold Courier` font	Represents commands entered from a terminal window.
Bold font	Represents commands entered into an ODBC input screen. For example, type **60** in the **Others: Row Buffer Size** field. Highlights references to file names and directories.
italic font	Represents variable information in commands.
Subsystem References	Are always initial capitalized. For example, set the access permissions for CMS users from the User Permissions subsystem.
Window Titles	Are always capitalized.

Terminology

The following terms are used in this document.

API	Application Programming Interface (API). For ODBC, an API is the user interface that can generate queries to a database.
CMS	Call Management System
DBMS	Database Management System (DBMS). CMS uses the Informix DBMS.
Driver	An implementation of an ODBC API that supports a particular DBMS; for example, Informix DBMS. More specifically, a driver is a dynamic link library.
Driver Manager	A driver manager loads specific drivers based on the type of data being accessed.
Informix	A relational database in which CMS stores much of its data, including administration and historical data. The database is an Informix Standard Engine (SE) for CMS loads that are R3V8 or earlier, and Informix Dynamic Server (IDS) for CMS R3V9.
ODBC	Open Database Connectivity. ODBC is an interface that allows applications to access data from a database using SQL.
SQL	Structured Query Language.

This section lists sources for related information about contact center products and features. Not all documents are supported for all CMS releases or equipment.

To order Avaya documentation, call the Avaya Publications Center at 1-800-457-1235 or +1-410-568-3680.

Related documents

Related Documents lists sources for related information about contact center products and features. Not all documents are supported for all CMS releases or equipment.

To order Avaya documentation, call the Avaya Publications Center at 1-800-457-1235 or +1-410-568-3680.

CMS software documents

Document title	Document number
Installing software on a CMS computer	
Avaya Call Management System Release 3 Version 11 Software Installation, Maintenance, and Troubleshooting	585-215-115
CentreVu Call Management System Release 3 Version 9 Software Installation, Maintenance, and Troubleshooting	585-215-956
CentreVu Call Management System Release 3 Version 8 Software Installation, Maintenance, and Troubleshooting	585-210-941
Setting up a disk-mirrored system	
Avaya Call Management System Release 3 Version 11 Software Installation, Maintenance, and Troubleshooting	585-215-115
CentreVu Call Management System Release 3 Version 9 Software Installation, Maintenance, and Troubleshooting	585-215-956
CentreVu Call Management System Release 3 Version 8 Disk-Mirrored Systems	585-210-940

Upgrade documents

There are several upgrade paths supported with CMS. For each of these upgrades, there is a document designed to support that upgrade. Note that none of the following upgrade documents are available from the publications center, but are available from the Avaya CMS documentation Web site.

- Base load upgrades

  A base load upgrade is used when upgrading CMS to the latest load of the same version (for example, R3V9 ak.g to R3V9 al.k). A specific set of instructions is written for the upgrade and is shipped to the customer site with the CMS software CD-ROM as part of a Quality Protection Plan Change Notice (QPPCN).

Document title
Avaya Call Management System Release 3 Version 11 Base Load Upgrade Procedures
CentreVu Call Management System Release 3 Version 9 Base Load Upgrade Procedures

- Platform upgrades and data migration

  A platform upgrade is used when upgrading to a new hardware platform (for example, upgrading from a SPARCserver 5 to an Enterprise 3500). The new hardware platform is shipped from the Avaya factory with the latest CMS load. Therefore, as part of the upgrade you will have the latest CMS load (for example, R3V9 to R3V11, or the latest load of the same CMS version). For R3V11, a specific set of instructions are written for the upgrade and are shipped to the customer site with the new hardware.

Document title
Avaya Call Management System Release 3 Version 11 Platform Upgrade and Data Migration Instructions
CentreVu Call Management System Release 3 Version 9 Platform Upgrade and Data Migration Instructions

- Avaya Call Management System Upgrade Express (CUE)

  CUE is used in the following conditions:

  — CMS is being upgraded from an earlier version (for example, R3V5u or R3V6) to the latest version (for example, R3V9 or R3V11).

  — The hardware platform is not changing.

  A specific set of upgrade instructions is written for the upgrade and is shipped to the customer site with the CUE kit.

Document title
Avaya Call Management System Release 3 Version 11 Sun Blade 100 Computer CUE Instructions
Avaya Call Management System Release 3 Version 11 Sun Blade 100 Computer Mirrored System CUE Instructions
Avaya Call Management System Release 3 Version 11 Sun Ultra 5 Computer CUE Instructions
Avaya Call Management System Release 3 Version 11 Sun Enterprise 3000 Computer CUE Instructions
Avaya Call Management System Release 3 Version 11 Sun Enterprise 3000 Computer Mirrored System CUE Instructions
Avaya Call Management System Release 3 Version 11 Sun Enterprise 3500 Computer CUE Instructions
Avaya Call Management System Release 3 Version 11 Sun Enterprise 3500 Computer Mirrored System CUE Instructions
Avaya Call Management System Release 3 Version 9 Sun Ultra 5 Computer CUE Instructions
Avaya Call Management System Release 3 Version 9 Sun Enterprise 3000 Computer CUE Instructions
Avaya Call Management System Release 3 Version 9 Sun Enterprise 3000 Computer Mirrored System CUE Instructions
Avaya Call Management System Release 3 Version 9 Sun Enterprise 3500 Computer CUE Instructions
Avaya Call Management System Release 3 Version 9 Sun Enterprise 3500 Computer Mirrored System CUE Instructions

Hardware documents

Document title	Document number
Avaya Call Management System Sun Blade 100 Computer Hardware Installation, Maintenance, and Troubleshooting	585-310-783
Avaya Call Management System Sun Blade 100 Computer Connectivity Diagram	585-310-782
Avaya Call Management System Sun Enterprise 3500 Computer Hardware Installation, Maintenance, and Troubleshooting	585-215-873
Avaya Call Management System Sun Enterprise 3500 Computer Connectivity Diagram	585-215-877
Avaya Call Management System Sun Ultra 5 Computer Hardware Installation, Maintenance, and Troubleshooting	585-215-871
Avaya Call Management System Sun Ultra 5 Computer Connectivity Diagram	585-215-872
Avaya Call Management System Sun Enterprise 3000 and SPARCserver Computers Hardware Maintenance and Troubleshooting	585-214-016
Avaya Call Management System Terminals, Printers, and Modems	585-215-874

Switch documents

Document title	Document number
CMS Switch Connections, Administration, and Troubleshooting	585-215-876

Administration documents

Document title	Document number
Avaya Call Management System Release 3 Version 11 Administration	585-215-515
CentreVu Call Management System Release 3 Version 9 Administration	585-214-015
CentreVu Call Management System Release 3 Version 8 Administration	585-210-910

Other documents

Document title	Document number
Avaya Call Management System Open Database Connectivity	585-780-701
Avaya Call Management System Release 3 Version 11 LAN Backup User Guide	585-215-715
Avaya CMS Release 3 Version 11 External Call History Interface	585-780-700
CentreVu CMS Release 3 Version 9 External Call History Interface	585-215-952
Avaya CMS Custom Reports	585-215-822
Avaya CMS Forecast	585-215-825
Avaya Visual Vectors Version 9 Installation and Getting Started	585-210-947
Avaya Visual Vectors Version 9 User Guide	585-210-944
Avaya Visual Vectors Release 11 Installation and Getting Started	585-210-706
Avaya Visual Vectors Release 11 User Guide	585-210-709

Documentation Web sites

For product documentation for all Avaya products and related documentation, go to http://www.avayadocs.com.

Important:
Additional information about new software or hardware updates will be contained in future issues of this book. New issues of this book will be placed on the web site when available.

Use the following web sites to view related support documentation:

- Sun hardware documentation

  http://docs.sun.com

- Okidata printer documentation

  http://www.okidata.com

- Informix documentation

  http://www.informix.com

- Tivoli Storage Manager documentation

  http://tivoli.com/support/documents/public_manuals.html

Chapter 1: Open Database Connectivity

Overview

"Open Database Connectivity" presents an overview of the Avaya Call Management System (CMS) Open Database Connectivity (ODBC) feature, its interaction with other CMS features, and its performance impact. An overview of ODBC functionality and its interaction with the CMS database is also included.

Contents

"Open Database Connectivity" contains the following topics:

ODBC features

Overview

An OpenLink™ ODBC driver provides the ODBC communication and connectivity that allows external data access to the CMS database, and is installed at the call center. The driver supports Solaris™, Windows 95®, Windows 98®, Windows 2000®, and Windows NT® clients. All historical CMS database tables, dictionary tables, and customer-provided tables can be accessed by ODBC clients through standard ODBC-enabled software applications.

All standard Structured Query Language (SQL) queries generated by the user applications are supported by the ODBC driver as limited by permissions. Table-level, read-only permissions restrict access to database tables. The tables accessible in the CMS Release 3 Version 11 database are described in detail in Chapter 3, Database tables. ODBC does not support double byte languages. ODBC only supports English.

Contents

Queries

Queries can be generated from client to server and from user applications. The ODBC driver is installed on the server, and the accompanying software is installed on the clients. The extracted data can be used in workforce management packages, network routers, and blended inbound/outbound applications. Any application that supports ODBC functionality can access permissible tables (for example, CMS Dictionary).

The same recommendations for composing queries in the CMS custom report editor also apply to ODBC queries. For example, accessing large tables such as the split/skill or agent tables, or joining tables in queries may have a negative impact on CMS performance. Use the exact table and database item names when querying the database. When performing calculations, keep in mind that arithmetic operations are performed in order from left to right. Multiplication and division operations are performed before addition and subtraction operations, unless the addition or subtraction operations are enclosed in parentheses. Operations in parentheses are always performed first. Therefore, it is very important to review queries before sending them to the database.

Additional information on queries

For more information on how to compose efficient database queries, see the “Edit | Queries” chapter in the *Avaya Report Designer Version 11 User Guide, 585-210-707*.

Performance impact

The number, size, and types of queries received by the CMS may impact performance. To keep system impact at a minimum, certain types of queries, such as accessing large tables or performing table joins, should be run during a period of low agent and real-time report activity; for example, during off hours.

Note:

During off-peak hours CMS runs it’s own activities, such as archiving and making backups. This can use a significant amount of resources and time when working with a large database.

Prioritize very involved queries the same way you prioritize reports during high business activity. The impact of complex or multiple queries on the database will be similar to running multiple reports. Another important performance impact consideration is the number of simultaneous database accesses allowed. To optimize system performance, minimize the number of database connects and disconnects from an application, and spread your ODBC activities throughout the day.

Security and port allocation

ODBC users log into the CMS server with password protection. Users have SQL access to Informix tables as limited by the table permissions in Chapter 3, Database tables. All historical and dictionary database tables have read-only access permission. The customer-created tables on the host (any table names beginning with "*c_*") have read and write permissions. All other tables are not accessible through ODBC.

If your network uses a firewall, it is common for unused ports to be locked down. ODBC uses UDP port 60001, and the TCP ports between PortLow and PortHigh. If these ports are locked, you will be unable to connect to the CMS database with ODBC.

Depending on the ODBC version, the ports will be defined under `[Protocol TCP]` in one of the following files:

- For version 3.2, the file is called **/cms/dc/odbc/cmsrqb3.2_init**
- For versions less than 3.2 the file is called **/cms/dc/odbc/cmsrqb_init**

CMS feature interactions requiring client support

The following CMS features require specific client administration and interaction. Most importantly, data received by the client application is in "raw" form; that is, it will need to be formatted for use within your application. Dictionary names and certain time fields are most affected by this formatting. Refer to Chapter 3, Database tables, for further information on data format and values.

- *Dictionary Names:* Clients can access CMS Dictionary names. The client must map the synonym to the report. Underlying data is numeric; for example, different splits are stored as numbers and not by their names.

  Note:

  To minimize performance impact when accessing synonyms, download the synonyms to your client application or database and perform the join at the client.

- *Permissions:* Applications which access Informix® externally, such as database access scripts, may not work if the table permission script tries to access a table to which permission is denied. Note that CMS-like permissions to individual entities such as splits and vectors are not available through the ODBC interface. It is up to the client application to create and enforce permissions at this level if this is required.
- *Field Display:* The time and date data received from the database may not be formatted. Generally, times may be shown in seconds or in military format. Review data for formatting when importing it into your software application. See your software's documentation for further information on formatting data.

ODBC background and functionality

Overview

Open Database Connectivity (ODBC) is an Application Programming Interface (API) that interfaces with one or many Database Management Systems (DBMSs) and allows queries to access data in the database for extraction and use in reports and other outside applications.

Contents

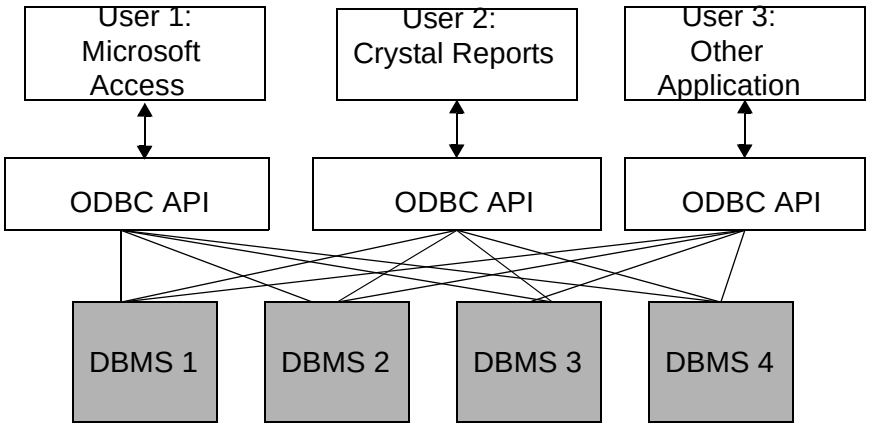
"ODBC background and functionality" contains the following topics:

Data access through ODBC

ODBC was developed as a solution to accessing different types of data. Within one call center, users may be working with different applications such as Microsoft Access™ or Seagate Crystal Reports™, that must access call center data. Users may need to import CMS data into many different applications. ODBC provides a standard method of database access that shields users from the underlying functionality of network software, naming conventions, and the other complexities involved in accessing data through a DBMS. The data only needs to be queried through the embedded SQL query function in the Windows® application that you are using. Refer to your specific application documentation for further information on its embedded SQL function.

Data access diagram

The following figure illustrates user data access through ODBC.



Structured query language (SQL)

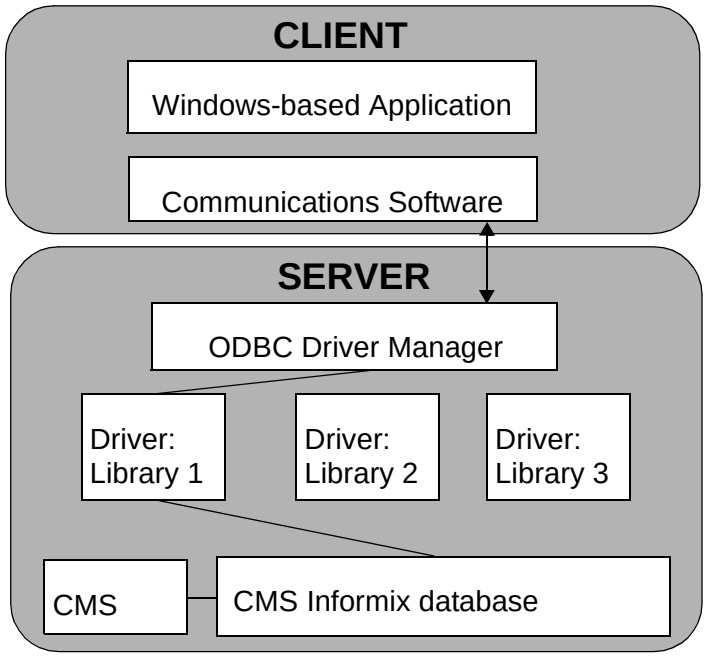
ODBC uses Structured Query Language (SQL) to query and access data. Because SQL is a language, queries written in SQL can be used to access data with different formats. SQL is the basis for relational database access. The simplest description of a relational database model is the table—data is stored in rows and columns, and relationships between tables are established through data items with matching values between the tables. SQL queries access the data stored in the relational database tables and extract it for use in other applications. SQL can also be used to construct data calculations. This is useful if you want to look at a sum of the data; for example, the total number of calls routed to a particular split or skill. An SQL query is composed from the Windows application for which you need the data. The SQL query function is embedded in the application itself.

CMS ODBC drivers

The CMS Informix database management system is supported by the OpenLink Multi-Tier ODBC Driver. A driver is an implementation of an ODBC application programming interface that supports a particular database management system; in this case, the Informix DBMS used by CMS. More specifically, a driver is a dynamic link library that is specific to a type of DBMS. The driver manager provides the link between the user's applications and the DBMS itself by selecting the dynamic link library using the format of the data being queried by processing the received ODBC function calls. More information on the OpenLink driver and its installation procedures can be found in Chapter 2, ODBC Driver and installation.

ODBC driver functionality

The following figure illustrates how drivers provide the interface between user applications and the data being accessed, assuming that driver dynamic link library 1 is the Informix specific library.



Uses for ODBC data

Data extracted and stored by an ODBC application can be used by ODBC-enabled programs, such as workforce management packages, network routers, and blended inbound/outbound applications. Data can also be extracted for use in ODBC-enabled programs, including spreadsheets and report writers. An example of an ODBC data application is generating reports using data from multiple call center sites and their databases.

Requesting data using ODBC

The ODBC driver installed on the CMS is compatible with the CMS Informix database. All queries in ODBC must be composed to ensure continued CMS performance. The query is invoked differently in each Windows application. For more information on how your application interfaces with ODBC, refer to your application's user documentation. For more information on how to compose efficient database queries, see the "Edit | Queries" chapter in the *Avaya Report Designer Version 11 User Guide,* Performance impact and Structured query language (SQL) sections in this chapter.

Chapter 2: ODBC Driver and installation

Overview

This chapter presents an overview of the driver used for the Open Database Connectivity (ODBC) feature and its function in relation to Avaya Call Management System (CMS) data. The second half of this chapter covers basic installation, configuration, and initialization procedures for the driver. Server connection procedures for the client are also given.

Procedures are included for personal computer clients, Solaris clients, and the CMS server. Troubleshooting tips and OpenLink ODBC Error Messages are included at the end of the chapter.

All server and client driver components are included on the Call Management System OpenLink ODBC Driver compact disk. You will use this disk for both client and server installation.

CAUTION:

If you choose to develop an application for the ODBC driver, Avaya cannot provide support for that application, or for any other third party software or related mapping.

Contents

"ODBC Driver and installation" contains the following topics:

ODBC driver

Overview

An OpenLink Multi-Tier ODBC driver facilitates client querying of data for extraction from the CMS database. The client driver supports client access from Solaris, Windows 95, Windows 98, Windows 2000, Windows XP, and Windows NT clients. All historical CMS database tables, dictionary tables, and customer-defined tables can be accessed by ODBC clients through standard Structured Query Language (SQL) queries. All standard SQL queries are supported by the ODBC driver as limited by permissions. Table-level permissions will restrict access to database tables. The tables are described in detail in Chapter 3, Database tables.

Function

OpenLink ODBC drivers are composed of two main components: a generic ODBC driver and the OpenLink Request Agent. The OpenLink Request Agent resides on the client and interfaces with the ODBC driver through proprietary protocol. ODBC drivers are controlled by an ODBC driver manager, in this case, the OpenLink Request Broker. The OpenLink Request Broker resides on the server. ODBC uses data source names as the link between the ODBC Request Broker and the relevant ODBC driver for a particular database.

The OpenLink ODBC driver is a multi-tier driver; that is, the driver controls both ODBC calls and the manner in which these calls are transferred to the relevant database management system (in CMS, Informix) over the communications infrastructure. The OpenLink Request Broker provides the link between the user's applications and the database management system (DBMS—in CMS, Informix). Once it is installed and administered, the ODBC driver and its components are transparent to the client. The CMS ODBC feature allows multiple, synchronous accesses from clients, users, and/or applications.

Supported logins

The CMS ODBC feature supports up to five or ten simultaneous logins, in increments of five, depending on your licensing agreement.

Additional information

For additional information on the ODBC driver, see the on-line help file that is included on the Call Management System OpenLink ODBC Driver compact disk.

Installation and configuration procedures

Overview

The server and the client ODBC driver software will be installed from the CMS Call Management System OpenLink ODBC Driver compact disk. This disk contains all necessary files for server, Solaris client, and Windows client installation. OpenLink ODBC driver installation and configuration procedures in this chapter are presented for the Sun Solaris/CMS server, Sun Solaris clients, and personal computer clients. Refer to the installation and configuration procedures that are specific to your client/server environment. Installation on your server will not impact system performance; the ODBC server components may be installed at any time.

- To install and configure the ODBC driver in a Sun Solaris/CMS client/server environment, see Installing the ODBC driver on the CMS server and Solaris client installation in this chapter.
- To install and configure the ODBC driver on your CMS server and on your personal computer for accessing CMS data, see Installing the ODBC driver on the CMS server, Desktop computer client installation, and Desktop computer client configuration in this chapter.
- For additional information on installing, configuring, and operating the ODBC driver, see the on-line help file that is included on the Call Management System OpenLink ODBC Driver compact disk.
- Once installed and configured, a valid CMS login has ODBC access.

Installing the ODBC driver on the CMS server

Introduction

"Installing the ODBC driver on the CMS server" describes system requirements, software installation procedures, configuration procedures, and initialization procedures for the OpenLink Multi-Tier ODBC driver on the CMS server.

System requirements

The following system requirements must be met prior to installing the software:

- The ODBC server is on a standard CMS hardware Sun platform with Solaris version 2.4 or later, and on the SPARCserver platform with CMS R3V5 or later.
- The network communication software is correctly installed and configured, and the network is fully functional, so that the server can communicate with the clients.

Installation requirements

Before installing the OpenLink Version 3.x ODBC driver on the server, determine which version of CMS is installed.

If you need to determine the CMS version, enter:

```
pkginfo  -x cms
```

The required ODBC version depends on the version of CMS installed on the system:

- if CMS R3V6, R3V8, R3V9 or R3V11 is installed see Installing ODBC version 3.2 on page 34
- if CMS R3V5 is installed see Installing ODBC version 3.0 on page 31

Installing ODBC version 3.0

Install ODBC version 3.0 only on CMS R3V5 systems. For the ODBC driver installation procedure for CMS R3V6, R3V8, R3V9 and R3V11 see Installing ODBC version 3.2 on page 34.

To install ODBC version 3.0:

1. Load the “Avaya CMS OPENLINK ODBC Driver” CD-ROM into the CD drive.
2. After about 15 seconds, enter **mount** to verify the name of the CD-ROM.

   The system displays a list of devices and file systems currently mounted. The last line should display the installed CD-ROM as shown below:

   ```
   /cdrom/odbc_driver on /vol/dev/dsk/c0t2d0/odbc_driver read
   only on Sat Jun  6 11:47:05 2002
   ```

3. If this is an upgrade of the ODBC software, enter the following commands to shut down the request broker and remove the old **openlink** directory; otherwise, proceed to Step 4.

   a. To shut down the request broker, enter:

      /cms/dc/odbc/odbc_init -r 0

   b. To remove the old **openlink** directory, enter:

      rm -fr /usr/openlink

4. Create the OpenLink ODBC driver directory by entering the following commands:

   mkdir /usr/openlink

   cd /usr/openlink

5. To confirm you are in **/usr/openlink**, enter:

   pwd

6. Enter the following commands to copy the files from the CD-ROM and install the files:

   cp /cdrom/cdrom0/server/cmsv5/* /usr/openlink

   ./install.sh

   The system displays the following message:

   ```
   Extracting (sladi5xx.taz) ...
   Extracting (sladi71x.taz) ...
   Extracting (slbrxxxx.taz) ...
   Extracting (slkoxxxx.taz) ...
   Enter the name of the user that owns the programs:
   ```

7. Enter **root** as the name of the user who will own the programs.

   The system displays the following message:

```
Enter the name of the group that owns the programs:
```

8. Enter **root** as the name of the group that will own the programs.

   The system displays the following message:

```
Changing ownership ...
Registering ...
oplrqb is now registered to Lucent Technologies BCS.
This is a 5 concurrent users license
that will not expire.
....
....
....
```

9. Press **Enter**.

   The system displays the following message:

```
Log File? [www_sv.log]
```

10. Press **Enter**.

    The system displays the following message:

```
Log all requests (y/n)? [n]
```

11. Press **Enter**.

    The system displays the following message

```
Administrator account? [admin]
```

12. Press **Enter**.

    The system displays the following message:

```
Administrator’s password? [admin]
```

13. Press **Enter**.

    The system displays the following message:

```
 Press return to proceed to the next phase of the install process:
```

14. Press **Enter**.

    The system displays the OpenLink Session Rules Book Configuration Utility menu.

```
Choose an item or type q to quit : q
```

15. Enter: **q**

    The system displays the following message:

```
End of installation.
```

16. Choose one of the following commands to configure and initiate the ODBC software:

    - If this is a new install, enter:

      /cms/dc/odbc/odbc_init

      The system displays the following message:

```
ODBC driver initialization complete
```

    - If this is an upgrade or reinstallation, enter:

      /cms/dc/odbc/odbc_init -r 1

      The system displays the following message:

```
oplrqb has been activated
```

17. Enter the following to verify that the ODBC Request Broker is active on the server:

    ps -ef | grep oplrqb

    The system displays a message similar to the following:

```
root  3354  3351  0 11:49:43 ?        0:00
/usr/openlink/bin/oplrqb -f +configfile
/cms/dc/odbc/cmsrqb_init +loglevel 5 +l
root  3359  3317  0 11:50:11 pts/5    0:00 grep oplrqb
```

18. Enter **cd** to move to the root directory.
19. Enter: **eject cdrom**

Installing ODBC version 3.2

For the ODBC driver installation procedure for R3V5 systems., see Installing ODBC version 3.0 on page 31.

To install ODBC version 3.2:

1. Load the “Avaya CMS OPENLINK ODBC Driver” CD-ROM.
2. After about 15 seconds, verify the name of the CD-ROM. Enter:

   mount

   The system displays a list of devices and file systems currently mounted. The last line should display the installed CD-ROM as shown below:

   ```
   . . .
   . . .
   /cdrom/openlink on /vol/dev/dsk/c0t2d0/openlink
   read only on (current date and time)
   ```

3. If this is an upgrade of the ODBC software, enter the following commands to shutdown the request broker and remove the old **openlink** directory; otherwise, proceed to Step 4.

   a. To shut down the request broker, enter:

      /cms/dc/odbc/odbc_init -r 0

   b. To remove the old **openlink** directory, enter:

      rm -fr /usr/openlink

4. Create the OpenLink directory and change to the new directory by entering the following commands:

   mkdir /usr/openlink

   cd /usr/openlink

5. To confirm you are in **/usr/openlink**, enter:

   pwd

6. To copy the server components from the CD-ROM, enter:

   cp /cdrom/cdrom0/server/cmsv6v8/* /usr/openlink

7. To install the server components on the system, enter:

   `./install.sh`

   The system displays the following message:

```
Extracting (smadi5zz.taz) ...
Extracting (smadi72z.taz) ...
Extracting (smaozzzz.taz) ...
Extracting (smbrzzzz.taz) ...
.........
.........
.........
TCP/IP Port to use? [8000]:
```

   Note:

   On some systems, the following message may be displayed:

   `Saving existing CMS odbc settings` - This file will be replaced with a new **cmsrqb_init** file for use with Openlink 3.2

8. To accept the default setting, press **Enter**.

   The system displays the following message:

```
Log File? [www_sv.log]
```

9. To accept the default setting, press **Enter**.

   The system displays the following message:

```
Log all requests (y/n)? [n]
```

10. To accept the default setting, press **Enter**.

    The system displays the following message:

```
Administrator account? [admin]
```

11. To accept the default setting, press **Enter**.

    The system displays the following message:

```
Administrator’s password? [admin]
```

12. To accept the default setting, press **Enter**.

    The system displays the following message:

```
The OpenLink Admin Assistant is now ready for use.
.........
.........
.........
Enter the name of the user that will own the programs [ENTER=Use
Current User Settings]
```

13. Enter: **`root`**

    The system displays the following message:

```
Enter the name of the group that will own the programs [ENTER=Use
Current Group Settings]
```

14. Enter: **`root`**

    The system displays the following message:

```
Changing ownership ...
Press return to proceed to the next phase of the
install process
```

15. Press **Enter**.

    The system displays the OpenLink Session Rules Book Configuration Utility menu.

```
 OpenLink Session Rules Book Configuration Utility
================================================================

1.  Request Broker              11. PostgreSQL
2.  Informix 5                  12. Progress 6
3.  Informix 6                  13. Progress 7
4.  Informix 7                  14. Progress 8
5.  Ingres 6                    15. Solid
6.  Virtuoso                    16. Sybase 4
7.  OpenIngres                  17. Sybase 10
8.  Oracle 6                    18. Sybase 11
9.  Oracle 7                    19. Unify 2000
10. Oracle 8                    20. Velocis

U.  Undo last change             V.  View the current Rules Book
C.  Clear log file               L.  View log file
B.  Backup Rules Book            R.  Restore Rules Book
I.  Verify Rules Book            N.  Reinitialize running Broker
S.  Startup  Request Broker      D.  Shutdown Request Broker

Choose an item or type q to quit :
```

16. Enter: **q**

    The system displays the following message:

```
End of installation.
```

17. Choose one of the following commands to configure and initiate the ODBC software:

    - If this is a new install, enter:

      /cms/dc/odbc/odbc_init

      The system displays the following message:

```
ODBC driver initialization complete
```

    - If this is an upgrade or reinstallation, enter:

      /cms/dc/odbc/odbc_init -r 1

      The system displays the following message:

```
oplrqb has been activated
```

18. To verify that the ODBC Request Broker is active on the server, enter:

    ps -ef | grep oplrqb

    One of the output lines should show the `oplrqb` process running from the **/usr/openlink/bin** directory, as shown in the following example:

```
root  1462  1459  0 14:41:38 ?
0:00 /usr/openlink/bin/oplrqb -f +configfile
/cms/dc/odbc/cmsrqb3.2_init +loglevel 5 +l
root  1475  1467  1 14:44:48 pts/4    0:00 grep oplrqb
```

19. Enter: **eject cdrom**

    At this point, the software is registered, installed, and running. If you do not see an *oplrqb* process running after completing Step 18, repeat the installation.

Server utilities

Introduction

Once the ODBC driver is installed on the server, a CMS utility program is available for setting the debug and log levels, and running and deactivating the ODBC feature by using the run feature. This utility program is located in the `/cms/dc/odbc` directory.

Contents

"Server utilities" contains the following procedures:

Debug levels

To set the server ODBC debug level:

1. Enter **`cd /cms/dc/odbc`** to change directories to **/cms/dc/odbc**.
2. Enter **`./odbc_init -d x`**

   Where ***x*** is either **`0`** or **`1`**

   - **To turn the debug utility off, enter: `0`**
   - To turn the debug utility on, enter: **`1`**

Log levels

To set the CMS server log level for ODBC:

1. Enter **`cd /cms/dc/odbc`** to change directories to **/cms/dc/odbc**.
2. Enter: **`./odbc_init -l 0-7`**

   Where **`0`** turns CMS server logging off, and **`1-7`** sets the log level, with **`1`** as the lowest level and **`7`** as the highest (**`5`** is the default and recommended setting).

   Note:

   It is recommended that users should not set the log level greater than 5, which is the default setting.

Run ODBC

To run or deactivate the ODBC feature on the server:

1. Enter **`cd /cms/dc/odbc`** to change directories to **/cms/dc/odbc**.
2. Enter **`./odbc_init -x`**

   Where ***x*** is either **`0`** or **`1`**

   - To turn the ODBC feature off, enter: **`0`**
   - To turn the ODBC feature on, enter: **`1`**

Solaris client installation

Overview

The OpenLink ODBC driver software can be installed on a Solaris client using the installation procedure in this section. Once the software is installed, it must also be configured using the procedure in this section. For additional information on installing and configuring the ODBC driver, see the client on-line help file that is included on the Call Management System OpenLink ODBC Driver compact disk.

System requirements

Before installing the OpenLink ODBC driver software on a Solaris client, verify that you have met the following list of requirements.

- *The client is communicating with the CMS server over the network.* Use the network protocol's *ping* utility to ensure that communication between the client and the CMS server is functional. Do not proceed if basic communications between the client and the server cannot be established. If the client cannot recognize the network, the ODBC driver will not function properly.
- The ODBC clients are appropriate Sun computers with the Solaris 2.4 (or later) operating systems installed.
- To develop an application using the ODBC API, the complete Solaris distribution (which provides the libraries and headers) and the SPARCCompiler™ must be installed. The development environment on the Sun client must be a Solaris 2.4 (or later) operating system.

Contents

"Solaris client installation" contains the following procedures:

Installation procedure

Important:

If the client is on the same machine as the server, the client software was installed with the server software. If this is your situation, go to Configuration procedure on page 42.

If you are installing a Solaris client on a machine other than the CMS server, continue with the following installation procedure.

To install the OpenLink ODBC software on the Solaris client.

1. Insert the “Avaya CMS OPENLINK ODBC Driver” compact disk into the CD-ROM drive.
2. Log in as **root**.
3. Enter:

   cd /usr
4. Enter:

   mkdir /usr/openlink

   The system creates the **/usr/openlink** directory
5. If an error occurs indicating that the **/usr/openlink** directory already exists, enter:

   rm /usr/openlink/*taz
6. Enter:

   cd openlink
7. Depending on the version of Solaris installed on the system, choose one of the following commands to copy the client components to **/usr/openlink**:

   - If Solaris version 2.4 is installed, enter:

     cp /cdrom/cdrom0/client/solaris2.4/* .
   - If Solaris version 2.5 (or later) is installed, enter:

     cp /cdrom/cdrom0/client/solaris2.5/* .
8. Enter **./install.sh** to run the Solaris client installation script.
9. At the prompt, enter **root** as the name of the user who will own the programs.
10. At the prompt, enter **root** as the name of the group that will own the programs.

The libraries and other files needed for ODBC application development are now installed. A test application to verify client/server connectivity is included. See Test ODBC connectivity on page 43 for more information.

Configuration procedure

The **cms_odbc.ini** file located in **/usr/openlink** must be modified and placed in the **$HOME** directory as **.odbc.ini** for each user that will initiate the client application.

CAUTION:

To correctly resolve database connectivity, the **.odbc.ini** file must exist in the **$HOME** directory of the initiating user on the ODBC client. The client application will fail if it is initiated by a user who does not have an **.odbc.ini** file.

1. First edit the **cms_odbc.ini** file, then copy it to **$HOME/.odbc.ini**. The **cms_odbc.ini** file resides in **/usr/openlink**.
2. The host parameter in the Informix connection segment must be changed.
3. The ServerType field should be set to the version of Informix that CMS is using:
   - **`Informix5.x`** for R3V5 CMS
   - **`Informix7.1`** for R3V5u CMS
   - **`Informix7.2`** for R3V6 and R3V8 CMS
   - **`Informix7.3`** for R3V9 CMS
   - **`Informix7.3`** for R3V11 CMS

   The modified Informix section should look similar to the following example:

```
[Informix7]
Driver = /usr/openlink/client/odbcsdk/lib/oplodbc.so
Host   = YOUR DEFAULT HOST NAME
ServerType= Informix7.2
ServerOptions=
Database= /cms/db/inf/cms
Options=
ReadOnly=
FetchBufferSize= 60
```

Note:

If the system is running R3V9 CMS, the **Database** field will be **`/cms`**

4. Enter **`cd /usr/openlink`** to change to the OpenLink directory.
5. Edit the host parameter in the Informix section of the **cms_odbc.ini** file (as described on the previous page).
6. Copy the **cms_odbc.ini** file to **$HOME/.odbc.ini**.
7. Enter **`cd $HOME`** to change to your home directory.
8. Enter **`vi .profile`** to edit your **.profile** file.
9. Save and exit the file.

10. Enter the following command on a single line at the command prompt:

    LD_LIBRARY_PATH =

    /usr/openlink/odbcsdk/lib:$LD_LIBRARY_PATH

11. **Enter: `export LD_LIBRARY_PATH`**
12. Enter: **`UDBCINI = $HOME/.odbc.ini`**
13. Enter: **`export UDBCINI`**
14. Press **ESC** and enter **`.wq!`** to exit your **.profile** edit.
15. Enter **`. .profile`** to source your **`.profile`**.

Once the Solaris client software is installed and configured, you can proceed with writing your application using the ODBC API.

Test ODBC connectivity

On the Solaris client, initiate the test application to verify connectivity with the following procedure:

1. Enter **`cd /usr/openlink/odbcsdk/examples`** to change directories to **/usr/openlink/odbcsdk/examples**.
2. Enter: **`./odbctest`**

   The system displays the following message:

   ```
   Enter ODBC connect string (? shows list):
   On a single line at the ODBC connect string prompt,
   enter with no spaces:
   DSN = InformixX;(the valid value for your CMS)
   UID = Valid _UNIX_User_on_Server;
   PWD = User_password
   ```

   Where `DSN` = data source name, `UID` = user ID, and `PWD` = password.

   An example would be:

   `DSN=Informix7;UID=cmssvc;PWD=cmspasswd`

3. At *SQL >*, enter a valid SQL query.

   To test data connection, for example:
   `select SPLIT, ACDCALLS from hsplit where ROW_DATE = “6/27/98” and STARTTIME = 1600`

4. Enter: **`exit`**

If an error occurs, an error message is reported and the test application closes.

Possible failure causes

A test application error could result from one of the following causes:

- The driver is not active on the CMS server. Verify that the ODBC driver is active on the server by entering
  `ps -ef | grep oplrqb`
  You should see an **`oplrqb`** process running.
- There was an error in the ODBC.connect string entry. Resolve this by re-entering the correct DSN, host, user ID and password at the ODBC.connect string prompt.
- The **.odbc.ini** file does not reside in the initiator's **$HOME** directory.
- The **.odbc.ini** file is incorrect. Verify that the entered host is correct and the database is **/cms/db/inf/cms**.
- The LD_LIBRARY_PATH is not set. Enter
  `echo $LD_LIBRARY_PATH`
  to ensure that LD_LIBRARY_PATH is included in the display. See Test ODBC connectivity on page 43 for more information.
- The UDBCINI is not set. Enter
  `echo $UDBCINI`
  to ensure that **$HOME/.odbc.ini** is included in the display. See Configuration procedure on page 42 for more information.
- The data source is not consistent with the **cmsrqb_init** entry. Verify the data source you have entered (for example, **`Informix7`**).
- The user name is invalid. OpenLink will return the following error message if the user name is invalid: "OpenLink UDBC; user name/password invalid." When the user logs in, the user name is verified from the **/etc/passwd** and **/etc/shadow** files. In addition, if an **/etc/shells** file exists, the user's shell is compared against the valid shells already entered in **/etc/shells**. If the user name is invalid, do the following steps:

  a. Verify the name of the shell they used to log in by going into **/etc/passwd** and searching for the user name. At the end of the appropriate user name line, the shell with which the user logged in will be listed.

  b. If the **/etc/shells** file exits, go into it and add an entry for the user shell if the user shell does not already exist.

  c. If the user shell already exists in the **/usr/shells** file or the **/etc/shells** file does not exist at all, refer the problem to the system administrator.

Review the **/usr/openlink/client/doc/STARTUP.DOC** file for other possible errors.

Build an ODBC application

Provided that the development environment is installed, all necessary components to build an ODBC application exist in the **/usr/openlink/client/odbcsdk** directory. Necessary headers and library routines reside in the respective **/include** and **/lib** directories. The actual application test code and associated makefile reside in **/usr/openlink/client/odbcsdk/examples**. This code illustrates the use of the ODBC API. Copy the code to another directory prior to your use and development.

The ODBC application is responsible for implementing the embedded ODBC code provided by the ODBC libraries linked to the driver manager for host/database connectivity, data access, and any other necessary processing. If the test connection is successful (Test ODBC connectivity on page 43 in this section), the client has been properly configured and the ODBC header and libraries have been verified to be fully accessible to an application. Avaya cannot provide support for client-generated applications, or other third-party software or related mapping. See the caution in this chapter's overview, or contact your Avaya account representative.

Desktop computer client installation

Overview

The OpenLink ODBC Request Agent software can be installed on your desktop computer or on your network for access by each client using the procedures in this section. Once the software is installed, it must also be configured using the procedure in this section. For additional information on installing and configuring the ODBC driver, see the on-line help file that is included on the Call Management System OpenLink ODBC Driver compact disk.

System requirements

Before installing the OpenLink ODBC driver software on your computer, verify that you have met the following list of requirements.

- *Client network software is installed:* Ensure that you have a “winsock”-compliant desktop TCP/IP product installed. Check for the existence of the *winsock.dll* file with the Windows Explorer function or through your file manager.
- *Your computer is communicating with the CMS server over the network:* Use your desktop TCP/IP product’s Packet Internet Groper (PING) utility to ensure that communication between your computer and the CMS server is functional. Do not proceed if basic communications between your computer and the server cannot be established.
- You are running Microsoft Windows 95, Microsoft Windows 98, Windows 2000, Windows NT 3.51, or Windows NT 4.0.

Contents

“Desktop computer client installation” contains the following procedures:

Installation on Windows 95, 98, 2000, XP or Windows NT 4.0

To install OpenLink ODBC client software on a Windows 95, 98, 2000, XP or Windows NT 4.0 system:

1. Start Microsoft Windows.
2. Insert the “Avaya CMS OPENLINK ODBC Driver” compact disk into the compact disk drive.
3. Open the Windows Explorer.
4. Select your compact disk drive.
5. Select the **Client** folder.
6. Select the **Win32** folder.
7. Double click on the **setup.exe** file to install the client software.
8. Follow the instructions on your screen.

   It is recommended that you accept the **Default** installation configuration.
9. When the OpenLink Data Access Drivers window is displayed select: **ODBC client has ODBC samples (0.3Mb)**
10. Configure your ODBC data sources using the ODBC Administrator utility resident within your Windows Control Panel and the procedures in this section.

Installation on Windows NT 3.51

To install OpenLink ODBC client software on a Windows NT 3.51 system:

1. Start Microsoft Windows.
2. Insert the “Avaya CMS OPENLINK ODBC Driver” CD-ROM into the CD drive.
3. From the Windows Program Manager, choose **File** (Alt+F). Then choose the **Run** command.
4. Type the compact disk drive letter, followed by **:\client\Win32\setup**. For example, if your compact disk drive is drive *d*, type **d:\client\Win32\setup**.
5. Follow the instructions on your screen. It is recommended that you use the **Normal** installation option.
6. Configure your ODBC data sources using the ODBC Administrator utility resident within your Windows Control Panel and the procedures in this section.

Install clients from a single network point

To install ODBC on a desktop computer client that does not have a compact disk drive, or to install ODBC from a single network point, complete the following steps:

1. Install the ODBC driver on a networked machine with a compact disk drive, using the procedure for the machine's operating system. See Installation on Windows 95, 98, 2000, XP or Windows NT 4.0 on page 47 or Installation on Windows NT 3.51 on page 47 for more information.
2. Open your computer's program manager or Windows Explorer and go to the directory for the networked machine.
3. **Copy** the OpenLink folder from the networked machine.
4. Change directories to your client computer's hard drive.
5. **Paste** the OpenLink folder to your hard drive.
6. Run the setup.exe file using the **Normal** option and configure your ODBC data sources using the procedures in this section.

Desktop computer client configuration

Overview

The procedures on the following pages use the OpenLink ODBC Driver Administration facility to administer data sources for the client(s).

Contents

“Desktop computer client configuration” contains the following procedures:

ODBC driver administration utility

The ODBC Driver Administration utility resides within your desktop environment’s control panel. This utility is responsible for adding and removing ODBC drivers.

Accessing the ODBC Driver Administration Utility

To access the ODBC Driver Administration utility after the OpenLink ODBC driver software has been installed, choose one of the following procedures depending upon your version of Windows.

For Windows NT 3.51

To access the ODBC Driver Administration utility in Windows NT 3.51:

1. Select **Main** from the **Program Manager**.
2. Select **Control Panel**.
3. Select **32 bit ODBC**.

For other Windows versions

To access the ODBC Driver Administration utility in Windows 95, Windows 98, Windows 2000, or Windows NT 4.0:

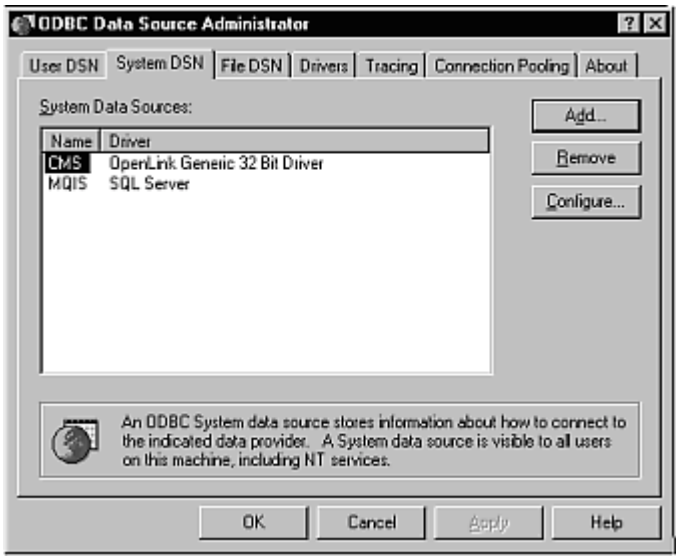
1. Select the **Start** label in the taskbar.
2. Select **Settings**.
3. Select **Control Panel**.
4. Select **ODBC Data Sources**.

ODBC data source dialog box

The ODBC Data Sources administrator dialog box displays a list of, available ODBC data sources. To access the CMS server, a data source must be administered, after it has been added and configured. See Define data sources on page 51 for more information. The ODBC Data Source administrator dialog box allow you to select an ODBC driver for which you want to add a data source. In this case, you will add the CMS (OpenLink Generic 32 bit*)* driver.

Data source administrator dialog box — 3.2 ODBC driver

The following figure shows an OpenLink Data Source Administrator dialog box with a version 3.2 ODBC driver.



Define data sources

The defined data sources for all of the currently installed drivers are listed in the User Data Sources (Driver) list on the Data Sources dialog box. You may define one or more data sources for each installed driver. The data source name should provide a unique description of the data; for example: CMS.

ODBC data source fields

The following table defines the version 3.*X* ODBC Data Source Administrator dialog box fields.

Field	Description
Close	Closes the dialog box and exits the ODBC Control Panel.
Help	Accesses the ODBC on-line Help file.
Configure	Allows you to configure an existing data source. You should select the name of the data source you want to configure before selecting the **Setup** button.
Remove	Removes an existing data source. You should select the name of the data source you want to delete from the list before selecting the **Delete** button.
Add	Adds a new data source. If you select this button, a dialog box prompts you for the driver for which you are adding a data source. After you select a driver, a driver-specific setup dialog box is displayed. If you select this button, a dialog box prompts you for the information required to set up a login ID-based data source. This is not recommended for CMS.
Drivers Tab	Displays information about an ODBC driver. If you select this button, a dialog box displays a list of the types of drivers currently installed on your computer.

Field	Description
System DSN Tab	Allows you to add, delete, or configure data sources local to a computer, rather than dedicated to a user. If you select this button, a dialog box prompts you for the information required to set up the system data source. This is recommended for CMS.
Tracing Tab	Allows you to set ODBC options. If you select this button, a dialog box prompts you, asking if the ODBC calls should be traced; and, if so, the name of the trace file.
User DSN Tab	Allows you to add, delete, or configure data sources that are dedicated to a specific user login ID. If you select this tab, the User DSN dialog box prompts you for the information required to set up a login ID-specific data source. This is not recommended for CMS, but is presented as an option in this document.
File DSN Tab	Allows you to add, delete, or configure file-based data sources in a shared folder that is accessible by every client on the network. If you select this tab, the File DSN dialog box and accompanying wizard prompts you for the information required to set up a login ID-specific data source. This is not recommended for CMS and is not described in this document; however, you may choose this option if you have a large number of networked *Windows NT* 4.0 clients.
About	Displays a dialog box with detailed information about the ODBC drivers available on your system.
Connection Pooling - version 3.2 only	Allows an application to reuse open connection handles, which save round trips to the servers.

Add system data sources

Before adding a data source, you should decide if you want to add a system-wide data source or a data source specific to a user login ID. Administering data sources on a per-user login ID basis is an optional procedure. Use the following procedure to administer a system data source. See Add login ID-based data sources on page 53 to administer data sources on a per-login ID basis.

Note:

It is recommended that you add system-wide data sources for CMS, rather than having to administer data sources on a per-login ID basis.

Procedure

To add a system data source, complete the following steps:

1. On the ODBC Data Sources administrator dialog box, select the **System DSN...** tab.

   The system displays the System Data Sources dialog box.

2. Select the **Add...** button.

   The system displays the create New Data Source Wizard.

3. On the Create New Data Source dialog box, select:

   OpenLink Generic 32 bit driver

4. To select the driver, press the **OK** button.

   Once you have selected the OpenLink ODBC driver, the OpenLink Setup dialog box displays. See Configure a new ODBC data source on page 54 for data source setup procedures.

Add login ID-based data sources

As an option, you may administer data sources on a per-user login ID basis. This procedure is useful if you are providing access for a specific user; however, it is strongly recommended that you administer system data sources.

To administer login ID-based data sources, complete the following steps:

1. On the ODBC Data Source Administrator dialog box, select the **User DSN... tab**.
2. On the ODBC Data Source Administrator dialog box, select the **Add...** button.

   The system displays the Add Data Source dialog box.

3. Select the generic ODBC driver:

   OpenLink Generic 32 bit driver

4. Press the **Finish** or **OK** button.

Once you have selected the OpenLink ODBC driver, the OpenLink Setup dialog box displays. See Configure a new ODBC data source on page 54 for data source setup procedures.

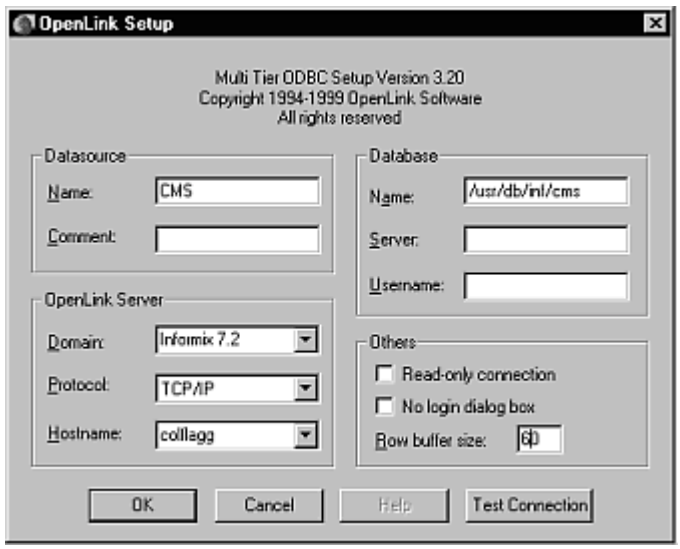
Remove a data source

To remove any data source, perform the following:

1. Select the appropriate ODBC Data Source
2. Press the **Delete** or **Remove** button on the appropriate ODBC Data Source dialog box and follow the prompts.

Configure a new ODBC data source

The CMS data source is now associated with the OpenLink Generic 32 bit driver. Specific server and database access must now be configured through the generic OpenLink Setup dialog box, shown in the following figure.



Procedure

To configure your ODBC driver software to access CMS data, complete the following steps:

1. In the **Datasource: Name** field, enter a descriptive name for your data source. In a multi-site call center situation, you can use this field to differentiate between call center locations.
2. In the **Datasource: Comment** field, you may optionally enter “Call Management System” (or other information related to the CMS data source).
3. In the **Provider: Type** field, enter one of the following or select from the pull-down list:
   - **`Informix5.x`** for R3V5 CMS
   - **`Informix7.1`** for R3V5u CMS
   - **`Informix7.2`** for R3V6 and R3V8 CMS
   - **`Informix7.3`** for R3V9 CMS
   - **`Informix7.3`** for R3V11 CMS
4. In the **Provider: Protocol** field, enter, or select from the pull-down list:

   TCP/IP
5. In the **Provider: Hostname** field, enter the name of your database host machine (example: CMS server).
6. Perform one of the following actions in the **Database: Path** field, depending on the version of CMS:
   - If you are running CMS R3V8 or earlier, enter:

     /cms/db/inf/cms
   - If you ar e running CMS R3V9, enter:

     /cms
7. Leave the **Database: Options** field blank.
8. In the **Database: User ID** field, you may optionally enter a valid CMS user login ID. Entering a CMS user login ID displays a default user name for each login to the data source.

9. Make sure the **following items in the Others** section are *not* selected:
   - **Read-only connection -** If the check box selected complete access ODBC to customer-defined database tables (**c_*** tables) will be denied.
   - **No login dialog box -** If the check box selected no login dialog box will display when you connect to the CMS database.

   Note:

   You may select this check box to prevent a login dialog box from displaying with each database access; however, this is recommended only if the software through which you are accessing the database, such as Crystal Reports, is password-enabled.
10. In the **Others: Row Buffer Size** field, enter a row buffer size of **60**.
11. Select the **OK** button, or select the **Cancel** button to cancel the configuration.

At this point, the ODBC driver software is installed on your computer. For further configuration options, continue with Administer ODBC options on page 59.

Once the driver is configured, the OpenLink ODBC driver is accessible to ODBC-enabled applications on your computer. Any queries that you send to the CMS database from client Windows applications, such as Microsoft Access, will use the ODBC feature to access data and copy it to your applications.

Note:

You will need to format the data within your application. The data returned from your SQL queries will be formatted in the manner described in Chapter 3, Database tables

OpenLink setup input fields

The following table describes the input fields on the OpenLink Setup dialog box.

Field	Description
Datasource	
Name	A representative name for the server/database you are connecting to, such as CMS.
Comment	A description of the data source you are connecting to, such as CMS.

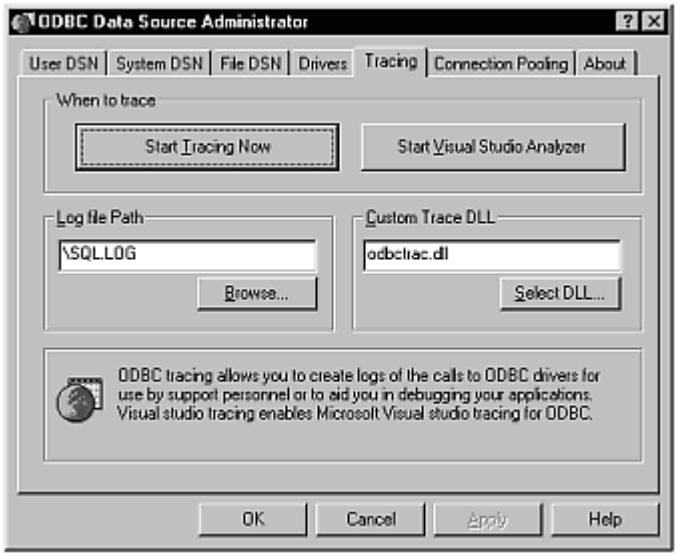
Field	Description
Provider	
Domain	The database type or user-defined logical representation of the underlying DBMS you are establishing an ODBC session with. The OpenLink ODBC driver allows the creation of custom providers, which enables you to mask the identity of the underlying database to which connections are made. This provides ODBC-based client/server infrastructure management flexibility.
Protocol	The network protocol you are using to access database data items; the default is TCP/IP.
Hostname	The network alias (or IP address) of the machine on your network hosting the database(s) to which you want to make ODBC connections.
Database	
Name	The field used to map the ODBC data source names to specific database names for a provider type. For example, the CMS Informix database has a path field value of **/cms/db/inf/cms**, identifying the CMS database to be associated with an ODBC data source name.
Server	An optional entry field used for database connections after the database environment has been initialized. These settings generally take the form of operating environment variables which cannot be entered in this field. An entry is not required or recommended in this field.
User Name	An optional entry field that contains a valid CMS user login ID for the database being represented by the provider type.

Field	Description
Others	
Read-only connection	A check box that allows you to choose if you do or do not want to enforce read-only database sessions for ODBC-compliant applications. This box should not be checked if you are accessing customer data, and should be checked if you are not accessing customer data.
No login dialog box	A check box that allows you to choose if you do or do not want to disable the pop-up login dialog box during ODBC session establishment with a selected ODBC data source.
Row buffer size	A variable-entry field that determines at ODBC run time the size of the OpenLink network array fetch buffer (the number of records transported from an OpenLink database agent to and ODBC-compliant application environment during a single network hop). This setting affects the performance of the OpenLink driver. The recommended setting for the CMS database is 60.

Administer ODBC options

You may specify how the ODBC driver traces ODBC function calls. If tracing is activated, a file is generated that contains the actual ODBC function calls. To set these ODBC options, complete the following steps:

1. In the ODBC Data Source Administrator dialog box, select the **Tracing** tab.



2. To trace ODBC calls or observe ODBC activity, select the **Start Tracing Now** button.
3. To stop tracing ODBC function calls automatically, select the **Stop Tracing Now** button. This will terminate the ODBC tracing upon completion of the ODBC session.
4. To select or change the file to which the OpenLink Request Broker writes tracing information, change the file name and path (or use the **Browse...** button). The default trace file is *\SQL.LOG*.

   The system displays the Select ODBC Trace File dialog box.
5. Select the file to which you want the OpenLink Request Broker to write tracing information.

View installed ODBC drivers

To view a list of installed ODBC drivers, complete the following steps. This is useful for verifying if the ODBC driver has been properly installed.

1. On the ODBC Data Source dialog box, select the **Drivers...** button or tab.

   The system displays the Drivers dialog box.

   Note:

   Use the Drivers dialog box to verify installation of the OpenLink Generic 32 bit driver. If the OpenLink Generic 32 bit driver does not display on the Drivers dialog box, return to Desktop computer client installation on page 46 and reinstall the drivers.

2. To view detailed information about an installed driver, select the driver from the list and select the **About** button or tab.

   The system displays the About dialog box.

3. When you have finished viewing information about the driver, select the **OK** button to exit the About dialog box.

4. Select the **Close** button or **OK** button to exit the Drivers dialog box.

   The system displays the ODBC Data Source dialog box.

   Note:

   The ODBC Driver Administration utility will close when you exit directly from any dialog box. You will need to reopen the ODBC Data Source dialog box using the procedure in ODBC driver administration utility on page 49.

Test desktop computer connectivity

Overview

Once you have installed the OpenLink ODBC driver software on both the client and the server, you may open a demonstration connection to a data source on the server to show connectivity and test SQL access. To connect to an administered data source, use the following procedure.

Contents

“Test desktop computer connectivity” contains the following procedures:

Connect to and access data

To connect to a data source from the client, complete the following steps:

1. From the **Start** button select Programs, **OpenLink Data Access Drivers** and **C++ Demo 32 bit** on your computer.

   The system displays the ODBC SDK 2.0 C++ Demo window.

2. From the **Environment** menu, select **Open Connection**.

   The system displays the SQL Data Sources dialog box.

   Note:

   Depending on the ODBC driver version you are using, the Select Data Source dialog box may display and you will be prompted to select a file or machine data source.

3. Select the **Machine Data Source** tab and select the CMS data source.

4. Select the data source that you want to use from the list.

   The system displays the OpenLink ODBC login dialog box.

   Note:

   This dialog box only displays if you did not check the **No Login Dialog Box** check box on the Setup dialog box.

   You should have already configured the CMS data source. If you do not see a data source on the SQL Data Sources dialog box, see Desktop computer client configuration on page 49.

5. Enter your CMS server user name and password (CMS server login).
6. Select **OK**.

   The system displays the ODBC SDK 2.0 C++ Demo screen.
7. Select **SQL** to begin writing your SQL query.
8. Select **Execute SQL**.

   The system displays the ODBC SDK 2.0 C++ Demo SQL dialog box.
9. Enter the following SQL query in the text box: **select count (*) from hsplit**.

   The system displays the queried data in the ODBC SDK 2.0 C++ Demo window. Any valid SQL query may be entered.

   You should see a count column on the ODBC SDK 2.0 C++ Demo window. The value in this column is the number of columns in the CMS `hsplit` table. This result confirms that you have successfully accessed the database from the client. Use this test as a troubleshooting tool in the future to verify connectivity and data access from the client.

Disconnect from a data source

Once you have completed test querying the database, you can disconnect from the data source through the ODBC driver software. To disconnect, complete the following steps:

1. From the **Environment** menu, select **Close Connection**.

   The system displays the ODBC SDK 2.0 C++ Demo Close Current Connection dialog box.
2. Select **OK** to close the connection, or **Cancel** to remain connected to the data source.

Troubleshooting

Overview

General troubleshooting procedures and error messages on the OpenLink ODBC driver are included in this section. For more detailed information, see the on-line help file that is included on the “Avaya CMS OPENLINK ODBC Driver” CD-ROM.

Network support

Since Avaya does not control customer network configuration or ODBC-enabled client applications, both installation and ongoing maintenance support is limited to determining if data is being transferred correctly in the most basic client/server relationship. This is defined as a CMS running ODBC on the same network hub as the client PC.

Verify that the trouble occurs on the same network subnet, then continue with troubleshooting procedures. If the trouble does not occur on the on the same network subnet, contact the Avaya help line.

Contents

“Troubleshooting” contains the following procedures:

Server log file

Once the ODBC driver is installed and initiated, the server log file, **odbc.log**, records the logging levels of all ODBC activities. The default log level is **5**. It is recommended that users should not set the log level greater than 5, which is the default setting. See Server utilities on page 38 for more information.

Review the **odbc.log** file for information about an ODBC sessions. Archives of past ODBC sessions are maintained in the **odbc.log.01** and **odbc.log.02** files. These logging levels are set by running **`./odbc_init -l`** located in the **/cms/dc/odbc** directory. See Log levels on page 38 for more information.

Client trace

The OpenLink client component configuration utility, located in the OpenLink group on your Windows desktop, allows you to enable or disable ODBC trace logging. Trace logging provides you with output on your entire ODBC session, including all ODBC calls made by the ODBC-compliant application you are using.

See Administer ODBC options on page 59 for complete information on this utility. This file also includes native database error messages that might not have been replaced by the ODBC-compliant application you were using.

OpenLink error messages

The error messages that you may receive from the OpenLink ODBC driver are detailed in the following list. These error messages are displayed on the client and in the server log file.

Invalid Username/Password	This error message is displayed when the operating system-level username and password verification is in use, and when you enter an invalid operating system-level username and password combination when connecting to your OpenLink ODBC Data Source, even though the username and password combination entered is valid at the database level.
Unable to Locate Requested Service	This error message is displayed as a result of the OpenLink Session Rules Book being incorrectly configured. It typically takes the form of an invalid reference to the OpenLink database agent executable program responsible for providing database access. This error will not occur with the default cmsrqb_init settings.
Unable to Load OpenLink Request Agent	This error message indicates a client machine problem at the network transport level, indicating the need to verify that your client machine's network software is correctly configured.
Remote Procedure Call (RPC) Unable to Send	This indicates a corruption of the communications channel being used by the OpenLink Request Agent. This error typically occurs when the server Request Broker has been shut down. Verify the status of the Request Broker, then try to reinitiate the ODBC session.

RPC Timed Out	This error message is displayed when timeout settings in either the client side or server side Session Rules Books have been exceeded. This message typically occurs when communication cannot be established from client to server or server to client. Verify that the *oplrqb* is running on the server side by entering **`ps -ef | grep oplrqb`**. You should see an oplrqb process running.
RPC Unable to Receive	This message indicates that the server Request Broker is no longer communicating with the client. This error occurs when the server Request Broker has been shut down or reinitiated during a session. Verify that the Request Broker is active on the server.
RPC Host Unknown	This message results from a network failure or invalid host entry in the Data Source Setup dialog box. Check the Hostname field entry in that window; if you suspect a network problem, verify this and correct it if necessary; or retry your ODBC session.
Unknown Database Agent Requested	This problem results when the client administration associated with the ODBC session does not resolve to the oplrqb rules in cmsrqb_init. To resolve this error, the Data Source Setup dialog box Type field should be set to Informix 7. Retry your ODBC session.
Unable to Start the Requested Database Agent	The oplrqb was able to resolve to a database agent, but was unable to execute the program. To resolve this, the Type field on the Data Source Setup dialog box should be set to Informix 7. Alternately, the generic_inf7 should be set to inf7_sv. Finally, the inf7_sv should exist in **/usr/openlink/bin/** as executable.
Database Errors	Database errors are displayed after failed database accesses. To resolve this, correct the database query and resubmit it.

Chapter 3: Database tables

Overview

This chapter defines the historical database items and tables populated by Avaya CMS that can be accessed through ODBC.

Contents

"Database tables" includes the following:

How database items are presented

Database item tables

The database items are presented in a table format, according to ACD entity (split/skill, agent, vector, VDN, trunk, trunk group, exception, synonym, call work code, and forecast).

Sample database item table

The following table is an example of how the table information is presented:

Database Item	Description	Data Type	Column Type	Length
DATABASE ITEM	The definition of the database item is given here. Any additional information, such as other database items that are included in the sum of the database item, or the specific ECS release to which the database item applies, is also listed.	C, A, I, N, M, or B	integer, smallint, date, smallfloat, or char(n)	length in bytes

Many database items are contained in more than one database table. When an item is in more than one table, the definition is probably not the same from table to table. Database items that are used in the description of another database item are in boldface type. Where applicable, Enterprise Communications Server (ECS) information has been included in the database item descriptions.

Note:

If you are using a G2.2 ECS without the Call Center Features (CCF = N), you should refer to information for the G2.1 ECS for functionality.

Index database items

The index database items in each table are marked. Indexes add structure to table rows so that CMS can retrieve data faster. The row search criteria you define for custom reports should be based on indexes whenever possible. For historical custom reports, always include a “where” clause based on the **ROW_DATE** database item.

Data types

Each database item contains one of the following types of data:

A =Administrative data: administered on the ECS or on CMS. For example, the database item INTRVL in the split/skill table contains the number of minutes in the intrahour interval (15, 30, 60) assigned for the specified ACD on CMS.

B = Busy Hour data: gives data that is only meaningful for the busy hour.

C =Cumulative data: accumulates throughout the collection interval. Most real-time database items contain cumulative data.

I = Row Identifier data: gives data that is common to all tables, such as time, date, split in the split/skill tables, and so on.

M = Maximum Interval Value data: gives data that is the maximum reached for any value in the specified interval.

N = Special Table data: belongs only to a specific table, such as the Historical Agent Login/Logout table or Current Day Forecast table.

S = Status data: shows the current status (a snapshot of a particular ACD element). For example, the database item INQUEUE in the split/skill real-time table contains the number of split/skill calls currently waiting in queue.

The letter A, B, C, I, M, N, or S is in the Data Type column for each database item.

Cumulative, **Administrative, Maximum Value, Row Identifier and Busy Hour** data items apply to historical and real-time database items. **Special Table** data items apply only to historical database items.

Call-based data

In addition to the types of data described above, items in the CMS database can be either call-based or interval-based. Most CMS database items are call-based. Call-based data is committed to the database after a call completes. Therefore, if a call starts and ends in different collection intervals, all of the data are recorded in the interval in which the call and any After Call Work (ACW) are completed.

Interval-based data

Interval-based data represents the amount of time during a collection interval that is used for a particular activity. Interval-based items are updated throughout the collection interval and timing is restarted at the end of the interval. Most interval-based items start with **I**_ or **TI**_. The database items **ALLINUSETIME** (trunk-group tables) and **MBUSYTIME** (trunk and trunk-group tables) are also interval-based. Each database item has a defined column type.

Column types

The database column type indicates the format of the information within that database item. The column type definition includes either the length in bits of the database item or the Informix data type. The column types are included in the database item tables, along with the length or Informix data type. Column type and length information is included for the user's reference. Because data gathered through ODBC can be used in a variety of applications, it is helpful to know what type of data you are accessing and how long it is in bytes. Any difference or exception in the column type or length between tables in a table group; such as, between the `hsplit` and `msplit` tables in the split/skill table group, are indicated in the **Column Type** and **Length** columns.

Informix column types

The following table defines the data column types in the open database.

Column Type	Definition
char(*n*)	An ASCII string of *n* characters, 1 byte per character.
date	Informix date type, 4 bytes long. The Informix date format is yyyy-mm-dd. For example, May 19, 1998 would display as 1998-05-19.
integer	4 byte integer
smallint	2 byte integer
smallfloat	Informix floating point numerical type, 4 bytes long. The Informix smallfloat format is a decimal type used for percentages, and includes a comma and a plus/minus sign.

ECS cross-reference tables

CMS database items apply to specific ECS releases. Following each database item table is an ECS cross-reference table. The ECS cross-reference tables list each database item by ECS release.

Sample ECS cross-reference table

The following table is an example of how the table information is presented:

Note:

If you are using the G2.2 ECS without the Call Center Features (CCF = N), you should refer to the G2.1 column for functionality

Database Item	G3V2/ G3V3	G3V4	DEFINITY ECS R5	DEFINITY ECS R6	DEFINITY ECS R7/R8	DEFINITY ECS R9/R10	MultiVantage R11
ITEM NAME	X	X	X	X	X	X	X
	The letter X indicates the ECS releases that this database item applies to.						

Database logic structure

CMS historical tables store information in one record per row format. This formatting affects the way data can be accessed through ODBC. For example, in the agent tables, a record will be created for each split/skill that an agent is logged into. If an agent is logged into four splits/skills, there will be four records for that agent. Similarly, if an agent starts the day with four splits/skills, and is added to a fifth split/skill before the end of the day, the agent's fifth record will be generated only from the point at which the additional split/skill was added. The other four records will reflect the total logon time. When accessing data in the historical tables, you may need to sum the information to retrieve complete data.

Agent tables

If an agent logged off and logged on more than once in a specified interval, another complete set of records will be created for that agent for each logon in the agent tables. If an agent logs into four split/skills, logs out, and then logs back on during a set interval, there will be two sets of four records for that agent, one set per logon.

VDN tables

The VDN tables store one record per vector on which a VDN terminates. Therefore, if the terminating vector for a specific VDN changes in a set interval, there will be two records for that VDN—one per terminating vector. This logic also applies to the Vector, Trunk, Trunk Group, and Split/Skill tables. If information is required from these tables, a sum structured query language (SQL) query may be necessary to access complete data for each VDN, vector, trunk, and so on.

Circular structure

The exceptions, call record, and agent trace tables are circular files. These tables will continuously populate until the table capacity plus ten percent has been reached; at which point, the oldest ten percent of the records will be deleted. For example, if an agent trace table has a capacity of 100 rows, and the total rows populated equals 110, the oldest ten rows will automatically be deleted. Therefore, the data in that table will change continuously as the table is updated.

Terminology

For definitions of terms used in database item descriptions, see the *Avaya CMS Database Items and Calculations,* 585-780-702 document, and the Glossary in this document.

CMS feature interactions

- **Dictionary Names:** Clients can access CMS Dictionary names. The client must map the synonym to the report.
- **Permissions:** Applications which access Informix externally, such as database access scripts, may not work if the table permission script tries to access a table to which permission is denied.
- **Field Display:** The time and date data you receive from the database may not be formatted. Generally, times may be shown in seconds or in military format. You will need to review data for formatting when you import it into your software application. The data returned from your SQL queries will be formatted in the manner described in your database interface specifications. See your software's documentation for further information on formatting data.

Interactions with ECS features and tracking of ECS capabilities

Some of the ECS features and capabilities have an impact on CMS open database items. For more information on these features and capabilities and their relationship with the CMS open database, see *Avaya CMS Database Items and Calculations,* 585-780-702.

Database table names

To select data for external use, you must use the names listed in the following table in your queries. The following table contains all the tables that are ODBC-accessible in CMS and a brief description of the data in each. If a table is not listed here, it is not accessible through ODBC, but will still be accessible by **root**. All tables included in this chapter have read-only external user access. The database items are described later in this chapter.

Name	Data Stored
hsplit	Split/Skill data for each intrahour interval.
dsplit	Split/Skill data summarized by day.
wsplit	Split/Skill data summarized by week.
msplit	Split/Skill data summarized by month.
hagent	Agent data for each intrahour interval.
dagent	Agent data summarized by day.
wagent	Agent data summarized by week.
magent	Agent data summarized by month.
htkgrp	Trunk group data for each intrahour interval.
dtkgrp	Trunk group data summarized by day.
wtkgrp	Trunk group data summarized by week.
mtkgrp	Trunk group data summarized by month.
htrunk	Trunk data for intrahour interval.
dtrunk	Trunk data summarized by day.
wtrunk	Trunk data summarized by week.
mtrunk	Trunk data summarized by month.
hvector	Vector data for each intrahour interval.
dvector	Vector data summarized by day.
wvector	Vector data summarized by week.
mvector	Vector data summarized by month.
hvdn	VDN data for each intrahour interval.
dvdn	VDN data summarized by day.

Name	Data Stored
wvdn	VDN data summarized by week.
mvdn	VDN data summarized by month.
hcwc	Call work code data for each intrahour interval.
dcwc	Call work code data summarized by day.
wcwc	Call work code data summarized by week.
mcwc	Call work code data summarized by month.
call_rec	Call record data.
d_secs	The number of seconds in the daily data collection period.
m_secs	The number of seconds in the monthly data collection period.
w_secs	The number of seconds in the weekly data collection period.
arch_stat	The status of archiver executions.
customer_log	The customer error log data.
agroups	Agent group definitions.
synonyms	Dictionary synonyms.
acd_shifts	Shift times and maximum agents logged in for each shift.
dbitems	Dictionary standard and custom database items, constants, and calculations.
f_cday	Forecast current day configuration data by split/skill.
f_cdayrep	Current day forecast data by split/skill.
haglog	Agent login and logout information.
ag_actv	Agent activity trace data.
agex	Agent exceptions.
fullex	Disk full exceptions.
spex	Split/skill exceptions.
tgex	Trunk group exceptions.
vecex	Vector exceptions.
vdnex	VDN exceptions.
linkex	Link down exceptions.
mctex	Malicious call trace exceptions.

Split/Skill database items

Overview

The Split/Skill database item descriptions apply to historical items.

The **Data Type** column refers to **Cumulative (C)**, **Administrative (A)**, **Status (S)**, **Row Identifier (I), Busy Hour (B), Special Table (N),** or **Maximum Value (M)** data. Cumulative, Administrative, Row Identifier, Busy Hour, and Maximum Value items apply to both the current and previous interval real-time tables. Special Table items are historical, and apply only to the table in which they are stored.

Historical split/skill database items apply to the Intrahour Split/Skill (`hsplit`), Daily Split/Skill (`dsplit`), Weekly Split/Skill (`wsplit`), and Monthly Split/Skill (`msplit`) tables. All items listed in the following table are included in all four tables, unless otherwise noted in the Column Type column. Any differences in the data format between the four split/skill tables are also noted in the Column Type column. The historical indexes are **SPLIT** and **ROW_DATE**.

Contents

"Split/Skill database items" contains the following topics:

Split/Skill database item table

The following table describes the data items in the Split/Skill CMS database tables.

Database Item	Description	Data Type	Column Type	Length
ABNCALLS	The number of **CALLSOFFERED** that were abandoned while in queue or ringing at an agent position. For Generic 3 Version 1 switches with the Vectoring feature, this also includes calls that were queued to the split/skill and abandoned while listening to a forced disconnect announcement.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
ABNCALLS (contd)	**Note:** When a call abandons while queued to multiple splits/skills and abandons from queue, only the primary split/skill increments **ABNCALLS** (calls that are ringing an agent and then abandon peg as abandons for the split/skill they were ringing). **This also includes calls with talk times less than the phantom-abandoned call timer value, if it is set.** ABNCALLS = ABNCALLS1 + ABNCALLS2 + ABNCALLS3 + ABNCALLS4 + ABNCALLS5 + ABNCALLS6 + ABNCALLS7 + ABNCALLS8 + ABNCALLS9 + ABNCALLS10 **ABNCALLS** includes **ABNCALLS1-10**, **ABNRINGCALLS**, **O_ABNCALLS**, **PHANTOMABNS**, and **SLVLABNS**.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
ABNCALLS1-10	The number of **ABNCALLS** that were abandoned during the collection interval in each of the service level increments **PERIOD1** through **PERIOD9** (as defined on the ACD Administration: Call Profile window). **ABNCALLS10** counts calls that abandoned after **PERIOD9**. **Note:** If call profiles are not set, the data gets stored into the first interval (**ABNCALLS1**).	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
ABNRINGCALLS	The number of split/skill or direct agent **ABNCALLS** that abandoned while ringing at an agent position. Available for ring tracking with Generic 2 and Generic 3 switches.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
ABNTIME	The amount of time callers spent waiting in queue and ringing at an agent's voice terminal before abandoning the call. For phantom abandons, **ABNTIME** includes the time until the agent releases the call.	C	integer	4 bytes
ACCEPTABLE	The number of **ACDCALLS** answered by an agent within the predefined acceptable service level (**SERVICELEVEL**), as defined on the ACD Administration: Split/Skill Call Profile window.	C	integer	4 bytes
ACD (index)	The ACD number for which data was collected.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
ACDAUXOUTCALLS	The number of **AUXOUTCALLS** agents in the split/skill made with at least one split/skill ACD call for this split/skill on hold. For agents in multiple skills with multiple call handling (Generic 3 Version 3 switch and later), the call is recorded for the skill of the last ACD call the agent put on hold. **ACDAUXOUTCALLS** includes calls made to transfer or conference the ACD call. Available with Generic 2.2 and Generic 3 switches.	A	smallint	2 bytes
ACDCALLS	The number of **CALLSOFFERED** calls that were answered by an agent in the split/skill. ACDCALLS = ACDCALLS1 + ACDCALLS2 + ACDCALLS3 + ACDCALLS4 + ACDCALLS5 + ACDCALLS6 + ACDCALLS7 + ACDCALLS8 + ACDCALLS9 + ACDCALLS10. **ACDCALLS** includes **ACCEPTABLE**, **ACDCALLS1-10**, **BACKUPCALLS**, **CONFERENCE**, **HIGHCALLS**, **HOLDCALLS**, **LOWCALLS**, **MEDCALLS**, **O_ACDCALLS**, **TOPCALLS**, and **TRANSFERRED**.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
ACDCALLS1-10	The number of **ACDCALLS** during the collection interval that were answered in each of the service level increments **PERIOD1** through **PERIOD9** (as defined on the ACD Administration: Call Profile window). **ACDCALLS10** is the number of calls answered after the last increment **PERIOD9**. **Note:** If call profiles are not set, the data gets stored into the first interval (**ACDCALLS1**).	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
ACDCALLS_R1	Number of **CALLSOFFERED** calls that were answered by a reserve1 agent in the split/skill. Found only in CMS R3V9.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
ACDCALLS_R2	Number of **CALLSOFFERED** calls that were answered by a reserve2 agent in the split/skill. Found only in CMS R3V9.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
ACDTIME	The length of talk time of all **ACDCALLS. ACDTIME** includes **O_ACDTIME**, but does not include **HOLDTIME**.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
ACWINCALLS	The number of inbound extension calls received by agents while in **ACW** for split/skill ACD calls or in **ACW**.	C	integer	4 bytes
ACWINTIME	The length of talk time of all **ACWINCALLS**. **ACWINTIME** does not include hold time on Generic 2.2 and Generic 3 switches. It does include time spent on calls received while in ACW not associated with an ACD call.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
ACWOUTADJCALLS	The number of **ACWOUTCALLS** that were placed by an adjunct on behalf of an agent (keyboard-dialed). If such calls are placed to off-switch destinations, they are also counted as **ACWOUTOFFCALLS**. Available for outbound calls on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	C	integer	4 bytes
ACWOUTCALLS	The number of outbound extension calls made by agents or on behalf of agents while in **ACW**. This includes ACW for split/skill ACD calls and ACW not associated with a call. **ACWOUTCALLS** includes **ACWOUTADJCALLS** and **ACWOUTOFFCALLS.**	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
ACWOUTOFFCALLS	The number of **ACWOUTCALLS** that were made to a an off-switch destination—a destination outside the switch. If such calls are placed by an adjunct on behalf of an agent while in ACW, they are also counted as **ACWOUTADJCALLS**. Available for external calls with Generic 2.2 and Generic 3 switches.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
ACWOUTOFFTIME	The length of talk time of all **ACWOUTOFFCALLS** (does not include time on hold). **ACWOUTOFFTIME** includes **ACWOUTTIME**. Available for external calls with Generic 2.2 and Generic 3 switches.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
ACWOUTTIME	The length of talk time of all **ACWOUTCALLS**. **ACWOUTTIME** does not include hold time on Generic 2.2 and Generic 3 switches. It does include time spent on calls made while in ACW not associated with an ACD call and on **ACWOUTADJCALLS** and on **ACWOUTOFFCALLS**.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
ACWTIME	The duration of all after call work associated with **ACDCALLS**. **Note:** **ACWTIME** does not include time spent in ACW not associated with an ACD call (that is, the agent pressed the ACW button while not on an ACD call). However, both **ACWINTIME** and **ACWOUTTIME** do include time spent on calls made or received while in ACW not associated with an ACD call. Therefore, the sum of **ACWINTIME** and **ACWOUTTIME** may be greater than **ACWTIME**. **ACWTIME** includes **ACWINTIME**, **ACWOUTTIME**, and **O_ACWTIME**.	C	integer	4 bytes
ANSTIME	The amount of time spent by callers in queue or ringing before being answered by an agent.	C	integer	4 bytes
ASSISTS	The number of times the supervisor was called (supervisor assists) by agents on split/skill calls, direct agent ACD calls, or in call-related ACW for this split/skill.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
AUXINCALLS	The number of inbound extension calls received by agents while in AUX (auxiliary work), **AVAILABLE**; or, for Generic 2.2 and Generic 3 switches, with an ACD or AUXIN/AUXOUT call on hold. **AUXINCALLS** are recorded in the **SPLIT** that is **OLDEST_LOGON** for agents in multiple splits/skills.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
AUXINTIME	The length of talk time of all **AUXINCALLS** (does not include hold time on Generic 2.2 and Generic 3 switches).	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
AUXOUTADJCALLS	The number of **AUXOUTCALLS** that were placed by an adjunct on behalf of an agent (keyboard-dialed). If such calls are placed to off-switch destinations, they are also counted as **AUXOUTOFFCALLS**. Available for outbound calls on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
AUXOUTCALLS	The number of outbound extension calls made by agents while in **AUX** (auxiliary work), **AVAILABLE**; or, for Generic 2.2 and Generic 3 switches with an ACD or **AUXIN/AUXOUT** call on hold. **AUXOUTCALLS** are recorded for the **SPLIT** which is the **OLDEST_LOGON**, unless the agent made the call with an ACD call on hold. In this case, they are recorded for the split/skill of the ACD call. **AUXOUTCALLS** includes **ACDAUXOUTCALLS**, **AUXOUTADJCALLS**, and **AUXOUTOFFCALLS**.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
AUXOUTOFFCALLS	The number of **AUXOUTCALLS** that were made to a destination outside the switch. If such calls are placed by an adjunct on behalf of an agent, they are also counted as **AUXOUTADJCALLS**. Available for external calls with Generic 2.2 and Generic 3 switches.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
AUXOUTOFFTIME	The length of talk time of all **AUXOUTOFFCALLS** (does not include **AUXOUTOFFCALLS** spent on hold). **AUXOUTOFFTIME** is included in **AUXOUTTIME**. Available for external calls on Generic 2.2 and Generic 3 switches.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
AUXOUTTIME	The length of talk time of all **AUXOUTCALLS. AUXOUTTIME** does not include hold time on Generic 2.2 and Generic 3 switches. **AUXOUTTIME** includes **AUXOUTOFFTIME**.	C	integer	4 bytes
BACKUPCALLS	The number of ACDCALLS that were delivered to and answered by this split/skill by a vector command other than "queue to main" and the number of ACDCALLS that were delivered to a split/skill by a “queue to” vector command answered by an agent that has neither reserve1 or reserve2 skill levels assigned for that skill. This allows tracking of calls answered by agents with a reserve1 or reserve2 skill level assigned for a particular skill. This includes calls delivered by messaging split/skill, check backup, route to split/skill, and redirect on no answer vector routing. Calls that are redirected back to the split/skill from ringing by the redirect on no answer feature that are subsequently answered by an agent in the split/skill are also counted as backup calls. Available on Generic 3 switches with the Vectoring feature. **Note:** The Redirect on No Answer VDN routing feature is also available on the DEFINITY ECS.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
BUSYCALLS	The number of **CALLSOFFERED** calls that were given a busy signal by the switch. This happens when a “busy” vector command is executed while the call is queued to this split/skill (and this is the primary split/skill the call is queued to) or if a call queued to this split/skill forwards to another split/skill whose queue is full. On Generic 3 and later switches, a busy is given because a non-vector controlled split has a full queue, no queue and no available agents, or no agents that are staffed.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
BUSYTIME	The amount of time callers waited in queue until hearing a busy tone for all **BUSYCALLS**.	C	integer	4 bytes
CALLSOFFERED	The number of calls that queued to the split/skill and that completed during the interval. This does NOT include calls on the Generic 3 switch that could not queue to the split/skill because the queue was full or there was no queue. **CALLSOFFERED** = **ACDCALLS** + **ABNCALLS** + **BUSYCALLS** + **DISCCALLS** + **OUTFLOWCALLS** + **DEQUECALLS.** **CALLSOFFERED** includes **ABNCALLS**, **RINGCALLS**, **OTHERCALLS**, and **INFLOWCALLS.**	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
CONFERENCE	The number of **ACDCALLS** that were conferenced at least once. Available on Generic 2.2 and Generic 3 switches.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
DA_ACWINCALLS	The number of inbound extension calls agents answered while in after call work mode for direct agent ACD calls that were queued through this split/skill. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
DA_ACWINTIME	The amount of talk time of inbound extension calls agents answered while in the after call work mode for direct agent ACD calls queued through this split/skill. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer	4 bytes
DA_ACWOCALLS	The number of outbound extension calls agents made while in the after call work mode for direct agent call ACD calls queued through this split/skill. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
DA_ACWOTIME	The amount of talk time of outbound extension calls the agent made while in the after call work mode for a direct agent ACD call. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer	4 bytes
DEQUECALLS	The number of calls that queued to this split/skill as a nonprimary split/skill, but whose disposition was recorded in another split/skill (as answered, abandoned, outflowed, busy, or forced disconnect). Requires vectoring for multiple split/skill queueing on a Generic 2.2 switch with EAS or a Generic 3 switch.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
DEQUETIME	The amount of time **DEQUECALLS** waited in this split/skill queue before dequeuing. Requires vectoring for multiple split/skill queueing on a Generic 2.2 switch with EAS or a Generic 3 switch.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
DISCCALLS	For Generic 2.2 switches, this is the number of **CALLSOFFERED** that were disconnected by the switch via the “disconnect” vector command. For the Generic 3 Version 2 and newer switch releases, this also includes the number of **CALLSOFFERED** that were disconnected by the switch when the vector disconnect timer expired. For the vectoring feature on Generic 3 Version 1 switches, this is the number of **CALLSOFFERED** that were given a forced disconnect announcement, then were disconnected by the switch. The disconnect announcement is for a “disconnect” vector command.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
DISCTIME	The amount of time all **DISCCALLS** spent in this split’s/skill’s queue. For Generic 2.2 switches, this is the time until the trunk drops, in the case where the caller hangs up without listening to the entire announcement. For the Generic 3 Version 2 and newer switches (if the call is disconnected due to the expiration of the vector disconnect timer), this is the time until the call is disconnected by the switch. For the vectoring feature on Generic 2.1 or Generic 3 Version 1 switches, this is the time until the announcement ends and the caller is disconnected by the switch.	C	integer	4 bytes
EVENT1-9	The number of times each event (stroke count) feature button (feature button 1 to 9) was pressed by agents on split/skill or direct agent ACD calls or in after call work associated with an ACD call for this split/skill. Available with Generic 2 and Generic 3 switches.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
HIGHCALLS	The number of **ACDCALLS** with high priority that were answered by agents in this split/skill (for example, answered calls that were queued to the split/skill with high priority by a “queue to main” or “check backup” vector command). For Generic 3 switches with the Vectoring feature, this includes calls that were queued to a split/skill with priority using the “route to” or “messaging split/skill” vector commands, and calls that queued directly to a split/skill with priority. (Priority in these cases is determined by the class of restriction of the originator, which is an agent, an extension, a trunk group or a VDN.) Available on Generic 2 and Generic 3 switches with the Vectoring feature.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
HOLDABNCALLS	The number of times split/skill ACD callers abandoned the call while on hold. Available on Generic 2 and Generic 3 switches.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
HOLDCALLS	The number of split/skill ACD calls that were placed on hold at least once. Available on Generic 2 and Generic 3 switches. **HOLDCALLS** includes **HOLDABNCALLS**.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
HOLDTIME	The amount of time spent by split/skill ACD callers on hold. Available on Generic 2 and Generic 3 switches.	C	integer	4 bytes
I_ACDAUXINTIME	The amount of time during the collection interval that **POSITIONS** were talking on **AUXIN** calls with a split/skill ACD call on hold where **SPLIT** is **OLDEST_LOGON**. Available on Generic 2.2 and Generic 3 switches.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_ACDAUX_ OUTTIME	The amount of time during the collection interval that **POSITIONS** spent dialing (Generic 2.2) and talking on **AUXOUT** calls with a split/skill ACD call for this split/skill on hold. **Note:** In a multiple call handling environment with agents in multiple skills, the ACD call for this skill must have been the last ACD call to have been put on hold before the agent made the AUXOUT call. Available on Generic 2.2 and Generic 3 switches.	C	integer	4 bytes
I_ACDOTHERTIME	The amount of time during the collection interval that **POSITIONS** spent in the OTHER state (dialing an outgoing call with a Generic 3 switch, with a ringing extension call with Generic 3 switch, or with calls on hold and with no other state selected) with a split/skill ACD call on hold. Available on Generic 2.2 and Generic 3 switches.	C	integer	4 bytes
I_ACDTIME	The amount of time during the collection interval that **POSITIONS** were on split/skill ACD calls. This includes time on **O_ACDCALLS** as well as on **ACDCALLS**.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_ACDTIME_R1	The time during the collection interval that Reserve Level 1 agents were on ACD calls for this skill. I_ACDTIME_R1 includes time on outgoing ACD calls placed by an adjunct. Reserve Level 1 agents will continue to be accumulate I_ACDTIME_R1 if the skill returns to Normal, while an ACD call is active. I_ACDTIME_R1 is available with Advocate Service Level Supervisor working with CMS R3V11 or later.	C	integer	4 bytes
I_ACDTIME_R2	The time during the collection interval that Reserve Level 2 agents were on ACD calls for this skill. This includes time on outgoing ACD calls placed by an adjunct. Reserve Level 2 agents will continue to accumulate I_ACDTIME_R2 if the skill returns to Normal, while an ACD call is active. I_ACDTIME_R2 is available with Advocate Service Level Supervisor working with CMS R3V11 or later.	C	integer	4 bytes
I_ACWINTIME	The amount of time during the collection interval that **POSITIONS** were in **ACW** for this split/skill, either associated with a split/skill ACD call or not associated with a call, and on inbound extension calls. This does not include time inbound extension calls spent on hold. Available on Generic 2.2 and Generic 3 switches.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_ACWOUTTIME	The amount of time during the collection interval that **POSITIONS** were in **ACW** for this split/skill, either associated with this split/skill ACD call or not associated with a call, and on outbound extension calls. This does not include time outbound extension calls spent on hold. Available on Generic 2.2 and Generic 3 switches.	C	integer	4 bytes
I_ACWTIME	The amount of time during the collection interval that **POSITIONS** were in **ACW** for this split/skill, either associated with a split/skill ACD call or not associated with a call. This **I_ACWTIME** includes **I_ACWINTIME** and **I_ACWOUTTIME**.	C	integer	4 bytes
I_ACWTIME_R1	The time during the collection interval that Reserve Level 1 agents were in ACW in this skill, either associated with a skill ACD call or not associated with a call. This item includes the time that agents in ACW spent on inbound extension calls or outbound extension calls. (ACWINTIME and ACWOUTIME are included and not broken out in separate R1 database items.) Reserve Level 1 Positions will continue to accumulate I_ACWTIME_R1 if the skill returns to Normal while the agent is in ACW or on ACD call and goes into ACW upon completing the call. I_ACWTIME_R1 is available with Advocate Service Level Supervisor working with CMS R3V11 or later. This is a cumulative item.	C	Integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_ACWTIME_R2	The time during the collection interval that Reserve Level 2 agents were in ACW in this skill, either associated with a skill ACD call or not associated with a call. This item includes the time that agents in ACW spent on inbound extension calls or outbound extension calls. (ACWINTIME and ACWOUTIME are included and not broken out into separate R2 database items.) Reserve Level 2 Positions will continue to be tracked in I_ACWTIME_R2 if the skill returns to Normal or on ACD call and goes into ACW upon completing the call. I_ACWTIME_R2 is available with Advocate Service Level Supervisor working with CMS R3V11 or later. This is a cumulative item.	C	Integer	4 bytes
I_ARRIVED	The number of calls that queued to this split/skill during this interval.	C	integer	4 bytes
I_AUXINTIME	The amount of time during the collection interval that **POSITIONS** were in **AUX** work, **AVAILABLE**, or, for Generic 2.2 and Generic 3 switches, had an ACD or **AUXIN/AUXOUT** call on hold and were on inbound extension calls. **I_AUXINTIME** includes **I_ACDAUXINTIME**.	C	integer	4 bytes
I_AUXOUTTIME	The amount of time during the collection interval that **POSITIONS** were in **AUX** work, **AVAILABLE;** or, for Generic 2.2 and Generic 3 switches, had an **ACD** or **AUXIN/AUXOUT** call on hold and were on outbound extension calls. **I_AUXOUTTIME** includes **I_ACDAUX_OUTTIME.**	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_AUXTIME	The amount of time during the collection interval that **POSITIONS** were in AUX in this split/skill. **I_AUXTIME** = **I_AUXTIME0** + **I_AUXTIME1** + **I_AUXTIME2** + **I_AUXTIME3** + **I_AUXTIME4** + **I_AUXTIME5** + **I_AUXTIME6** + **I_AUXTIME7** + **I_AUXTIME8** + **I_AUXTIME9.** **I_AUXTIME** includes **I_AUXTIME0**, **I_AUXTIME1-9**, **I_AUXINTIME**, **I_AUXOUTTIME**, and **I_TAUXTIME**.	C	integer	4 bytes
I_AUXTIME0	The amount of time during the collection interval that **POSITIONS** were in **AUX** for reason code 0 in this split/skill. This includes time on extension calls from this **AUX** state. For switches with **AUX** reason codes active, this represents time agents spent in “system” **AUX**. For switches without **AUX** reason codes active, **I_AUXTIME0** is the same as **I_AUXTIME**.	C	integer	4 bytes
I_AUXTIME1-9	The amount of time during the collection interval that **POSITIONS** were in **AUX** for each reason code 1-9 in this skill. This includes time on extension calls from each **AUX** state. Available for Generic 3 Version 5 and later Generic 3 switches with EAS.	C	integer	4 bytes
I_AVAILTIME	The amount of time during the collection interval that **POSITIONS** were available for calls from this split/skill. **I_AVAILTIME** includes **I_TAVAILTIME.**	C	integer	4 bytes
I_DA_ACDTIME	The amount of time during the collection interval that the agent spent talking on direct agent ACD calls queued through this split/skill. **I_DA_ACDTIME** is a subset of **I_OTHERTIME**. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_DA_ACWTIME	The amount of time that **POSITIONS** spent in ACW for direct agent **ACD** calls queued through this split/skill. **I_DA_ACWTIME** is a subset of **I_OTHERTIME**. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer	4 bytes
I_NORMTIME	The amount of time in seconds (0-3600) that this skill spent under all administered thresholds. Requires a DEFINITY ECS R6 with EAS.	C	integer	4 bytes
I_OL1TIME	The amount of time, in seconds (0-3600), that the skill spent over threshold 1. Requires a DEFINITY ECS R6 with EAS.	C	integer	4 bytes
I_OL2TIME	The amount of time, in seconds (0-3600), that this skill spent over threshold 2. Requires a DEFINITY ECS R6 with EAS.	C	integer	4 bytes
I_OTHERTIME	The amount of time during the collection interval that **POSITIONS** were doing other work. **I_OTHERTIME** is collected for the time period after the link to the switch comes up or after the agent logs in and before the CMS receives notification of the agent's state from the switch. For Generic 3 switches, other work includes: while in Auto-In or Manual-In mode, an agent put any call on hold and performed no further action; the agent had a direct agent call ringing, was on a direct agent call or in ACW for a direct agent call; the agent dialed to place a call or activate a feature; or an extension-in call rang at the agent's voice terminal with no other activity.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_OTHERTIME (contd.)	For Generic 2.2 with EAS and Generic 3 switches, the other work includes the time agents were logged into multiple splits/skills and doing work for a split/skill other than this one (with an ACD call ringing, talking on an ACD call, or in ACW for a split/skill other than this one). For Generic 3 switches with EAS and multiple call handling, agents are available in other, multiple call handling skills, but not in this skill. **I_OTHERTIME** includes **I_ACDOTHERTIME**, **I_DA_ACDTIME**, and **I_DA_ACWTIME**.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_OTHERTIME_R1	The time during the collection interval that Reserve Level 1 agents were doing other work while the skill is in overload 1. I_OTHERTIME_R1 is collected for the time period after the link to the switch comes up or after the agent logs in and before the CMS receives notification of the agent's state from the switch. Other work includes: while in Auto-In or Manual-In mode, the agent put any call on hold and performed no further action; the agent had a direct agent call ringing, was on a direct agent call or in ACW for a direct agent call; the agent dialed to place a call or activate a feature; or an extension-in call rang at the agent's terminal with no other activity. Also includes the time Reserve agents were logged into multiple skills and doing work for a skill other than this one (with an ACD call ringing, talking on an ACD call, or in ACW for a skill other than this one.) If the skill goes from overload 1 to Normal the agent will stop accumulating I_OTHERTIME_R1 and start accumulating I_OTHERSTBYTIME_R1. I_OTHERTIME_R1 does not include I_OTHERSTBYTIME_R1. I_OTHERTIME_R1 is available with Advocate Service Level Supervisor working with CMS R3V11 or later.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_OTHERTIME_R2	The time during the collection interval that Reserve Level 2 agents were doing other work while the skill is over threshold 2. I_OTHERTIME_R2 is collected for the time period after the link to the switch comes up or after the agent logs in and before the CMS receives notification of the agent's state from the switch. Other work includes: while in Auto-In or Manual-In mode, the agent put any call on hold and performed no further action; the agent had a direct agent call ringing, was on a direct agent call or in ACW for a direct agent call; the agent dialed to place a call or activate a feature; or an extension-in call rang at the agent's terminal with no other activity. Also includes the time Reserve agents were logged into multiple skills and doing work for a skill other than this one (with an ACD call ringing, talking on an ACD call, or in ACW for a skill other than this one.) If the skill goes from overload 2 to overload 1 the agent will stop accumulating I_OTHERTIME_R2 and start accumulating I_OTHERSTBYTIME_R2. I_OTHERTIME_R2 does not include I_OTHERSTBYTIME_R2. I_OTHERTIME_R2 is available with Advocate Service Level Supervisor working with CMS R3V11 or later.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_RINGTIME	The amount of time during the collection interval that agents were in the ringing state for calls to this split/skill. If the agent changes work modes or answers/makes another call instead of answering the ringing call, **I_RINGTIME** will stop accumulating. **RINGTIME** is the time the caller spends ringing and is independent of agent activity. **Note:** With forced multiple call handling (Generic 3 Version 4 and later), if an ACD call rings at the agent's voice terminal while the agent is talking on another call, **I_RINGTIME** does not accumulate. Available on Generic 2 and Generic 3 switches for ring tracking.	C	integer	4 bytes
I_RINGTIME_R1	The time during the collection interval that Reserve Level 1 agents were in the ringing state for calls to this skill. Reserve Level 1 agents will continue to accumulate I_RINGTIME_R1 if the skill returns to Normal while a call is ringing. I_RINGTIME_R1 is available with Advocate Service Level Supervisor working with CMS R3V11 or later.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_RINGTIME_R2	The time during the collection interval that Reserve Level 2 agents were in the ringing state for calls to this skill. Reserve Level 2 agents will continue to accumulate I_RINGTIME_R2 if the skill returns to Normal while a call is ringing. I_RINGTIME_R2 is available with Advocate Service Level Supervisor working with CMS R3V11 or later.	C	integer	4 bytes
I_STAFFTIME	The amount of time during the collection interval that **POSITIONS** were staffed (logged in). **I_STAFFTIME** = **I_AVAILTIME** + **I_ACDTIME** + **I_ACWTIME** + **I_AUXTIME** + **I_RINGTIME** + **I_OTHERTIME.** **I_STAFFTIME** includes **I_ACDTIME**, **I_ACWTIME**, **I_AUXTIME**, **I_AVAILTIME**, **I_OTHERTIME**, and **I_RINGTIME**.	C	integer	4 bytes
I_TAUXTIME	The amount of time top agents in this split/skill were in AUX mode. This includes time on **AUXIN/AUXOUT** calls, received or made without an ACD call on hold. (Time on **AUXIN/AUXOUT** calls with an ACD call on hold is tracked in **I_ACDAUXINTIME** and **I_ACDAUX_OUTTIME**.) Available with a Generic 3 switch with the EAS feature for top skills.	C	integer	4 bytes
I_TAVAILTIME	The amount of time top agents in this split/skill were available to receive calls for this split/skill. Available with a Generic 3 switch with the EAS feature for top skills.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
INCOMPLETE	This item indicates if data is complete for this collection interval. Data is incomplete whenever the link goes down and whenever tracking is aborted for calls, due to trunk failures (Generic 2 switches), the trunk going maintenance busy with a call active (Generic 3 switches), protocol failures with data collection active, or when split/skill or VDN call profile is changed with data collection active. The value for interval tables indicates whether data is incomplete for the interval (0 = NO, 1 = YES). The value in the daily, weekly, and monthly tables indicates the number of incomplete intervals in the day, week, or month.	C	smallint	2 bytes

Database Item	Description	Data Type	Column Type	Length
INFLOWCALLS	The number of calls that were redirected to the split's/skill's queue from another queue. When a call leaves the VDN (for example, by routing to another VDN) or leaves vector processing (for example, by routing to a split/skill), the next split/skill to which a call queues will not be credited with an inflow. Calls that ring at an agent and are then requeued to the same split/skill by the Redirect on No answer feature are counted as inflows to that split/skill. On Generic 2 and Generic 3 switches with vectoring, an inflow is counted for calls that intraflow from one split's queue to another (that is, call that queue to a split after having previously been queued to another split). For Generic 2 with vectoring, the calls that queue to a split and subsequently queue to a new split by a "queue to main" or "check backup" split vector command and are counted as inflows to the new split. On Generic 2.2 switches with EAS, an inflow is counted for this skill when a call is answered by an agent in a non-primary skill. Calls that are queued to one skill group and are subsequently queued to another are not counted as inflows to the subsequent skill group. Calls that queue to the "zero" skill after having been queued to a "nonzero" skill are not counted as inflows to the "zero" skill. Similarly, calls that queue to a "nonzero" skill after having been queued to a "zero" skill are not counted as inflows to the "nonzero" skill. For Generic 3 with vectoring and Generic 2.2 with EAS and multiple split/skill queueing, calls answered by an agent in a non-primary split/skill are counted as inflows to that split/skill. Calls that abandon from ringing at an agent's voice terminal in a non-primary split/skill are also counted as inflows to that skill.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
INFLOWCALLS (contd)	On Generic 3 Version 2 and newer switches, calls that ring at an agent in this split/skill and then requeue to the same split/skill by the Redirection on No Answer to a Split/Skill feature are counted as inflows.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
INTERFLOWCALLS	The number of **OUTFLOWCALLS** that were redirected to a destination outside the switch.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
INTRVL	The number of minutes in the intrahour interval (15, 30, or 60). **INTRVL** applies to intrahour tables only.	A	smallint; only in hsplit table	2 bytes
LOWCALLS	For switches with vectoring, this is the number of **ACDCALLS** with low priority that were answered by this split/skill. For switches without vectoring, this is the number of **ACDCALLS** with no priority that were answered by this split/skill.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
MAXINQUEUE	The maximum number of simultaneous calls in this split's/skill's queue during the collection interval.	M	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
MAXOCWTIME	The maximum amount of time that a call, recorded during the collection interval, waited in queue and ringing before an agent answered in this split/skill, the caller abandoned, or the call was redirected, received a busy signal, or was disconnected.	M	integer	4 bytes
MAXSTAFFED	The maximum number of agent **POSITIONS** that were simultaneously staffed during the collection interval. **MAXSTAFFED** includes **MAXTOP**.	M	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
MAXTOP	The maximum number of top agents that were staffed during the collection interval in this split/skill.	M	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
MEDCALLS	For switches with vectoring, the number of **ACDCALLS** with medium priority that were answered by agents in the split/skill. For example, answered calls that were queued to the split/skill with medium priority by a “queue to main” or “check backup” vector command. For Generic 3 switches with vectoring, **MEDCALLS** includes calls that were queued to a split/skill with no priority using the “route to” or “messaging split” vector commands, calls that queued directly to a non-vector-controlled split with no priority, and calls that intraflowed to a split/skill with no priority. For switches without vectoring, the number of **ACDCALLS** with “yes” priority that were answered by agents in the split/skill.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
NOANSREDIR	The number of split/skill ACD calls that rang at agent positions in the split/skill and were automatically redirected back to the split/skill queue or to a VDN by the Redirection on No Answer feature because they were not answered. Redirection On No Answer to a split/skill is available on DEFINITY ECS or Generic 3 Version 2 or later ECS. Redirection On No Answer to a VDN is available only on DEFINITY ECS R5 and later. **Note:** When a call is requeued to the same split/skill using the Redirection on No Answer feature, it is counted as an outflow from the split/skill and an inflow to the same split/skill. This is NOT true for calls that are redirected to a VDN using the Redirection on No Answer feature, rather than redirecting the call back to the same split/skill. Such calls count as outflows from the original split or skill, but do not count as inflows to the next split/skill to which they are queued through the new VDN.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
NOANSREDIR (contd.)	They are also counted as **NOANSREDIR** calls and can be subtracted out from the split/skill outflows and inflows to calculate the number of outflows and inflows that were not due to requeuing the call to the same split. This does not apply to Redirection on No Answer calls to a VDN. Such calls count as outflows from the original split or skill, but do not count as inflows to the next split/skill to which they are queued through the new VDN. It is also counted as a **NOANSREDIR** call and so can be subtracted out from the outflows and from the inflows to calculate the number of outflows and inflows that were not due to requeuing the call to the same split.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
O_ABNCALLS	The number of **ABNCALLS** that were placed by an adjunct, that is, the number of outbound predictive dialing calls that were abandoned by the far end. **O_ABNCALLS** is a subset of **ABNCALLS**. Available for outbound calls on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
O_ACDCALLS	The number of **ACDCALLS** that were placed by an adjunct (outbound predictive dialing). **O_ACDCALLS** includes **DA_ACDCALLS**. Available for outbound calls on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
O_ACDTIME	The amount of talk time of all **O_ACDCALLS** (does not include time calls spent on hold). Available for outbound calls on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature. **ACDTIME** includes **O_ACDTIME**.	C	integer	4 bytes
O_ACWTIME	The duration of all after call work associated with **O_ACDCALLS**. Available for outbound calls on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature. **O_ACWTIME** is included in **ACWTIME**.	C	integer	4 bytes
O_OTHERCALLS	The number of outbound calls queued to this split/skill that were not answered or abandoned as ACD split/skill calls. These include forced busy calls and calls with unknown dispositions. Available for outbound calls on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
OTHERCALLS	The number of calls offered to this split/skill that did not abandon and were not answered by an ACD agent for this split/skill. **OTHERCALLS** = **BUSYCALLS** + **DISCCALLS** + **OUTFLOWCALLS** + **DEQUEUECALLS.**	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
OTHERTIME	The amount of time **OTHERCALLS** waited in queue until the disposition was known and the call left the split/skill. **Note:** **OTHERTIME** relates to time for **OTHERCALLS** and is not related to **I_OTHERTIME**, which is the time agents spent in the OTHER state. **OTHERTIME** = **BUSYTIME** + **DEQUETIME** + **DISCTIME** + **OUTFLOWTIME.**	C	integer	4 bytes
OUTFLOWCALLS	The number of **CALLSOFFERED** that were redirected to another destination while queued to this split/skill. This can happen under different circumstances, depending on the switch release and on whether vectoring is active or not.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
OUTFLOWCALLS (contd)	For Generic 3 switches without vectoring, this is the number of **CALLSOFFERED** that were redirected to another destination while queued to this split/skill. This can happen if: • the call intraflowed or interflowed • the split/skill call forwarding was active • a ringing ACD call was answered using call pickup • a ringing ACD call redirects on no answer. For Generic 3 switches with vectoring, the number of outflow calls are counted if: • a ringing ACD call redirects on no answer • the call rang at an agent in this split/skill and was answered using call pickup • the call was routed to another VDN • the call routed to a number or digit • the call queued to a messaging split/skill • the call queued to this split/skill as the primary split/skill and was answered by an agent in another split/skill, rang at an agent in another split/skill and then abandoned or was redirected by the Redirection on No Answer feature (for Generic 3 Version 2 and later switch releases). For Generic 3 Version 2 switches, this is the number of **CALLSOFFERED** that were redirected to another destination while queued to this split/skill. This can happen by requeueing to the same split/skill via the Redirect on No Answer feature.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
OUTFLOWCALLS (contd)	For Generic 2 switches without vectoring, outflows are counted when calls intraflow to another split or extension. For Generic 2 switches with vectoring (except for Generic 2.2 with EAS), outflows are counted if: • the call was routed to another VDN • the call is routed to a number • the call queued to another split (using “queue to main” or “check backup”). For Generic 2 switches with vectoring, without EAS, this is the number of **CALLSOFFERED** that were redirected to another destination while queued to this split/skill. This can happen if: • the call intraflowed or interflowed • the split/skill call forwarding was active • the call queued to another split. For Generic 2.2 switches with EAS, outflows are counted if: • the call was routed to another VDN • the call is routed to a number • the call is queued to this skill which is a “nonzero” skill as primary and then is queued to the “zero” skill • a call is queued to this skill which is a “zero” skill and then is queued to a “nonzero” skill • the call is queued to this skill as primary and then is subsequently queued to another skill. **OUTFLOWCALLS** includes **INTERFLOWCALLS**, **NOANSREDIR**, and **SLVLOUTFLOWS.**	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
OUTFLOWTIME	The time all **OUTFLOWCALLS** waited in queue or ringing before being redirected.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
PERIOD1-9	The length, in seconds, of each service level increment as defined in the ACD Administration: Split/Skill Call Profile window. Each increment represents a progressively longer wait time. CMS counts answered or abandoned calls that wait beyond the last increment (**PERIOD9**) in either **ACDCALLS10** or **ABNCALLS10**.	A	smallint	2 bytes
PERIODCHG	This item indicates if service level increments **PERIOD1-9** (as defined on the ACD Administration: Split/Skill Call Profile window) changed during the collection interval. Valid values for **PERIODCHG** are 1 = YES and 0= NO.	A	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
PHANTOMABNS	The number of split/skill ACD calls with talk time less than the value of the phantom-abandoned call timer. Available on Generic 3 and newer switches.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
RINGCALLS	The number of this split's/skill's calls that rang at agent positions. Available on a Generic 2 and Generic 3 switch for ring tracking. **RINGCALLS** includes **ACDCALLS** and **NOANSREDIR.**	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
RINGTIME	The time this split's/skill's calls spent ringing at agent positions independent of final disposition and other agent activity. **I_RINGTIME** is the time the agent spends with ringing calls and is affected by other agent activity. **RINGTIME** is the time the caller spends ringing and is independent of agent activity. Available on a Generic 2 and Generic 3 switch for ring tracking.	C	integer	4 bytes
ROW_DATE (index)	The date on which data was collected.	I	date	Informix date, 4 bytes

Database Item	Description	Data Type	Column Type	Length
RSV_LEVEL	RSV_LEVEL provides an indication of whether the skill assigned to the agent is a non-reserve or a reserve level skill. 0 is for a non-reserve skill, 1 is for a reserve level 1 skill, and 2 is for a reserve level 2 skill. RSV_LEVEL is available with Advocate Service Level Supervisor working with CMS R3V11 or later.		small int	2 bytes
SERVICELEVEL	The number of seconds within which calls must be answered/connected in order to be considered acceptable (as defined on the ACD Administration: Split/Skill Call Profile window).	A	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
SLVLABNS	The number of **ABNCALLS** whose time to abandon was less than or equal to this split's/skill's **SERVICELEVEL**.	C	integer	4 bytes
SLVLOUTFLOWS	The number of **OUTFLOWCALLS** whose time to outflow was less than or equal to this split's/skill's **SERVICELEVEL**.	C	integer	4 bytes
SPLIT (index)	The split/skill number for which data was collected.	I	smallint	2 bytes
STARTTIME	The start time of the interval for which data was collected. **STARTTIME** applies only to the interval table.	I	smallint; only in hsplit table	2 bytes
SVCLEVELCHG	This item indicates if the service level was changed during the collection interval. Valid values for **SVCLEVELCHG** are 1 = YES and 0 = NO.	A	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
TOPCALLS	The number of **ACDCALLS** with top priority that were answered by agents in this split/skill. Available with Generic 2 and Generic 3 switches with vectoring.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

Database Item	Description	Data Type	Column Type	Length
TRANSFERRED	The number of **ACDCALLS** that were transferred to another destination. For Generic 2.1 switches, includes transfers to a measured VDN or split. For Generic 2.2 or Generic 3 switches, includes all split/skill calls transferred.	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table

ECS cross-reference

The following table lists which ECS release supports each Split/Skill database item.

Note:

The following is a key to the database items tables:

- Items marked “X” indicate that the database item is supported by the specified ECS release.
- Items marked “EAS” require that the EAS feature be active on the ECS for the items to be populated.

Items marked “e” are populated for the releases shown, but the values are only meaningful for EAS releases.

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ABNCALLS	X	X	X	X	X	X	X
ABNCALLS1-10	X	X	X	X	X	X	X
ABNRINGCALLS	X	X	X	X	X	X	X
ABNTIME	X	X	X	X	X	X	X
ACCEPTABLE	X	X	X	X	X	X	X
ACD (index)	X	X	X	X	X	X	X
ACDAUXOUTCALLS	X	X	X	X	X	X	X
ACDCALLS	X	X	X	X	X	X	X
ACDCALLS1-10	X	X	X	X	X	X	X
ACDCALLS_R1						X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ACDCALLS_R2						X	X
ACDTIME	X	X	X	X	X	X	X
ACWINCALLS	X	X	X	X	X	X	X
ACWINTIME	X	X	X	X	X	X	X
ACWOUTADJCALLS	X	X	X	X	X	X	X
ACWOUTCALLS	X	X	X	X	X	X	X
ACWOUTOFFCALLS	X	X	X	X	X	X	X
ACWOUTOFFTIME	X	X	X	X	X	X	X
ACWOUTTIME	X	X	X	X	X	X	X
ACWTIME	X	X	X	X	X	X	X
ANSTIME	X	X	X	X	X	X	X
ASSISTS	X	X	X	X	X	X	X
AUXINCALLS	X	X	X	X	X	X	X
AUXINTIME	X	X	X	X	X	X	X
AUXOUTADJCALLS	X	X	X	X	X	X	X
AUXOUTCALLS	X	X	X	X	X	X	X
AUXOUTOFFCALLS	X	X	X	X	X	X	X
AUXOUTOFFTIME	X	X	X	X	X	X	X
AUXOUTTIME	X	X	X	X	X	X	X
BACKUPCALLS	X	X	X	X	X	X	X
BUSYCALLS	X	X	X	X	X	X	X
BUSYTIME	X	X	X	X	X	X	X
CALLSOFFERED	X	X	X	X	X	X	X
CONFERENCE	X	X	X	X	X	X	X
DA_ACWINCALLS	X	X	X	X	X	X	X
DA_ACWINTIME	X	X	X	X	X	X	X
DA_ACWOCALLS	X	X	X	X	X	X	X
DA_ACWOTIME	X	X	X	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
DEQUECALLS	X	X	X	X	X	X	X
DEQUETIME	X	X	X	X	X	X	X
DISCCALLS	X	X	X	X	X	X	X
DISCTIME	X	X	X	X	X	X	X
EVENT1-9	X	X	X	X	X	X	X
HIGHCALLS	X	X	X	X	X	X	X
HOLDABNCALLS	X	X	X	X	X	X	X
HOLDCALLS	X	X	X	X	X	X	X
HOLDTIME	X	X	X	X	X	X	X
I_ACDAUXINTIME	X	X	X	X	X	X	X
I_ACDAUX_OUTTIME	X	X	X	X	X	X	X
I_ACDOTHERTIME	X	X	X	X	X	X	X
I_ACDTIME	X	X	X	X	X	X	X
I_ACDTIME_R1							X
I_ACDTIME_R2							X
I_ACWINTIME	X	X	X	X	X	X	X
I_ACWOUTTIME	X	X	X	X	X	X	X
I_ACWTIME	X	X	X	X	X	X	X
I_ACWTIME_R1							X
I_ACWTIME_R2							X
I_ARRIVED	X	X	X	X	X	X	X
I_AUXINTIME	X	X	X	X	X	X	X
I_AUXOUTTIME	X	X	X	X	X	X	X
I_AUXSTBYTIME_R1							X
I_AUXSTBYTIME_R2							X
I_AUXTIME	X	X	X	X	X	X	X
I_AUXTIME_R1							X
I_AUXTIME_R2							X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
I_AUXTIME0	X	X	X	X	X	X	X
I_AUXTIME1-9			EAS	X	X	X	X
I_AVAILTIME	X	X	X	X	X	X	X
I_DA_ACDTIME	X	X	X	X	X	X	X
I_DA_ACWTIME	X	X	X	X	X	X	X
I_NORMTIME				EAS	EAS	EAS	EAS
I_OL1TIME				EAS	EAS	EAS	EAS
I_OL2TIME				EAS	EAS	EAS	EAS
I_OTHERSTBYTIME_R1							X
I_OTHERSTBYTIME_R2							X
I_OTHERTIME	X	X	X	X	X	X	X
I_OTHERTIME_R1							X
I_OTHERTIME_R2							X
I_RINGTIME	X	X	X	X	X	X	X
I_RINGTIME_R1							X
I_RINGTIME_R2							X
I_STAFFTIME	X	X	X	X	X	X	X
I_TAUXTIME	X	X	X	X	X	X	X
I_TAVAILTIME	X	X	X	X	X	X	X
I_TOTHERTIME							X
INCOMPLETE	X	X	X	X	X	X	X
INFLOWCALLS	X	X	X	X	X	X	X
INTERFLOWCALLS	X	X	X	X	X	X	X
INTRVL	X	X	X	X	X	X	X
LOWCALLS	X	X	X	X	X	X	X
MAXINQUEUE	X	X	X	X	X	X	X
MAXOCWTIME	X	X	X	X	X	X	X
MAXSTAFFED	X	X	X	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
MAXTOP	*	*	*	X	X	X	X
MEDCALLS	X	X	X	X	X	X	X
NOANSREDIR	X	X	X	X	X	X	X
O_ABNCALLS	X	X	X	X	X	X	X
O_ACDCALLS	X	X	X	X	X	X	X
O_ACDTIME	X	X	X	X	X	X	X
O_ACWTIME	X	X	X	X	X	X	X
O_OTHERCALLS	X	X	X	X	X	X	X
OTHERCALLS	X	X	X	X	X	X	X
OTHERTIME	X	X	X	X	X	X	X
OUTFLOWCALLS	X	X	X	X	X	X	X
OUTFLOWTIME	X	X	X	X	X	X	X
PERIOD 1-9	X	X	X	X	X	X	X
PERIODCHG	X	X	X	X	X	X	X
PHANTOMABNS	X	X	X	X	X	X	X
RINGCALLS	X	X	X	X	X	X	X
RINGTIME	X	X	X	X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
SERVICELEVEL	X	X	X	X	X	X	X
SLVLABNS	X	X	X	X	X	X	X
SLVLOUTFLOWS	X	X	X	X	X	X	X
SPLIT	X	X	X	X	X	X	X
STARTTIME	X	X	X	X	X	X	X
SVCLEVELCHG	X	X	X	X	X	X	X
TOPCALLS	X	X	X	X	X	X	X
TRANSFERRED	X	X	X	X	X	X	X

Agent database items

Overview

The Agent database item descriptions apply to historical items.

The **Data Type** column refers to **Cumulative (C)**, **Administrative (A)**, **Status (S)**, **Row Identifier (I), Busy Hour (B), Special Table (N),** or **Maximum Value (M)** data. Cumulative, Administrative, Row Identifier, Busy Hour, and Maximum Value items typically apply to both the current and previous interval real-time tables. Special Table items are historical, and apply only to the table in which they are stored.

Historical agent database items apply to the Intrahour Agent (`hagent`), Daily Agent (`dagent`), Weekly Agent (`wagent`), and Monthly Agent (`magent`) tables. All items listed in the following table are included in all four tables, unless otherwise noted in the Column Type column. Any differences in the data format between the four agent tables are also noted in the Column Type column. The historical indexes are **LOGID**, **SPLIT,** and **ROW_DATE**.

Contents

“Agent database items” contains the following topics:

Agent database item table

The following table describes the data items in the CMS Agent database tables.

Database Item	Description	Data Type	Column Type	Length
ABNCALLS	The number of split/skill ACD calls that were abandoned while ringing the agent's voice terminal (after being directed to the agent voice terminal, but before being answered). This includes calls considered abandoned because their talk time was less than the phantom-abandoned call timer. For Generic 3 switches, **ABNCALLS** includes **PHANTOMABNS**. Available on Generic 2 and Generic 3 switches.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
ABNTIME	The amount of time split/skill ACD callers waited while ringing the agent's voice terminal before the call was abandoned. For Generic 3 switches, **ABNTIME** includes the time until the agent releases the call for phantom-abandoned calls. Also available on Generic 2 and Generic 3 switches.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
ACD (index)	The ACD number for which data was collected.	I	smallint	2 bytes
ACD_RELEASE	The number of split/skill ACD calls that the agent released or dropped before the far end released. **Note:** The transfers and conferences are always recorded as agent-released calls. Available for Generic 3 switches.	C	integer; not in hagent table	4 bytes
ACDAUXOUTCALLS	The number of **AUXOUTCALLS** the agent made with at least one split/skill or direct agent ACD call on hold. This includes calls made to transfer or conference the ACD call. Available on Generic 2.2 and Generic 3 switches.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)

Database Item	Description	Data Type	Column Type	Length
ACDCALLS	The number of calls that were queued to **SPLIT** and answered by this agent in this **SPLIT**. **ACDCALLS** includes **O_ACDCALLS** and **ACD_RELEASE.**	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
ACDCALLS_R1	The number of calls that were queued to **SPLIT** and answered by this reserve1 agent in this **SPLIT**. Found only in CMS R3V9.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
ACDCALLS_R2	The number of calls that were queued to **SPLIT** and answered by this reserve2 agent in this **SPLIT**. Found only in CMS R3V9.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
ACDTIME	The amount of talk time of all **ACDCALLS**. **ACDTIME** includes **O_ACDTIME**. It does not include **HOLDTIME** except on Generic 2.1 switches.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
ACWINCALLS	The number of inbound extension calls received by the agent while in ACW. This includes ACW for split/skill and direct agent ACD calls and ACW not associated with a call.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
ACWINTIME	The amount of talk time of all **ACWINCALLS.** **ACWINTIME** includes **DA_ACWINCALLS**, but does not include **HOLDTIME** except on Generic 2.1 switches.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
ACWOUTADJCALLS	The number of **ACWOUTCALLS** that were placed by an adjunct on behalf of an agent (keyboard-dialed). If such calls are placed to off-switch destinations, they are also counted as **ACWOUTOFFCALLS**. Available on the Generic 2.2 switch with the ASAI Gateway Interface feature and on the Generic 3 switch with the ASAI feature.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
ACWOUTCALLS	The number of outbound extension calls made by the agent or on behalf of the agent while in ACW. This includes ACW for split/skill ACD calls and ACW not associated with a call. **ACWOUTCALLS** includes **ACWOUTADJCALLS, ACWOUTOFFCALLS,** and **DA_ACWOCALLS**.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
ACWOUTOFFCALLS	The number of **ACWOUTCALLS** that were made to an off-switch destination—a destination outside the switch. If these calls were placed by an adjunct on behalf of the agent (keyboard-dialed), they are counted as **ACWOUTADJCALLS**. Available for external calls on Generic 2.2 and Generic 3 switches.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
ACWOUTOFFTIME	The amount of talk time of all **ACWOUTOFFCALLS** (does not include time on hold). **ACWOUTTIME** includes **ACWOUTOFFTIME**. Available for external calls on Generic 2.2 and Generic 3 switches.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
ACWOUTTIME	The amount of talk time of all **ACWOUTCALLS**. ACWOUTTIME does not include **HOLDTIME** (except for Generic 2.1 switches). **ACWOUTTIME includes time** spent on calls made while in ACW that was not associated with an ACD call and on **ACWOUTADJCALLS** and on **ACWOUTOFFCALLS**. For Generic 2.2 and generic 3 switches, **ACWOUTTIME** does not include time **ACWOUTCALLS** spent on hold. It does include time spent on calls made while in ACW not associated with an ACD call.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
ACWTIME	The duration of all after call work associated with **ACDCALLS**, including **ACWINTIME and ACWOUTCALLS** received/made during call-associated ACW. **Note:** **ACWTIME** does not include the time spent in ACW not associated with an ACD call (that is, the agent pressed the ACW button while not on an ACD call). However, both **ACWINTIME** and **ACWOUTTIME** do include time spent on calls made or received while in ACW not associated with an ACD call. Therefore, the sum of **ACWINTIME** and **ACWOUTIME** may be greater than **ACWTIME**. **ACWTIME** includes **ACWINTIME**, **ACWOUTTIME**, **DA_ACWTIME**, and **O_ACWTIME**.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
ANSRINGTIME	The amount of time split/skill and direct agent ACD calls spent ringing at the agent's voice terminal before being answered. Available for ring-tracking on Generic 2 and Generic 3 switches.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
ASSISTS	The number of times the supervisor was called (supervisor assists) by agents on a split/skill direct agent ACD calls, or in call-related ACW for this split/skill.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
AUXINCALLS	The number of inbound extension calls received by agents while in **AUX** (auxiliary work), **AVAILABLE**; or, for Generic 2.2 and Generic 3 switches, with an **ACD** or **AUXIN/AUXOUT** call on hold.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
AUXINTIME	The amount of talk time of all **AUXINCALLS** (does not include **HOLDTIME** except on Generic 2.1 switches).	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
AUXOUTADJCALLS	The number of **AUXOUTCALLS** that were placed by an adjunct on behalf of an agent (keyboard dialed). If such calls are placed to off-switch destinations, they are also counted as **AUXOUTOFFCALLS.** Available for outbound calls on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)

Database Item	Description	Data Type	Column Type	Length
AUXOUTCALLS	The number of outbound extension calls that were made by the agent or on behalf of the agent while in AUX (auxiliary work), AVAILABLE; or, for Generic 2.2 and Generic 3 switches with an ACD or AUXIN/AUXOUT call on hold. **Note:** Calls the agent makes to transfer or conference an ACD call are included as AUXOUT calls. **AUXOUTCALLS** includes **AUXOUTADJCALLS, AUXOUTOFFCALLS,** and **ACDAUXOUTCALLS**.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
AUXOUTOFFCALLS	The number of **AUXOUTCALLS** that were made to a destination outside the switch. If such calls were placed by an adjunct on behalf of the agent (keyboard-dialed), they are also counted as **AUXOUTADJCALLS**. Available for external calls on Generic 2.2 and Generic 3 switches.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
AUXOUTOFFTIME	The amount of talk time of all **AUXOUTOFFCALLS** (does not include **HOLDTIME**). This time is included in **AUXOUTTIME**. Available for external calls on Generic 2.2 and Generic 3 switches.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
AUXOUTTIME	The amount of talk time of all **AUXOUTCALLS**. **AUXOUTTIME** does not include **HOLDTIME** except on Generic 2.1 switches). **AUXOUTTIME** includes **AUXOUTOFFTIME**, **AUXOUTOFFCALLS**, and **AUXADJCALLS**.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
CONFERENCE	The number of times the agent completed a conference; that is, pushed the conference key a second time). Available on Generic 2.2 and Generic 3 switches.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
DA_ABNCALLS	The number of direct agent ACD calls that were abandoned by callers while in queue or ringing the agent's voice terminal. Includes calls considered abandoned because their talk time was less than the phantom abandon call timer. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
DA_ABNTIME	The amount of time **DA_ABNCALLS** were waiting in queue or ringing before being abandoned. Includes the time until the agent releases the call for phantom abandoned calls. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
DA_ACDCALLS	The number of direct agent ACD calls that the agent answered. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling. **DA_ACDCALLS** includes **DA_RELEASE**.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
DA_ACDTIME	The amount of talk time of all **DA_ACDCALLS** (does not include **HOLDTIME**). Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
DA_ACWINCALLS	The number of inbound extension calls answered by the agent while in ACW for direct agent ACD calls. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
DA_ACWINTIME	The amount of talk time of all **DA_ACWINCALLS** (does not include **HOLDTIME**). Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
DA_ACWOADJCALLS	The number of **DA_ACWOCALLS** that were placed by an ASAI adjunct on behalf of the agent (keyboard-dialed). If these calls were placed to off-switch destinations, they are also counted as **DA_ACWOOFFCALLS**. Available on Generic 3 switches with ASAI.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
DA_ACWOCALLS	The number of outbound extension calls agents made while in ACW for direct agent ACD calls queued through this split/skill. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling. **DA_ACWOCALLS** includes **DA_ACWOADJCALLS** and **DA_ACWOOFFCALLS.**	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
DA_ACWOOFFCALLS	The number of **DA_ACWOCALLS** that were made to an off-switch location. If these calls were placed by an adjunct on behalf of the agent (keyboard-dialed), they are also counted as **DA_ACWOADJCALLS**. Requires a Generic 3 switch with ASAI.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)

Database Item	Description	Data Type	Column Type	Length
DA_ACWOOFFTIME	The amount of talk time of all **DA_ACWOOFFCALLS** (does not include **HOLDTIME**). **DA_ACWOOFFTIME** is included in **DA_ACWOTIME**. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
DA_ACWOTIME	The amount of talk time of all DA_ACWOCALLS (does not include HOLDTIME). **DA_ACWOTIME** includes **DA_ACWOOFFTIME**. Available on Generic 3 switches with the ASAI or EAS feature for direct agent calling.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
DA_ACWTIME	The duration of ACW associated with **DA_ACDCALLS**, including time on **DA_ACWINCALLS** and **DA_ACWOCALLS**. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling. **DA_ACWTIME** includes **DA_ACWINTIME** and **DA_ACWOTIME**.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
DA_ANSTIME	The amount of time spent by callers in direct agent queue and ringing before being answered. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
DA_OTHERCALLS	The number of direct agent calls that were redirected to another destination before being answered; for example, by call pickup, coverage or Redirection on No Answer. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)

Database Item	Description	Data Type	Column Type	Length
DA_OTHERTIME	The amount of time spent in queue or ringing by **DA_OTHERCALLS** before being redirected. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
DA_RELEASE	The number of direct agent ACD calls released or dropped by the agent before the far end released. Requires an ECS with the ASAI or EAS feature for direct agent calling.	C	integer	4 bytes
EVENT1-9	The number of times each event (stroke count) feature button (1 to 9) was pressed while the agent was on an ACD call or in call-related after call work. Available on Generic 2 and Generic 3 switches.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
EXTENSION	The extension number for which data were collected.	A	char(6)	6 byte ASCII text string
HOLDABNCALLS	The number of times callers abandoned from hold. For Generic 2.2 and Generic 3 switches, **HOLDABNCALLS** applies to all calls the agent put on hold. For Generic 2.1 switches, **HOLDABNCALLS** applies to split ACD calls held.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
HOLDACDTIME	The amount of time split/skill and direct agent ACD calls spent on hold at the agent's voice terminal. This includes time on **AUXIN** or **AUXOUT** calls with the ACD calls on hold.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
HOLDCALLS	The number of calls that were placed on hold at least once. **HOLDCALLS** includes **HOLDABNCALLS**. For Generic 2.1 switches, **HOLDCALLS** applies to split ACD calls held. For Generic 3 and Generic 2.2 switches, **HOLDCALLS** applies to all calls the agent put on hold. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
HOLDTIME	The amount of time spent by callers on hold. For Generic 2.1 switches, **HOLDTIME** applies to split ACD calls held. For Generic 3 and Generic 2.2 switches, **HOLDTIME** applies to all calls the agent put on hold. **HOLDTIME** includes **HOLDACDTIME**.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
I_ACDAUXINTIME	The amount of time during the collection interval that the agent spent talking on AUXIN calls with at least one split/skill or direct agent ACD call on hold. For agents in multiple splits/skills, this time is recorded in the record in which **SPLIT** is **OLDEST_LOGON**. Available on Generic 2.2 and Generic 3 switches.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
I_ACDAUX_OUTTIME	The amount of time during the collection interval that the agent spent dialing (Generic 2.2) and talking on AUXOUT calls with at least one split/skill or direct agent ACD call for this split/skill with the call on hold. Available on Generic 2.2 and Generic 3 switches.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
I_ACDOTHERTIME	The amount of time during the collection interval that the agent spent in the OTHER state (dialing an outgoing call, with a ringing personal call [Generic 3 switches], or with calls on hold and with no other state selected) with at least one split/skill or direct agent ACD call on hold. Available on Generic 2.2 and Generic 3 switches.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
I_ACDTIME	The amount of time during the collection interval that the agent was talking on ACD calls for **SPLIT** or the time ACD calls spent on hold. **I_ACDTIME** includes time spent on **O_ACDCALLS**, but does not include **HOLDTIME** (except for Generic 2.1 switches).	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
I_ACWINTIME	The amount of time during the collection interval that the agent was in ACW and on inbound extension calls. **I_ACWINTIME** includes ACW for split/skill ACD calls and ACW not associated with a call, but does not include the time inbound ACW calls spent on hold (except for Generic 2.1 switches).	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
I_ACWOUTTIME	The amount of time during the collection interval that the agent was in **ACW** and on outbound extension calls. **I_ACWOUTTIME** includes ACW for split/skill ACD calls and ACW not associated with a call, but does not include the time **ACWOUT** calls spent on hold (except for Generic 2.1 switches).	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
I_ACWTIME	The amount of time during the collection interval that the agent was in **ACW**. This includes ACW for split/skill ACD calls and ACW not associated with a call. **Note:** **I_ACWINTIME** and **I_ACWOUTTIME** include time in **ACW** for direct agent calls, but **I_ACWTIME** does not include this time. Therefore, the sum of **I_ACWINTIME** and **I_ACWOUTTIME** may be greater than **I_ACWTIME**. **I_ACWTIME** includes **I_ACWINTIME** and **I_ACWOUTTIME.**	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
I_AUXINTIME	The amount of time during the collection interval that the agent was in AUX work, **AVAILABLE**; or, for Generic 2.2 and Generic 3 switches, including time when an ACD or AUXIN/AUXOUT call is on hold and on inbound extension calls and **SPLIT** was the **OLDEST_LOGON.** **I_AUXINTIME** includes **I_ACDAUXINTIME** but does not include time calls spent on hold except for Generic 2.1 switches, which include time spent on hold unless the agent makes an outgoing call with an AUXIN call on hold.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
I_AUXOUTTIME	The amount of time during the collection interval that the agent was in AUX work, AVAILABLE, or, for Generic 2.2 and Generic 3 switches, including time when an ACD or **AUXIN/AUXOUT** call on hold and on outbound extension calls. In the cases where the agent was in AUX work, **AVAILABLE** or had an **AUXIN/AUXOUT** call on hold, the **AUXOUT** time and calls are recorded for the **SPLIT** that is the **OLDEST_LOGON**. In cases where the agent had an ACD call on hold, **SPLIT** is the split or skill associated with the last ACD call put on hold. **I_AUXOUTTIME** includes **I_ACDAUX_OUTTIME**, but does not include time calls spent on hold (except for Generic 2.1 switches).	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
I_AUXSTBYTIME	The time during the collection interval that an Agent was in AUX Work state while the skill was Normal. This database item is valid only for agents administered as Reserve Level 1 or Reserve Level 2 for a skill. **Note:** When the skill is Normal, the individual Reserve Agent's time for that skill is tracked as I_OTHERSTBYTIME and I_AUXSTBYTIME. I_ACWTIME_R2 is available with Advocate Service Level Supervisor working with CMS R3V11 or later.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
I_AUXTIME	The amount of time the agent spent in AUX work in SPLIT. When an agent is in AUX work in multiple splits/skills, this time is recorded in each split or skill in which the agent is in AUX. **I_AUXTIME** includes **I_AUXINTIME** and **I_AUXOUTTIME**.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
I_AUXTIME_R1	The time during the collection interval that Reserve Level 1 agents spent in AUX Work for this skill while the skill was in overload 1 condition. Includes all AUX time including time on extension calls from the AUX state. Unlike I_AUXTIME, this item is not broken down into separate items, call direction, or reason codes. If the skill goes from overload 1 to Normal the agent will stop accumulating I_AUXTIME_R1 and start accumulating I_AUXSTBYTIME_R1. I_AUXTIME_R1 is available with Advocate Service Level Supervisor working with CMS R3V11 or later.	C	integer	4 bytes
I_AUXTIME_R2	The time during the collection interval that Reserve Level 2 agents were in AUX in this skill while the skill was in overload 2. Includes all AUX time including time on extension calls from the AUX state. Unlike I_AUXTIME, this item is not broken down into separate items, call direction, or reason codes. If the skill goes from overload 2 to overload 1 the agent will stop accumulating I_AUXTIME_R2 and start accumulating I_AUXSTBYTIME_2. I_AUXTIME_R2 is available with Advocate Service Level Supervisor working with CMS R3V11 or later.	C	integer	4 bytes
I_AVAILTIME	The amount of time during the collection interval that the agent was available for ACD calls in this split/skill.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
I_DA_ACDTIME	The amount of time during the collection interval that the agent spent talking on direct agent calls. Does not include **HOLDTIME**. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
I_DA_ACWTIME	The amount of time during the collection interval that the agent was doing after call work associated with direct agent ACD calls. Requires a Generic 3 switch with the ASAI or EAS feature for direct agent calling.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
I_OTHERSTBYTIME	The time during the collection interval that the Agent was logged into the skill and in standby because the skill was not in overload 1 or 2. This database item is valid only for agents administered as Reserve Level 1 or Reserve Level 2 for a skill. **Note:** Reserve Level 1 agents that are working on a call for another skill when the reserve skill goes into an overload condition will stop being tracked as I_OTHERSTBYTIME_R1 and start being tracked as I_OTHERTIME_R1. I_OTHERSTBYTIME is available with Advocate Service Level Supervisor working with CMS R3V11 or later. I_OTHERSTBYTIME is not included in I_STAFFTIME or I_OTHERTIME.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
I_OTHERSTBYTIME_R1	The time during the collection interval that Reserve Level 1 agents were logged into this skill but not needed. I_OTHERSTBYTIME_R1 accumulates while the skill (for which the agent is Reserve Level 1) is not in overload 1. I_OTHERSTBYTIME is not included in I_STAFFTIME, I_OTHERTIME or I_OTHERTIME_R1. **Note:** Reserve Level 1 agents that are working on a call for another skill when the reserve skill goes into an overload condition will stop being tracked as I_OTHERSTBYTIME_R1 and start being tracked as I_OTHERTIME_R1. I_OTHERSTBYTIME_R1 is available with Advocate Service Level Supervisor working with CMS R3V11 or later.	C	Integer	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
I_OTHERSTBYTIME_R2	The time during the collection interval that Reserve Level 2 agents were staffed in this skill and in standby. I_OTHERSTBYTIME_R2 accumulated while the skill (for which the agent is Reserve Level 2) is not in overload 2. **Note:** Reserve Level 2 agents that are working on a call for another skill when the reserve skill goes into an overload condition will stop being tracked as I_OTHERSTBYTIME_R2 and start being tracked as I_OTHERTIME_R2. I_OTHERSTBYTIME_R2 is available with Advocate Service Level Supervisor working with CMS R3V11 or later.	C	Integer	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
I_OTHERTIME	The amount of time during the collection interval that **POSITIONS** were doing other work. **I_OTHERTIME** is collected for the time period after the link to the switch comes up or after the agent logs in and before the CMS receives notification of the agent's state from the switch. For Generic 3 switches, other work includes: while in Auto-In or Manual-In mode, an agent put any call on hold and performed no further action; the agent had a direct agent call ringing, was on a direct agent call or in ACW for a direct agent call; the agent dialed to place a call or activate a feature; or an extension-in call rang at the agent's voice terminal with no other activity. For Generic 2.2, agents were in AUTO-IN or MANUAL-IN, put a call on hold using the Hold button or switchhook flash and performed no further action. For Generic 2.2 with EAS and Generic 3 switches, the other work includes the time agents were logged into multiple splits/skills and doing work for a split/skill other than this one (with an ACD call ringing, talking on an ACD call, or in ACW for a split/skill other than this one). For Generic 3 switches with EAS and multiple call handling, agents are available in other, multiple call handling skills, but not in this skill. **I_OTHERTIME** includes **I_ACDOTHERTIME**, **I_DA_ACDTIME**, and **I_DA_ACWTIME**.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
I_RINGTIME	The amount of time during the collection interval that the agent had split/skill and direct agent ACD calls ringing. If the agent changes work modes or makes/receives another call instead of answering the ringing call, **I_RINGTIME** will stop accumulating. **RINGTIME** is the time the caller spends ringing and is independent of agent activity. Available on a Generic 2 and Generic 3 switch for ring tracking.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
I_STAFFTIME	The amount of time during the collection interval that the agent was staffed (logged in) in this split/skill. **I_STAFFTIME** includes **I_AUXTIME, I_AVAILABLE, I_ACDTIME, I_ACWTIME, I_DA_ACDTIME, I_DA_ACWTIME, I_OTHERTIME**, and **I_RINGTIME**.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
INCOMPLETE	This item indicates if data is complete for this collection interval. Data is incomplete whenever the link goes down and whenever tracking is aborted for calls, due to trunk failures (Generic 2), the trunk going maintenance busy with a call active (Generic 3), protocol failures with data collection active, or when split/skill or VDN call profile is changed with data collection active. The value for interval tables indicates whether data is incomplete for the interval (0 = NO, 1 = YES). The value in the daily, weekly, and monthly tables indicates the number of incomplete intervals in the day, week, or month. Changing split/skill or VDN call profile data while data collection is active only affects the respective split/skill or VDN data.	C	smallint	2 bytes

Database Item	Description	Data Type	Column Type	Length
INTRVL	The number of minutes in the intrahour interval (15, 30, or 60). **INTRVL** applies to intrahour intervals only.	A	smallint; only in hagent table	2 bytes
LOC_ID	The switch port network location ID that is associated with an agent upon login to the ACD. A location ID is not directly assigned to an agent; instead, it is associated with the equipment location of the voice terminal that the agent uses to log into the ACD. Therefore, only when an agent logs into the ACD can the agent become associated with a location ID. Valid values are 01-44. **Important:** CMS R3V11 only supports location IDs from 1 to 44. MultiVantage 11.1 software supports location IDs from 1 to 64. When running CMS reports that include location IDs (LOC_ID), those IDs defined on the MultiVantage system that are greater than 44 will return a default location ID of 0 to CMS. Support for location IDs above 44 is expected in a future release of CMS.	C	integer	4 bytes
LOGID (index)	The Login ID that was used to staff the **EXTENSION**. Agents in multiple splits/skills have one **LOGID**.	A	char(10)	10 byte ASCII text string

Database Item	Description	Data Type	Column Type	Length
NOANSREDIR	The number of split/skill and direct agent ACD calls that rang at this agent's voice terminal and then were automatically redirected by the Redirection on No Answer feature because they were not answered. Split/skill ACD calls are requeued to the split/skill or VDN; direct agent ACD calls are redirected to the agent's coverage path. Redirection on no Answer to a split/skill is available on DEFINITY ECS or Generic 3 Version 2 or later switches. Redirection On No Answer to a VDN is only available on DEFINITY ECS.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
O_ACDCALLS	The number of **ACDCALLS** and **DA_ACDCALLS** that were placed by an adjunct (predictive dialing). Available for outbound calling on a Generic 2.2 switch with the ASAI Gateway Interface feature and on a Generic 3 switch with the ASAI feature.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
O_ACDTIME	The amount of talk time of all **O_ACDCALLS** (does not include time calls spent on hold). This time is included in **ACDTIME**. Available for outbound calling on Generic 2.2 switches with the ASAI Gateway Interface feature and on Generic 3 switches with the ASAI feature.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
O_ACWTIME	The duration of all ACW associated with **O_ACDCALLS**. **O_ACWTIME** is included in **ACWTIME**. Available for outbound calling on Generic 2.2 switches with the ASAI Gateway Interface feature and on Generic 3 switches with the ASAI feature.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
PHANTOMABNS	The number of ACD calls with talk time less than the value of the phantom-abandoned call timer. Available on Generic 3 switches.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
RINGCALLS	The number of split/skill (Generic 2 and Generic 3 switches) and direct agent ACD calls (Generic 3 switches) that rang at the agent's position. **RINGCALLS** includes **NOANSREDIR**. Available on Generic 2 and Generic 3 switches for ring tracking.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)
RINGTIME	The amount of time split/skill and direct agent ACD calls spent ringing at the agent's position (independent of disposition or other agent activity). **RINGTIME** is the time the caller spends ringing and is independent of agent activity. **I_RINGTIME** is the time the agent spends in the ringing state and is affected by other agent activity. **RINGTIME** includes **ANSRINGTIME**. Available on Generic 2 and Generic 3 switches.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
ROW_DATE (index)	The day for which the data was collected or the exception occurred.	I	date	4 byte Informix date
SPLIT (index)	The split number to which the **EXTENSION** is assigned to or the skill number that the agent logged into.	I	smallint	2 bytes
STARTTIME	The start time for the interval for which data was collected. **STARTTIME** applies only to the Interval table.	I	smallint; only in hagent table	2 bytes

Database Item	Description	Data Type	Column Type	Length
TI_AUXTIME	**TI_AUXTIME** includes **TI_AUXTIME0**, **TI_AUXTIME1-9, I_AUXINTIME**, and **I_AUXOUTTIME.** **sum(TI_AUXTIME) = sum(TI_AUXTIME0 + TI_AUXTIME1 + TI_AUXTIME2 + TI_AUXTIME3 + TI_AUXTIME4 + TI_AUXTIME5 + TI_AUXTIME6 + TI_AUXTIME7 + TI_AUXTIME8 + TI_AUXTIME9)**, over all splits/skills the agent was logged into. Requires an ECS with the EAS feature.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
TI_AUXTIME0	The amount of time the agent spent in **AUX** with reason code 0 (zero). This is time in “system” AUX for switches with AUX reason codes active. It is the same as **TI_AUXTIME** for switches without AUX reason codes active. “TI_” time is only stored for the skill logged in to the longest. “TI_” time needs to be summed across the skills the agents may log in to, in case the login order changes during the collection interval. Requires an ECS with the EAS feature.	C	integer	4 bytes
TI_AUXTIME1-9	The amount of time the agent spent in AUX with reason codes 1-9. “TI_” time is only stored for the skill logged in to the longest. “TI_” time needs to be summed across the skills the agents may log in to, in case the login order changes during the collection interval. Requires an ECS with the EAS feature.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
TI_AVAILTIME	The amount of time during the collection interval that the agent was in the available state for split/skill or direct agent ACD calls in any split/skill. **TI_AVAILTIME** is recorded for the split/skill that was the **OLDEST_LOGON**. For a non-EAS operation, if an agent logged into multiple splits and is in AUX mode in one split and is available for ACD calls in another split, the agent will accrue **I_AVAILTIME** for the split in which the agent is available and **TI_AVAILTIME** in the split logged into the longest. **Note:** "TI_" time is only stored for the split/skill the agent has been logged into the longest. "TI_" time needs to be summed across the splits/skills the agents may log in to, in case the logon order changes during the collection interval.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
TI_OTHERTIME	The amount of time during the collection interval that the agent was doing other work in all splits/skills. For all switches, **TI_OTHERTIME** is collected for the time period after the link to the switch comes up or after the agent logs in and before CMS receives notification of the agent's state from the switch. For Generic 3 switches, other work includes: while in AUTO-IN or MANUAL-IN mode, the agent put any call on hold and performed no further action, the agent dialed to place a call or to activate a feature, or an extension call rang with no other activity. For Generic 2.2 switches, the agent pushed the Hold button or flashed the switchhook from auto-in or manual-in mode and performed no further action. **Note:** "TI_" time is only stored for the split/skill the agent has been logged into the longest. "TI_" time needs to be summed across the splits/skills the agents may log in to, in case the logon order changes during the collection interval. **TI_OTHERTIME** includes **I_ACDOTHERTIME**.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table

Database Item	Description	Data Type	Column Type	Length
TI_STAFFTIME	The amount of time during the collection interval that the agent was staffed in any split/skill. “TI_” time is only stored for the split/skill logged into the longest. “TI_” time needs to be summed across the splits/skills the agents may log in to, in case the login order changes during the collection interval. **sum(TI_STAFFTIME) = sum(I_ACDTIME + I_ACWTIME + I_DA_ACDTIME + I_DA_ACWTIME + I_RINGTIME + TI_AUXTIME + TI_AVAILTIME + TI_OTHERTIME),** over all splits/skills the agent was logged into.	C	integer; smallint in hagent table	4 bytes; 2 bytes in hagent table
TRANSFERRED	The number of calls the agent transferred to another destination. Note that **TRANSFERRED** calls include both inbound and outbound calls. Therefore, **OTHERCALLS** and **O_OTHERCALLS** may each include some **SHORTCALLS**. For Generic 2.1 switches, includes transfers to measured VDNs or splits/skills. For Generic 2.2 and Generic 3 switches, this includes transferring all calls.	C	smallint (dagent, hagent); integer (magent, wagent)	2 bytes (dagent, hagent); 4 bytes (magent, wagent)

ECS cross-reference

The following table lists which ECS releases support each Agent database item.

The following is a key to the database items tables:

- Items marked “X” indicate that the database item is supported by the specified ECS release.
- Items marked “EAS” require that the EAS feature be active on the ECS for the items to be populated.

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ABNCALLS	X	X	X	X	X	X	X
ABNTIME	X	X	X	X	X	X	X
ACD	X	X	X	X	X	X	X
ACD_RELEASE	X	X	X	X	X	X	X
ACDAUXOUTCALLS	X	X	X	X	X	X	X
ACDCALLS	X	X	X	X	X	X	X
ACDCALLS_R1						X	X
ACDCALLS_R2						X	X
ACDTIME	X	X	X	X	X	X	X
ACWINCALLS	X	X	X	X	X	X	X
ACWINTIME	X	X	X	X	X	X	X
ACWOUTADJCALLS	X	X	X	X	X	X	X
ACWOUTCALLS	X	X	X	X	X	X	X
ACWOUTOFFCALLS	X	X	X	X	X	X	X
ACWOUTOFFTIME	X	X	X	X	X	X	X
ACWOUTTIME	X	X	X	X	X	X	X
ACWTIME	X	X	X	X	X	X	X
ANSRINGTIME	X	X	X	X	X	X	X
ASSISTS	X	X	X	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
AUXINCALLS	X	X	X	X	X	X	X
AUXINTIME	X	X	X	X	X	X	X
AUXOUTADJCALLS	X	X	X	X	X	X	X
AUXOUTCALLS	X	X	X	X	X	X	X
AUXOUTOFFCALLS	X	X	X	X	X	X	X
AUXOUTOFFTIME	X	X	X	X	X	X	X
AUXOUTTIME	X	X	X	X	X	X	X
CONFERENCE	X	X	X	X	X	X	X
DA_ABNCALLS	X	X	X	X	X	X	X
DA_ABNTIME	X	X	X	X	X	X	X
DA_ACDCALLS	X	X	X	X	X	X	X
DA_ACDTIME	X	X	X	X	X	X	X
DA_ACWINCALLS	X	X	X	X	X	X	X
DA_ACWINTIME	X	X	X	X	X	X	X
DA_ACWOADJCALLS	X	X	X	X	X	X	X
DA_ACWOCALLS	X	X	X	X	X	X	X
DA_ACWOOFFCALLS	X	X	X	X	X	X	X
DA_ACWOOFFTIME	X	X	X	X	X	X	X
DA_ACWOTIME	X	X	X	X	X	X	X
DA_ACWTIME	X	X	X	X	X	X	X
DA_ANSTIME	X	X	X	X	X	X	X
DA_OTHERCALLS	X	X	X	X	X	X	X
DA_OTHERTIME	X	X	X	X	X	X	X
DA_RELEASE	X	X	X				
EVENT1-9	X	X	X	X	X	X	X
EXTENSION	X	X	X	X	X	X	X
HOLDABNCALLS	X	X	X	X	X	X	X
HOLDACDTIME	X	X	X	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
HOLDCALLS	X	X	X	X	X	X	X
HOLDTIME	X	X	X	X	X	X	X
I_ACDAUXINTIME	X	X	X	X	X	X	X
I_ACDAUX_OUTTIME	X	X	X	X	X	X	X
I_ACDOTHERTIME	X	X	X	X	X	X	X
I_ACDTIME	X	X	X	X	X	X	X
I_ACWINTIME	X	X	X	X	X	X	X
I_ACWOUTTIME	X	X	X	X	X	X	X
I_ACWTIME	X	X	X	X	X	X	X
I_AUXINTIME	X	X	X	X	X	X	X
I_AUXOUTTIME	X	X	X	X	X	X	X
I_AUXTIME	X	X	X	X	X	X	X
I_AVAILTIME	X	X	X	X	X	X	X
I_DA_ACDTIME	X	X	X	X	X	X	X
I_DA_ACWTIME	X	X	X	X	X	X	X
I_OTHERSTBYTIME							X
I_OTHERTIME	X	X	X	X	X	X	X
I_RINGTIME	X	X	X	X	X	X	X
I_STAFFTIME	X	X	X	X	X	X	X
INCOMPLETE	X	X	X	X	X	X	X
INTRVL	X	X	X	X	X	X	X
LOC_ID					X	X	X
LOGID	X	X	X	X	X	X	X
NOANSREDIR	X	X	X	X	X	X	X
O_ACDCALLS	X	X	X	X	X	X	X
O_ACDTIME	X	X	X	X	X	X	X
O_ACWTIME	X	X	X	X	X	X	X
PHANTOMABNS	X	X	X	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
RINGCALLS	X	X	X	X	X	X	X
RINGTIME	X	X	X	X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
RSV_LEVEL							X
SPLIT	X	X	X	X	X	X	X
STARTTIME	X	X	X	X	X	X	X
TI_AUXTIME	X	X	X	X	X	X	X
TI_AUXTIME0	X	X	X	X	X	X	X
TI_AUXTIME1-9			EAS	X	X	X	X
TI_AVAILTIME	X	X	X	X	X	X	X
TI_OTHERTIME	X	X	X	X	X	X	X
TI_STAFFTIME	X	X	X	X	X	X	X
TRANSFERRED	X	X	X	X	X	X	X

Trunk group database items

Overview

The Trunk Group database item descriptions apply to historical items.

The **Data Type** column refers to **Cumulative (C)**, **Administrative (A)**, **Status (S)**, **Row Identifier (I), Busy Hour (B), Special Table (N),** or **Maximum Value (M)** data. Cumulative, Administrative, Row Identifier, Busy Hour, and Maximum Value items apply to both the current and previous interval real-time tables. Special Table items are historical, and apply only to the table in which they are stored.

Historical trunk group database items apply to the Intrahour Trunk Group (htkgrp), Daily Trunk Group (dtkgrp), Weekly Trunk Group (`wtkgrp`), and Monthly Trunk Group (`mtkgrp`) tables. All items listed in the following table are included in all four tables, unless otherwise noted in the Column Type column. Any differences in the data format between the four trunk group tables are also noted in the Column Type column. The historical indexes are **ROW_DATE** and **TKGRP**.

Contents

“Trunk group database items” contains the following topics:

Trunk group database item table

The following table describes the data items in the CMS Trunk Group database tables.

Database Item	Description	Data Type	Column Type	Length
ABNCALLS	The number of calls carried by this trunk that were abandoned by the caller before being answered by an agent. Calls directly to unmeasured stations that did not go through a measured VDN or split/skill are not recorded. For Generic 2.2 and Generic 3 switches, **ABNCALLS** includes all calls abandoned by the caller that were carried by this trunk, except for calls directly to unmeasured stations that did not go through a measured VDN or split/skill. This includes ACD calls and calls that routed to an agent or extension with talk times less than the phantom-abandoned call timer value. For Generic 2.1 switches, this is ACD calls that abandon from the split queue or from ringing, and calls that abandon from vector processing. Calls that abandon while listening to a forced disconnect are also included in **ABNCALLS**. For Generic 3 Version 1 switches, **ABNCALLS** includes all calls abandoned by the caller that were carried by this trunk, except for calls that direct to unmeasured stations that did not go through a VDN or split/skill. It also includes calls that abandon while listening to a forced disconnect. **ABNCALLS** includes **ABNVECCALLS**, **ABNQUEUECALLS**, and **ABNRINGCALLS**.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
ABNQUECALLS	The number of **ABNCALLS** that abandoned while in a split/skill or direct agent ACD queue. Available on Generic 2 and Generic 3 switches.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table

Database Item	Description	Data Type	Column Type	Length
ABNRINGCALLS	The number of split/skill or direct agent **ABNCALLS** that abandoned by the caller while ringing at an agent position. Available on Generic 2 and Generic 3 switches.	C	integer	4 bytes
ABNVECCALLS	The number of **ABNCALLS** that abandoned while in vector processing. This includes vector calls that abandoned while in queue or while ringing at an agent position. Available on Generic 2 and Generic 3 switches with vectoring. **ABNVECCALLS** includes **ABNQUECALLS** and **ABNRINGCALLS**.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
ACD (index)	The ACD number for which data was collected	I	smallint	2 bytes
ACDCALLS	The number of **INCALLS** that were answered by an agent as a split/skill or direct agent ACD call. **ACDCALLS** includes **BACKUPCALLS**.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
ACDCALLS_R1	The number of **INCALLS** that were answered by a reserve1 agent as a split/skill ACD call. Found only in CMS R3V9.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
ACDCALLS_R2	The number of **INCALLS** that were answered by a reserve2 agent as a split/skill ACD call. Found only in CMS R3V9.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
ALLINUSETIME	The length of time during the interval that all trunks in the trunk group are in use (on calls or maintenance busy).	C	integer	4 bytes
AUDIO	The number of calls for which audio difficulty problems were reported for a trunk or for trunks in this trunk group. Available on Generic 2 and Generic 3 switches.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table

Database Item	Description	Data Type	Column Type	Length
BACKUPCALLS	The number of ACDCALLS that were delivered to and answered by this split/skill by a vector command other than "queue to main" and the number of ACDCALLS that were delivered to a split/skill by a “queue to” vector command answered by an agent that has neither reserve1 or reserve2 skill levels assigned for that skill. This allows tracking of calls answered by agents with a reserve1 or reserve2 skill level assigned for a particular skill. This includes calls delivered by "messaging split/skill", "check backup", and "route to split/skill" vector commands, direct agent calls, and redirect on no answer routing. Calls answered in a main split/skill can be calculated as ACDCALLS - BACKUPCALLS. Available on Generic 3 switches with vectoring. **Note:** The Redirect on No Answer to VDN routing feature is available on the DEFINITY ECS.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_ABNCALLS	The number of incoming calls carried by the trunk group that abandoned during the busy hour.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_ACDCALLS	The number of incoming calls carried by this trunk group during the busy hour that were answered by an agent as split/skill or direct agent ACD calls.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_ALLINUSETIME	The length of time during the busy hour that all trunks in the trunk group were in use.	B	integer	4 bytes
BH_BUSYCALLS	The number of incoming calls carried by the trunk group during the busy hour that were given a busy signal by the switch.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table

Database Item	Description	Data Type	Column Type	Length
BH_DISCCALLS	The number of incoming calls carried by the trunk group during the busy hour that were forced to disconnect by the switch.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_INCALLS	The number of incoming calls carried by this trunk group that completed during the busy hour. **BH_INCALLS** includes **BH_ABNCALLS**, **BH_ACDCALLS**, and **BH_OTHERCALLS**.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_INTERVAL	The defined interval length, with the start time, of the busy hour.	B	integer; only in dtkgrp table	4 bytes
BH_INTIME	The trunk holding time of all incoming calls carried by this trunk group that completed during the busy hour.	B	integer	4 bytes
BH_OABNCALLS	The number of outgoing adjunct-originated calls carried by the trunk group that abandoned during the busy hour. Available on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_OACDCALLS	The number of outgoing adjunct-originated ACD calls carried by the trunk group and answered by an agent as split/skill or direct agent ACD calls that completed during the busy hour. Available on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_OOTHERCALLS	The number of outgoing calls carried by the trunk group during the busy hour that were not answered or abandoned as ACD calls. **BH_OOTHERCALLS** include extension out calls, outbound call management calls forced busy or forced disconnect, short outgoing calls, and outgoing calls with unknown disposition.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table

Database Item	Description	Data Type	Column Type	Length
BH_OTHERCALLS	The number of incoming calls carried by the trunk group during the busy hour that were not answered or abandoned. **BH_OTHERCALLS** include extension in calls, calls forced busy or disconnected, calls that outflowed off the switch, short inbound calls, and inbound calls of unknown disposition. **BH_OTHERCALLS** includes **BH_BUSYCALLS** and **BH_DISCCALLS**.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_OUTCALLS	The number of outgoing calls carried by the trunk group that completed during the busy hour. **BH_OUTCALLS** includes **BH_OABNCALLS**, **BH_OACDCALLS**, and **BH_OOTHERCALLS**.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_OUTTIME	The trunk holding time of all outgoing calls carried by the trunk group that completed during the busy hour.	B	integer	4 bytes
BH_STARTTIME	The starting time of the hour for which busy hour data was collected. The busy hour is that set of contiguous intervals during the day totaling an hour in which the trunk holding time for the trunk group was a maximum.	B	integer	4 bytes
BLOCKAGE	The number of outbound call attempts that were blocked because all trunks were busy. Available on Generic 2 switches.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table

Database Item	Description	Data Type	Column Type	Length
BUSYCALLS	The number of **INCALLS** that were given a busy signal by the switch. This can occur on all switches via the “busy" vector command. On Generic 3 switches without vectoring**,** **BUSYCALLS** can occur if a call is routed to a split/skill with coverage set to “yes” where there are no agents available, the queue is full (or there is no queue), there is no coverage, and an announcement has played or the trunk is not a CO trunk. Also on Generic 3 switches, **BUSYCALLS** can occur if a call is routed to a direct agent with coverage set to “yes”, the agent is not logged in and there is no coverage path administered, and an announcement has played or the trunk is not a CO trunk. **BUSYCALLS** can occur on Generic 3 switches without vectoring when a split queue is full or there are no queue slots, no busy coverage is administered and an announcement has played or the trunk is not a CO trunk.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
COMPLETED	The number of **OUTCALLS** that were completed (far end answered). Available on Generic 3 switches.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
CONNECTCALLS	The number of **INCALLS** that were answered at a station and were not split/skill or direct agent ACD calls.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
DISCCALLS	With Generic 2.1, Generic 3 Version 1 and Generic 3 Version 2 (prior to load 100) switches, **DISCCALLS** is the number of **INCALLS** that were given a forced disconnect announcement by the “disconnect” vector command, listened to the entire announcement, then were disconnected by the switch. With Generic 2.2 switches, this is the number of **INCALLS** that were disconnected by the switch by the “disconnect" vector command. With Generic 3 Version 2 and later Generic 3 switches, this is the number of **INCALLS** that were disconnected by the switch by the “disconnect” vector command. **DISCCALLS** also includes calls that were disconnected by the switch when the vector disconnect timer expired or that reached the end of vector processing without being queued. **DISCCALLS** includes **VDISCCALLS**.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
FAILURES	The number of trunk failures for this **TKGRP**. No time or call is recorded in any of the CMS tables. Trunk failures can be due to hardware problems on the trunk, incompatible trunk types on either end of a call, or internal switch errors (such as errors in call processing or vectoring translations). This item does not include calls with short holding times. Available on Generic 2 switches. The **FAILURES** database item is not populated for Generic 3 switches because trunks that fail are automatically placed in the maintenance busy state.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table

Database Item	Description	Data Type	Column Type	Length
I_INOCC	The total time during the collection interval that ALL trunks in the trunk group were occupied by incoming calls. If an incoming call on a measured trunk is transferred off the switch, the incoming trunk remains in use for the call and accrues trunk holding time until the caller drops or the call is released.	C	integer	4 bytes
I_OUTOCC	The amount of time during the collection interval that trunks in this trunk group were occupied by outgoing calls.	C	integer	4 bytes
INCALLS	The number of inbound calls that were carried by this **TKGRP** and that completed during the collection interval. **INCALLS** includes **ABNCALLS**, **ACDCALLS**, **CONNECTCALLS**, and **TRANSFERRED.** **INCALLS** = **ACDCALLS** + **ABNCALLS + OTHERCALLS.**	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
INCOMPLETE	This item indicates if data is complete for this collection interval. Data is incomplete whenever the link goes down and whenever tracking is aborted for calls, due to trunk failures (Generic 2), the trunk going maintenance busy with a call active (Generic 3), protocol failures with data collection active, or when split/skill or VDN call profile is changed with data collection active. The value for interval tables indicates whether data is incomplete for the interval (0 = N0, 1 = YES). The value in the daily, weekly, and monthly tables indicates the number of incomplete intervals in the day, week, or month.	C	smallint	2 bytes

Database Item	Description	Data Type	Column Type	Length
INTIME	The trunk holding time for all **INCALLS** carried by trunks in this trunk group that completed during the collection interval. Trunk holding time is the time from the initial trunk seizure until the trunk goes idle (that is, until the caller drops, the agent releases the call, or the switch disconnects the call). If an incoming call on a measured trunk is transferred off the switch, the incoming trunk remains in use for the call and accrues trunk holding time until the caller drops or the call is released.	C	integer	4 bytes
INTRVL	The number of minutes in the intrahour interval (15, 30, or 60). **INTRVL** applies to intrahour tables only.	A	smallint; only in htkgrp table	2 bytes
MBUSYTIME	The total time (in seconds) during the collection interval that trunks in the trunk group were maintenance busy.	C	integer	4 bytes
O_ABNCALLS	The number of **OUTCALLS** on this trunk group that were offered by an adjunct as split/skill or direct agent ACD calls and were answered then abandoned by the far end. Available on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
O_ACDCALLS	The number of **OUTCALLS** from this trunk group that were offered by an adjunct to one or more splits/skills and were answered by an agent. Available on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table

Database Item	Description	Data Type	Column Type	Length
O_OTHERCALLS	The number of **OUTCALLS** on this trunk group that were not answered or abandoned as ACD split/skill calls. These include extension out calls, calls forced busy and forced disconnected, short outgoing calls, and calls with unknown dispositions. **O_OTHERCALLS** includes **SHORTCALLS**.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
OTHERCALLS	The number of **INCALLS** carried by this trunk group that were not answered as split/skill or direct agent ACD calls or abandoned. These include forced busy calls, forced disconnect calls, calls that were connected to a non-ACD destination, short inbound calls, calls that outflowed off the switch, and calls with unknown dispositions. **OTHERCALLS** includes **BUSYCALLS**, **DISCCALLS**, **SHORTCALLS, and CONNECTCALLS**. **OTHERCALLS** = **INCALLS** - **ACDCALLS** - **ABNCALLS**	C	Integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
OUTCALLS	The number of outbound calls that were carried by this **TKGRP** and were completed during the collection interval. **OUTCALLS** includes **COMPLETED**, **O_ABNCALLS**, **O_ACDCALLS**, **O_OTHERCALLS**, **TRANSFERRED, and SHORTCALLS.** **OUTCALLS** = **O_ACDCALLS** + **O_ABNCALLS** + **O_OTHERCALLS.**	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table

Database Item	Description	Data Type	Column Type	Length
OUTTIME	The trunk holding time for all **OUTCALLS** carried by trunks in this trunk group that completed during the collection interval. Trunk holding time is the time from the initial trunk seizure until the trunk goes idle (that is, until the far end drops, the agent releases the call, or the switch disconnects the call). **OUTTIME** includes **SETUPTIME**.	C	integer	4 bytes
ROW_DATE (index)	The day for which data was collected or the exception occurred.	I	date	4 byte Informix date
SETUPTIME	The amount of time from trunk seizure until **OUTCALLS** completed at the far end. Available on Generic 3 switches.	C	integer	4 bytes
SHORTCALLS	The number of inbound and outbound calls that occupied a trunk in the trunk group for less than 2 seconds and that did not queue to a split/skill, forward to a split/skill, get answered by an agent, get a forced busy or forced disconnect from the switch, or produce a trunk failure or maintenance busy. Note that **SHORTCALLS** includes both inbound and outbound calls. Therefore, **OTHERCALLS** and **O_OTHERCALLS** may each include some **SHORTCALLS**.	C	integer	4 bytes
SPLIT	The split/skill to which this **TKGRP** terminates.	A	smallint	2 bytes
TKGRP (index)	The trunk group number for which data was collected. This will be zero if the trunk group carrying the call is not measured.	I	smallint	2 bytes

Database Item	Description	Data Type	Column Type	Length
TRANSFERRED	The number of calls that the agent transferred to another destination. Note that **TRANSFERRED** includes both inbound and outbound calls. Therefore, **OTHERCALLS** and **O_OTHERCALLS** may each include some **TRANSFERRED**. For Generic 2.1 switches, **TRANSFERRED** includes transfers to measured VDNs or splits/skills. For Generic 3 and Generic 2.2 switches, **TRANSFERRED** includes all calls that transferred.	C	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
TRUNKS	The current number of trunks assigned to this **TKGRP**.	A	smallint	2 bytes
VDN	The VDN to which the **TKGRP** terminates. Available on Generic 2 and Generic 3 switches with vectoring.	A	char(6)	6 byte ASCII text string
VECTOR	The vector to which this trunk group's **VDN** terminates. Available on Generic 2 and Generic 3 switches with vectoring.	A	smallint	2 bytes

ECS cross-reference

The following table lists which of the ECS releases support each of the Trunk Group database items.

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ABNCALLS	X	X	X	X	X	X	X
ABNQUECALLS	X	X	X	X	X	X	X
ABNRINGCALLS	X	X	X	X	X	X	X
ABNVECCALLS	X	X	X	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ACD (index)	X	X	X	X	X	X	X
ACDCALLS	X	X	X	X	X	X	X
ACDCALLS_R1						X	X
ACDCALLS_R2						X	X
ALLINUSETIME	X	X	X	X	X	X	X
AUDIO	X	X	X	X	X	X	X
BH_ABNCALLS	X	X	X				
BH_ACDCALLS	X	X	X				
BH_ALLINUSETIME	X	X	X				
BH_BUSYCALLS	X	X	X				
BH_DISCCALLS	X	X	X				
BH_INCALLS	X	X	X				
BH_INTERVAL	X	X	X				
BH_INTIME	X	X	X				
BH_OABNCALLS	X	X	X				
BH_OACDCALLS	X	X	X				
BH_OOTHERCALLS	X	X	X				
BH_OTHERCALLS	X	X	X				
BH_OUTCALLS	X	X	X				
BH_OUTTIME	X	X	X				
BH_STARTTIME	X	X	X				
BACKUPCALLS	X	X	X	X	X	X	X
BLOCKAGE				X	X	X	X
BUSYCALLS	X	X	X	X	X	X	X
COMPLETED	X	X	X	X	X	X	X
CONNECTCALLS	X	X	X	X	X	X	X
DISCCALLS	X	X	X	X	X	X	X
FAILURES				X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
I_INOCC	X	X	X	X	X	X	X
I_OUTOCC	X	X	X	X	X	X	X
INCALLS	X	X	X	X	X	X	X
INCOMPLETE	X	X	X	X	X	X	X
INTIME	X	X	X	X	X	X	X
INTRVL	X	X	X	X	X	X	X
MBUSYTIME	X	X	X	X	X	X	X
O_ABNCALLS	X	X	X	X	X	X	X
O_ACDCALLS	X	X	X	X	X	X	X
O_OTHERCALLS	X	X	X	X	X	X	X
OUTCALLS	X	X	X	X	X	X	X
OTHERCALLS	X	X	X	X	X	X	X
OUTTIME	X	X	X	X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
SETUPTIME	X	X	X	X	X	X	X
SHORTCALLS	X	X	X	X	X	X	X
SPLIT	X	X	X	X	X	X	X
TKGRP	X	X	X	X	X	X	X
TRANSFERRED	X	X	X	X	X	X	X
TRUNKS	X	X	X	X	X	X	X
VDN	X	X	X	X	X	X	X
VECTOR	X	X	X	X	X	X	

Trunk database items

Overview

The Trunk database item descriptions apply to historical items.

The **Data Type** column refers to **Cumulative (C)**, **Administrative (A)**, **Status (S)**, **Row Identifier (I), Busy Hour (B), Special Table (N),** or **Maximum Value (M)** data. Cumulative, Administrative, Row Identifier, Busy Hour, and Maximum Value items typically apply to both the current and previous interval real-time tables. Special Table items are historical, and apply only to the table in which they are stored.

Historical trunk database items apply to the Intrahour Trunk (`htrunk`), Daily Trunk (`dtrunk`), Weekly Trunk Group (`wtrunk`), and Monthly Trunk (`mtrunk`) tables. All items listed in the following Trunk Database Items table are included in all four tables, unless otherwise noted in the Column Type column. Any differences in the data format between the four trunk tables are also noted in the Column Type column. The historical indexes are **EQLOC**, **ROW_DATE**, and **TKGRP**.

Contents

“Trunk database items” contains the following topics:

Trunk database item table

The following table describes the data items in the CMS Trunk database tables.

Database Item	Description	Data Type	Column Type	Length
ABNCALLS	The number of calls carried by this trunk that were abandoned by the caller before being answered by an agent. Calls directed to unmeasured stations that did not go through a measured VDN or split/skill are not recorded. For Generic 2.2 and Generic 3 switches, **ABNCALLS** includes all calls abandoned by the caller that were carried by this trunk, except for calls directed to unmeasured stations that did not go through a measured VDN or split/skill. This includes ACD calls and calls that routed to an agent or extension with talk times less than the phantom-abandoned call timer value. For Generic 2.1 switches, this is ACD calls that abandon from the split queue or from ringing, and calls that abandon from vector processing. Calls that abandon while listening to a forced disconnect are also included in **ABNCALLS**. For Generic 3 Version 1 switches, **ABNCALLS** includes all calls abandoned by the caller that were carried by this trunk, except for calls that directed to unmeasured stations that did not go through a VDN or split/skill. It also includes calls that abandon while listening to a forced disconnect. **ABNCALLS** includes **ABNVECCALLS**, **ABNQUEUECALLS**, and **ABNRINGCALLS**.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
ACD (index)	The ACD number for which data was collected.	I	smallint	2 bytes
ACDCALLS	The number of **INCALLS** that were answered by an agent as a split/skill or direct agent ACD call.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table

Database Item	Description	Data Type	Column Type	Length
ACDCALLS_R1	The number of **INCALLS** that were answered by a reserve1 agent as a split/skill ACD call. Found only in CMS R3V9.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
ACDCALLS_R2	The number of **INCALLS** that were answered by a reserve2 agent as a split/skill ACD call. Found only in CMS R3V9.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
AUDIO	The number of calls for which audio difficulty problems were reported for this trunk. Available on Generic 2 and Generic 3 switches.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
EQLOC (index)	The physical equipment location (trunk number) for which data was collected. For the DEFINITY ECS R8, this field is eight characters. For previous switch releases, it is nine characters.	A	char(8)	8 byte ASCII text string
FAILURES	The number of trunk failures for this trunk. No time or call is recorded in any of the CMS tables. Trunk failures can be due to hardware problems on the trunk, incompatible trunk types on either end of a call, or to internal switch errors (such as errors in call processing or vectoring translations). This item does not include calls with short holding times. Available on Generic 2 switches. The FAILURES database item is not populated for Generic 3 switches because trunks that fail are automatically placed in the maintenance busy state.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
I_INOCC	The total time during the collection interval that the trunk was occupied by incoming calls. If an incoming call on a measured trunk is transferred off the switch, the incoming trunk remains in use for the call and accrues trunk holding time until the caller drops or the call is released.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table

Database Item	Description	Data Type	Column Type	Length
I_OUTOCC	The total time during the collection interval that this trunk was occupied by outbound calls.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
INCALLS	The number of inbound calls carried by this trunk that completed during the collection interval. This includes calls with short holding times (**SHORTCALLS)** but does not include calls that had a trunk failure (**FAILURES)**. **INCALLS** = **ABNCALLS** + **ACDCALLS** + **OTHERCALLS.**	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
INCOMPLETE	This item indicates if data is complete for this collection interval. Data is incomplete whenever the link goes down and whenever tracking is aborted for calls, due to trunk failures (Generic 2), the trunk going maintenance busy with a call active (Generic 3), protocol failures with data collection active, or when split/skill or VDN call profile is changed with data collection active. The value for interval tables indicates whether data is incomplete for the interval (0 = N0, 1 = YES). The value in the daily, weekly, and monthly tables indicates the number of incomplete intervals in the day, week, or month. Changing split/skill or VDN call profile data while data collection is active only affects the respective split/skill or VDN data.	C	smallint	2 bytes
INTIME	The trunk holding time for all **INCALLS** carried by this trunk that completed during the collection interval. Trunk holding time is the time from the initial trunk seizure until the trunk goes idle (that is, until the caller drops, the agent releases the call, or the switch disconnects the call). If an incoming call on a measured trunk is transferred off the switch, the incoming trunk remains in use for the call and accrues trunk holding time until the caller drops or the call is released.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table

Database Item	Description	Data Type	Column Type	Length
INTRVL	The number of minutes in the intrahour interval (15, 30, or 60). **INTRVL** applies to intrahour intervals only.	A	smallint; only in htrunk table	2 bytes
LOC_ID	The switch location ID that is associated with a trunk. A location ID is not directly assigned to a trunk; instead, it is assigned to a port network (on the 'ch cabinet x' form), and therefore, each trunk whose equipment location belongs to that port network will be associated with that port network's location ID. Valid values are 01-44.	C	integer	4 bytes
MBUSYTIME	The total time during the collection interval that this trunk was maintenance busy.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
O_ABNCALLS	The number of **OUTCALLS** on this trunk that were offered by an adjunct as split/skill or direct agent ACD calls and were answered then abandoned by the far end before talking to an agent. Available on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
O_ACDCALLS	The number of **OUTCALLS** from this trunk that were offered by an adjunct as split/skill or direct agent ACD calls and were answered by an agent. Available on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
O_OTHERCALLS	The number of **OUTCALLS** on this trunk that were not answered as ACD split/skill calls or abandoned. These include extension out calls, forced busy calls, short outgoing calls, and calls with unknown dispositions. **O_OTHERCALLS** includes **SHORTCALLS**.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table

Database Item	Description	Data Type	Column Type	Length
OUTCALLS	The number of outbound calls that were carried by the trunk and were completed during the collection interval. **OUTCALLS** = **O_ACDCALLS** + **O_ABNCALLS** + **O_OTHERCALLS.**	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
OTHERCALLS	The number of **INCALLS** on this trunk that were not answered or abandoned as split/skill or direct agent **ACD** calls. These include forced busy calls, forced disconnect calls, calls that outflowed off the switch, calls that were connected to a non-**ACD** destination, short inbound calls, and calls with unknown dispositions.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
OUTTIME	The trunk holding time for all **OUTCALLS** carried by this trunk that completed during the collection interval. Trunk holding time is the time from the initial trunk seizure until the trunk goes idle (that is, until the far end drops, the agent releases the call, or the ECS disconnects the call). The format is the number of seconds that have accumulated.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
ROW_DATE (index)	The day for which data was collected or the exception occurred.	I	date	4 byte Informix date
SHORTCALLS	The number of inbound and outbound calls that occupied a trunk for less than 2 seconds and that did not queue to a split/skill, forward to a split/skill, get answered by an agent, get a forced busy or forced disconnect from the switch, or produce a trunk failure or maintenance busy. Note that **SHORTCALLS** includes both inbound and outbound calls. Therefore, **OTHERCALLS** and **O_OTHERCALLS** may each include some **SHORTCALLS**.	C	integer; smallint in htrunk table	4 bytes; 2 bytes in htrunk table
TKGRP (index)	The trunk group number to which the trunk is assigned.	A	smallint	2 bytes

ECS cross-reference

The following table lists which of the ECS releases support each Trunk database item

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ABNCALLS	X	X	X	X	X	X	X
ACD	X	X	X	X	X	X	X
ACDCALLS	X	X	X	X	X	X	X
ACDCALLS_R1						X	X
ACDCALLS_R2						X	X
AUDIO	X	X	X	X	X	X	X
EQLOC	X	X	X	X	X	X	X
FAILURES				X	X	X	X
I_INOCC	X	X	X	X	X	X	X
I_OUTOCC	X	X	X	X	X	X	X
INCALLS	X	X	X	X	X	X	X
INCOMPLETE	X	X	X	X	X	X	X
INTIME	X	X	X	X	X	X	X
INTRVL	X	X	X	X	X	X	X
ITN	X	X	X				
MBUSYTIME	X	X	X	X	X	X	X
O_ABNCALLS	X	X	X	X	X	X	X
O_ACDCALLS	X	X	X	X	X	X	X
O_OTHERCALLS	X	X	X	X	X	X	X
OUTCALLS	X	X	X	X	X	X	X
OTHERCALLS	X	X	X	X	X	X	X
OUTTIME	X	X	X	X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
SHORTCALLS	X	X	X	X	X	X	X
STARTTIME	X	X	X	X	X	X	X
TKGRP	X	X	X	X	X	X	X

Vector database items

Overview

The Vector database item descriptions apply to historical items.

Note:

Vector database items are available only if the Vectoring feature has been purchased and authorized for you to use.

The **Data Type** column refers to **Cumulative (C)**, **Administrative (A)**, **Status (S)**, **Row Identifier (I), Busy Hour (B), Special Table (N),** or **Maximum Value (M)** data. Cumulative, Administrative, Row Identifier, Busy Hour, and Maximum Value items apply to both the current and previous interval real-time tables. Special Table items are historical, and apply only to the table in which they are stored.

Historical vector database items apply to the Intrahour Vector (`hvector`), Daily Vector (`dvector`), Weekly Vector (`wvector`), and Monthly Vector (`mvector`) tables. All items listed in the following Vector Database Items table are included in all four tables, unless otherwise noted in the Column Type column. Any differences in the data format between the four vector tables are also noted in the Column Type column. The historical indexes are **ROW_DATE** and **VECTOR**.

Contents

“Vector database items” contains the following topics:

Vector database item table

The following table describes the data items in the CMS Vector database tables.

Database Item	Description	Data Type	Column Type	Length
ABNCALLS	The number of **INCALLS** that were abandoned while **INPROGRESS** for this vector. This includes split/skill and direct agent ACD calls that abandon from queue or from ringing, calls that abandon from vector processing, and for the Generic 2.1 and Generic 3 Version 1 switches, calls that abandoned while listening to a forced disconnect announcement. **ABNCALLS** includes **ABNQUECALLS**, **ABNRINGCALLS**, and **PHANTOMABNS**.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
ABNQUECALLS	The number of **ABNCALLS** that hung up while in a split/skill or direct agent ACD queue. Available on Generic 2 and Generic 3 switches.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
ABNRINGCALLS	The number of split/skill or direct agent **ABNCALLS** that were abandoned while ringing at an agent position. Available on Generic 2 and Generic 3 switches and on the ECS.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
ABNTIME	The amount of time caller waited while vector steps were executed, the call was queued, and ringing, before abandoning. For phantom abandons, **ABNTIME** includes the total time until the agent releases the call.	C	integer	4 bytes
ACD (index)	The ACD number for which data was collected.	I	smallint	2 bytes

Database Item	Description	Data Type	Column Type	Length
ACDCALLS	The number of split/skill and direct agent ACD calls that were answered by an agent from “queue to main, “check backup”, “messaging split/skill”, “route to” split/skill or direct agent, and “adjunct routing” to a split/skill or direct agent. **ACDCALLS** includes **BACKUPCALLS**.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
ACDCALLS_R1	The number of split/skill and direct agent ACD calls that were answered by a reserve1 agent from “queue to skill or best, “check skill or best”, “messaging split/skill”, “route to” split/skill or direct agent, and “adjunct routing link” to a split/skill. Found only in CMS R3V9.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
ACDCALLS_R2	The number of split/skill and direct agent ACD calls that were answered by a reserve2 agent from “queue to skill or best, “check skill or best”, “messaging split/skill”, “route to” split/skill or direct agent, and “adjunct routing link” to a split/skill. Found only in CMS R3V9.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
ADJATTEMPTS	The number of adjunct routing attempts for calls in this **VECTOR**. Available on the ECS and Generic 3 switches with the ASAI feature. **ADJATTEMPTS** includes **ADJROUTED**.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
ADJROUTED	The number of adjunct-routing calls that were redirected by an adjunct processor or host computer. Available on the ECS and Generic 3 switches with vectoring and the ASAI feature.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
ANSTIME	The amount of time that split/skill and direct agent ACD calls waited while executing steps in this vector, queuing, and ringing before being answered by an agent. **ANSTIME** includes **RINGTIME**.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
BACKUPCALLS	The number of ACDCALLS that were delivered to and answered by this split/skill by a vector command other than "queue to main" and the number of ACDCALLS that were delivered to a split/skill by a “queue to” vector command answered by an agent that has neither reserve1 or reserve2 skill levels assigned for that skill. This allows tracking of calls answered by agents with a reserve1 or reserve2 skill level assigned for a particular skill. Calls answered in a main split/skill (MAINCALLS) can then be calculated as ACDCALLS - BACKUPCALLS However, MAINCALLS does not include direct agent calls. BACKUPCALLS includes "messaging split/skill" calls, "check backup" calls, and calls that route to a split/skill or direct agent, either by the "route to" vector command or by adjunct routing. Calls that are redirected back to the split/skill using the redirection on no answer feature and are subsequently answered are also counted as BACKUPCALLS. **Note:** The Redirect to No Answer to VDN routing feature is available on the switch. Available on Generic 2 and Generic 3 switches and on the ECS with the vectoring feature.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table

Database Item	Description	Data Type	Column Type	Length
BUSYCALLS	The number of **INCALLS** that were given a busy signal by the switch. This can occur on all switches when the “busy'' vector command is executed. On Generic 3 switches, **BUSYCALLS** can occur if a call is routed to a split with coverage set to “yes” where there are no agents available, the queue is full (or there is no queue), there is no coverage, and an announcement has played or the trunk is not a Central Office (CO) trunk. Also on Generic 3 switches, **BUSYCALLS** can occur if a call is routed to a direct agent with coverage set to “yes”, the agent is not logged in and there is no coverage path administered and an announcement has played or the trunk is not a CO trunk.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
BUSYTIME	The amount of time callers waited in queue until hearing a busy tone for all **BUSYCALLS**.	C	integer	4 bytes
DEFLECTCALLS	The number of calls deflected to the network through Best Service Routing (BSR). Requires the DEFINITY *ECS R6*.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table

Database Item	Description	Data Type	Column Type	Length
DISCCALLS	With Generic 2.1 and Generic 3 Version 1 switches, **DISCCALLS** is the number of **INCALLS** that were given a forced disconnect announcement by the “disconnect” vector command, listened to the entire announcement, then were disconnected by the switch. With the Generic 2.2 switch, Generic 3 Version 2, and later Generic 3 switches and with the ECS, the number of **INCALLS** that executed the “disconnect" vector command. With Generic 3 Version 2 and newer switches (and with the ECS), **DISCCALLS** also includes calls disconnected by the switch when the vector disconnect timer expired or that reached the end of vector processing without being queued. **DISCCALLS** includes **VDISCCALLS**.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
DISCTIME	The amount of time all **DISCCALLS** spent in this **VECTOR**. The time until the trunk drops following the forced disconnect command for those calls recorded as **DISCCALLS**. For Generic 2.1 and Generic 3 Version 1 switches, this is the time until the announcement ends and the caller is disconnected by the switch. For Generic 2.2, Generic 3 Version 2 and later Generic 3 switches, and for the ECS, this is the time until the trunk drops, in the case where the caller hangs up without listening to the entire announcement. For Generic 3 Version 2 and newer switches, and for the ECS, this is the time when the call is disconnected due to the expiration of the vector disconnect timer or the time until the caller is disconnected by the switch.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
GOTOCALLS	The number of **OUTFLOWCALLS** that were redirected to another vector by way of a “go to vector” command. Available on Generic 2.2 and Generic 3 switches and on the ECS.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
GOTOTIME	The amount of time all **GOTOCALLS** spent in this vector before being redirected to another vector. Available on Generic 2.2 and Generic 3 switches and on the ECS.	C	integer	4 bytes
INCALLS	The number of inbound calls that were processed by this vector. **INCALLS** includes **ABNCALLS**, **RINGCALLS, INFLOWCALLS**, and **OTHERCALLS**. **INCALLS** = **ACDCALLS** + **ABNCALLS** + **OTHERCALLS.**	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
INCOMPLETE	This item indicates if data is complete for this collection interval. Data is incomplete whenever the link goes down and whenever tracking is aborted for calls due to trunk failures (Generic 2), the trunk going maintenance busy with a call active (Generic 3 and the ECS), protocol failures with data collection active, or when split/skill or VDN call profile is changed with data collection active. The value for interval tables indicates whether data is incomplete for the interval (0 = N0, 1 = YES). The value in the daily, weekly, and monthly tables indicates the number of incomplete intervals in the day, week, or month.	C	smallint	2 bytes
INFLOWCALLS	The number of calls that were redirected to this vector by way of a “go to vector” or a “route to” VDN command, or by redirection on no answer to a VDN.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table

Database Item	Description	Data Type	Column Type	Length
INTERFLOWCALLS	The number of **OUTFLOWCALLS** that were directed to an off-switch location. **INTERFLOWCALLS** includes **LOOKFLOWCALLS**.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
INTIME	The amount of time all **DISCCALLS** spent in this **VECTOR**. The time until the trunk drops following the forced disconnect command for those calls recorded as **DISCCALLS**. For Generic 2.1 and Generic 3 Version 1 switches, this is the time until the announcement ends and the caller is disconnected by the switch. For Generic 2.2, Generic 3 Version 2 and later Generic 3 switches, and for the ECS, this is the time until the trunk drops, in the case where the caller hangs up without listening to the entire announcement. For Generic 3 Version 2 and newer switches, and for the ECS, this is the time when the call is disconnected due to the expiration of the vector disconnect timer or the time until the caller is disconnected by the switch.	C	integer	4 bytes
INTRVL	The number of minutes in the timed period (15, 30, or 60). **INTRVL** applies to intrahour tables only.	A	smallint; only in hvdn table	2 bytes
LOOKATTEMPTS	The number of look-ahead interflow attempts for calls in this vector. Available on Generic 2.2 and Generic 3 switches and on the ECS with the vectoring and look-ahead interflow features. **LOOKATTEMPTS** includes **LOOKFLOWCALLS**.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
LOOKFLOWCALLS	The number of **INTERFLOWCALLS** that were redirected by way of the Lookahead Interflow feature. Available on Generic 2.2 and Generic 3 switches with the Lookahead Interflow feature.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table

Database Item	Description	Data Type	Column Type	Length
NETDISCCALLS	The number of disconnected calls for the reply step in BSR. Requires the DEFINITY *ECS R6*.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
NETPOLLS	The number of network polls for the consider steps in BSR. Requires the DEFINITY *ECS R6*.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
OTHERCALLS	The number of **INCALLS** that were redirected out of the vector, given a busy signal, or were disconnected. **OTHERCALLS** includes **BUSYCALLS**, **DISCCALLS**, and **OUTFLOWCALLS**. **OTHERCALLS** = **INCALLS** - **ACDCALLS** - **ABNCALLS**	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
OTHERTIME	The amount of time **OTHERCALLS** spent in the vector until the disposition was known and the call left the vector. **OTHERTIME** includes **BUSYTIME**, **DISCTIME**, and **OUTFLOWTIME**.	C	integer	4 bytes
OUTFLOWCALLS	The number of **INCALLS** that were redirected to another destination by way of a “go to vector'' command or by a ''route to'' or ''adjunct routing'' command to a destination other than a split/skill or direct agent. (Calls that route to a split/skill or direct agent by way of a “route to,'' “adjunct routing,'' or “messaging split/skill'' command are still tracked in the vector.) **OUTFLOWCALLS** includes **GOTOCALLS** and **INTERFLOWCALLS**.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
OUTFLOWTIME	The amount of time all **OUTFLOWCALLS** spent in the **VECTOR** before being redirected. **OUTFLOWTIME** includes **GOTOTIME**.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
PHANTOMABNS	The number of split/skill and direct agent ACD calls and calls that routed to an agent or extension with talk time less than the value of the phantom abandoned call timer. Available on Generic 3 and newer switches and the ECS.	C	integer	4 bytes
RINGCALLS	The number of split/skill and direct agent ACD calls that rang at agent positions. **RINGCALLS** includes **ACDCALLS**. Available on Generic 2 and Generic 3 switches and the ECS.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvector table
RINGTIME	The amount of time split/skill and direct agent ACD that rang at agent positions. Available on Generic 2 and Generic 3 switches and the ECS.	C	integer	4 bytes
ROW_DATE (index)	The date for which data was collected or the exception occurred.	I	date	4 byte Informix date
VDISCCALLS	The number of calls forced to disconnect because the vector disconnect timer timed out or because the call reached a vector stop without being queued. “Vector stop” means an explicit “stop” vector command, the end of the vector, or the call executed 1000 vector steps. Available on Generic 3 Version 2 and newer switches and the ECS.	C	integer; smallint in hvector table	4 bytes; 2 bytes in hvdn table
VECTOR (index)	The vector number that this row represents. Available on Generic 2 and Generic 3 switches and on the ECS with vectoring.	I	smallint	2 bytes

ECS cross-reference

The following table lists which of the ECS releases support the Vector database items.

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ABNCALLS	X	X	X	X	X	X	X
ABNQUECALLS	X	X	X	X	X	X	X
ABNRINGCALLS	X	X	X	X	X	X	X
ABNTIME	X	X	X	X	X	X	X
ACD	X	X	X	X	X	X	X
ACDCALLS	X	X	X	X	X	X	X
ACDCALLS_R1						X	X
ACDCALLS_R2						X	X
ADJATTEMPTS	X	X	X	X	X	X	X
ADJROUTED	X	X	X	X	X	X	X
ANSTIME	X	X	X	X	X	X	X
BACKUPCALLS	X	X	X	X	X	X	X
BUSYCALLS	X	X	X	X	X	X	X
BUSYTIME	X	X	X	X	X	X	X
DEFLECTCALLS				X	X	X	X
DISCCALLS	X	X	X	X	X	X	X
DISCTIME	X	X	X	X	X	X	X
GOTOCALLS	X	X	X	X	X	X	X
GOTOTIME	X	X	X	X	X	X	X
INCALLS	X	X	X	X	X	X	X
INCOMPLETE	X	X	X	X	X	X	X
INFLOWCALLS	X	X	X	X	X	X	X
INTERFLOWCALLS	X	X	X	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
INTIME	X	X	X	X	X	X	X
INTRVL	X	X	X	X	X	X	X
LOOKATTEMPTS	X	X	X	X	X	X	X
LOOKFLOWCALLS	X	X	X	X	X	X	X
NETDISCCALLS				X	X	X	X
NETPOLLS				X	X	X	X
OTHERCALLS	X	X	X	X	X	X	X
OTHERTIME	X	X	X	X	X	X	X
OUTFLOWCALLS	X	X	X	X	X	X	X
OUTFLOWTIME	X	X	X	X	X	X	X
PHANTOMABNS	X	X	X	X	X	X	X
RINGCALLS	X	X	X	X	X	X	X
RINGTIME	X	X	X	X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
STARTTIME							X
VDISCCALLS	X	X	X	X	X	X	X
VECTOR	X	X	X	X	X	X	X

VDN database items

Overview

VDN database items are available only if the Vectoring feature has been purchased and authorized for you to use. The VDN Database Item descriptions apply to historical items.

The **Data Type** column refers to **Cumulative (C)**, **Administrative (A)**, **Status (S)**, **Row Identifier (I), Busy Hour (B), Special Table (N),** or **Maximum Value (M)** data. Cumulative, Administrative, Row Identifier, Busy Hour, and Maximum Value items apply to both the current and previous interval real-time tables. Special Table items are historical, and apply only to the table in which they are stored.

Historical VDN database items apply to the Intrahour VDN (`hvdn`), Daily VDN (`dvdn`), Weekly VDN (`wvdn`), and Monthly VDN (`mvdn`) tables. All items listed in the following VDN Database Items table are included in all four tables, unless otherwise noted in the Column Type column. Any differences in the data format between the four VDN tables are also noted in the Column Type column. The historical indexes are **ROW_DATE** and **VDN**.

Contents

"VDN database items" contains the following topics:

VDN database item table

The following table describes the data items in the CMS VDN database tables.

Database Item	Description	Data Type	Column Type	Length
ABNCALLS	The number of **INCALLS** that were abandoned while **INPROGRESS** for this VDN. This includes split/skill and direct agent ACD calls that abandon from queue or from ringing, calls that abandon from vector processing, calls that abandon after being routed to an extension via the “route to" vector command, and for the Generic 2.1 and Generic 3 (prior to Generic 3 Version 2 load 100) switches, calls that abandoned while listening to a forced disconnect announcement. **ABNCALLS** includes ACD calls and calls routed to an agent or extension with talk times less than the value of the phantom abandoned call timer. **ABNCALLS** includes **ABNCALLS1** through **ABNCALLS10, ABNQUECALLS, ABNRINGCALLS, PHANTOMABNS**, and **SLVLABNS** are pegged as **ABNCALLS**.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
ABNCALLS1-10	The number of **INCALLS** that abandoned in each of the service level increments **PERIOD1** through **PERIOD9** (as defined IN the ACD Administration: VDN Call Profile Setup window). **ABNCALLS10** counts calls that abandoned after **PERIOD9**.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
ABNQUECALLS	The number of **ABNCALLS** that were abandoned while in a split/skill or direct agent ACD queue. Available on Generic 2 and Generic 3 switches and the ECS.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table

Database Item	Description	Data Type	Column Type	Length
ABNRINGCALLS	The number of split/skill and direct agent **ABNCALLS** that were abandoned by the caller while ringing at an agent position. Available on Generic 2 and Generic 3 switches and on the ECS.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
ABNTIME	The amount of time caller waited while vector steps were executed, the call was queued, and ringing before abandoning. For phantom-abandon calls, **ABNTIME** is the total time from entering the VDN until the agent released the call.	C	integer	4 bytes
ACCEPTABLE	The number of **ACDCALLS** and **CONNECTCALLS** that were answered within the acceptable service level (**SERVICELEVEL**) as defined on the ACD Administration: VDN Call Profile Setup window.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
ACD (index)	The ACD number for which data was collected.	I	smallint	2 bytes
ACDCALLS	The number of split/skill and direct agent ACD calls that were answered by an agent from “queue to main,” “check backup,” “messaging split/skill,” “route to” split/skill or direct agent, and “adjunct routing” to a split/skill or direct agent. **ACDCALLS** includes **ACDCALLS1-10, ACCEPTABLE, ANSCONNCALLS1-10, BACKUPCALLS,** and **TRANSFERRED**.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
ACDCALLS _R1	The number of split/skill and direct agent ACD calls that were answered by a reserve1 agent from “queue to skill or best,” “check skill or best,” “messaging split/skill,” “route to” split/skill or direct agent, and “adjunct routing link” to a split/skill.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table

Database Item	Description	Data Type	Column Type	Length
ACDCALLS _R2	The number of split/skill and direct agent ACD calls that were answered by a reserve2 agent from “queue to skill or best,” “check skill or best,” “messaging split/skill,” “route to” split/skill or direct agent, and “adjunct routing link” to a split/skill.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
ACDTIME	The amount of talk time of all **ACDCALLS**, not including **HOLDTIME**. **ACDTIME** includes **SKILLTIME1, SKILLTIME2,** and **SKILLTIME3**.	C	integer	4 bytes
ACWTIME	The amount of time that agents spent in ACW associated with **ACDCALLS**. **ACWTIME** includes **SKILLACWTIME1-3**.	C	integer	4 bytes
ADJATTEMPTS	The number of adjunct-routing attempts for calls in this VDN. **ADJATTEMPTS** includes **ADJROUTED**. Available on Generic 3 switches and on the ECS with the ASAI gateway.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
ADJROUTED	The number of adjunct routing calls that were redirected by an adjunct processor or host computer. Available on Generic 3 switches and on the ECS with vectoring and the ASAI feature.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
ANSCONNCALLS 1-10	The number of times that callers were answered (**ACDCALLS**) and connected (**CONNECTCALLS**) during each of the service level increments **PERIOD1** through **PERIOD9** as defined in the ACD Administration: VDN Call Profile Setup window. **ANSCONNCALLS10** counts calls answered or connected after **PERIOD9**. Answered/connected calls include split/skill and direct agent ACD calls and extension calls by a “route to” or “adjunct routing” vector command.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table

Database Item	Description	Data Type	Column Type	Length
ANSTIME	The amount of time split/skill and direct agent ACD calls spent waiting to be answered in vector processing, in queue, and while ringing. For extension calls on Generic 2.1 switches, **ANSTIME** is the time until ringing starts. **ANSTIME** includes **RINGTIME**.	C	integer	4 bytes
BACKUPCALLS	The number of ACDCALLS that were delivered to and answered by this split/skill by a vector command other than "queue to main" and the number of ACDCALLS that were delivered to a split/skill by a “queue to” vector command answered by an agent that has neither reserve1 or reserve2 skill levels assigned for that skill. This allows tracking of calls answered by agents with a reserve1 or reserve2 skill level assigned for a particular skill. Calls answered in the main split/skill can then be calculated as ACDCALLS - BACKUPCALLS. However, this calculation does not include direct agent calls. BACKUPCALLS includes "messaging split/skill" calls, "check backup" calls, and calls that route to a split/skill or direct agent, either by the "route to" vector command or by adjunct routing. Calls that are redirected back to the split/skill using the Redirection on No Answer feature and then answered are also counted as BACKUPCALLS. **Note:** The Redirect on No Answer to VDN routing feature is available on the switch. Available on Generic 2 and Generic 3 switches and the ECS with the vectoring feature.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
BH_ABNCALLS	The number of incoming calls carried by the trunk group that abandoned during the busy hour.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table

Database Item	Description	Data Type	Column Type	Length
BH_ACDCALLS	The number of incoming calls carried by this trunk group during the busy hour that were answered by an agent as split/skill or direct agent ACD calls.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_ACDTIME				
BH_BUSYCALLS	The number of incoming calls carried by the trunk group during the busy hour that were given a busy signal by the switch.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_DISCCALLS	The number of incoming calls carried by the trunk group during the busy hour that were forced to disconnect by the switch.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_INTERVAL	The defined interval length, with the start time, of the busy hour.	B	integer; only in dtkgrp table	4 bytes
BH_OTHERCALLS	The number of incoming calls carried by the trunk group during the busy hour that were not answered or abandoned. **BH_OTHERCALLS** include extension in calls, calls forced busy or disconnected, calls that outflowed off the switch, short inbound calls, and inbound calls of unknown disposition. **BH_OTHERCALLS** includes **BH_BUSYCALLS** and **BH_DISCCALLS**.	B	integer; smallint in htkgrp table	4 bytes; 2 bytes in htkgrp table
BH_STARTTIME	The starting time of the hour for which busy hour data was collected. The busy hour is that set of contiguous intervals during the day totaling an hour in which the trunk holding time for the trunk group was a maximum.	B	integer	4 bytes
BH_VDNCALLS				
BSRPLAN	This item consists of information for the specified Best Service Routing (BSR) plan. Available on the R6 ECS.	A	smallint	2 bytes

Database Item	Description	Data Type	Column Type	Length
BUSYCALLS	The number of **INCALLS** that were given a busy signal by the ECS. This can occur on all ECS releases through the “busy" vector command. On the Generic 3 ECS, **BUSYCALLS** can occur if a call is routed to a split/skill with coverage set to “yes” where there are no agents available, the queue is full (or there is no queue), there is no coverage, and an announcement has played or the trunk is not a CO trunk. Also on the Generic 3 ECS, **BUSYCALLS** can occur if a call is routed to a direct agent with coverage set to “yes,” the agent is not logged in and there is no coverage path administered, an announcement has played, or the trunk is not a CO trunk.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
BUSYTIME	The duration of all **BUSYCALLS** (until the trunk goes idle). The format is the number of seconds that have accumulated.	C	integer	4 bytes
CONNECTCALLS	The number of non-ACD **INCALLS** that were delivered to a station extension (other than a VDN or direct agent login ID) by a “route to” or “adjunct routing” vector command and were not abandoned by the callers. For Generic 2.1 ECS, non-ACD abandons are not tracked, so all calls that route to a station extension (other than a VDN) are included in **CONNECTCALLS** for that ECS release. **CONNECTCALLS** includes **ANSCONNCALLS1-10**.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
CONNECTTIME	The amount of time **CONNECTCALLS** waited before being answered (for the Generic 3 ECS). For Generic 2 ECS, **CONNECTTIME** is the time before ringing starts. The format is the number of seconds that have accumulated.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
CONNTALKTIME	The amount of talk time for all **CONNECTCALLS,** not including **HOLDTIME (except on Generic 2.1)**. The format is the number of seconds that have accumulated.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
DEFLECTCALLS	The number of calls that were deflected to the network by BSR. Requires the R6 ECS.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
DISCCALLS	With Generic 2.1 and Generic 3 Version 1 switches, **DISCCALLS** is the number of **INCALLS** that were given a forced disconnect announcement by the "disconnect" vector command, listened to the entire announcement, then were disconnected by the switch. With Generic 2.2, Generic 3 Version 2 and later Generic 3 switches, and with the ECS, this is the number of **INCALLS** that were disconnected by the switch by the "disconnect'' vector command. With Generic 3 Version 2 and newer Generic 3 switches and with the ECS, **DISCCALLS** also includes calls disconnected by the switch when the vector disconnect timer expired or that reached the end of vector processing without being queued. **DISCCALLS** includes **VDISCCALLS**.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
DISCTIME	The amount of time all **DISCCALLS** spent in this VDN. The time until the trunk drops following the forced disconnect command for those calls recorded as **DISCCALLS**. For Generic 2.2, Generic 3 Version 2, and newer Generic 3 switches, and for the ECS, if the caller hangs up during the forced disconnect announcement, this is the time until the caller hangs up. For Generic 2.1 and Generic 3 switches, this is the time until the announcement ends and the caller is disconnected by the switch.	C	integer	4 bytes

Database Item	Description	Data Type	Column Type	Length
HOLDABNCALLS	The number of times that callers abandoned from on hold. For Generic 2.1 switches, **HOLDABNCALLS** applies to split ACD calls held. For Generic 2.2 and Generic 3 switches, and for the ECS, **HOLDABNCALLS** applies to all calls the agent put on hold.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
HOLDACDCALLS	The number of split/skill or direct agent ACD calls placed on hold at least one time.	C	integer	4 bytes
HOLDACDTIME	The amount of time spent by split/skill or direct agent ACD callers spent on hold.	C	integer	4 bytes
HOLDCALLS	The number of calls that were placed on hold at least once. **HOLDCALLS** also includes **HOLDABNCALLS** and **HOLDACDCALLS**. For Generic 2.1 switches, **HOLDCALLS** applies to split ACD calls held. For Generic 2.2 and Generic 3 switches, and for the ECS, **HOLDCALLS** applies to all calls the agent put on hold.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
HOLDTIME	The amount of time spent by callers on hold. **HOLDTIME** includes **HOLDACDTIME**. For Generic 2.1 switches, **HOLDTIME** applies to split ACD calls. For Generic 2.2 and Generic 3 switches and for the ECS, **HOLDTIME** applies to all calls the agent put on hold.	C	integer	4 bytes
I_ARRIVED	The number of calls that reached this VDN during this interval.	C	integer	4 bytes
INCALLS	The number of inbound calls that were directed to this VDN. **INCALLS** includes **ABNCALLS**, **INFLOWCALLS**, **OTHERCALLS,** **RETURNCALLS**, and **RINGCALLS** (which includes **ACDCALLS**). **INCALLS** = **ABNCALLS** + **ACDCALLS** + **OTHERCALLS.**	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table

Database Item	Description	Data Type	Column Type	Length
INCOMPLETE	This item indicates if data is complete for this collection interval. Data is incomplete whenever the link goes down and whenever tracking is aborted for calls, due to trunk failures (Generic 2), the trunk going maintenance busy with a call active (Generic 3 and the ECS), protocol failures with data collection active, or when the split/skill or VDN call profile is changed with data collection active. The value for interval tables indicates whether data is incomplete for the interval (0 = NO, 1 = YES). The value in the daily, weekly, and monthly tables indicates the number of incomplete intervals in the day, week, or month.	C	smallint	2 bytes
INFLOWCALLS	The number of calls that were redirected to the split's/skill's queue from another queue. When a call leaves the VDN (for example, by routing to another VDN) or leaves vector processing, (for example, by routing to a split/skill) the next split/skill to which a call queues will not be credited with an inflow. Calls that ring at an agent and are then requeued to the same split/skill by the Redirect on No answer feature are counted as inflows to that split/skill. On Generic 2 and Generic 3 switches with vectoring, an inflow is counted for calls that intraflow from one split's queue to another (that is, call that queue to a split after having previously been queued to another split).	C	integer; smallint in hsplit table	4 bytes, 2 bytes in hsplit table
INTERFLOWCALLS	The number of **OUTFLOWCALLS** that were redirected to a destination outside the switch. **INTERFLOWCALLS** includes **LOOKFLOWCALLS**.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table

Database Item	Description	Data Type	Column Type	Length
INTIME	The amount of time spent by **INCALLS** in this VDN. **INTIME** = **ACDTIME** + **ABNTIME** + **ANSTIME** + **HOLDTIME** + **OTHERTIME** However, on Generic 2.1 switches, there are multiple call handling scenarios in which call-based **ACDTIME** is stopped before the call ends. In these scenarios, **INTIME** does not add up to **ACDTIME + ABNTIME + ANSTIME + OTHERTIME + HOLDTIME**. (The scenarios occur when an agent puts an ACD call on hold using the HOLD key when another ACD call is already on hold, or when a call is dropped while an ACD call is on hold.)	C	integer	4 bytes
INTRVL	The number of minutes in the timed period (15, 30, or 60). **INTRVL** applies to intrahour tables only.	A	smallint; only in hvdn table	2 bytes
LOOKATTEMPTS	The number of look-ahead interflow attempts for calls in this VDN. Available on Generic 2.2 and Generic 3 switches and on the ECS with the vectoring and look-ahead interflow features. **LOOKATTEMPTS** includes **LOOKFLOWCALLS**.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
LOOKFLOWCALLS	The number of **INTERFLOWCALLS** that were redirected by way of the Lookahead Interflow feature. Available on Generic 2.2 and Generic 3 switches with the Lookahead Interflow feature.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
MAXOCWTIME	The maximum time that a call, recorded during the collection interval, waited in the VDN before being answered (ACD calls) or connected (non-ACD calls), abandoning, being redirected, receiving a busy signal or being disconnected. This applies only to the first disposition of the call.	M	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table

Database Item	Description	Data Type	Column Type	Length
MAXWAITING	The maximum number of calls simultaneously in progress in the VDN during the collection interval.	M	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
NETDISCCALLS	The number of calls that disconnected from the BSR reply step. Requires the R6 ECS.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
NETINCALLS	The amount of calls that interflowed in from the network in BSR. Requires the R6 ECS.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
NETINTIME	The amount of time, in seconds, that the call was in a VDN somewhere else in the network. Requires the R6 ECS.	C	integer	4 bytes
NETPOLLS	The number of network polls for BSR consider steps. Requires the R6 ECS.	C	integer	4 bytes
NOANSREDIR	The number of split/skill and direct agent ACD calls that rang at agent stations and then were automatically redirected by the Redirection on No Answer feature because they were not answered. Available on Generic 3 Version 2 and newer switches and on the ECS.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
NUMTGS	The number of trunk groups assigned to this VDN.	A	integer	4 bytes
OTHERCALLS	The number of calls that were given a forced busy, forced disconnect, or outflowed from the switch, and non-ACD calls that were answered (**CONNECTCALLS**). **OTHERCALLS** includes **BUSYCALLS**, **CONNECTCALLS**, **DISCCALLS**, and **OUTFLOWCALLS**. **OTHERCALLS** = **INCALLS** - **ACDCALLS** - **ABNCALLS.**	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table

Database Item	Description	Data Type	Column Type	Length
OTHERTIME	The duration of all **OTHERCALLS** until the calls leave the VDN (the calls drop, are sent to another VDN, are transferred, or are sent outside the switch). **OTHERTIME** includes **BUSYTIME**, **CONNECTTIME**, **CONNTALKTIME**, **DISCTIME**, and **OUTFLOWTIME**.	C	integer	4 bytes
OUTFLOWCALLS	The number of **INCALLS** that were redirected to another VDN or to a destination outside the switch by way of a “route to” or “adjunct routing” command, or were redirected to another VDN by the Redirect on No Answer feature. Note that calls are only counted as outflows from the VDN when they are redirected to another VDN or to an off-switch destination. Calls in the VDN that route to other destinations, such as split/skills or extensions, are not counted as outflows from the VDN. **OUTFLOWCALLS** includes **INTERFLOWCALLS** and **SLVLOUTFLOWS**.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
OUTFLOWTIME	The amount of time all **OUTFLOWCALLS** spent in the VDN before being redirected.	C	integer	4 bytes
PERIOD1-9	The length, in seconds, of each service level increment as defined in the ACD administration. Each increment represents a progressively longer wait time. CMS counts answered or abandoned calls that wait beyond the last increment (**PERIOD9**) in **ANSCONNCALLS10**.	A	smallint	2 bytes
PERIODCHG	This item indicates if service level increments **PERIOD1** through **PERIOD9** (as defined on the ACD Administration: VDN Call Profile window) changed during the data collection interval. Valid values for **PERIODCHG** are 1 = YES and 0 = NO.	A	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table

Database Item	Description	Data Type	Column Type	Length
PHANTOMABNS	The number of split/skill and direct agent ACD calls and calls that routed to an agent or extension with talk time of less than the value set for the phantom abandoned call timer. Available on Generic 3 and newer switches and on the ECS.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
RETURNCALLS	The number of calls that reached this VDN via the VDN return destination feature. Available on Generic 3 Version 3 and newer switches and on the ECS.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
RINGCALLS	The number of split/skill and direct agent ACD calls that rang at agent positions. Available on Generic 2 and Generic 3 switches and on the ECS. **RINGCALLS** includes **ACDCALLS**.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
RINGTIME	The amount of time split/skill and direct agent ACD calls spent ringing at agent positions, independent of final disposition. Available on Generic 2 and Generic 3 switches and on the ECS.	C	integer	4 bytes
ROW_DATE (index)	The date for which the data was collected or the exception occurred.	I	date	4 byte Informix date
SERVICELEVEL	The number of seconds within which calls must be answered/connected to be considered acceptable (as defined on the ACD Administration: VDN Call Profile Setup window).	A	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
SKILLACWTIME1-3	The amount of ACW time spent by agents for calls answered in each VDN skill preference. Available on Generic 2.2 and Generic 3 switches with EAS.	C	integer	4 bytes
SKILLCALLS1-3	The number of calls answered by agents in each VDN skill preference. Available on Generic 2.2 and Generic 3 Version 2 switches with EAS.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table

Database Item	Description	Data Type	Column Type	Length
SKILLTIME1-3	The amount of time agents spent talking on calls they answered in each VDN skill preference. Available on Generic 2.2 and Generic 3 switches with EAS.	C	integer	4 bytes
SKILL1-3	The first, second, and third VDN skill assigned to this VDN. Available on Generic 2.2 and Generic 3 and newer switches and on the ECS with EAS.	A	smallint	2 bytes
SLVLABNS	The number of **ABNCALLS** whose time to abandon was less than or equal to this VDN's **SERVICELEVEL**.	C	integer	4 bytes
SLVLOUTFLOWS	The number of **OUTFLOWCALLS** whose time to outflow was less than or equal to this VDN's **SERVICELEVEL**.	C	integer	4 bytes
SVCLEVELCHG	This item indicates whether the service level was changed during the data collection interval. Valid values for **SVCLEVELCHG** are 1 = YES and 0 = NO.	A	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
TRANSFERRED	The number of calls that were transferred to another destination. For Generic 2.1 switches, **TRANSFERRED** includes all VDN calls blind transferred to a measured VDN or split. For Generic 2.2 and Generic 3 switches and for the ECS, **TRANSFERRED** includes all VDN calls transferred.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table
VDISCCALLS	The number of calls forced to disconnect because the vector disconnect timer timed out, or because the call reached a vector stop without being queued. ("Vector stop" means a "stop" vector command, the end of the vector, or the call executed 1000 vector steps.) Available on Generic 3 Version 2 and newer switches and on the ECS.	C	integer; smallint in hvdn table	4 bytes; 2 bytes in hvdn table

Database Item	Description	Data Type	Column Type	Length
VDN (index)	The vector directory number associated with this VDN.	I	char(6)	6 byte ASCII text string
VECTOR (index)	The vector number associated with this VDN.	A	smallint	2 bytes

ECS Cross-Reference

The following table lists which of the ECS releases support each VDN database item.

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ABNCALLS	X	X	X	X	X	X	X
ABNCALLS1-10	X	X	X	X	X	X	X
ABNQUECALLS	X	X	X	X	X	X	X
ABNRINGCALLS	X	X	X	X	X	X	X
ABNTIME	X	X	X	X	X	X	X
ACCEPTABLE	X	X	X	X	X	X	X
ACD	X	X	X	X	X	X	X
ACDCALLS	X	X	X	X	X	X	X
ACDCALLS_R1						X	X
ACDCALLS_R2						X	X
ACDTIME	X	X	X	X	X	X	X
ACWTIME	X	X	X	X	X	X	X
ADJATTEMPTS	X	X	X	X	X	X	X
ADJROUTED	X	X	X	X	X	X	X
ANSCONNCALLS1-10	X	X	X	X	X	X	X
ANSTIME	X	X	X	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
BH_ACDCALLS	X	X	X				
BH_BUSYCALLS	X	X	X				
BH_DISCCALLS	X	X	X				
BH_INTERVAL	X	X	X				
BH_OTHERCALLS	X	X	X				
BH_OUTCALLS	X	X	X				
BH_OUTTIME	X	X	X				
BH_STARTTIME	X	X	X				
BH_VDNCALLS	X	X	X				
BACKUPCALLS	X	X	X	X	X	X	X
BSRPLAN	X	X	X	X	X	X	X
BUSYCALLS	X	X	X	X	X	X	X
BUSYTIME	X	X	X	X	X	X	X
CONNECTCALLS	X	X	X	X	X	X	X
CONNECTTIME	X	X	X	X	X	X	X
CONNTALKTIME	X	X	X	X	X	X	X
DEFLECTCALLS				X	X	X	X
DISCCALLS	X	X	X	X	X	X	X
DISCTIME	X	X	X	X	X	X	X
HOLDABNCALLS	X	X	X	X	X	X	X
HOLDACDCALLS	X	X	X	X	X	X	X
HOLDACDTIME	X	X	X	X	X	X	X
HOLDCALLS	X	X	X	X	X	X	X
HOLDTIME	X	X	X	X	X	X	X
I_ARRIVED	X	X	X	X	X	X	X
INCALLS	X	X	X	X	X	X	X
INCOMPLETE	X	X	X	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
INFLOWCALLS	X	X	X	X	X	X	X
INTERFLOWCALLS	X	X	X	X	X	X	X
INTIME	X	X	X	X	X	X	X
INTRVL	X	X	X	X	X	X	X
LOOKATTEMPTS	X	X	X	X	X	X	X
LOOKFLOWCALLS	X	X	X	X	X	X	X
MAXOCWTIME	X	X	X	X	X	X	X
MAXWAITING	X	X	X	X	X	X	X
NETDISCCALLS				X	X	X	X
NETINCALLS				X	X	X	X
NETINTIME				X	X	X	X
NETPOLLS				X	X	X	X
NOANSREDIR	X	X	X	X	X	X	X
OTHERCALLS	X	X	X	X	X	X	X
OTHERTIME	X	X	X	X	X	X	X
OUTFLOWCALLS	X	X	X	X	X	X	X
OUTFLOWTIME	X	X	X	X	X	X	X
PERIOD1-9	X	X	X	X	X	X	X
PERIODCHG	X	X	X	X	X	X	X
PHANTOMABNS	X	X	X	X	X	X	X
RETURNCALLS	X (V3)	X	X	X	X	X	X
RINGCALLS	X	X	X	X	X	X	X
RINGTIME	X	X	X	X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
SERVICELEVEL	X	X	X	X	X	X	X
SKILL1-3	(EAS)	(EAS)	(EAS)	X	X	X	X
SKILLACWTIME1-3	(EAS)	(EAS)	(EAS)	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
SKILLCALLS1-3	(EAS)	(EAS)	(EAS)	X	X	X	X
SKILLTIME1-3	(EAS)	(EAS)	(EAS)	X	X	X	X
SLVLABNS	X	X	X	X	X	X	X
SLVLOUTFLOWS	X	X	X	X	X	X	X
STARTTIME	X	X	X	X	X	X	X
SVCLEVELCHG	X	X	X	X	X	X	X
TRANSFERRED	X	X	X	X	X	X	X
VDISCCALLS	X	X	X	X	X	X	X
VDN	X	X	X	X	X	X	X
VECTOR	X	X	X	X	X	X	X

Call work codes database items

Overview

The Call Work Codes database item descriptions apply to historical items.

Note:

Call work codes are only available with Generic 3 and Generic 2.2 ECS.

The **Data Type** column refers to **Cumulative (C)**, **Administrative (A)**, **Status (S)**, **Row Identifier (I), Busy Hour (B), Special Table (N),** or **Maximum Value (M)** data. Cumulative, Administrative, Row Identifier, Busy Hour, and Maximum Value items apply to both the current and previous interval real-time tables. Special Table items are historical, and apply only to the table in which they are stored.

Historical call work codes database items apply to the Intrahour Call Work Codes (`hcwc`), Daily Call Work Codes (`dcwc`), Weekly Call Work Codes (`wcwc`), and Monthly Call Work Codes (`mcwc`) tables. All items listed in the following Call Work Codes Database Items table are included in all four tables, unless otherwise noted in the Column Type column. Any differences in the data format between the four call work codes tables are also noted in the Column Type column. The indexes are **ACD**, **ROW_DATE** and **CWC**.

Contents

“Call work codes database items” contains the following topics:

Call work codes database item table

The following table describes the data items in the CMS Call Work Codes database table.

Database Item	Description	Data Type	Column Type	Length
ACD (index)	The ACD number for which data was collected.	I	smallint	2 bytes
ACDCALLS	The number of times this call work code was entered while an agent was on a split/skill or direct agent ACD call or in call-related ACW.	C	integer; smallint in hcwc table	4 bytes; 2 bytes in hcwc table
ACDTIME	The amount of talk time of all **ACDCALLS** (not including **HOLDTIME**) associated with this call work code.	C	integer	4 bytes
ACWTIME	The amount of time that the agent spent in ACW for **ACDCALLS** that were associated with this call work code.	C	integer	4 bytes
CWC (index)	The call work code for which data was collected.	I	char(16)	16 byte ASCII text string
INCOMPLETE	This item indicates if data is complete for this interval. Data is incomplete whenever the link goes down and whenever tracking is aborted for calls, due to trunk failures (Generic 2), the trunk going maintenance busy with a call active (Generic 3), protocol failures with data collection active, or when split/skill or VDN call profile is changed with data collection active. The value for interval tables indicates whether data collection is incomplete for the interval (0 = N0, 1 = YES). The value in the daily, weekly, and monthly tables indicates the number of incomplete intervals in the day, week, or month. Changing split/skill or VDN call profile data while data collection is active only affects the respective split/skill or VDN data.	C	smallint	2 bytes

Database Item	Description	Data Type	Column Type	Length
INTRVL	The number of minutes in the intrahour interval (15, 30, or 60). **INTRVL** applies to intrahour tables only.	A	smallint; in hcwc table only	2 bytes
ROW_DATE (index)	The day for which data was collected or the exception occurred.	I	date	4 byte Informix date
STARTTIME	The start time of the interval for which data was collected. **STARTTIME** applies to the only interval table.	I	smallint; in hcwc table only	2 bytes

ECS cross-reference

The following table lists which of the ECS releases support each Call Work Code database item.

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ACD	X	X	X	X	X	X	X
ACDCALLS	X	X	X	X	X	X	X
ACDTIME	X	X	X	X	X	X	X
ACWTIME	X	X	X	X	X	X	X
CWC	X	X	X	X	X	X	X
INCOMPLETE	X	X	X	X	X	X	X
INTRVL	X	X	X	X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
STARTTIME	X	X	X	X	X	X	X

Agent Login/Logout database items

Overview

The Agent Login/Logout database item descriptions are **historical** items that apply specifically to the Agent Login/Logout (`haglog`) table. The indexes are **ACD**, **SPLIT** and **ROW_DATE**.

Contents

"Agent Login/Logout database items" contains the following topics:

Agent Login/Logout database item table

The following table describes the data items in the CMS Agent Login/Logout database table.

Database Item	Description	Column Type	Length
ACD (index)	The ACD number for which data was collected.	smallint	2 bytes
EXTN	the extension number of the station that the agent staffed.	char(6)	6 byte ASCII text string
INFLAG	If not null, indicates that agent was already logged in when the link came up. Values are NULL and "<."	char(1)	1 byte ASCII text string

Database Item	Description	Column Type	Length
LOC_ID	The switch port network location ID that is associated with an agent upon login to the ACD. A location ID is not directly assigned to an agent; instead, it is associated with the equipment location of the voice terminal that the agent uses to log into the ACD. Therefore, only when an agent logs into the ACD can the agent become associated with a location ID. Valid values are 01-44.	integer	4 bytes
LOGID	The login ID that was used to staff the **EXTENSION**. Agents in multiple splits/skills have one **LOGID**.	char(10)	10 byte ASCII text string
LOGIN	The time at which the agent logged into this extension and split/skill with the given login ID. This field is a standard UNIX time field; that is, the time is stored as the number of seconds since January 1, 1970.	integer	4 bytes
LOGONSKILL2-20	The second through twentieth skills the agent logged in with. **Note:** The number of skills per agent depends on the type of switch. For example, prior to the ECS, LOGONSKILL5 is only available on Generic 2.2 switches. Available on the ECS with the EAS feature and the Generic 2.2 and later Generic 3 switches with the EAS feature.	smallint	2 bytes
LOGOUT	The time at which the agent logged out. This field is a standard UNIX time field; that is, the time is stored as the number of seconds since January 1, 1970.	integer	4 bytes
LOGOUT_DATE	The date on which the agent logged out.	date	4 byte Informix date
LOGOUT_REASON	The reason code (0 through 9) associated with the agent's logout. For switch releases earlier than the ECS or switch releases that do not have the EAS feature and reason codes active, this field will always contain a 0 when the agent has logged out.	smallint	2 bytes

Database Item	Description	Column Type	Length
OUTFLAG	If not null, this item indicates that the agent logged out while the link was down. Values are NULL and ">."	char(1)	1 byte ASCII text string
PREFERENCE	The agent's call handling preference. Values are NEED (greatest need) LVL (skill level), and PCNT (percent allocation). Requires a DEFINITY ECS R5 or R6 with EAS. PCNT is only available on the ECS R6.	integer	4 bytes
SKLEVEL	This item indicates the agent's skill level (1-16) for a normal skill or reserve level (1 or 2) for a reserve skill. This **SKLEVEL** applies to **LOGONSKILL.** Requires a DEFINITY ECS R5 or R6 with EAS. Reserve levels are only available on the ECS R6.	integer	4 bytes
SKLEVEL2-20	This item indicates the agent's skill level (1-16) for a normal skill or reserve level (1 or 2) for a reserve skill. This **SKLEVEL2-20** applies to **LOGONSKILL2-20.** Requires a DEFINITY ECS R5 or R6 with EAS. Reserve levels are only available on the ECS R6.	integer	4 bytes
SKPERCENT	The agent's percent allocation for **LOGONSKILL.** Requires a DEFINITY ECS R6 with EAS.	integer	4 bytes
SKPERCENT2-20	The agent's percent allocation for **LOGONSKILL2-20.** Requires a DEFINITY ECS R6 with EAS.	integer	4 bytes
ROW_DATE (index)	The day for which data was collected or the exception occurred.	date	4 byte Informix date

Database Item	Description	Column Type	Length
SKILLTYPE	The type (“p” for primary or “s” for secondary) of the second, third, and fourth skills the agent logged into. **Note:** For the ECS and later switch releases with the EAS feature, skill level 1 will be represented by “p,” skill level 2 will be represented by “s,” and skill levels 3 through 16 by blank. Users of more than 2 skill levels should use **SKLEVEL** instead of **SKILLTYPE** items. Available for Generic 3 switches with EAS and requires Generic 2.2 or Generic 3 Version 2 switches with EAS.	char(1)	1 byte ASCII text string
SKILLTYPE2-4	The type (“p” for primary or “s” for secondary) of the second, third, and fourth skills the agent logged into. Requires Generic 2.2 or Generic 3 Version 2 switches with EAS. **Note:** For the ECS and newer EAS releases, skill level 1 will be represented by “p”, skill level 2 by “s” and skill levels 3-16 by blank. Users of more than 2 skill levels should use **SKLEVEL** items instead of **SKILLTYPE** items.	char(1)	1 byte ASCII text string
SKLEVEL	The skill level (from 1 through 16) associated with the first skill the agent logged into.	smallint	2 bytes
SKLEVEL2-20	The skill levels (from 1 through 20) associated with the second through twentieth skills (**LOGONSKILL2** through **LOGONSKILL20**) the agent logged into.	smallint	2 bytes
SPLIT (index)	The split number to which the extension is assigned or skill number the agent logged into.	smallint	2 bytes

ECS cross- reference

The following table lists which of the ECS releases support each Agent Login/Logout database items.

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ACD	X	X	X	X	X	X	X
EXTN	X	X	X	X	X	X	X
INFLAG	X	X	X	X	X	X	X
LOC_ID							X
LOGID	X	X	X	X	X	X	X
LOGIN	X	X	X	X	X	X	X
LOGONSKILL2-4	(EAS)	(EAS)	(EAS)	X	X	X	X
LOGONSKILL5			(EAS)	X	X	X	X
LOGONSKILL6-20			(EAS)	X	X	X	X
LOGOUT	X	X	X	X	X	X	X
LOGOUT_DATE	X	X	X	X	X	X	X
LOGOUTREASON			(EAS)	X	X	X	X
OUTFLAG	X	X	X	X	X	X	X
PREFERENCE				X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
SKILLTYPE	X (EAS)	X (EAS)	[1]	X	X	X	X
SKILLTYPE2-4	X (EAS)	X (EAS)	[1]	X	X	X	X
SKLEVEL	X (EAS)	X (EAS)	X (EAS)	X	X	X	X
SKLEVEL2-4	X (EAS)	X (EAS)	X (EAS)	X	X	X	X
SKLEVEL5-20			X (EAS)	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
SKPERCENT				X	X	X	X
SKPERCENT2-20				X	X	X	X
SPLIT	X	X	X	X	X	X	X

1. Not recommended.

Agent trace database items

Overview

The Agent Trace database item descriptions are **historical** items that apply specifically to the Agent Trace (`ag_actv`) table. The indexes are **ACD**, **LOGID,** and **ROW_DATE**.

Contents

"Agent trace database items" contains the following topics:

Agent trace database item table

The following table describes the data items in the CMS Agent Trace database table.

Database Item	Description	Column Type	Length
ACD (index)	The ACD number for which data was collected.	smallint	2 bytes
AGT_RELEASED	The agent who released or dropped the split/skill or direct agent ACD call. This is always true for ACD calls the agent transferred or conferenced. Available on Generic 3 and newer switches.	char(1)	1 byte ASCII text string
ASSIST_ACTV	The agent requested supervisor assistance (pressed the ASSIST button).	char(1)	1 byte ASCII text string
AUXREASON	The reason code associated with the agent's state. This is blank if the agent is not in the AUX state. For agents in AUX on switch releases that are earlier than the ECS or that do not have EAS and reason codes active, this will be 0 (zero).	smallint	2 bytes

Database Item	Description	Column Type	Length
CALLER_HOLD	The agent put the current call on hold. For Generic 2.1 switches, **CALLER_HOLD** applies to split ACD calls. For Generic 2.2 and Generic 3 switches, **CALLER_HOLD** applies to all calls the agent put on hold.	char(1)	1 byte ASCII text string
CALLING_II	The Information Indicator (II) digits associated with the call. These digits supply information about the originator location; for example, pay phone, hospital, or prison. Available on the ECS and newer switches.	char(2)	2 byte ASCII text string
CALLING_PTY	The calling party identification, which is the Automatic Number Identification (ANI)/Station Identification (SID) (for Generic 2.2 and Generic 3 Version 4 and later switches with ISDN ANI delivery), extension or trunk equipment location identifying the originator of the call. The field is blank if the trunk is not measured or, for internal calls, if the originating extension is not measured.	char(12)	12 byte ASCII text string
CONFERENCE	The agent who activated a conference. Available on Generic 3 and Generic 2.2 switches.	char(1)	1 byte ASCII text string
DIGITS_DIALED	The digits the agent dialed to originate a call. Trunk access codes, feature access codes, account and authorization codes are not included. Available on Generic 2.2 and Generic 3 switches.	char(16)	16 byte ASCII text string
DIRECTION	The direction of the call the agent is currently handling for any split/skill. Valid values are IN, OUT, or as defined in Dictionary. If the agent is not on a call, the value is blank (NULL).	smallint	2 bytes
DURATION	The duration of current **WORKMODE** and **DIRECTION** for this split (for example, length of time in current **AGSTATE** for this split).	integer	4 bytes
EVENT_TIME	The time of day (hour, minute, and second) the **WORKMODE** or **DIRECTION** changed.	integer	4 bytes

Database Item	Description	Column Type	Length
EXT_CALL_ORIG	The agent who originated an external (off-switch) call. Available on Generic 2.2 and Generic 3 switches.	char(1)	1 byte ASCII text string
KEYBD_DIALED	The call was keyboard dialed. Available on Generic 2.2 switches with the ASAI Gateway Interface feature and Generic 3 switches with the ASAI feature.	char(1)	1 byte ASCII text string
LOC_ID	The switch port network location ID that is associated with an agent upon login to the ACD. A location ID is not directly assigned to an agent; instead, it is associated with the equipment location of the voice terminal that the agent uses to log into the ACD. Therefore, only when an agent logs into the ACD can the agent become associated with a location ID. Valid values are 01 through 44.	integer	4 bytes
LOGID (index)	The login ID that was used to staff the **EXTENSION**. Agents in multiple splits/skills have one **LOGID**.	char(10)	10 byte ASCII text string
LOGOUTREASON	The reason code (0 through 9) associated with the agent's logout. For switch releases earlier than the ECS or switch releases that do not have the EAS feature and reason codes active, this field will always contain 0 when the agent has logged out.	smallint	2 bytes
MCT	The agent activated a malicious call trace. Available on Generic 2 and Generic 3 (except for Generic 3i Version 1) switches.	char(1)	1 byte ASCII text string
RECONNECT	This event represents the agent reconnecting to the call after putting it on hold. Available on Generic 2 and Generic 3 (except for Generic 3 Version 1) switches.	char(1)	1 byte ASCII text string
ROW_DATE (index)	The day for which data was collected or the exception occurred.	date	4 byte Informix date
SEQ_NUM	The sequence number of this exception record.	integer	4 bytes

Database Item	Description	Column Type	Length
SPLIT	The split number to which the EXTENSION is assigned or the skill number the agent logged into.	smallint	2 bytes
STARTTIME	The time of day (hour and minute) for which the agent trace is being ordered. This is the time of day you enter to request the report.	smallint	2 bytes
TRANSFERRED	This item represents if an answering agent initiated a transfer for this call. Valid values are YES and NO. For Generic 2.1 switches, **TRANSFERRED** includes transfers to measured VDNs or splits/skills. For Generic 3 and Generic 2.2 switches, **TRANSFERRED** includes all calls that are transferred.	char(1)	1 byte ASCII text string
UCID	The UCID is the Universal Call Identifier—a unique number assigned to this call segment within the customer network. Requires the DEFINITY ECS R6.	char(20)	20 byte ASCII text string
WMODE_SEQ	The sequence number for events that occur in the same second.	smallint	2 bytes
WORKCODE	The call work code the agent entered for the call. Available on Generic 3 and Generic 2.2 switches with call work codes.	char(16)	16 byte ASCII text string
WORKMODE	The work mode in which the agent was working during the trace. Agent work modes include: AVAIL, ACD, ACW, AUX, DACD, DACW, RING, UNKNOWN, OTHER, and UNSTAFF. If the agent has not been logged in during the collection interval, the value is blank.	smallint	2 bytes

ECS cross-reference

The following table lists which of the ECS releases support each Agent Trace database item.

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ACD	X	X	X	X	X	X	X
AGT_RELEASED	X	X	X	X	X	X	X
ASSIST_ACTV	X	X	X	X	X	X	X
AUXREASON	X	X	X	X	X	X	X
CALLER_HOLD	X	X	X	X	X	X	X
CALLING_II			X	X	X	X	X
CALLING_PTY	X	X	X	X	X	X	X
CONFERENCE	X	X	X	X	X	X	X
DIGITS_DIALED	X	X	X	X	X	X	X
DIRECTION	X	X	X	X	X	X	X
DURATION	X	X	X	X	X	X	X
EVENT_TIME	X	X	X	X	X	X	X
EXT_CALL_ORIG	X	X	X	X	X	X	X
KEYBD_DIALED	X	X	X	X	X	X	X
LOCID							X
LOGID (index)	X	X	X	X	X	X	X
LOGOUTREASON	X	X	X	X	X	X	X
MCT	X	X	X	X	X	X	X
RECONNECT	X	X	X	X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
SEQ_NUM							X
SPLIT	X	X	X	X	X	X	X
STARTTIME	X	X	X	X	X	X	X
TRANSFERRED	X	X	X	X	X	X	X
UCID				X	X	X	X
WMODE_SEQ	X	X	X	X	X	X	X
WORKCODE	X	X	X	X	X	X	X
WORKMODE	X	X	X	X	X	X	X

Current day configuration forecast database items

Overview

The Current Day Configuration database item descriptions are **historical** items used specifically to collect values entered in the Forecast: Current Day window. They apply to the Current Day (`f_cday`) table. The indexes are **ACD**, **ROW_DATE** and **SPLIT**.

Contents

"Current day configuration forecast database items" contains the following topics:

Current day configuration forecast database item table

The following table describes the data items in the CMS Current Day Configuration Forecast database table.

Database Item	Description	Column Type	Length
ACD (index)	The ACD number for which data was collected.	smallint	2 bytes
CHANGE	The additional change factor (percent).	smallfloat	4 byte Informix floating point type
CHPROF	The number of the call handling profile to use.	smallint	2 bytes
FMETHOD	The type of trending to use for forecast. Values are 0 = none, 1 = seasonal, 2 = current trending.	smallint	2 bytes
HDATE1-4	The date of first (**HDATE1**), second (**HDATE2**), third (**HDATE3**), and fourth (**HDATE4**) days of historical data to be used.	date	4 byte Informix date

Database Item	Description	Column Type	Length
ROW_DATE (index)	The day for which data was collected or the exception occurred.	date	4 byte Informix date
SPLIT (index)	The split/skill number for which data was collected	smallint	2 bytes
TRENDBASE	The base date for seasonal trending.	date	4 byte Informix date
WT1-4	The weight given to date 1 (**WT1**), date 2 (**WT2**), date 3 (**WT3**), and date 4 (**WT4**), respectively.	smallint	2 bytes

ECS cross-reference

The following table lists which of the ECS releases support each of the Current Day Configuration database items.

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ACD	X	X	X	X	X	X	X
CHANGE	X	X	X	X	X	X	X
CHPROF	X	X	X	X	X	X	X
FMETHOD	X	X	X	X	X	X	X
HDATE1-4	X	X	X	X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
SPLIT	X	X	X	X	X	X	X
TRENDBASE	X	X	X	X	X	X	X
WT1-4	X	X	X	X	X	X	X

Current day forecast report database items

Overview

The Current Day Forecast Report database item descriptions are **historical** items used specifically to collect values displayed in the Current Day Forecast report, and they apply to the Current Day Report (`f_cdayrep`) table. The indexes are **ACD**, **ROW_DATE** and **SPLIT**.

Forecast data for a split/skill are automatically generated when the Forecast Manager runs (if you have also completed a Current Day Configuration for the split/skill).

Contents

"Current day forecast report database items" contains the following topics:

Current day forecast report database item table

The following table describes the data items in the CMS Current Day Forecast Report database table.

Database Item	Description	Column Type	Length
ACD (index)	The ACD number for which data was collected.	smallint	2 bytes
AGOCC	The objective maximum percentage of time that an agent will be on ACD calls (agent occupancy).	smallfloat	4 byte Informix floating point type
AVGAGSERV	The objective average number of seconds for an agent to service a call.	smallint	2 bytes
AVGSPEEDANS	The objective average speed of answer in seconds for this type of call.	smallint	2 bytes
FCALLS	The number of forecast calls carried.	integer	4 bytes

Database Item	Description	Column Type	Length
INTRVL	The length of intrahour interval (15, 30, or 60). **INTRVL** applies to intrahour tables only.	smallint	2 bytes
NUMAGREQ	The number of agents required to handle **FCALLS**.	smallint	2 bytes
RAGOCC	The resulting maximum percentage of time that an agent will be on ACD calls.	smallfloat	4 byte Informix floating point type
RAVGSPEEDANS	The resulting average speed of answer in seconds for this type of call.	smallint	2 bytes
ROW_DATE (index)	The day for which data was collected or the exception occurred.	date	4 byte Informix date
RSERVLEVELP	The resulting percentage of calls to be handled within **SERVLEVELT** seconds.	smallfloat	4 byte Informix floating point type
SERVLEVELP	The objective percentage of calls to be handled within **SERVLEVELT** seconds.	smallfloat	4 byte Informix floating point type
SERVLEVELT	The number of seconds within which **SERVLEVELP** percent of calls are to be answered (service level time).	smallint	2 bytes
SPLIT (index)	The split/skill number for which data was collected.	smallint	2 bytes
STARTTIME	The start time of the intrahour interval for which data was collected. **STARTTIME** applies only to the interval table.	smallint	2 bytes

ECS cross-reference

The following table lists which of the ECS releases support each Current Day Report database items.

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ACD	X	X	X	X	X	X	X
AGOCC	X	X	X	X	X	X	X
AVGAGSERV	X	X	X	X	X	X	X
AVGSPEEDANS	X	X	X	X	X	X	X
FCALLS	X	X	X	X	X	X	X
INTRVL	X	X	X	X	X	X	X
NUMAGREQ	X	X	X	X	X	X	X
RAGOCC	X	X	X	X	X	X	X
RAVGSPEEDANS	X	X	X	X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
RSERVLEVELP	X	X	X	X	X	X	X
SERVLEVELP	X	X	X	X	X	X	X
SERVLEVELT	X	X	X	X	X	X	X
SPLIT	X	X	X	X	X	X	X
STARTTIME	X	X	X	X	X	X	X

Call record database items

Overview

The Call Record database item descriptions are **historical** items that apply specifically to the Call Record (`call_rec`) table. The indexes are **ACD** and **ROW_DATE**.

Contents

"Call record database items" contains the following topics:

Call record database item table

The following table describes the database items in the CMS Call Record database table.

Database Item	Description	Column Type	Length
ACD (index)	The ACD number for which data was collected.	smallint	2 bytes
ACWTIME	The time spent, in seconds, in After Call Work (ACW) associated with this call by the answering agent in this segment.	integer	4 bytes
AGT_RELEASED	The agent who released or dropped the split/skill or direct agent ACD call. This is always true for ACD calls the agent transferred or conferenced. (0=NO, 1=YES). Available on Generic 3 and newer switches.	smallint	2 bytes

Database Item	Description	Column Type	Length
ANSHOLDTIME	The total time, in seconds, the call was put on hold by the answering agent in this call segment. In agent-to-agent calls, **ANSHOLDTIME** is accrued for the answering agent if the agent puts the call on hold, but not for the other agent (who continues to accrue talk time). For Generic 2.1 switches, hold time is accrued only for ACD calls. For Generic 2.2 and Generic 3 switches, and DEFINITY ECS Release 5, hold time is accrued for any type of call.	integer	4 bytes
ANSLOCID	The location ID that is associated with the EXTENSION at which the answering agent logged in. The location ID is always 0 for Definity R7 and older switches. This item requires DEFINITY R8 switches through Definity R10, MultiVantage R11, and later switches.	smallint	2 bytes
ANSLOGIN	The login ID of the agent who answered the call in this segment. This field is blank for unmeasured extensions when EAS is not active.	char(9)	9 byte ASCII text string
ANSREASON	The reason code (0 through 9) associated with the answering agent's mode, if the agent is in the AUX mode. For agents in AUX on switches with releases prior to the ECS or switches that do not have EAS and reason codes active, **ANSREASON** is always 0.	smallint	2 bytes
ASSIST	This item indicates if the answering agent in this segment requested supervisor assistance on this call. Valid values for **ASSIST** are 0=NO, 1=YES.	smallint	2 bytes
AUDIO	This item indicates if an agent in this segment reported an audio difficulty problem. Valid values for **AUDIO** are 0=NO, 1=YES.	smallint	2 bytes
CALLID	A unique number assigned to this call and all its call segments. For conferenced/transferred calls, two (or more) calls are tied together. When the entire call is recorded, one call ID is used to tie together all call segments. In "meet-me" conferences, this may result in a "later" segment of the call starting earlier than the first segment. Call IDs are not necessarily strictly sequential, but will be unique for calls over a day.	integer	4 bytes

Database Item	Description	Column Type	Length
ANSHOLDTIME	The total time, in seconds, the call was put on hold by the answering agent in this call segment. In agent-to-agent calls, **ANSHOLDTIME** is accrued for the answering agent if the agent puts the call on hold, but not for the other agent (who continues to accrue talk time). For Generic 2.1 switches, hold time is accrued only for ACD calls. For Generic 2.2 and Generic 3 switches, and DEFINITY ECS Release 5, hold time is accrued for any type of call.	integer	4 bytes
ANSLOCID	The location ID that is associated with the EXTENSION at which the answering agent logged in. The location ID is always 0 for Definity R7 and older switches. This item requires DEFINITY R8 switches through Definity R10, MultiVantage R11, and later switches.	smallint	2 bytes
ANSLOGIN	The login ID of the agent who answered the call in this segment. This field is blank for unmeasured extensions when EAS is not active.	char(9)	9 byte ASCII text string
ANSREASON	The reason code (0 through 9) associated with the answering agent's mode, if the agent is in the AUX mode. For agents in AUX on switches with releases prior to the ECS or switches that do not have EAS and reason codes active, **ANSREASON** is always 0.	smallint	2 bytes
ASSIST	This item indicates if the answering agent in this segment requested supervisor assistance on this call. Valid values for **ASSIST** are 0=NO, 1=YES.	smallint	2 bytes
AUDIO	This item indicates if an agent in this segment reported an audio difficulty problem. Valid values for **AUDIO** are 0=NO, 1=YES.	smallint	2 bytes
CALLID	A unique number assigned to this call and all its call segments. For conferenced/transferred calls, two (or more) calls are tied together. When the entire call is recorded, one call ID is used to tie together all call segments. In "meet-me" conferences, this may result in a "later" segment of the call starting earlier than the first segment. Call IDs are not necessarily strictly sequential, but will be unique for calls over a day.	integer	4 bytes

Database Item	Description	Column Type	Length
CALLING_II	The Information Indicator (II) digits associated with the call. These digits are a two-digit string provided by the ISDN to indicate the type of originating line of the caller. These digits supply information about the originator location; for example, pay phone, hospital, or prison. The column is blank if the call does not contain II digits. Available on the ECS and newer switches.	char(2)	2 byte ASCII text string
CALLING_PTY	The calling party identification, which is the Automatic Number Identification (ANI)/Station Identification (SID) (for Generic 2.2 or Generic 3 Version 4 switches or the ECS with ISDN ANI delivery), extension or trunk equipment location identifying the originator of the call. This field is blank if the trunk is not measured or, for internal calls, if the originating extension is not measured. (Up to 12 digits in this field.)	char(12)	12 byte ASCII string
CONFERENCE	This item indicates if the answering agent initiated a conference on this segment. Valid values for **CONFERENCE** are 0=NO, 1=YES. Available on the ECS, Generic 2.2 and Generic 3 switches.	smallint	2 bytes
CONSULTTIME	The amount of time an agent talked on any outbound call while in AUX work, ACW, or in Other with a call on hold. This includes the time the originating agent spent talking to the destination party while establishing a conference or transferring a call. (This is the time between presses of the transfer or conference button.) It includes wait time if the agent is calling a Vector Directory Number (VDN) or split/skill extension, but the wait time can be subtracted out by subtracting the **DISPTIME** item from **CONSULTTIME**.	integer	4 bytes
CWC1 through CWC5	The first, second, third, fourth, or fifth Call Work Code entered by an agent for the Call Segment. Items CWC1 through CWC5 are supported by the Multiple Call Work Codes per Call Record feature implemented in CMS R3V11 and later.	char(16)	16 byte ASCII text string
DA_QUEUED	This item indicates if the call was queued as a direct agent call Valid values for **DA_QUEUED** are 0=NO, 1=YES. Applies to the ECS and Generic 3 switches only.	smallint	2 bytes

Database Item	Description	Column Type	Length
DIALED_NUM	The number the caller dialed (up to 24 digits). This will be the VDN for inbound vectoring calls, blank for inbound calls without vectoring, and dialed digits for outbound calls.	char(24)	24 byte ASCII text string
DISPIVECTOR	The number of the first vector associated with the disposition VDN (DISPVDN).	smallint	2 bytes
DISPOSITION	This item represents the call disposition and indicates whether the call in the segment was: 1=connected (CONN, non-ACD call to a measured agent) 2=answered (ANS, split/skill or direct agent call answered by an agent) 3=abandoned (ABAN) 4=interflowed (IFLOW) 5=forced busy (FBUSY) 6=forced disconnect (FDISC) 7=other (OTHER)	smallint	2 bytes

Database Item	Description	Column Type	Length
DISPOSITION (contd)	A connected call is a non-ACD call to a measured agent for which CMS receives an indication that the call was connected. An answered call is any split/skill or direct agent ACD call for which CMS receives an indication that the call was answered by an agent and was not a phantom abandon. An abandoned call is any ACD call in which a caller hangs up before receiving an answer from an agent and for which CMS receives notification that the caller abandoned. Phantom abandons (**PHANTOMABNS**) are included as abandoned calls. Interflowed calls are calls that are interflowed to an off-switch destination. Forced busy calls are calls that CMS records as **BUSYCALLS** for the trunk group that carried them. These calls can be VDN calls that received a forced busy from the vector command or, on the ECS and G3 switches, a split/skill call for a nonvector-controlled split that received a busy indication from the switch because the split queue was full. For Generic 2.2, Generic 3 Version 2 and newer Generic 3 switches, and the DEFINITY ECS, forced disconnect calls are VDN calls that are disconnected by the switch due to the execution of a disconnect vector command. For the ECS, and Generic 3 Version 2 and later Generic 3 switches, forced disconnect calls also include calls disconnected because of the vector disconnect timer or because they reached the end of vector processing without being queued. For Generic 3 Version 1 switches, forced disconnect calls are calls that were given a forced disconnect announcement and listening to the entire announcement, then were disconnected by the switch. Other calls include any other calls that do not fall into categories such as answered or abandoned. See definitions for individual tables for **OTHERCALLS**.	smallint	2 bytes

Database Item	Description	Column Type	Length
DISPPRIORITY	The priority the call had at its disposition in this segment. Priorities can be 1=NO or 2=YES (without vectoring), or 3=LOW, 4=MED, 5=HIGH, or 6=TOP (with vectoring). If the call never gets queued to a split/skill, the priority will not be set. For the ECS, and Generic 3 switches with vectoring, calls directed to split/skills using “route to” or “messaging split/skill” commands and calls directly routed to splits/skills without going through a vector will have MED (no priority) or HIGH (priority) priority, depending on the class of restriction of the originator of the call (agent, extension, trunk group, or VDN).	smallint	2 bytes
DISPSKLEVEL	The skill level (1 through 16) associated with the skill for which the agent answered the call or, for calls that abandoned from ringing or from a direct agent queue, with the agent from whom the call abandoned.	smallint	2 bytes
DISPSPLIT	The number of the split/skill associated with the call at its disposition in this call segment. Calls that were not queued to a split or skill at the time of disposition will have **DISPSPLIT** set to null. Calls that were queued to an unmeasured split/skill at the time of disposition will have **DISPSPLIT** set to zero.	smallint	2 bytes
DISPTIME	The wait time (in the vector, in queue, and ringing) until the disposition is recorded in **DISPOSITION** for the segment. For extension calls made directly to agents (not through a VDN), this will always be zero.	integer	4 bytes
DISPVDN	The number of the VDN associated with the call at its disposition for this call segment. **DISPVDN** will be blank for calls that are not associated with a VDN at their disposition.	char(5)	5 byte ASCII text string
DURATION	The total time the trunk was in use. This is the overall trunk holding time from the beginning of the call segment until the caller is disconnected. For the first segment of a call, this will be the trunk holding time for the caller for the entire call (from seized until idle). With a transfer, the original trunk remains associated with both call segments until the call ends.	integer	4 bytes

Database Item	Description	Column Type	Length
EQLOC	The physical equipment location (trunk number) for which data was collected or for which the exception occurred. This will be blank if the trunk is not measured. For the DEFINITY ECS R8, this field is eight characters. For previous switch releases, it is nine characters.	char(8)	8 byte ASCII text string
EQLOCID	The switch location ID that is associated with the trunk. It can be 0 to 44. The location ID is not directly assigned to a trunk but rather is assigned to a port network location on the switch. Each trunk whose equipment location belongs to a specific port network will be associated with that port network's location ID.	smallint	2 bytes
EVENT1-9	The number of times each event (stroke count) button (buttons 1 to 9) was entered for this call segment. Available with the ECS, Generic 2, and Generic 3 switches.	smallint	2 bytes
FIRSTVECTOR	The number of the first vector associated with the first VDN for the call segment. This will be blank if no vector is involved.	smallint	2 bytes
FIRSTVDN	The number of the first VDN associated with the call segment. This will be blank for calls not associated with a VDN.	char(5)	5 byte ASCII text string
HELD	The total number of times this call was placed on hold by the answering agent in this call segment. With agent-to-agent calls, this count is incremented for the agent who puts the call on hold, but not for the calling agent. (For the ECS, Generic 2.2, and Generic 3 switches, applies to all calls the agent put on hold.)	smallint	2 bytes
HOLDABN	This item indicates if this call abandoned from hold in this call segment. Valid values for **HOLDABN** are 0=NO, 1=YES. With the ECS, Generic 2.2, and Generic 3 switches, this applies to all calls the agent put on hold.	smallint	2 bytes
LASTCWC	The last call work code (up to 16 digits) entered by the answering agent in this segment. This database item applies to Generic 2.2 and Generic 3 switches and the ECS only.	char(16)	16 byte ASCII text string

Database Item	Description	Column Type	Length
LASTDIGITS	The last set of collected digits sent to the CMS by the switch for this call. These are digits the switch sends to CMS when it executes a “collect” vector command. The digits may be digits the caller was prompted to enter, either through the prompting feature on the switch or through network-prompted digits (“caller-entered digits (CED)”), customer database-provided digits (“CDPD” from the network), or digits collected through a “converse” vector command. Available on ECS switches.	char(16)	16 byte ASCII text string
LASTOBSERVER	The login ID of the last agent who service-observed or bridged on to this call.	char(9)	9 byte ASCII text string
MALICIOUS	This item indicates if a malicious call trace was activated for this call segment. Valid values for **MALICIOUS** are 0=NO, 1=YES. Applies to the ECS, Generic 2, and Generic 3 (except Generic 3i Version 1) switches.	smallint	2 bytes
NETINTIME	The amount of time the call spent in a VDN processing at another switch located elsewhere in the network. Requires the DEFINITY *ECS R6*.	integer	4 bytes
OBSERVINGCALL	This item indicates whether this call represents an agent observing or bridging on to an existing call. Valid values for **OBSERVINGCALL** are 0=NO, 1=YES.	smallint	2 bytes
OBSLOCID	The location ID of an agent observing or bridging on to an existing call. OBSLOCID is available with DEFINITY R8 and later.	smallint	2 bytes
ORIGHOLDTIME	The total time the call was put on hold by the originating agent. Requires the DEFINITY *ECS R6*.	integer	4 bytes
ORIGLOCID	The location ID of the agent who is originating the call.	smallint	2 bytes
ORIGLOGIN	The login ID of the agent originating the call. This is used for calls an agent originates to another agent, to an on-switch extension, or to an external destination.	char(9)	9 byte ASCII text string

Database Item	Description	Column Type	Length
ORIGREASON	The reason code (0 through 9) associated with the originating agent's mode, if the agent is in the AUX mode. For agents in AUX on switches with releases prior to the ECS or switches that do not have EAS and reason codes active, **ORIGREASON** is always 0.	smallint	2 bytes
ROW_DATE	The date for which data was collected or the exception occurred.	date	4 byte Informix date
ROW_TIME	The starting time for this segment.	smallint	2 bytes
SEGMENT	The number identifying the call segment. Segment numbers are from 1 up to the number of segments in the call.	smallint	2 bytes
SEGSTART	The *UNIX* time and date when the call segment started. (*UNIX* time and date is the number of seconds since midnight, 01/01/70.) Call segments start when CMS receives the first message for the call, since each call segment represents a call. (When an agent transfers or conferences a call, the agent makes another call to bring about the transfer/conference.)	integer	4 bytes
SEGSTOP	The *UNIX* time and date when the call segment ended. (*UNIX* time and date is the number of seconds since midnight, 01/01/70.) A call segment ends when all trunks and agents associated with the call segment have dropped off the call. This means that after call work time for the agent(s) is included when calculating the call segment stop time.	integer	4 bytes
SEQNUM	The sequence number of this exception record.	integer	4 bytes
SPLIT1	The first split/skill the call queued to in the first VDN with which it was associated in the call segment.	smallint	2 bytes
SPLIT2	The second split/skill the call was also queued to in the first VDN with which it was associated in the call segment. Applies to Generic 2.2 with Expert Agent Selection (EAS), the ECS, and Generic 3 switches with vectoring only.	smallint	2 bytes
SPLIT3	The third split/skill the call was also queued to in the first VDN with which it was associated in the call segment. Applies to Generic 2.2 with EAS, the ECS, and Generic 3 switches with vectoring only.	smallint	2 bytes

Database Item	Description	Column Type	Length
TALKTIME	The total talk time for the answering agent in this segment.	integer	4 bytes
TKGRP	The trunk group number for which data was collected (or for which an exception occurred). This will be null if the trunk group carrying the call is not measured.	smallint	2 bytes
TRANSFERRED	This item indicates if an answering agent initiated a transfer for this call segment. Valid values are 0=NO, 1=YES. For the ECS, Generic 2.2, and Generic 3 switches, **TRANSFERRED** includes all calls that are transferred.	smallint	2 bytes
UCID	The UCID is the Universal Call Identifier—a unique number assigned to this call segment within the customer network. Requires the DEFINITY ECS R6.	char(20)	20-byte ASCII text string

ECS cross-reference

The following table lists which of the ECS releases support the Call Record database items.

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
ACD	X	X	X	X	X	X	X
ACWTIME	X	X	X	X	X	X	X
AGT_RELEASED	X	X	X	X	X	X	X
ANSHOLDTIME	X	X	X	X	X	X	X
ANSLOGIN	X	X	X	X	X	X	X
ANSREASON	X	X	X	X	X	X	X
ASSIST	X	X	X	X	X	X	X
AUDIO	X	X	X	X	X	X	X
CALLID	X	X	X	X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
CALLING_II			X	X	X	X	X
CALLING_PTY	X	X	X	X	X	X	X
CONFERENCE	X	X	X	X	X	X	X
CONSULTTIME	X	X	X	X	X	X	X
DA_QUEUED	X	X	X	X	X	X	X
DIALED_NUM	X	X	X	X	X	X	X
DISPIVECTOR	X	X	X	X	X	X	X
DISPOSITION	X	X	X	X	X	X	X
DISPPRIORITY	X	X	X	X	X	X	X
DISPSKLEVEL			X (EAS)	X	X	X	X
DISPSPLIT	X	X	X	X	X	X	X
DISPTIME	X	X	X	X	X	X	X
DISPVDN	X	X	X	X	X	X	X
DURATION	X	X	X	X	X	X	X
EQLOC	X	X	X	X	X	X	X
EQLOCID	X	X	X	X	X	X	X
EVENT1-9	X	X	X	X	X	X	X
FIRSTIVECTOR	X	X	X	X	X	X	X
FIRSTVDN	X	X	X	X	X	X	X
HELD	X	X	X	X	X	X	X
HOLDABN	X	X	X	X	X	X	X
LASTCWC	X	X	X	X	X	X	X
LASTDIGITS			X	X	X	X	X
LASTOBSERVER	X	X	X	X	X	X	X
MALICIOUS	X	X	X	X	X	X	X
NETINTIME				X	X	X	X

Database Item	G3V2/ G3V3	G3V4	ECS R5	ECS R6/R7	ECS R8	ECS R9/R10	MultiVantage R11
OBSLOCID					X	X	X
OBSERVINGCALL	X	X	X	X	X	X	X
ORIGHOLDTIME				X	X	X	X
ORIGLOCID				X	X	X	X
ORIGLOGIN	X	X	X	X	X	X	X
ORIGREASON	X	X	X	X	X	X	X
ROW_DATE	X	X	X	X	X	X	X
ROW_TIME	X	X	X	X	X	X	X
SEGMENT	X	X	X	X	X	X	X
SEGSTART	X	X	X	X	X	X	X
SEGSTOP	X	X	X	X	X	X	X
SEQNUM				X	X	X	X
SPLIT1	X	X	X	X	X	X	X
SPLIT2	X	X	X	X	X	X	X
SPLIT3	X	X	X	X	X	X	X
TALKTIME	X	X	X	X	X	X	X
TKGRP	X	X	X	X	X	X	X
TRANSFERRED	X	X	X	X	X	X	X
UCID				X	X	X	X

Data collection period database items

Overview

The **Administrative** data collection period database items apply to the Daily Data Collection Period (`d_secs`), Weekly Data Collection Period (`w_secs`), and Monthly Data Collection Period (`m_secs`) tables. All items listed in the following table are included in all three tables, unless otherwise noted in the Column Type column. Any differences in the data format between the three data collection period tables are also noted in the Column Type column. The indexes are **ROW_DATE** and **ACD.** The tables indicate the number of seconds in the data collection period (daily, weekly, monthly).

Because the data contained in the Data Collection periods tables are administrative, the table does not rely on ECS releases.

Data collection period database item table

The following table describes the data items in the CMS Data Collection Periods database tables.

Database Item	Description	Data Type	Column Type	Length
ACD (index)	The ACD number for which data was collected.	A	smallint	2 bytes
ROW_DATE	The day for which data was collected or the exception occurred.	A	date	4 byte Informix date
SECSPERDAY	The number of seconds in the daily data collection period	A	integer; only in d_secs table	4 bytes
SECSPERMN	The number of seconds in the monthly data collection period.	A	integer; only in m_secs table	4 bytes
SECSPERWK	The number of seconds in the weekly data collection period.	A	integer; only in w_secs table	4 bytes

Archiver execution status database items

Overview

The Archiver Execution Status database item descriptions apply specifically to items in the arch_stat table. The table contains status information on recent archiver executions, and displays status and the next run scheduled. The indexes are **ACD** and **ARCH_TYPE**. Because the data contained in the arch_stat table are administrative, the table is not dependent on ECS releases.

Archiver execution status database item table

The following table describes the data items in the CMS Archiver Execution Status database table.

Database Item	Description	Data Type	Column Type	Length
ACD (index)	The ACD number for which data was collected.	A	smallint	2 bytes
ARCH_TYPE (index)	The type of archiver executions being run. Values are: 1 = interval, 2 = daily, 3 = weekly, 4 = monthly.	A	char(20)	20 byte ASCII text string
LAST_TIME	The last time the archiver execution was run.	A	char(20)	20 byte ASCII text string
STATUS	The status of the archiver execution. This field indicates if the execution was not run, is currently running, or has finished. Values are: 1 = not run 2 = finished, the archive was successful 3 = finished, the archive had a failure 4 = running. If the status field displays 3, for finished, but had a failure, you should consult the error log (customer_log table) and the archive log for the reason prior to troubleshooting.	A	char(9)	9 byte ASCII text string

Customer log database items

Overview

The Customer Log database item descriptions apply specifically to items in the customer_log table. The table contains customer error log information on recent archiver executions. The information includes the error code, the date the error occurred, the severity, the associated event, and a description of the error. Because the data contained in the customer_log table are administrative, the table is not dependent on ECS releases. The customer_log table is not backed up by the CMS Maintenance backup.

Customer log database item table

The following table describes the data items in the CMS Customer Log database table.

Database Item	Description	Data Type	Column Type	Length
ACD_ID	The ACD number for which data was collected.	A	integer	4 bytes
COUNTS	The number of occurrences of the error.	A	integer	4 bytes
DATE_OCCURRED	The date that the error occurred.	A	date	4 byte Informix date
DESCRIPTION	A text description of the error.	A	char(256)	256 byte ASCII text string
ERROR_CODE	The error code number.	A	integer	4 bytes
LAST_TIME	The last time the error occurred.	A	integer	4 bytes
SEVERITY	The level of severity of the error.	A	char(10)	10 byte ASCII text string

Agent group database items

Overview

The Agent Group database items descriptions in the following able apply to items in the agroups table. The table contains dictionary information on agent groups. The indexes are ACD_NO, ITEM_TYPE, ITEM_NAME, and VALUE. because the data contained in the agroups table are administrative and are used to define agent groups, the table is not dependent on ECS releases.

Agent group database item table

The following table describes the data items in the CMS Agent Group database dictionary table.

Database Item	Description	Data Type	Column Type	Length
ACD_NO (index)	The ACD number for which data was collected.	A	smallint	2 bytes
ITEM_TYPE (index)	The type “agent group.”	A	char(20)	20 byte ASCII text string
ITEM_NAME (index)	The name of the agent group.	A	char(20)	20 byte ASCII text string
VALUE (index)	An agent login ID belonging to the agent group.	A	char(9)	9 byte ASCII text string

Synonyms database items

Overview

The Synonyms database item descriptions apply specifically to items in the synonyms table. The table contains dictionary synonyms. The indexes are **ACD_NO**, **ITEM_TYPE**, **ITEM_NAME**, and **VALUE**. Because the data contained in the synonyms table are administrative, and are used to define dictionary entries, the table is not dependent on ECS releases.

Synonyms database item table

The following table describes the data items in the CMS Synonyms database dictionary table.

Database Item	Description	Data Type	Column Type	Length
ACD_NO (index)	The ACD number for which data was collected.	A	smallint	2 bytes
DESCR	The description, or definition, of the dictionary synonym.	A	char(150)	5 byte ASCII text string
ITEM_TYPE (index)	The type of synonym. The following synonyms are used in reports: **TypeDatabase Item** acdACD agnameLOGID (login ID) tkgrpTKGRP (trunk group) splitSPLIT vdnVDN (vector directory number)	A	char(20)	20 byte ASCII text string

Database Item	Description	Data Type	Column Type	Length
ITEM_TYPE (index) **(contd)**	vectorVECTOR aux_rsnAUXREASON (AUX work state reason) logout_rsnLOGOUTREASON (agent logout reason) cwcCWC (call work code) workmodeWORKMODE (agent work mode in both Agent and Agent Trace tables) ag_originORIGIN ag_dirDIRECTION ag_destDESTINATION ag_prefPREFERENCE tkstateTKSTATE (trunk state) tk_priPRIORITY tk_queueQUETYPE (AUX work state reason) tk_vpriPRIORITY (only available with vectoring) tk_dirDIRECTION all_busyALLINUSE slvl_chgSVCLEVELCHG per_chgPERIODCHG	A	char(20)	20 byte ASCII text string
ITEM_NAME (index)	The name of the synonym. There can be many **ITEM_NAME**s for a specific **ITEM_TYPE**.	A	char(60) Only first 20 are significant	60 byte ASCII text string. Only first 20 are significant
STANDARD	This item indicates if the item is a standard or custom synonym; Values are: 1 = standard, not 1 = custom.	A	smallint	2 bytes
VALUE (index)	The item name's corresponding value. Because each **ITEM_TYPE** can have many different **ITEM_NAME**s, a discrete value is assigned to each synonym **ITEM_NAME**.	A	char(40) Only first 9 are significant	40 byte ASCII text string. Only first 20 are significant

ACD shifts database items

Overview

The ACD shifts database item descriptions apply specifically to items in the acd_shifts table. The table contains information on ACD shift times and the maximum number of agents logged in for each shift. The indexes are **ACD** and **SHIFT_ID**. Because the data contained in the acd_shifts table are administrative, the table is not dependent on ECS releases.

ACD shifts database item table

The following table describes the data items in the CMS ACD Shifts database dictionary table.

Database Item	Description	Data Type	Column Type	Length
ACD (index)	The ACD number for which data was collected.	A	smallint	2 bytes
SHIFT_ID	The identification number of the ACD shift. Values are 1 - 4.	A	smallint	2 bytes
START_TIME	The ACD shift start time.	A	smallint	2 bytes
STOP_TIME	The ACD shift stop time.	A	smallint	2 bytes
MAX_AGENTS	The maximum number of agents logged in per shift.	A	smallint	2 bytes

Database items

Overview

The Database Items descriptions apply specifically to items in the dbitems table. The table contains definitions for Dictionary standard and custom database items, constants, and calculations. The index is **ITEM_NAME**. Because the data contained in the dbitems table are administrative, and are used to define dictionary entries, the table is not dependent on ECS releases.

Database item table

The following table describes the data items in the CMS Database Items dictionary table.

Database Item	Description	Data Type	Column Type	Length
ITEM_TYPE (index)	The type of data for the row. Valid values are: dbase = database item calc = calculation constant = constant cust_def = customer-defined database item	A	char(8)	8 byte ASCII text string
ITEM_NAME (index)	The name of the data item. There can be many **ITEM_NAME**s for a specific **ITEM_TYPE**.	A	char(20)	20 byte ASCII text string
FORMULA	The formula for the database constant or calculation.	A	char(70)	70 byte ASCII text string
STANDARD	This item indicates if the item is a standard or custom database item. Values are: 1 = standard, not 1 (null) = custom.	A	smallint	2 bytes
DESCR	The description of the database calculation, constant, or standard/custom database item.	A	char(50)	50 byte ASCII text string

Exceptions historical database items

Overview

In the tables in this section, the database item **EXTYPE** lists numerical values associated with exception types. The database item **REASON** in the Database Collection Exceptions Database Items table lists numerical values associated with exception types. CMS stores exception types using the numerical values, then translates the numbers into the text you see in standard exception reports.

To select specific exception types for a custom report, you must enter the numerical value(s) in the Select rows where: statement.

Contents

"Exceptions historical database items" contains the following topics:

Agent exceptions database item table

The following table describes the historical Agent Exceptions database items. They apply to the Agent Exceptions (agex) table.

Database Item	Description	Column Type	Length
ACD	The ACD number for which data was collected.	smallint	2 bytes
AGLOCID	The equipment location ID that is associated with a particular agent. This is the location ID of the terminal the agent is logged into. It is associated with a port network location ID on the switch.	smallint	2 bytes
EXTYPE	The type of exception that occurred: **Numerical Value Type** 1 Time Available 2Time on inbound ACD call (min) 3 Time on inbound ACD call (max) 4 Time in after call work 5 Time on outbound ACW call 6 Time on inbound ACW call 7 Time in AUX work 8 Time on outbound AUX call 9 Time on inbound AUX call 10 Number of outbound ACW calls/agent 11 Number of inbound ACW calls/agent 12 Number of outbound AUX calls/agent 13 Number of inbound AUX calls/agent 14 Login identification 15 Time ACD call spent on hold*	smallint	2 bytes

Database Item	Description	Column Type	Length
EXTYPE (Contd)	16 Number ACD calls placed on hold* 17 Number ACD calls abandoned while on hold* 18 Time on outbound ACD call (min) 19 Time on outbound ACD call (max)† 20 Number calls transferred** 21 Time on external outbound ACW call† 22 Time on external outbound AUX call† 23 Time on direct agent call‡ 24 Number external outbound ACW calls/agent** 25 Number external outbound AUX calls/agent** 26 Time ACD call spends ringing** 27 Multiple logins on same extension 28 Ringing call automatically redirected from agent 29 Agent logged out with active/held calls 30 Number of calls in direct agent queue‡ 31 Time call waited in direct agent queue‡ 32 Number calls abandoned from direct agent queue‡ 34 Number calls outflowed from direct agent queue‡ 38 Number of calls transferred 48 Logout attempt without valid reason code 49 Could not be logged in 59 AUX attempt without valid reason code 60 Time in AUX with reason code 0 (default) 61 Time in AUX with reason code 1 62 Time in AUX with reason code 2 63 Time in AUX with reason code 3 64 Time in AUX with reason code 4 65 Time in AUX with reason code 5 66 Time in AUX with reason code 6 67 Time in AUX with reason code 7 68 Time in AUX with reason code 8 69 Time in AUX with reason code 9 98 Agent denied login to some skills 99 Invalid call work code	smallint	2 bytes

Database Item	Description	Column Type	Length
LOGID	The login ID of the agent who had the exception.	char(10)	10 byte ASCII text string
REASON_ CODE	The reason code that the agent was in when the exception occurred.	smallint	2 bytes
ROW_DATE (index)	The day for which data was collected or the exception occurred.	date	4 byte Informix date
ROW_TIME	The time at which the exception occurred.	smallint	2 bytes
SEQNO	The sequence number of this exception record.	integer	4 bytes
SPLIT	The split/skill in which the agent was doing work when the exception occurred.	smallint	2 bytes
THRESHOLD	The limit, as a number of occurrences, administered for the exception type. An exception occurs when the agent's activity falls outside of that limit.	smallint	2 bytes
TIME	The limit, as a number of seconds, administered for timed exceptions types. An occurrence is logged against the threshold when the agent's activity falls outside of that limit.	smallint	2 bytes

* Available only on Generic 2 and Generic 3 ECS.

† Available only with Generic 2.2 ECS with the ASAI Gateway Interface feature and Generic 3 ECS with the ASAI feature.

** Available only with Generic 2.1 or later and Generic 3 ECS.

‡ Available only with Generic 3 ECS with the ASAI or EAS feature.

Disk full exceptions database item table

The following table describes the historical Disk Full Exceptions database items. They apply to the Disk Full Exceptions (`fullex`) table.

Database Item	Description	Column Type	Length
PROC_NAME	The name of the process that failed because the disk was full.	char(30)	30 byte ASCII text string
ROW_DATE	The date at which the disk was full.	date	4 byte Informix date
ROW_TIME	The time at which the disk was full.	smallint	2 bytes
SEQNO	The sequence number of this record.	smallint	2 bytes
TASK_GRP	The activity that failed because the disk was full.	char(2)	2 byte ASCII text string

Split/Skill exceptions database item table

The following table describes the historical Split/Skill Exceptions database items. They apply to the Split/Skill Exceptions (spex) table.

Database Item	Description	Column Type	Length
ACD	The ACD number for which data was collected.	smallint	2 bytes
EXTYPE	The type of exception that occurred: **Numerical Value — Type** 30 Number calls waiting 31 Time call has waited in queue 32 Number calls abandoned 33 Number intraflowed-in calls 34 Number intraflowed-out calls 35 Number interflowed-out calls 36 Number calls offered while queue full* 37 Number calls handled as backup† 38 Number calls transferred** 39 Average speed of answer (seconds) 40 Rolling average speed of answer (seconds) 41 Expected wait time (priority top) 42 Expected wait time (priority high) 43 Expected wait time (priority medium) 44 Expected wait time (priority low)	smallint	2 bytes
ROW_DATE (index)	The day for which data was collected or the exception occurred.	date	4 byte Informix date
ROW_TIME	The time at which the exception occurred.	smallint	2 bytes
SEQNO	The sequence number of this exception record.	integer	4 bytes
SPLIT	The split/skill in which the exception occurred.	smallint	2 bytes
THRESHOLD	The limit, as a number of occurrences, administered for the exception type. An exception occurs when the split's/skill's activity falls outside of that limit.	smallint	2 bytes
TIME	The limit, as a number of seconds, administered for timed exception types. An occurrence is logged against the threshold when the split's/skill's activity falls outside of that limit.	smallint	2 bytes

* Available only on Generic 2 and Generic 3 ECS

† Available only with Generic 2.2 ECS with the ASAI Gateway Interface feature and Generic 3 ECS with the ASAI feature.

** Available only with Generic 2.1 or later and Generic 3 ECS.

Trunk group exceptions database item table

The following table describes the historical Trunk Group Exceptions database items. They apply to the Trunk Group Exceptions (`tgex`) table.

Database Item	Description	Column Type	Length
ACD (index)	The ACD number for which data was collected.	smallint	2 bytes
AGLOCID	The equipment location ID that is associated with a particular agent. This is the location ID of the terminal the agent is logged into. It is associated with a port network location ID on the switch.	smallint	2 bytes
EXTYPE	The type of exception that occurred: **Numerical Value Type** 50 Time trunk in use (min) 51 Time trunk in use (max) 52 Number of trunks in use 53 Time any trunk maintenance busy 54 Number of trunks maintenance busy 55 Length of time all trunks busy 56 Number trunk failures in group 57 Number failures on a single trunk 58 Audio difficulty on a trunk	smallint	2 bytes
EQLOC	The physical equipment location (trunk number) for which data was collected or the exception occurred. For the DEFINITY ECS R8, this field is eight characters. For previous switch releases, it is nine characters.	char(9)	9 byte ASCII text string
EQLOCID	The switch location ID that is associated with the trunk. It can be 0 to 44. The location ID is not directly assigned to a trunk but rather is assigned to a port network location on the switch. Each trunk whose equipment location belongs to a specific port network will be associated with that port network's location ID.	smallint	2 bytes
LOGID	The login ID of the agent reporting audio difficulty.	char(10)	10 byte ASCII text string

Database Item	Description	Column Type	Length
ROW_DATE	The day for which data was collected or the exception occurred.	date	4 byte Informix date
ROW_TIME	The time at which the exception occurred.	smallint	2 bytes
SEQNO	The sequence number of this exception record.	integer	4 bytes
THRESHOLD	The limit, as a number of occurrences, administered for the exception type. An exception occurs when the trunk group's activity falls outside of that limit.	smallint	2 bytes
TIME	The limit, as a number of seconds, administered for timed exception types. An occurrence is logged against the threshold when the trunk group's activity falls outside of that limit.	smallint	2 bytes
TKGRP	The trunk group number for which data was collected (or for which an exception occurred). This will be zero if the trunk group carrying the call is not measured.	smallint	2 bytes

VDN exceptions database items

The following table describes the historical VDN Exceptions database items. They apply to the VDN Exceptions (vdnex) table. VDN exceptions are available only with the Vectoring feature.

Database Item	Description	Column Type	Length
ACD (index)	The ACD number for which data was collected.	smallint	2 bytes

Database Item	Description	Column Type	Length
EXTYPE	The type of exception that occurred: **Numerical Value Type** 2 Time at agent (min) 3 Time at agent (max) 30 Number calls in an ACD split queue 32 Number calls abandoned while in vector 33 Number calls that flowed into VDN 34 Number calls that flowed out of VDN 35 Number calls interflowed out of VDN 37 Number calls handled by backup split 71 Time in vector (max) 72 Number calls forced busy 73 Number calls disconnected 74 Number unsuccessful lookahead attempts 75 Adjunct routing 76 Rolling average speed of answer	smallint	2 bytes
ROW_DATE	The day for which data was collected or the exception occurred.	date	4 byte Informix date
ROW_TIME	Time at which the exception occurred.	smallint	2 bytes
SEQNO	The sequence number of this exception record.	integer	4 bytes
THRESHOLD	Limit, as a number of occurrences, administered for the exception type. An exception occurs when the VDN activity falls outside of that limit.	smallint	2 bytes
TIME	Limit, as a number of seconds, administered for timed exceptions types. An occurrence is logged against the threshold when the VDN activity falls outside of that limit.	smallint	2 bytes
VDN	VDN for which the exception occurred or that carried the malicious call.	char(6)	6 byte ASCII text string
VECTOR	Vector number associated with this VDN or for which the exception occurred.	smallint	2 bytes

Vector exceptions database item table

The following table describes the historical Vector Exceptions database items. They apply to the Vector Exceptions (vecex) table. Vector exceptions are available only with the Vectoring feature.

Database Item	Description	Column Type	Length
ACD (index)	The ACD number for which data was collected.	smallint	2 bytes
EXTYPE	The type of exception that occurred: **Numerical Value Type** 30 Number calls in an ACD split/skill queue 32 Number calls abandoned while in the vector 72 Number calls forced busy 73Number calls disconnected 74Number unsuccessful lookahead interflow attempts 75 Number unsuccessful adjunct routing attempts 80Time in vector (min) 81Time in vector (max)	smallint	2 bytes
ROW_DATE	The date for which data was collected or which the exception occurred.	date	4 byte Informix date
ROW_TIME	The time at which the exception occurred.	smallint	2 bytes
SEQNO	The sequence number of this exception record.	integer	4 bytes
THRESHOLD	The limit, as a number of occurrences, administered for the exception type. An exception occurs when the vector activity falls outside of that limit.	smallint	2 bytes
TIME	The limit, as a number of seconds, administered for timed exceptions types. An occurrence is logged against the threshold when the vector activity falls outside of that limit.	smallint	2 bytes
VECTOR	The vector number that this row represents, or for which the exception occurred.	smallint	2 bytes

Malicious call trace exceptions database item table

The following table describes the historical Malicious Call Trace Exceptions database items. The exception type will always be 90 = Malicious Call. They apply to the Malicious Call Trace Exceptions (`mctex`) table.

Database Item	Description	Column Type	Length
ACD (index)	The ACD number for which data was collected.	smallint	2 bytes
AGLOCID	The equipment location ID that is associated with a particular agent. This is the location ID of the terminal the agent is logged into. It is associated with a port network location ID on the switch.	smallint	2 bytes
ANI_SID	The billing number or phone number from which the malicious call originated (available only if the switch has ANI/SID service).	char(8)	8 byte ASCII text string
EQLOC	The physical equipment location (trunk number) for which data was collected or for which the exception occurred.	char(9)	9 byte ASCII text string
EQLOCID	The switch location ID that is associated with the trunk. It can be 0 to 44. The location ID is not directly assigned to a trunk but rather is assigned to a port network location on the switch. Each trunk whose equipment location belongs to a specific port network will be associated with that port network's location ID.	smallint	2 bytes
II_DIGITS	The type of exception that occurred. The value for Malicious Call Trace exceptions is 90.	smallint	2 bytes
LOGID	The Information Indicator digits. Indicates type of originating line the call used.	char(10)	10 byte ASCII text string
ROW_DATE	The login ID of the agent initiating a malicious call trace.	date	4 byte Informix date
ROW_TIME	The date for which data was collected or the exception occurred.	smallint	2 bytes
SEQNO	The sequence number of this exception record.	integer	4 bytes

Database Item	Description	Column Type	Length
SPLIT	The split/skill of the agent reporting the malicious call.	smallint	2 bytes
TKGRP	The trunk group number for which data was collected (or for which an exception occurred). This will be zero if the trunk group carrying the call is not measured.	smallint	2 bytes
VDN	The VDN for which the exception occurred or that carried the malicious call. Available on Generic 2, Generic 3, and newer switches with vectoring.	char(6)	6 byte ASCII text string

Data collection exceptions database item table

The following table describes the historical Data Collection Exceptions database items. They apply to the Data Collection Exceptions (`linkex`) table.

Database Item	Description	Column Type	Length
ACD (index)	The ACD number for which data was collected.	smallint	2 bytes
DURATION	The length of time for which data collection was off.	integer	4 bytes
REASON	The reason for the interruption of data collection. The reasons may be as follows: **Numerical Value Reason** 91 Data collection started 92 Data collection of new translations started 93 Data collection turned off 94 Data collection busied out 95 Data collection timed out 96 Data collection clock was reset 97 Data collection session down	smallint	2 bytes
ROW_DATE	The day for which data was collected or the exception occurred.	date	4 byte Informix date
ROW_TIME	The time at which data collection was interrupted.	smallint	2 bytes
SEQNO	The sequence number of this exception record.	integer	4 bytes
THRESHOLD	The limit, as a number of occurrences, administered for the exception type. An exception occurs when the activity falls outside of that limit.	smallint	2 bytes

Glossary

Abandoned Call

A call in which a caller hangs up before receiving an answer from an agent. The call could be queued to a split/skill or in a vector/vector directory number (VDN) or ringing at an agent before it is abandoned.

Abandoned Call Search

An Automatic Call Distribution (ACD) capability that enables the system to make sure on certain trunk types that the caller is on the line before passing the call to an agent.

Acceptable Service Level

A target value set to define the acceptable amount of time for an agent to answer a call. Target values are normally set as objectives by management.

A percentage of calls answered within a set amount of time (for example, 80 percent of calls answered within 20 seconds).

Access Permissions

Permissions assigned to a Call Management System (CMS) user so that the user can access different subsystems in CMS or administer specific elements (splits/skills, trunks, vectors, and so on) of the ACD. Access permissions are specified as read or write permission. Read permission means the CMS user can access and view data (for example, run reports or view the Dictionary subsystem). Write permission means the CMS user can add, modify, or delete data and execute processes.

ACD

See *Automatic Call Distribution*.

ACD Call

A call that queued to a split/skill and was answered by an agent in that split/skill, or a call that queued as a direct agent call and was answered by the agent for whom it was queued.

Acknowledgement

A window that requires you to confirm an action or to acknowledge a system message (for example, system going down, warning, or fatal error for the user window). This window cannot be moved, sized, or scrolled and disappears only when you confirm the message.

Action List

A menu in the upper-right corner of most user windows. The menu lists the actions available for that particular user window (for example, add, modify, delete, and so on). You select an action after entering necessary data in the user window.

Activate Agent Trace

From this window you can start CMS tracing of agent activities. These activities include all agent state changes until the trace is turned off. You can activate traces for a maximum of 100 agents for an *Intel* computer, or 250 agents for a *Sun* computer, at any one time. You must activate an agent trace to obtain an Agent Trace report.

Active VDN Calls — Also known as counted-calls to a VDN. A Call Vectoring capability available with G3V4 or later switches. Counted-calls to a VDN is a parameter of the “go to step” and “go to vector” commands that provides conditional branching (to a different step in the same vector or to a different vector) based on the number of incoming trunk calls a VDN is currently processing in a vector or at an agent.

ACW — See *After Call Work*.

Add — A CMS Action that adds the data entered in the given window to the database.

Adjunct/Switch Applications Interface (ASAI) — An open application interface through which processors and switches can jointly provide services that require applications to initiate, receive, and control calls or make use of switch features. (See *Open Application Interface*.)

After Call Work (ACW) — An agent state generally representing work related to the preceding ACD call. Going on-hook after an ACD call during MANUAL-IN operation places the agent in ACW. With Generic 3, ACW is also accessible by a button on the agent's set and may not be related to an ACD call.

Agent — A person or Voice Response Unit (VRU) port that answers calls to an ACD split/skill. The agent is known to CMS by a login identification keyed into a voice terminal.

Agent Login ID — A 1- to 4-digit number (Generic 2) or a 1- to 9-digit number (Generic 3) entered by an ACD agent from a voice terminal to activate the agent’s position. Agent logins are required for all CMS-measured ACD agents.

Agent Occupancy — The percentage of time that you are expecting or targeting for each split/skill agent to spend, while logged in, on ACD calls and in ACW.

Agent Position (EAS) — The combination of the agent login ID and the skills the agent is assigned. Data are collected for the agent by skill, so the total work for the agent must be summed over all skills in which the agent worked.

Agent Position (Non-EAS) — The combination of agent login ID and split the agent logged into. Agents logged into multiple splits have multiple positions associated with them. Call data are collected separately for each agent/split combination.

Agent Role

A description of the kind of service an agent in multiple skills gives to one of their skills. This is a combination of call handling preference and skill/reserve levels. The five roles are:

- *Top*: top agents logged into their highest priority skill.
- *Allocated*: agents with percent allocation call handling preference administered (see the *Avaya Advocate User Guide*, 585-210-711).
- *Backup*: agent is assigned to a skill, but not as the top skill.
- *Roving*: an agent answers the skill's calls when this skill has the greatest need.
- *Reserve*: an agent who normally does not answer calls for this skill answers calls in the skill because the skill has surpassed its pre-set over-threshold conditions (see the *Avaya® Advocate User Guide*, 585-210-711).

Agent Site Tracking Feature

Agent Site Tracking provides reporting on a location/site basis. It enables a customer with the Multi-Location feature to associate each agent with a particular location ID upon login to be able to run reports that provide data on agent activity at a particular location.

Agent Skill

An attribute that is associated with an ACD agent. Agent Skills can be thought of as the ability for an agent with a particular set of skills to handle a call that requires one of a set of skills. An agent can be assigned up to 20 skills. The meaning of each Agent Skill is defined by the customer. Examples of what could be considered skills are: the ability to speak a particular language or the expertise to handle a certain product.

See *Primary Skill, Secondary Skill*, and *Skill Level*.

Agent State

A feature of agent call handling. Agent states are the different call work modes and call states an agent can be in (ACD, ACW, AVAIL, AUX, UNSTAFF, DACD, DACW, OTHER, UNKNOWN, RING). Data about these states is displayed in real-time and historical reports.

See the definition of each state for additional information.

Agent Terminal

The voice terminal used by a call center agent.

Agent Trace

You must start an agent trace before you can obtain an Agent Trace report. You can select the dates that the report will cover. This report lists each agent activity and the time it occurred.The Agent Trace report can be helpful when evaluating how well individual agents are using their time.

AI

See *Auto-In*.

Algorithm

A prescribed set of well-defined rules or instructions for the solution of a problem; for example, the performance of a calculation, in a finite number of steps. Expressing an algorithm in a formal notation is one of the main parts of a software program.

ANI

See *Automatic Number Identification*.

Announcement	A recorded voice message that normally tells the caller what destination the call has reached. The announcement also often tries to persuade the caller to stay on the line. With Call Vectoring, announcements can be part of a vector's call processing. An announcement is assigned to a vector by entering an announcement number.
Answered Call	The agent's state changes to ACD or Direct ACD (DACD). The term answered is used only for split/skill and direct agent ACD calls. (See **Connected** for non-ACD calls.) For manual answer agents, the call is answered when the agent selects the ringing line appearance. For automatic answer agents, the call is answered directly after the zip tone is applied.
API	See *Application Programming Interface*.
Application Programming Interface (API)	A set of related functions that a computer programmer uses to obtain some kind of service from another piece of software. Programmers of Windows based applications use the Windows API to create windows, draw text on the screen, access files, and perform all other services provided by Windows. Despite the use of the word application in this term, applications might not be the only software using an API; lower-level software components such as network drivers also have APIs, but these components are not "applications" and are not used directly by applications.
ASA	See *Average Speed of Answer*.
ASAI	See *Adjunct/Switch Applications Interface*.
Asynchronous Transport Mode (ATM)	A high-speed, connection-oriented switching and multiplexing technology that uses 53 byte cells (5-byte header, 48-byte payload) to transmit different types of traffic simultaneously, including voice, video, and data.
Auto-Available Split	An ACD capability that ensures that after a power failure or a system restart, Voice Response Units (for example, the CONVERSANT® Voice Information System) are brought on line again immediately, without time-consuming reprogramming.
Auto-In (AI)	An ACD work mode that makes the agent available to receive calls and allows the agent to receive a new ACD call immediately after disconnecting from the previous call.
Automatic Call Distribution (ACD)	A switch feature using software that channels high-volume incoming and outgoing call traffic to agent groups (splits or skills). Also an agent state where the extension is engaged on an ACD call. See *Redirect On No Answer* and *Auto-Available Split*.
Automatic Number Identification (ANI)	A general industry term referring to knowledge of the calling party number (CPN). When the calling party is behind a switch, the number provided can be either a billing number for the switch or the station identification (SID) number.
AUX	See *Auxiliary Work*.

AUX Reason Codes

AUX reason codes enable a call center to track an agent's time more precisely when the agent is in the AUX state. The agent can specify exactly why the AUX state is used, such as lunch or meetings.

Auxiliary Work (AUX)

An agent work mode. For example, the agent is engaged in non-ACD work, is on break, in a meeting, or at lunch. An agent can reach this state by pressing the AUX WORK button or dialing the proper access code from the voice terminal. The agent can also reach the state by going off-hook to make or answer an extension call while in AVAIL or with a call on hold while in AI/MI mode.

AVAIL

See *Available*.

Available (AVAIL)

An agent state in which the extension is able to accept an ACD call. The agent enters this state by selecting the AI or MI work mode.

Average Agent Service Time

The average time you are expecting or targeting each agent to spend on an ACD call, including talk time and after-call-work time.

Average Speed of Answer (ASA)

The average amount of time a caller waits before connecting to an agent. ASA is usually an objective set by your call center's management.

The actual ASA for a split/skill includes the time spent in queue and the time ringing an agent. ASA for a VDN includes the time spent in vector processing, in queue, and the time ringing an agent.

Interval ASA is used for Basic Call Management System (BCMS) and CMS reporting where the ASA is calculated on reporting interval boundaries and the ASA is cleared to zero at the start of each reporting interval.

See also *Rolling ASA*.

B Channel

An Integrated Services Digital Network (ISDN) channel that carries voice and circuit-switched data at a bandwidth of 64,000 bits per second. It will carry packet data at 19,200 bits per second (dial-up) or 64,000 bits per second (dedicated). "B" is an abbreviation of Bearer Channel.

Backup

The process of protecting data by writing the contents of the disk to an archive (or tape) that can be removed from the computer environment and stored safely.

Basic Call Management System (BCMS)

Call center management information available on the G3 switches and the Enterprise Communications Server (ECS) that can provide statistics on a limited number of agents, splits/skills, trunk groups and VDNs.

Basic Rate Interface

An ISDN channel arrangement that provides multiple channels over the existing two-wire local loop. This service has up to three channels: one D and a maximum of two B channels.

Best Service Routing (BSR)

A method of automatic call distribution between switches based on expected wait time. BSR can be used as a single-site as well as a multi-site feature.

BCMS See *Basic Call Management System*.

BRI See *Basic Rate Interface*.

BSR See *Best Service Routing*.

Calculation The abbreviated name (calculation name) for the formula calculation that generates the data for a field in a report.

Call-Based Items The category of database items in CMS that are committed to the database after the call completes. If a call starts and ends in different intrahour intervals, all of the call-based data is recorded in the interval in which the call completed. Most database items are call-based.

Call Handling Preference A parameter of agent administration in the EAS environment that specifies how calls are selected for the agent.

Call-Handling Profile A set of objectives describing how you want a split/skill to handle calls. Call-handling profiles are part of the Forecasting subsystem.

Call Management System (CMS) A software product used by business customers that have Avaya telecommunications switches/ECS and receive a large volume of telephone calls that are processed through the Automatic Call Distribution (ACD) feature of the switch/ECS. The CMS collects call-traffic data, formats management reports, and provides an administrative interface to the ACD feature in the switch/ECS.

Call Management System Query Language (CMSQL) A tool that allows direct queries of the historical database. This tool is the interactive interface typically used to view the Informix database. For CMS purposes, CMSQL is used instead of Informix SQL.

Call Prompting A switch feature that routes incoming calls based on information entered by the calling party, such as an account number. The caller receives an announcement and is prompted to select an option from those listed in the announcement or to enter numeric information using the telephone dial pad.

Call Vectoring A switch feature that provides a highly flexible method for processing ACD calls.

A call vector is a set of instructions that controls the routing of incoming and outgoing calls based on current conditions. Examples of call vector conditions include time of day and the number of calls in queue.

Call Work Code (CWC) An ACD capability that allows the agent to enter a string of digits during or after the call and send them to CMS for management reporting.

Caller Needs The reasons a customer calls your call center.

Calls Carried Trunk data. The number of inbound/outbound calls that occupied the trunk during a given time period.

Cancel A CMS action that cancels the current Backup, Restore, or Migration that is in progress.

Caution Condition A graph term. The bars in the graph report change color or intensity indicating that the data being displayed met your defined first threshold limit.

Central Office (CO) The location housing telephone switching equipment that provides local telephone service and access to toll facilities for long-distance calling.

Avaya Supervisor The Call Management System application for the *Microsoft Windows* operating environment.

Change Agent Skills An agent's skills may be changed from CMS by using the Change Agent Skills Window. Through this window a user can quickly see what skills are currently assigned to a single agent and the user can easily change a single agent's skill assignment.

Close Users can close a window using the Close command on the Control menu. Closing an application's main window is the same as quitting it.

CMS See *Call Management System*.

CMS Large One of two hardware configurations for the CMS product line. This configuration has 128MB RAM, 4GB of disk storage space, and one processor.

CMS Options The optional features available with CMS. CMS options include graphics, forecasting, and vectoring.

CMS X-Large One of two hardware configurations for the CMS product line. This configuration has 128MB RAM, 4GB of disk storage space, and two processors.

CO See *Central Office*.

Configuration The way that the computer is set up to allow for particular uses or situations.

CONN See *Connected*.

Connected (CONN) A trunk state in which a caller and an agent are connected on an ACD call.

Connected Call A non-ACD call (that is connected to an agent through a VDN) for which CMS receives an indication that the call rang or was answered.

Continue A CMS action that resumes the suspended Restore or Migration operation.

CONVERSANT CONVERSANT® is a powerful voice response system which interacts with the caller. This system may include: automated call routing, announcement storage, message retrieval, and callback.

Copy Group or Profile A CMS action that creates a new group or profile from an existing one.

Current A CMS screen-labeled key (SLK) that allows the user to cycle through open windows.

Current Interval Represents the current intrahour interval, which can be 15, 30, or 60 minutes. The current interval is part of the real-time database. CMS starts collecting ACD cumulative data at the beginning of the interval (on the hour, half-hour, or quarter hour) and continues collecting ACD cumulative data until the end of the interval. When the current interval changes, all cumulative data is cleared and CMS begins counting cumulative data again starting from zero. The length of the interval is set in the System Setup: Storage Intervals window and is called the *intrahour interval*.

Current Wait Time The time a call has waited for service in a call queue adjusted for queue priority.

Current Window The user window in which you are actively working.

CWC See *Call Work Code*.

D Channel An ISDN channel that transmits at 16,000 bits per second. The signaling information connects, monitors, and disconnects calls. It can also carry low-speed packet data at 9,600 bits per second. “D” is an abbreviation of Data Channel.

DABN See *Dequeued and Abandoned*.

DACD See *Direct Agent ACD*.

DACW See *Direct Agent ACW*.

Daily Data Interval data that has been converted to a 1-day summary.

Data Collection Off CMS is not collecting ACD data. Data already collected will not be lost when turning data collection off.

Data Collection On CMS is collecting ACD data.

Data Points Dates of historical data used in Forecasting. A data point should include data for each interval of the working day.

Database A group of files that store ACD data according to a specific time frame: current and previous intrahour real-time data and intrahour, daily, weekly, and monthly historical data.

Database Item A name for a specific type of data stored in one of the CMS databases. A database item may store ACD identifiers (split numbers or names, login IDs, VDNs, and so on) or statistical data on ACD performance (number of ACD calls, wait time for calls in queue, current states of individual agents and so on).

Database Management System (DBMS)

The software that manages access to structured data. For example, the Microsoft SQL Server is a database management system. Database management system can also be used generally to include PC database products such as Microsoft Access, as well as any other software that can provide data access services.

Database Tables

CMS uses these tables to collect, store, and retrieve ACD data. Standard CMS items (database items) are names of columns in the CMS database tables.

Date Format

The standard format for entering dates on CMS reports. Acceptable formats are:

- Month/day/year (for example, 3/21/93)
- A "-" offset based on today's date (for example, -1 for yesterday)
- Separating individual data entry items using a semicolon (for example, 3/21/93;3/23/93;3/25/93)
- Entering ranges by placing a hyphen between entries (for example, 3/21/93-3/25/93).

When you specify a date for a weekly report, that date must correspond to the week start day selected in the System Setup—Storage Intervals window. If the date and day do not match, the message *No records found* displays in the status line.

The month start date must be the first day of the month.

DBMS

See *Database Management System*.

DDC

See *Direct Department Calling*.

Default Skill (Generic 2.2 with EAS)

Every skill that ends with a "0" is called a default skill, since every staffed agent in the skill group is logged into this skill by default. The default skill is the first skill for each skill group.

Delete

A CMS action that removes the entry on the window from the database.

Dequeued and Abandoned (DABN)

A trunk state in which the trunk quickly goes to idle after the caller abandons the call.

Designer Reports

Customized reports that can be created using Avaya Supervisor's Report Designer feature, and which are run from Avaya Supervisor.

Dialed Number Identification Service (DNIS)

A network capability that identifies, for each call, the number dialed or the area from which the call originated (for example, a specific 800 number set up for a promotion).

Dictionary

A CMS subsystem that can be used to assign names to various call center elements such as login IDs, splits/skills, trunk groups, VDNs and vectors. These names are displayed on reports, making them easier to interpret. Dictionary also allows customized calculations to be created for use in reports.

DID

See *Direct Inward Dialing*.

Dimmed Indicates unavailable or disabled. A dimmed menu item, button, or command is displayed in a lighter shade than the standard display and cannot be selected. See *Grayed Out*.

Direct Agent ACD (DACD) An agent state in which the agent is on a direct agent ACD call.

Direct Agent ACW (DACW) An agent state in which the agent is in the after call work (ACW) state for a direct agent ACD call.

Direct Agent Calling An EAS capability that makes it possible for a caller to reach the same agent every time and still include the call as an ACD call in the management tracking of the call center. This is ideal for claims processing where a client needs to speak with the agent handling the claim. This flexibility ensures a high level of customer service without reducing management control.

Direct Department Calling (DDC) A process of selecting an agent when more than one agent is available. With DDC, the call will go to the agent closest to the top of an ordered list. (This is a non-EAS option only.)

Direct Inward Dialing (DID) An incoming trunk used for dialing directly from the public network into a communications system without help from the attendant.

DNIS See *Dialed Number Identification Service*.

Driver A dynamic link library that processes ODBC function calls received from the driver manager, submits the resultant SQL requests to a specific data source, and returns the results to the requesting application. If necessary, a driver modifies an application's request so that the request conforms to the syntax supported by the associated DBMS. See *Structured Query Language, Database Management System*, and *Dynamic Link Library* for more information.

Driver Manager A dynamic link library that loads drivers on behalf of an application.

Dynamic Link Library A dynamic link library is another name for a driver or a driver manager. A dynamic link library is specific to the DBMS of the data being accessed. For example, an Informix specific dynamic link library will be used to access data in an Informix database, such as the CMS database.

EAD See *Expert Agent Distribution*.

EAS See *Expert Agent Selection*.

ECS See *Enterprise Communications Server*.

Enterprise Communications Server (ECS)
An Enterprise Communications Server is another name for a switch release, for example, the G3 ECS, or switch.

Entity
A generic term that refers to one of the following: Agent, Split/Skill, Trunk, Trunk Group, VDN, or Vector.

Error Message
A response from a program indicating that a problem has arisen or something unexpected has happened, requiring your attention.

EWT
See *Expected Wait Time*.

Exception
A type of activity in the ACD which falls outside the limits you have defined. An exceptional condition is defined in the CMS Exceptions subsystem, and usually indicates abnormal or unacceptable performance of the ACD (by agents, splits/skills, VDNs, vectors, trunks, or trunk groups).

Exception Reports
Display occurrences of conditions that fall outside the limits you defined.

Expected Delay
See *Expected Wait Time*.

Expected Wait Time (EWT)
An estimate of how long a caller will have to wait in queue to be served by a call center considering the current and past traffic, handling time, and staffing conditions. (Also referred to as *expected delay*.) The time spent in vector processing before being queued and the time spent ringing an agent with manual answering operation are not included in the EWT prediction. EWT is a switch-based calculation that can be used in vector processing decisions and can be viewed from CMS (release R3V4 and later). The EWT feature is available on the G3V4 switch, and the DEFINITY® Enterprise Communications Server (ECS) Release 5.

Expert Agent Distribution (EAD)
An EAS process that selects an agent when more than one agent is available. With EAS, a call will go to the most idle agent with the skill as primary (skill level one). If none are available, the call goes to an agent who is idle and has the skill as secondary. If none are available, the selection process continues with progressively lower skill levels until an idle agent is found.

Expert Agent Selection (EAS)

Expert Agent Selection (EAS) is an optional switch feature that builds on the power of the Call Vectoring and ACD features of the switch to match the skills required to handle a particular call to an agent who has at least one of the skills that a caller requires. Agents are assigned a single set of work mode buttons, rather than one set per skill. This simplifies the agent's interface to the work mode buttons. When the "MI" (Manual In) or "AI" (Auto In) button is lighted, the agent is available to take a call in any assigned skills or, in the case of Multiple Call Handling (MCH), in any MCH skills. The ACD queuing and the vector commands *Queue-to-Main* and *Check-Backup* are used to route a call to an agent with the appropriate skill to handle that call.

CMS collects data on skills in the same manner as it collects data on splits. Real-time Agent reports generally indicate the skill in which agents are currently working. Skill reports show the performance of the skill overall, displaying such items as the ASA, the number of calls, and the percentage of calls answered within the target service level for the skill.

CMS also reports VDN data by VDN skill preference, so that customers can assess the call center performance relative to calls requiring particular skills. CMS reports how many calls were handled, how long these calls waited for service, and the average talk time for calls queued to a particular skill preference in a particular VDN.

External Call

Calls made to off-switch destinations. This includes calls to other switches in a DCS network.

Extension Call

Extension calls are any calls originated by agents and non-ACD calls received by agents. For the Generic 2.2 and Generic 3 switches and the ECS, these include calls an agent makes to set up a conference or transfer.

FBUSY

See *Forced Busy*.

FDISC

See *Forced Disconnect*.

Find One

A CMS action that searches the database for entries that match the input value and displays the matching entries one at a time.

First Threshold

A graph term for the lower limit of a particular condition in a graph report. The bar(s) change color/intensity when the defined limit is met, notifying you that a Caution condition could exist.

Flex Agents

Agents who have a role of Roving, Backup, or Allocated. Top and Reserve agents are not Flex agents. See the *Avaya Business Advocate User Guide*, 585-210-711, for more information.

Forced Busy (FBUSY)

A trunk state in which the switch sends a busy signal to a caller when the call center is too busy to handle the incoming call.

Forced Disconnect (FDISC)

A trunk state in which the switch disconnects the caller.

Forced Multiple Call Handling (FMCH)
A feature available with the G3V4 switch, and the ECS, that, when activated for a split/skill, allows calls to be automatically delivered to an idle line appearance if the agent is in the Auto-In/Manual-In work mode and an unrestricted line appearance is available on the voice terminal, even if the agent is talking on an ACD call.

Function Call
Another name for a request for data generated within an ODBC-enabled application. The embedded SQL function in the application submits the request to the ODBC driver manager as a function call, where it is translated and a DBMS-specific driver manager (dynamic link library) is selected based on the type of data requested.

Get Contents
A CMS action that provides access to the members of an agent group, or to tasks on a shortcut or timetable.

Graphics
A CMS reporting option that allows you to view some reports in bar graph format. Avaya Supervisor supports other graphic formats.

Grayed Out
When you do not have access to a menu or action list item, it will be grayed out (that is, dimmed or displayed in a different color from the rest of the menu or action list).

See *Dimmed*.

Historical Database
A database that contains intrahour records for up to 62 days in the past, daily records for up to 5 years in the past, and weekly/monthly records for up to 10 years in the past for each CMS-measured agent, split/skill, trunk, trunk group, vector, and VDN.

Historical Reports
Reports that display past ACD data for various agent, split/skill, trunk, trunk group, vector, or VDN activities.

HOLD
A trunk state in which the agent has put the call on hold.

ICM
See *Inbound Call Management*.

IDLE
A trunk state indicating that the trunk is not in use.

II
See *Information Indicator*.

Inbound Call Management (ICM)
A set of switch and adjunct features using ASAI to enable the adjunct to provide automatic screen delivery and call routing.

Information Indicator (II)
A 2-digit code that identifies the type of originating line (for example: hotel or pay phone) for incoming ISDN primary rate interface (PRI) calls.

Informix
A relational database management system used to organize CMS historical data.

Informix SQL
A query language tool that is used to extract data from an Informix database. For the CMS historical database, CMSQL is used in place of Informix SQL.

Input Field

An input field is an area on a user window into which a CMS user enters one or more valid field values. For example, the valid values for the input field `Split` are integers from 1 through 255 (Generic 3r Version 2 to Version 4) and split names that have been assigned in the Dictionary subsystem.

Integrated Services Digital Network (ISDN)

A digital standard for telephony that enables, among other things, telephone, television, and computer signals on the same lines.

Interval-Based Items

A category of database items. These items generally represent the amount of time during a collection interval spent doing a particular activity. Interval-based items are updated throughout the collection interval and timing is restarted at the end of the interval. Interval-based items should only be used to show the amount of time in an interval for an activity or to calculate percentages of time spent in an interval. Interval-based items should not be used to calculate averages (such as average hold time).

Intrahour Interval

A 15-, 30-, or 60-minute segment of time starting on the hour. An intrahour interval is the basic unit of CMS report time.

ISDN

See *Integrated Services Digital Network*.

LAI

See *Look Ahead Interflow*.

LAN

See *Local Area Network*.

List All

A CMS action that lists all the entries that matched the current field values.

List Devices

A CMS action that lists all the devices which have been specified in the Backup/Restore Devices window.

Local Area Network (LAN)

A private interactive communication network that allows computers to communicate over short distances, usually less than one mile, at high data transfer rates from 1 Mbps to as high as 100 Mbps.

Location ID for Agents

The switch port network location ID (01-44) that is associated with an agent upon login to the ACD. A location ID is not directly assigned to an agent; instead, it is associated with the equipment location of the voice terminal that the agent uses to log into the ACD. Therefore, only when an agent logs into the ACD can the agent become associated with a location ID.

Location ID for Trunks

The switch location ID (01-44) associated with a trunk. A location ID is not directly assigned to a trunk; instead, it is assigned to a port network (on the ‘ch cabinet x’ form), and, therefore each trunk whose equipment location belongs to that port network will be associated with that port network’s location ID.

Location Number

A number, between 1 and 44, that is associated with one or more port network locations on switch. All hardware residing on a port network will be associated with that port network's location number (on the 'ch cabinet x' form). Location number was introduced in support of ATM WAN PNC, however, use of a location number does not require ATM WAN PNC. Location number is supported by the optional Multiple Locations feature and is only available on the large DEFINITY platform (the *r* platform). For the DEFINITY ECS R7.1r, the location numbers used with the Multiple Locations feature can range from 10-044. If the Multiple Locations feature is not used on the R7.1r platform, or if the platform is smaller than *r*, all port network locations will be assigned "1" as a location number. If R8.1r is used in this case, a location number of "0" is used instead. Each location number also has an associated time zone offset, daylight savings plan number, and if the North American Numbering Plan is supported, an area code. Both logged-in agents and individual trunks will be associated with a location number by their location IDs; the same location number can apply to both a logged-in agent and a trunk. VDNs, hunt groups, and vectors will not have a location number associated with them; these resources are considered location-dependent.

Logical Agent

An EAS feature that associates the agent's login ID with the physical extension when the agent logs in. Properties such as the assigned skills, class of restriction, and coverage path are associated with the login ID rather than the physical extension. This allows agents to log in at any available set.
The Logical Agent capability allows agents to be called by dialing their login IDs. Calls to login IDs may be treated as direct agent ACD calls, given the proper class of restriction, or may be treated as extension (personal) calls. Treating the calls as direct agent calls can be used to help distinguish business-related from personal calls.

LOGOFF

An agent work mode indicating that an agent has logged out and is not available to take ACD calls.

LOGON

An agent work mode indicating that an agent logged in or is staffed.

Logout Reason Codes

Logout reason codes enable an agent to specify the reason for logging out, such as the end of a shift or training.

Look Ahead Interflow (LAI)

A switch feature that can be used to balance the call load among multiple call centers. The LAI feature works with Call Vectoring and ISDN PRI trunks to intelligently route calls between call centers. This allows multiple call centers to share work loads, expands hours of coverage, and allows calls to be transparently handled by call centers in different time zones.

Maintenance

A CMS subsystem that is used for routine maintenance of the CMS, such as backing up data, checking on the status of the connection to the switch, and scanning the error log.

Maintenance Busy (MBUSY)

A trunk state in which the trunk is out of service for maintenance purposes.

Manual In (MI) An ACD work mode. The Manual In (MI) mode makes the agent available to receive an ACD call and automatically places the agent into the ACW state upon release from the call.

MBUSY See *Maintenance Busy*.

MCH See *Multiple Call Handling*.

Measured A term that means an ACD element (agent, split/skill, trunk, trunk group, vector, VDN) has been identified to the switch as of interest to the CMS. The switch sends messages to CMS only for ACD elements that are measured. If the ACD element is not measured on the switch, no data is collected.

Messages Temporary windows used only for displaying information like “field help” and syntactical field errors. Message windows cannot be moved, sized, or scrolled and do not count in the user window count. Message windows are automatically removed when you correct the error or move to the next field.

MI See *Manual In*.

MIA See *Most Idle Agent*.

Migration The process of migrating CMS data when upgrading your CMS. (Execute the “R3 Migrate Data” or “R2 Migrate Data” menu item after selecting System Setup from the main menu.) Migration is sometimes necessary in order to move the customer’s data to a new release. Migration is usually, but not always, driven by a hardware platform change.

Modify A CMS action that changes the database entry to reflect the new values entered in the given window.

Monthly Data Daily data that has been converted to a monthly summary.

More Help More help displays a user window containing expanded information about the current primary user window or menu selection in CMS. More help is for primary windows only, not output windows (like list all windows).

Most Idle Agent (MIA) This is an ACD distribution method that maintains a queue of idle agents and distributes a call to the split/skill to the agent closest to the head of the queue who is not marked as “busy.” (“Busy” is defined as being in ACW, on an AUXIN/OUT call, or on an ACD call for another split/skill.) Agents in multiple splits/skills are in multiple “eligible agent” lists. There is one list for each skill. If MIA across splits/skills is enabled, agents are put at the bottom of all agent lists after completing an ACD call for any split/skill. If MIA across skills is *not* enabled, an agent who is on an AUXIN/OUT extension call from the AVAIL wait or on an ACD call for another split/skill continues to move up the list of eligible agents for other splits/skills. Agents in ACW may or may not be in the eligible agent lists, depending upon the setting of an option (DEFINITY ECS R5 and later).

Multi-Agent Skill Change Window — This window allows the user to perform the following administration actions: add up to 32 agents to a skill; move up to 32 agents from one skill to another; and remove up to 32 agents from a skill.

Multiple Call Handling (MCH) — Allows an agent to receive an ACD call while other calls are active on the agent's station. Unless forced MCH is in operation, the agent must put the current call on hold and press AI/MI in order to receive another ACD call.

Multiple Split/Skill Queuing — With Call Vectoring, a call can be queued to up to three splits/skills at the same time. The first agent who becomes free in any of the splits/skills gets the call.

Multiuser Mode — Any administered CMS user can log into CMS. Data continues to be collected if data collection is "on." This is the normal CMS operating mode.

Name (Synonym) Fields — Fields in which you may enter a name (synonym) that has been entered in the Dictionary subsystem (for example, names of agents, splits/skills, agent groups, trunk groups, vectors, VDNs).

Next — A CMS action that displays the next match found following a Find One, Next, or Previous action.

Night Service — A switch capability that assists calls that come in after business hours or on weekends to be automatically rerouted to a split, an announcement, or an alternate destination specifically set up for after-hours coverage.

Nonprimary Split/Skill — When a call is queued to multiple splits/skills, the second and third splits/skills to which the call queues in a VDN are called nonprimary splits/skills. They are also referred to as secondary and tertiary splits/skills, respectively.

Nonzero (0) Skill (Generic 2.2 with EAS) — Any skill that does not end in "0" is called a nonzero skill.

Normal Condition — A graph term in which the bars in the graph report are within your defined acceptable limits.

OCM — See *Outbound Call Management*.

ODBC — See *Open Database Connectivity.*

Open Application Interface — An open applications interface that allows the same application to be supported by a wide range of products and avoids the need to develop specialized interfaces for new applications.

Open Database Connectivity (ODBC) — Open Database Connectivity is a standard application programming interface (API) for accessing data in both relational and non-relational databases.

Open Window — A user window that remains open because you have not yet closed it with the **Exit** SLK. An open window becomes the current window when it initially is displayed on the screen or when you make it the current window using the **Current** SLK.

OTHER — An agent state in which the agent is working on a call for another split or skill, or has put a call on hold and has not chosen another work mode. When the link to the switch first comes up or when the agent has just logged in, the agent state is OTHER until the switch notifies CMS of the agent's state.

Outbound Call Management (OCM) — A set of switch and adjunct features using ASAI, that distributes outbound calls initiated by an adjunct to internal extensions (usually ACD agents).

PEC — See *Price Element Code*.

Percent (%) Within Service Level — The percentage of calls that were answered by an agent within a specific number of seconds. This observed percentage is used in reports as a statistical value.
See *Acceptable Service Level*.

Previous — A CMS action that displays the previous match found.

Previous Interval — Represents one intrahour interval and is part of the real-time database. At the end of each intrahour interval, the contents of the current intrahour interval are copied to the previous intrahour interval portion of the real-time database.

Price Element Code (PEC) — The set of numbers that Avaya has assigned to each part that may be ordered.

Primary Skill — The skills assigned to an agent. Primary skills are the areas in which the agent has the most expertise. (This is used in G3V2 through G3V4 with EAS.)
See *Agent Skill*, *Skill Level*.

Primary Window — The first window opened in response to a menu selection. A primary window may also generate another user window (secondary window). A primary window can be moved, sized, or scrolled, and counts in the window count.

Private Report — A custom or designer report that only the creator can access.

Pseudo-ACD — An area you create on your CMS to place previously backed-up ACD data. A pseudo-ACD is not a *live* (real) ACD and does not communicate with any switch.

Queue — A holding area for calls waiting to be answered in the order in which they were received. Calls in a queue may have different priority levels, in which case, calls with a higher priority are answered first.

QUEUED — A trunk state in which an ACD call has seized the trunk and is queued to a split/skill waiting for an agent to answer.

R3V5 See *Release 3 Version 5*.

R3V6 See *Release 3 Version 6*.

R3V8 See *Release 3 Version 8*.

R3V9 See *Release 3 Version 9*.

R3V11 See *Release 3 Version 11*.

Read Permission The CMS user with read permission can access and view data (for example, run reports or view the Dictionary subsystem). Read permission is granted from the User Permissions subsystem.

Real-Time Database Consists of the current and previous intrahour data on each CMS-measured agent, split/skill, trunk, trunk group, vector, and VDN.

Real-Time Reports Reports that display current ACD call activity on agents, splits/skills, trunks, trunk groups, vectors, and VDNs for the current or previous intrahour interval. Current intrahour interval real-time reports are periodically updated as data changes during the interval. Previous intrahour interval real-time reports show data totals for activity that occurred in the previous intrahour interval.

Recorded Announcements Prerecorded greetings and information played to callers as they wait for service.

Redirect On No Answer (RONA) An ACD capability that removes an unanswered call from the voice terminal at which it is ringing, busies out the port or makes the agent unavailable, and requeues the call at top priority or sends it to a VDN.

Refresh Rate The number of seconds CMS should wait for each update of the real-time report data. A user's fastest allowable refresh rate is defined in the User Permissions—User Data window as a minimum refresh rate. The default refresh rate when a user brings up the report input window is the administered minimum refresh rate plus 15 seconds.

Release 3 Version 5 (R3V5) R3V5 can refer to a software version of CMS.

Release 3 Version 6 (R3V6) R3V6 can refer to a software version of CMS.

Release 3 Version 8 (R3V8) R3V8 can refer to a software version of CMS.

Release 3 Version 9 (R3V9) R3V9 can refer to a software version of CMS.

Release 3 Version 11 (R3V11)

R3V11 can refer to a software version of CMS.

Request Agent

The OpenLink Request Agent resides on the client and interfaces with the ODBC driver via proprietary protocol. The Request Agent links the client applications and the OpenLink Request Broker on the server to facilitate transparent database access.

Request Broker

The OpenLink Request Broker is an ODBC driver manager that resides on the server. ODBC uses data source names as the link between the ODBC Request Broker and the relevant ODBC driver for a particular database. The OpenLink Request Broker provides the link between the user's applications and the DBMS itself by selecting a dynamic link library specific to the database being queried.

RING

An agent state in which a call rings at an agent's voice terminal after leaving the queue and before the agent answers the call. (*This agent state is available only with Generic 2.2, and with Generic 3 and ECS*.)

A trunk state in which a call is ringing at the agent's voice terminal.

Rolling ASA

Rolling ASA is a running weighted average calculation without regard to any interval boundaries. A rolling ASA calculated by the switch or ECS can be used, beginning with R3V4 CMS, for vector routing. Rolling ASA is calculated on the G3V4 switch, and the ECS, and sent to R3V4 and later CMS releases.

An additional Rolling ASA calculated by the switch is also available as a real time database item for G3V4 and later G3 switches.

RONA

See *Redirect On No Answer*.

Run

A CMS action that starts the process for the given window.

Screen-Labeled Key (SLK)

The first eight function keys at the top of your keyboard that correspond to the screen labels at the bottom of your terminal screen in CMS. The screen labels indicate the function each key performs.

Scrolling

Moving backward and forward within a window in CMS.

Second Threshold

A graph term for the upper limit you enter for a particular condition in a graph report. When this limit is met, the bar(s) change color/intensity, indicating that a possible Warning condition exists.

Secondary Split/Skill (G3 Vectoring, G2.2 EAS)

The second split/skill the call queues to in a VDN is called the secondary split/skill.

Skill Group (Generic 2.2 EAS)

A group of ten skills. Each consecutive ten skills ending with digits 0 through 9 constitute a skill tens group. For example, skills 10-19 form a skill tens group, as do skills 340-349.

Secondary Skill

Skills assigned to an agent. Secondary skills are the areas in which the agent does not have extensive expertise, or is not the agent's preference. (Used in G3V2 through G3V4 with EAS.)

See *Agent Skill, Skill Level.*

Secondary Window

A user window that is generated from a primary window. Secondary windows can be moved, sized, or scrolled and do not count in the user window count.

SEIZED

A trunk state in which the trunk is being used for either an incoming or an outgoing call.

Select Tables

A CMS action that allows you to select specific tables to Back Up or Restore.

Service Observing—Remote

A feature that allows a user to dial into the switch and monitor a call.

Service Observing—VDNs

A feature available with the G3V4 switch and the ECS that gives a voice terminal user the ability to monitor the treatment a call receives as it is processed by a VDN, routes to another VDN or agent, or transfers to another VDN or agent.

Shortcut

A CMS capability that enables a series of tasks to run immediately on your screen. Shortcut is a fast, easy way to select windows that you might look at every day.

Single-User Mode

Only one person can log into CMS. Data continues to be collected if data collection is "on." This mode is required to change some CMS administration.

Site

A site refers to a physical location. This can be a building, a section of a building, or it can be what was once a separate ACD before the ACD WAN PNC capability was used to merge this separate ACD together with other ACDs into one large call center. A site will typically be assigned one or more location IDs. A site, despite being part of a larger call center, may continue to have sole responsibility for handling certain 800 numbers. A site may also share responsibility for handling an 800 number by having some of its agents be part of a larger split/skill that includes agents from other sites.

Skill

An attribute that is assigned to an ACD Agent when EAS is enabled. An agent skill is a particular expertise or speciality enabling an agent to handle a call which requires someone with that particular area of expertise. You define skills based on specific customer needs and call center requirements.

Skill Hunt Group

When EAS is enabled, calls route to specific skill hunt groups. These skill hunt groups are usually based on the needs of your customers. Agents are not assigned to a skill group (like split hunt groups), but agents are assigned specific skills that become active when they log in.

Skill Level

A priority level from 1 (highest) to 16 (lowest) indicating an agent's level of expertise or ability to handle calls to the given skill. (ECS Version 5 and later.)

Skill, Primary	See *Primary Skill*.
Skill, Secondary	See *Secondary Skill*.
SLK	See *Screen-Labeled Key*.
Solaris System	See *UNIX System/Solaris System*.
SPARCserver 5	See *Sun Computers*.
SPARCserver 10	See *Sun Computers*.
SPARCserver 20	See *Sun Computers*.
SPARCstation	A workstation client for the SPARCserver.
Split	A group of extensions (referred to as agents) that receives special-purpose calls in an efficient, cost-effective manner. Calls automatically go to a split and can queue if no agents are available.
Split/Skill ACD Call	A split/skill ACD call is a call that routed to a split/skill and was answered by an agent in that split/skill.
SQL	See *Informix SQL*.
Staffed Agent	An agent who is currently logged in to the switch.
Standard Reports	The set of reports that are delivered with the CMS or Avaya Supervisor software.
Station	An unmeasured voice terminal extension. An extension that is not currently staffed by an agent or that is a member of an unmeasured split/skill or hunt group.
Stop	A CMS action that stops the Restore or Migration that is in progress. Stop is also a vector command.
String Values	The descriptive words that are displayed on reports dealing with agents, splits/skills, and trunks. A *word* is used to describe the value of the data (for example, HOLD, AVAIL, YES).
Structured Query Language (SQL)	A language used to interrogate and process data in a relational database (such as *Informix*). See *Informix SQL*.
Submenu	A menu that is displayed as a result of a menu selection. All menu selections followed by a “>” have submenus.

Subsystem — Each CMS main menu selection (for example, Reports, Dictionary, System Setup, Exceptions, and so on) along with Timetable and Shortcut is referred to as a subsystem of the Call Management System.

***Sun* Computers** — A host computer that is attached to a network and provides services other than simply acting as a store-and-forward processor or communication switch.

Switch — A private switching system providing voice-only or voice and data communications services (including access to public and private networks) for a group of terminals within a customer's premises.

System — A general term for a computer and its software and data.

System Setup — A CMS subsystem that allows users to initialize their CMS (for example, selecting the size of the intrahour interval based on how much data storage will be used for CMS data).

Technical Service Center (TSC) — Provisioning, maintenance, and helpline support for Avaya call center customers.

Terminal — A combination of monitor (video display) and keyboard used to communicate with a computer to enter and display information.

See *Agent Terminal*.

Tertiary Split/Skill — Generic 3 and ECS with vectoring, Generic 2.2 with EAS only. When a call is queued to multiple splits/skills, the third split/skill the call queued to in a VDN is called the tertiary split/skill.

Time Format — The standard format for entering times on CMS reports. Acceptable formats are:

- A 12-hour time format with AM/PM (for example, 7:30AM, 5:00PM).
- A 24-hour time format (for example, 7:30, 17:00).
- A "-" offset based on the current interval date (for example, -1 for the previous hour or -0:30 for the previous half hour interval).

Timetable — A task or group of tasks (such as reports) scheduled for completion at a time that is convenient and nondisruptive for your call center's operation.

Top Skill

The agent's top skill is the agent's first-administered, highest-level skill. This concept is the most useful when you have a Generic 3 switch (with EAS) and with agents who are using skill level call handling preference. In this case, the agent's top skill represents the skill for which the agent is most likely to receive a call. Agents for whom a given skill is the top skill are the agents that a skill supervisor can count on to handle calls for the skill.

NOTE: This concept is not useful for agents using the greatest need call handling preference or for agents who are not Generic 3 (with EAS) agents. In these cases, the top skill data is still populated. When using the Generic 2.2 (with EAS) switch, the agent's top skill is always the "zero skill." For non-EAS agents, the top "skill" is the split the agent has been logged into the longest.

Translations

A CMS action that requests a full set of information about measured entities from the switch.

Trunk

A telephone line that carries calls between two switches, between a Central Office (CO) and a switch, or between a CO and a phone.

Trunk Group

A group of trunks that are assigned the same dialing digits—either a phone number or a Direct Inward Dialed (DID) prefix.

TSC

See *Technical Service Center*.

UCD

See *Uniform Call Distribution*.

Ultra Enterprise 3000

A Sun Microsystems computer capable of hosting CMS.

See *Sun Computers*.

Uniform Call Distribution (UCD)

A process that selects an agent when more than one agent is available. With UCD, the most idle agent for the skill/split receives the call.

See *Direct Department Calling and Expert Agent Distribution*.

Universal Call Identifier (UCID)

A number that uniquely identifies a call in a network of nodes that support UCID. This number will be a part of the records in the Call History feature of CMS.

UNIX System/ Solaris System

A multi-user computer operating system that supports CMS. A user can access the UNIX system from the **Commands** SLK.

UNKNOWN

An agent state in which CMS does not recognize the current state.
A trunk state in which CMS does not recognize the state of the trunk.

UNSTAF

See *Unstaffed*.

Unstaffed (UNSTAF)

An agent state in which the agent is not logged in and, therefore, is not being tracked by CMS.

Update A process used to modify a customer's existing software release in order to give the customer additional functionality or to fix a problem. The update process involves downloading CMS update files from a cartridge tape to hard disk, and installing the new files.

Upgrade A process used to move an existing CMS customer from one release/load to another, giving the customer the additional functionality provided in the new release. Depending on which release the customer is upgrading from, and to, the upgrade may involve upgrading the software only, or may involve upgrading software and hardware. When an upgrade involves a major release, then data migration may be a necessary part of the upgrade procedure.

User Application A user application is the software on the user's PC that the data is being accessed for. For example, a user may want to access data in the CMS database for use in Microsoft Excel. The user generates a query from the embedded data querying function in the application. The data returned from the CMS database can then be used to generate a spreadsheet.

User ID The login ID for a CMS user.

User Permissions A CMS subsystem that allows the CMS administrator to define user access permissions.

User Window A window you can move, size, or scroll. It may contain input fields, reports, or help information.

VDN See *Vector Directory Number*.

VDN Counted-Calls Also known as counted-calls to VDN and active VDN calls. A Call Vectoring capability available with the G3V4 switch and the ECS. Counted-calls to VDN is a parameter of the "go to step" and "go to vector" commands that provides conditional branching (to a different step in the same vector or to a different vector) based on the number of incoming trunk calls currently in a VDN (in vector processing or at an agent).

VDN of Origin Announcement (VOA) A short announcement that is assigned to a VDN through switch administration. The VOA identifies the origin or purpose of a call for the call center agent who answers the call.

VDN Skill Preference A prioritized list of agent skills assigned to a VDN. Up to three skills can be assigned. VDN skill preferences are referred to in the vector as "1st," "2nd," and "3rd." Vectors use VDN skills to queue calls based on your preference. CMS tracks calls by VDN skill preference.

Vector	A list of steps that process calls in a user-defined manner. The steps in a vector can send calls to splits/skills, play announcements and music, disconnect calls, give calls a busy signal, or route calls to other destinations. Calls enter vector processing via VDNs, which may have received calls from assigned trunk groups, from other vectors, or from extensions connected to the switch.
Vector Command	A vector step that describes the action to be executed for a call (for example, “Queue to main”, “check backup”, “disconnect”).
Vector Directory Number (VDN)	An extension number that enables calls to connect to a vector for processing. A VDN is not assigned an equipment location. It is assigned to a vector. A VDN can connect calls to a vector when the calls arrive over an assigned automatic-in trunk group, dial-repeating (DID) trunk group, or ISDN trunk group. The VDN by itself may be dialed to access the vector from any extension connected to the switch.
Vector Step	One processing step in a vector. A vector step consists of a command and one or more conditions or parameters.
Vector Step Condition	A condition accompanying a vector command that defines the circumstances in which the command will be applied to a call.
VOA	See *VDN of Origin Announcement*.
Voice Response Unit (VRU)	A switch that routes calls to a VRU adjunct computer that provides interactive voice related services to inbound callers.
Voice Terminal	A telephone set, usually with buttons, that gives an agent some control over the way calls are handled.
VRU	See *Voice Response Unit*.
Warning Condition	A graph term. The bars in the graph report change color or intensity indicating that the data being displayed met your defined second threshold limit.
Weekly Data	Daily data that has been converted to a weekly summary.
Window	Any rectangle on your CMS screen that encloses a menu, data entry fields, reports, or messages.
Window Count	The number of primary windows that can be open at any one time.
Write Permission	The CMS user can add, modify, or delete data and execute processes. Write permission is granted from the User Permissions subsystem.
Zero (0) Skill (Generic 2.2 with EAS)	Every skill that ends with a “0” is called a zero skill. The zero skill is the first skill for each skill group. This is the same as the default skill.

Index

Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W