AVAYA

Avaya Call Management System
Release 14
External Call History Interface

Contents

Contents

Contents

Preface

Avaya Call Management System (CMS) is an application for businesses and organizations that use Avaya communication servers to process large volumes of telephone calls using the Automatic Call Distribution (ACD) feature. Avaya CMS supports solutions for routing and agent selection, multi-site contact centers, remote agents, reporting, interfaces to other systems, workforce management, desktop applications, system recovery, and quality monitoring.

Avaya CMS is part of the Operational Effectiveness solution of the Avaya Customer Interaction Suite.

This section includes the following topics:

Purpose

The purpose of this document is to describe the format of the External Call History data files, and how to transfer these files to another computer. External Call History Interface (ECHI) customers are responsible for the storage, formatting, printing, and any additional processing of the data transferred to the computer.

Intended users

This document is written for:

- Avaya support personnel
- Contact center administrators

Users of this document must be familiar with Avaya CMS.

Overview

This document includes the following topics:

- [Overview of the External Call History Interface](#) on page 15

  Provides an overview of the ECHI feature.

- [Setting up the ECHI package](#) on page 55

  Outlines the procedures for installing the External Call History Interface software and connecting the CMS computer to the receiving computer.

- [Connecting the CMS to the receiving computer](#) on page 69

  Outlines the procedures for configuring the receiving computer.

Conventions and terminology

If you see any of the following safety labels in this document, take careful note of the information presented.

⚠ **CAUTION:**
Caution statements call attention to situations that can result in harm to software, loss of data, or an interruption in service.

⚠ **WARNING:**
Warning statements call attention to situations that can result in harm to hardware or equipment.

⚠ **DANGER:**
Danger statements call attention to situations that can result in harm to personnel.

⚠ **SECURITY ALERT:**
Security alert statements call attention to situations that can increase the potential for unauthorized use of a telecommunications system.

Reasons for reissue

This is the first issue of this document.

Availability

Copies of this document *are* available from one or both of the following sources:

Note:
Although there is no charge to download documents through the Avaya Web site, documents ordered from the Avaya Publications Center must be purchased.

● The Avaya online support Web site, http://www.avayadocs.com

- The Avaya Publications Center, which you can contact by:

  Voice:

  +1-207-866-6701
  +1-800-457-1764 (Toll-free, U.S. and Canada only)

  Fax:

  +1-207-626-7269
  +1-800-457-1764 (Toll-free, U.S. and Canada only)

  Mail:

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

  E-mail:

  totalware@gwsmail.com

Related documentation

You might find the following Avaya CMS documentation useful. This section includes the following topics:

-
-
-
-
-
-
-

Change description

For information about recent changes made in Avaya CMS, see:

- *Avaya Call Management System Release 14 Change Description*, 07-601579

Administration documents

For more information about Avaya CMS administration, see:

- *Avaya Call Management System Release 14 Administration,* 07-601585
- *Avaya Call Management System (CMS) Release 14 Database Items and Calculations,* 07-601591
- *Avaya Call Management System Supervisor Release 14 Reports*, 07-601590
- *Avaya Call Management System (CMS) Supervisor Release 14 Installation and Getting Started*, 07-601587
- *Avaya Call Management System High Availability User Guide*, 07-300066
- *Avaya Call Management System High Availability Connectivity, Upgrade and Administration*, 07-600957

Software documents

For more information about Avaya CMS software, see:

- *Avaya Call Management System Release 14 Software Installation, Maintenance, and Troubleshooting Guide,* 07-601578
- *Avaya CMS Open Database Connectivity Version 5.2,* 07-601580
- *Avaya Call Management System Release 14 LAN Backup User Guide,* 07-601589
- *Avaya Call Management System Release 14 External Call History Interface,* 07-601586
- *Avaya CMS Custom Reports,* 585-215-822
- *Avaya CMS Forecast User Guide,* 585-215-825
- *Avaya Call Management System (CMS) Supervisor Release 14 Report Designer*, 07-601588
- *Avaya Business Advocate Reports*, 07-601618

Hardware documents

For more information about Avaya CMS hardware, see:

- *Avaya Call Management System Sun Netra 210 Computer Hardware Installation, Maintenance, and Troubleshooting,* 07-600963
- *Avaya Call Management System Sun Fire V880/V890 Computer Hardware Installation, Maintenance, and Troubleshooting,* 07-600965
- *Avaya Call Management System Sun Blade 100/150 Workstation Hardware Installation, Maintenance, and Troubleshooting,* 07-600964
- *Avaya Call Management System Terminals, Printers, and Modems,* 585-215-874

Call Center documents

For more information about Avaya Call Center documents, see:

- *Avaya Call Management System Switch Connections, Administration*, and *Troubleshooting*, 07-601582

Avaya CMS upgrade documents

There are several upgrade paths supported with Avaya CMS. There is a document designed to support each upgrade.

This section includes the following topics:

- Base load upgrades on page 13
- Platform upgrades and data migration on page 13
- Avaya Call Management System Upgrade Express (CUE) on page 13

Base load upgrades

Use a base load upgrade when upgrading CMS to the latest load of the same version (for example, r14ak.g to r14al.k). A specific set of instructions is included with the upgrade. The *Avaya Call Management System Release 14 Base Load Upgrade* document is shipped to the customer site with the CMS software CD-ROM as part of a Product Correction Notice (PCN).

Platform upgrades and data migration

Use a platform upgrade when upgrading to a new hardware platform (for example, upgrading from a SPARCserver 5 to a Sun Netra 210). 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, R3V11 to R14).

For more information about platform upgrades and data migration, see:

- *Avaya Call Management System Release 14 Platform Upgrade and Data Migration*, 07-601581

Avaya Call Management System Upgrade Express (CUE)

Use CUE when CMS is being upgraded from an earlier version (for example, R3V11) to the latest version (for example, R14).

A specific set of upgrade instructions is included with the upgrade. The *Avaya Call Management System Release 14 CMS Upgrade Express (CUE) for Sun Computers* document is included on the CUE software CD-ROM that is shipped to the customer site with the CUE kit.

For information about customer requirements for CUE upgrades, see:

- *Avaya Call Management System Release 14 CMS Upgrade Express (CUE) Customer Requirements, 700419930*

Documentation Web sites

For Avaya product documentation, go to http://www.avayadocs.com. 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:

- Information about Avaya products and service

  http://www.avaya.com

- Sun hardware documentation

  http://docs.sun.com

Support

Contacting Avaya technical support

Avaya provides support telephone numbers for you to report problems or ask questions about your product.

For United States support:

1- 800- 242-2121

For international support:

See the 1-800 Support Directory listings on the Avaya Web site.

Escalating a technical support issue

Avaya Global Services Escalation Management provides the means to escalate urgent service issues. For more information, see the Escalation Management listings on the Avaya Web site.

Overview of the External Call History Interface

External Call History Interface (ECHI) is an optional Avaya Call Management System (CMS) feature that allows you to transfer the CMS call record data to another computer for processing. This feature is especially useful for call centers with a high volume of calls. A CMS system with ECHI can store a greater volume of call record data than a CMS system alone.

This section includes the following topics:

- Space allocation on page 15
- Call record processing on page 15
- CMS ECHI call record format options on page 16
- About the call record file format on page 16
- Call record field descriptions on page 18
- Example call scenario and call records on page 33

Space allocation

You will need to allocate space on CMS for call records to be buffered (in System Setup: Data Storage Allocation, up to 99,999 segments in a buffer over all ACD's).

Call record processing

As calls are processed, call records will be recorded in a file (one record per call segment) until either the file size reaches approximately 1 MB or the end of an interval is reached. Then the file of call record segments will be transferred to another computer via uucp or other designated software. If the computers are in close proximity, the data is transferred at speeds of up to 38,400 bps. A new file and its first record will be created as the first segment of the next call is processed.

CMS ECHI call record format options

The External Call History Interface (ECHI) software allows you to choose either the standard call record format or the extended record format. You can choose to use the standard call record format if you have a reporting application that supports only the R3V11 call record format and still upgrade your CMS system to R14. If your reporting application supports the R12 record format you can use the extended call record enhancements. For more information, see Call record field descriptions on page 18.

About the call record file format

This section describes the format of the CMS call record files that are transferred by External Call History Interface (ECHI).

This section includes the following information:

- Data characteristics on page 16
- File name format on page 17
- File header on page 17

Data characteristics

Integers are stored with the least significant byte first and the most significant byte last. All integers are unsigned.

Data items of type char are stored with the most significant byte first and the least significant byte last. Type char data items are ASCII character strings of the length specified in the Length column of the Standard call record field descriptions on page 18.

Note:
The following sections explain how CMS sends integers to the receiving computer. The receiving computer may have different descriptions for integers. Once your software has parsed and stored the data on the receiving computer, it may be transformed to a different representation.

File name format

The call history files are named *chrxxxx* (*chr* followed by a 4-digit number) on the CMS computer. The transfer process adds an extension of up to three digits so that the file name on the receiving computer is *chrxxxx.x, chrxxxx.xx or chrxxxx.xxx*.

File header

Each call record file header includes a **VERSION** field and a **SEQUENCE _NUMBER** field.

When using the NICE Analyzer with CMS, you need to inform the NICE Analyzer of new call record formats through the version field in the External Call History file header and the External Call History Call Record header.

This section includes the following topics:

About the VERSION field

The **VERSION** field is a 4-byte Integer.

If you use the standard call record format, the **VERSION** field will contain an 11 for every file header.

If you use the extended call record format, the **VERSION** field will contain a 12 for every file header.

About the SEQUENCE _NUMBER

The **SEQUENCE _NUMBER** field is a 4-byte Integer. The **SEQUENCE _NUMBER** field identifies a particular call record file so that duplicates can be recognized when retransmission has occurred. These files are sequential. The **SEQUENCE_NUMBER** restarts at zero when its 4-byte integer reaches its limit (when all bits are high [ones] in its binary equivalent).

Example call record file header

The following table is an example of the information contained in a call record header. The least significant bytes are stored first and most significant bytes are stored last.

	VERSION	**SEQUENCE_NUMBER**
Decimal	11	1
Binary	00000101 00000000 00000000 00000000	00000001 00000000 00000000 00000000

Call record field descriptions

The file header is followed by a file containing a variable number of fixed-length records. ECHI does not use CMS database items. The standard call record data items are represented by their CMS database equivalents. If a field is not measured, the call record field will be populated with a "null", "0", or "-1". A standard format record contains 65 data fields, and an extended format record contains 77 data fields.

This section includes the following topics:

Standard call record field descriptions

ACD

Type: integer

Length: 1 byte

Description: The ACD number for which data was collected.

ACWTIME

Type: integer

Length: 4 bytes

Description: The time spent, in seconds, in After Call Work (ACW) associated with this call by the answering agent in this segment.

AGT_RELEASED

Type: bit

Length: 1 bit

Description: The agent 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). This value is padded with seven 0s to maintain byte alignment.

ANS_LOCID

Type: integer

Length: 2 bytes

Description: The location ID of the answering agent. This ID number is not assigned to an agent, but rather to the agent terminal and is associated with the communication server port network ID. An agent may be associated with a location ID only upon logging into the ACD. Valid values are 0 through 250.

> ⚠ **Important:**
> If the location IDs (LOC_ID) defined on the Communication Manager server are greater than the valid CMS values, then a default location ID of 0 will be assigned.

ANSHOLDTIME

Type: integer

Length: 4 bytes

Description: 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).

Hold time is accrued for any type of call.

ANSLOGIN

Type: char

Length: 10 bytes

Description: 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.

ANSREASON

Type: integer

Length: 1 byte

Description: The reason code associated with the answering agent's mode, if the agent is in the AUX mode. For agents in AUX for communication servers that do not have EAS and reason codes active, **ANSREASON** is always 0.

ASSIST

Type: bit

Length: 1 bit

Description: Whether the answering agent in this segment requested supervisor assistance on this call. Valid values are 0=NO, 1=YES.

AUDIO

Type: bit

Length: 1 bit

Description: Whether an agent in this segment reported an audio difficulty problem. Valid values are 0=NO, 1=YES.

CALLID

Type: integer

Length: 4 bytes

Description: TA 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. For additional information on "meet-me" conferences, see the appropriate Avaya Communication Manager administrator guide.

CALLING_II

Type: char

Length: 3 bytes

Description: The Information Indicator (II) digits associated with the call. These digits are a two-digit string provided by ISDN Primary Rate Interface (PRI) 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.

CALLING_PTY

Type: char

Length: 13 bytes

Description: The calling party identification, (which is the Automatic Number Identification (ANI)/Station Identification (SID) for Integrated Services Digital Network (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.)

CONFERENCE

Type: bit

Length: 1 bit

Description: Whether the answering agent initiated a conference on this segment. Valid values are 0=NO, 1=YES.

CONSULTTIME

Type: integer

Length: 4 bytes

Description: The 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 obtained by subtracting the **DISPTIME** item from **CONSULTTIME**.

CWC1 through CWC5

Type: char

Length: 17 bytes

Description: The Call Work Code entered by an agent for the Call Segment.

The first five Call Work Codes entered by an agent are stored in the call segment of a call record. The last Call Work Code entered by an agent for a call segment will continue to be stored in the LASTCWC column. The following data items contain Call Work Codes 1 through 5:

- **CWC1**
- **CWC2**
- **CWC3**
- **CWC4**
- **CWC5**

DA_QUEUED

Type: bit

Length: 1 bit

Description: Whether the call was queued as a direct agent call Valid values are 0=NO, 1=YES.

DIALED_NUM

Type: char

Length: 25 bytes

Description: 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.

CALL_DISP

Type: integer

Length: 1 byte

Description: This field represents the call disposition and indicates whether the call in the segment was:

- **1=connected** (CONN, non-ACD call to a measured agent)

  A connected call is a non-ACD call to a measured agent for which CMS receives an indication that the call was connected.

- **2=answered** (ANS, split/skill or direct agent call answered by an agent)

  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.

- **3=abandoned** (ABAN)

  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.

- **4=interflowed** (IFLOW)

  Interflowed calls are calls that are interflowed to an off-switch destination.

- **5=forced busy** (FBUSY)

  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.

- **6=forced disconnect** (FDISC)

  Forced disconnect calls are VDN calls that are disconnected by the communication server due to the execution of a disconnect vector command.

- **7=other** (OTHER)

  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.

  Other calls include any other calls that do not fall into categories such as answered or abandoned. See definitions for individual tables for **OTHERCALLS**.

DISPIVECTOR

Type: integer

Length: 2 bytes

Description: The number of the first vector associated with the disposition VDN (DISPVDN).

DISPPRIORITY

Type: integer

Length: 1 byte

Description: 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.

DISPSKLEVEL

Type: integer

Length: 1 byte

Description: 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.

DISPSPLIT

Type: integer

Length: 2 bytes

Description: 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.

DISPTIME

Type: integer

Length: 4 bytes

Description: The wait time (in the vector, in queue, and ringing) until the disposition is recorded in **CALL_DISP** for the segment. For extension calls made directly to agents (not through a VDN), this will always be zero.

DISPVDN

Type: char

Length: 8 bytes

Description: 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.

DURATION

Type: integer

Length: 4 bytes

Description: The total time the trunk was in use. An alternate description would be the length of time the trunk has been in the TK state. 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.

EQ_LOCID

Type: integer

Length: 2 bytes

Description: The location ID of the trunk. This ID number is not assigned directly to a trunk, but rather to the communication server port network. Therefore, each trunk on the network will have the same location ID number. Valid values are ID numbers 0 through 250.

EQLOC

Type: char

Length: 10 bytes

Description: 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. The value for this field is eight characters followed by two null characters.

EVENT1-9

Type: integer

Length: 1 byte each

Description: The number of times each event (stroke count) button (buttons 1 to 9) was entered for this call segment.

FIRSTVDN

Type: char

Length: 8 bytes

Description: The number of the first VDN associated with the call segment. This will be blank for calls not associated with a VDN.

FIRSTVECTOR

Type: integer

Length: 2 bytes

Description: The number of the first vector associated with the first VDN for the call segment. This will be blank if no vector is involved.

HELD

Type: integer

Length: 1 byte

Description: 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.

HOLDABN

Type: bit

Length: 1 bit

Description: Whether this on-hold call was abandoned from hold in this call segment. Valid values are 0=NO, 1=YES.

LASTCWC

Type: char

Length: 17 bytes

Description: The last call work code (up to 16 digits) entered by the answering agent in this segment.

LASTDIGITS

Type: char

Length: 17 bytes

Description: The last set of collected digits sent to the CMS by the communication server for this call. These are digits the communication server 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 communication server 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.

LASTOBSERVER

Type: char

Length: 10 bytes

Description: The login ID of the last agent who service-observed or bridged on to this call.

MALICIOUS

Type: bit

Length: 1 bit

Description: Whether a malicious call trace was activated for this call segment. Valid values are 0=NO, 1=YES.

NETINTIME

Type: integer

Length: 4 bytes

Description: The time the call spent in a VDN processing at another communication server located elsewhere in the network.

OBS_LOCID

Type: integer

Length: 2 bytes

Description: The location ID of the observing agent. This ID number is not assigned to an agent, but rather to the agent terminal and is associated with the communication server port network ID. An agent may be associated with a location ID only upon login to the ACD. Valid values are ID numbers from 0 through 250.

OBSERVINGCALL

Type: bit

Length: 1 bit

Description: Whether this call represents an agent observing or bridging on to an existing call. Valid values for **OBSERVINGCALL** are 0=NO, 1=YES. Some reports will display only a 1 (YES).

ORIG_LOCID

Type: integer

Length: 2 bytes

Description: The location ID of the calling agent. This ID number is not assigned to an agent, but rather to the agent terminal and it is associated with the communication server port network ID. An agent may be associated with a location ID only upon logging into the ACD. Valid values are ID numbers from 0 through 250.

ORIGHOLDTIME

Type: integer

Length: 4 bytes

Description: The total time the call was put on hold by the originating agent.

ORIGLOGIN

Type: char

Length: 10 bytes

Description: 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.

ORIGREASON

Type: integer

Length: 1 byte

Description: The reason code associated with the originating agent's mode, if the agent is in the AUX mode.

SEGMENT

Type: integer

Length: 1 byte

Description: The number identifying the call segment. Segment numbers are from 1 up to the number of segments in the call.

SEGSTART

Type: integer

Length: 4 bytes

Description: The UNIX time and date when the call segment started. 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.)

SEGSTOP

Type: integer

Length: 4 bytes

Description: The UNIX time and date when the call segment ended. 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 agents is included when calculating the call segment stop time.

SPLIT1

Type: integer

Length: 2 bytes

Description: The first split/skill the call queued to in the first VDN with which it was associated in the call segment.

SPLIT2

Type: integer

Length: 2 bytes

Description: The second split/skill the call was also queued to in the first VDN with which it was associated in the call segment.

SPLIT3

Type: integer

Length: 2 bytes

Description: The third split/skill the call was also queued to in the first VDN with which it was associated in the call segment.

TALKTIME

Type: integer

Length: 4 bytes

Description: The total talk time for the answering agent in this segment.

TKGRP

Type: integer

Length: 2 bytes

Description: 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.

TRANSFERRED

Type: bit

Length: 1 bit

Description: Whether an answering agent initiated a transfer for this call segment. Valid values are 0=NO, 1=YES.

UCID

Type: char

Length: 21 bytes

Description: The Universal Call Identifier - a unique number assigned to this call segment within the customer network.

Extended call record field descriptions

The extended call record enhancements will not be available for use if you choose to use the standard call record format.

The following table contains the call record field description location information.

Field name	Field location in the call record
VDN2 through **VDN9**	Found at the end of the call record in numeric order.
QUEUETIME	Found after **ORIGHOLDTIME**
RINGTIME	Found after **QUEUETIME**
ASAIUUI	Found after **VDN9**
UUI_LEN	Found after **OBS_LOCID**

VDN2 through VDN9

Type: char

Length: 8 bytes

Description: The first 9 VDNs and the last VDN associated with a call segment.

The existing **FIRSTVDN** data item contains the first VDN and **DISPVDN** contains the final VDN. The following data items contain VDNs 2 through 9:

- **VDN2**
- **VDN3**
- **VDN4**
- **VDN5**
- **VDN6**
- **VDN7**
- **VDN8**
- **VDN9**

VDN2 through **VDN9** are populated only when a call touches more than one measured VDN. Data items that are not populated are NULL. **DISPVDN** is usually populated with the same values as the last VDN populated. If the last VDN populated was **VDN7**, then **DISPVDN** will include the same values as **VDN7**. The exception to this rule is when there are 10 or more VDNs associated with a call.

QUEUETIME

Type: integer

Length: 4 bytes

Description: The time a call spends in queue for a call segment

QUEUETIME is the time from when a call first queues to a skill until it starts ringing at an agent. If a call consists of multiple segments, the queue time includes the time associated with that segment. The default is 0 if there is no queue time. **QUEUETIME** is stored for all Avaya CMS supported communication server releases.

RINGTIME

Type: integer

Length: 4 bytes

Description: The time a call spends ringing at an agent station for a call segment

RINGTIME includes all ring time. This is the time a call spends ringing at an agent position and is independent of the final disposition of the call and ring time associated with RONA. If a call consists of multiple segments, each segment contains its associated ring time. The default is 0 if there is no ring time. **RINGTIME** is stored for all Avaya CMS supported communication server releases.

ASAIUUI

Type: unsigned char

Length: 96 bytes

Description: The last ASAI user-to-user information associated with a call segment

If an **ASAIUUI** is not sent, then this field will be NULL.

UUI_LEN

Type: short integer

Length: 2 bytes

Description: The length of the UUI information in bytes

If an **ASAIUUI** is not provided by the communication server, the default length is 0.

Example call scenario and call records

This section presents an example of a record scenario in standard record format. In this example an inbound vectored call is answered by an ACD agent and is then transferred to a VDN. After the transfer, two call records are generated. One call record is generated for every call segment.

> ⚠ **Important:**
> These record scenarios are examples of what a typical call record would look like. The call records you see for your location might vary from this example.

This section includes:

First call segment

The following occurs during the first call segment:

1. A call comes into ACD 1 on TG 32 (carried on the trunk located at 0101A0102) to VDN 43211 which points to Vector 33 at 07:37:10 on 04/16/02.

2. The call queues to Split 1 at medium priority via a "queue to" command, and then queues to backup Split 2 at low priority via a "backup" command.

3. The call waits 10 seconds, then rings for 5 seconds, and then is answered by agent 5018 in Split 1.

4. The caller and agent talk for 44 seconds. The agent transfers the call to VDN 43712 which points to Vector 37. (The call is held for 4 seconds while transferring.)

5. The agent has 42 seconds of after call work during which the agent enters call work code 12345.

First call segment record

The first column in the First call segment record table on page 34 shows whether the field is part of the header or a record field. See Call record field descriptions on page 18 for the field type, field length and field description. The Call report entry column shows data that would appear on a standard CMS call report for this call segment. The Call record entry column shows the binary equivalent of the Call report entry data. This binary data appears in an actual call record file in the order shown in the following table. Spaces are shown between bytes for clarity in the Call record entry column.

First call segment record table

Field	Call report entry	Call record entry
VERSION (header)	11	00001011 00000000 00000000 00000000
SEQUENCE_NUM (header)	1	00000001 00000000 00000000 00000000
CALLID	212	11010100 00000000 00000000 00000000
ACWTIME	00:42	00101010 00000000 00000000 00000000
ANSHOLDTIME	00:04	00000100 00000000 00000000 00000000
CONSULTTIME	00:00	00000000 00000000 00000000 00000000
DISPTIME	00:15	00001111 00000000 00000000 00000000

First call segment record table (continued)

Field	Call report entry	Call record entry
DURATION	04:25	00001001 00000001 00000000 00000000
SEGSTART	07:37:10 04/16/02	10110110 10111110 01010110 00100110
SEGSTOP	07:42:35 04/16/02	10110110 10111110 01011010 11101011
TALKTIME	00:44	00101100 00000000 00000000 00000000
NETINTIME	98	01100010 00000000 00000000 00000000
ORIGHOLDTIME	55	00110111 00000000 00000000 00000000
DISPIVECTOR	33	00100001 00000000
DISPSPLIT	1	00000001 00000000
FIRSTIVECTOR	33	00100001 00000000
SPLIT1	1	00000001 00000000
SPLIT2	2	00000010 00000000
SPLIT3	0	00000000 00000000
TKGRP	32	00100000 00000000

First call segment record table (continued)

Field	Call report entry	Call record entry
EQ_LOCID	33	00100001 00000000
ORIG_LOCID	13	00001101 00000000
ANS_LOCID	23	00010111 00000000
OBS_LOCID	42	00101010 00000000
ASSIST	N	0
AUDIO	N	0
CONFERENCE	N	0
DA_QUEUED	N	0
HOLDABN	N	0
MALICIOUS	N	0
OBSERVINGCALL	N	0
TRANSFERRED	Y	1
AGT_RELEASED	Y	10000000
ACD	1	00000001
CALL_DISP	Ans	00000010
DISPPRIORITY	4 (MED)	00000100
HELD	1	00000001
SEGMENT	1	00000001
ANSREASON	0	00000000
ORIGREASON	0	00000000
DISPSKLEVEL	0	00000000

First call segment record table (continued)

Field	Call report entry	Call record entry
EVENT1-9	0	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
UCID	0006300089088 5813486	00000000 00000000 00000000 00000110 00000011 00000000 00000000 00000000 00001000 00001001 00000000 00001000 00001000 00000101 00001000 00000001 00000011 00000100 00001000 00000110 00000000
DISPVDN	43211	00110100 00110011 00110010 00110001 00110001 00110000 00110000 00000000

First call segment record table (continued)

Field	Call report entry	Call record entry
EQLOC	0101A01	00110000 00110001 00110000 00110001 01000001 00110000 00110001 00110000 00000000 00000000
FIRSTVDN	43211	00110100 00110011 00110010 00110001 00110001 00110000 00110000 00000000
ORIGLOGIN	0	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
ANSLOGIN	5018	00110101 00110000 00110001 00111000 00000000 00000000 00000000 00000000 00000000 00000000

First call segment record table (continued)

Field	Call report entry	Call record entry
LASTOBSERVER	0	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
DIALED_NUM	43211	00110100 00110011 00110010 00110001 00110001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

First call segment record table (continued)

Field	Call report entry	Call record entry
CALLING_PTY	0101A0102	00110000 00110001 00110000 00110001 01000001 00110000 00110001 00110000 00110010 00000000 00000000 00000000 00000000
LASTDIGITS	1	00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
LASTCWC	12345	00110001 00110010 00110011 00110100 00110101 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

First call segment record table (continued)

Field	Call report entry	Call record entry
CALLING_II	0	00000000 00000000 00000000
CWC1	12345	00110001 00110010 00110011 00110100 00110101 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
CWC2	12345	00110001 00110010 00110011 00110100 00110101 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

First call segment record table (continued)

Field	Call report entry	Call record entry
CWC3	12345	00110001 00110010 00110011 00110100 00110101 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

First call segment record table (continued)

Field	Call report entry	Call record entry
CWC4	12345	00110001 00110010 00110011 00110100 00110101 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
CWC5	12345	00110001 00110010 00110011 00110100 00110101 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Second call segment

The scenario continues with the second call segment of the inbound vectored call being transferred to a VDN. The call was transferred by agent 5018 to a VDN. The explains the call record for the second call segment. This record has no header information because it follows the first call segment record in the same file.

The following occurs during the second call segment:

1. The caller is transferred to VDN 43712 which points to Vector 37.

2. The call is queued to Splits 4 and 5 at high priority using a "queue to" step to queue to Split 4, and a "check" step to queue to Split 5.

3. The call waits 1 second and then rings for 2 seconds before being answered by agent 2139 in Split 5.

4. The caller and agent talk for 3 minutes, 19 seconds, and then the agent releases the call.

5. The agent has 1 minute of after call work during which the agent enters call work code 67890. Stop time is 7:42:35 (includes the ACW time that extends beyond the time at which the caller dropped).

Second call segment record

The first column in the Second call segment record table on page 44 shows that all fields in this record are record fields. See Call record field descriptions on page 18 of this document for the field type, length, and description. The Call report entry column shows data that would appear on a standard CMS call report for this call segment. The Call record entry column shows the binary equivalent of the Call report entry data. This binary data appears in an actual call record file in the order shown in the following table. Spaces are shown between bytes for clarity in the Call record entry column.

Second call segment record table

Field	Call report entry	Call record entry
CALLID	212	11010100 00000000 00000000 00000000
ACWTIME	01:00	00111100 00000000 00000000 00000000
ANSHOLDTIME	00:00	00000000 00000000 00000000 00000000
CONSULTTIME	00:00	00000000 00000000 00000000 00000000

Second call segment record table (continued)

Field	Call report entry	Call record entry
DISPTIME	00:03	00000011 00000000 00000000 00000000
DURATION	03:22	11001010 00000000 00000000 00000000
SEGSTART	07:38:13 04/16/94	10110110 10111110 01010111 00100101
SEGSTOP	07:42:35 04/16/94	10110110 10111110 01011010 11101011
TALKTIME	03:19	11000111 00000000 00000000 00000000
NETINTIME	98	01100010 00000000 00000000 00000000
ORIGHOLDTIME	55	00110111 00000000 00000000 00000000
DISPIVECTOR	37	00100101 00000000
DISPSPLIT	5	00000101 00000000
FIRSTIVECTOR	37	00100101 00000000
SPLIT1	4	00000100 00000000
SPLIT2	5	00000101 00000000

Second call segment record table (continued)

Field	Call report entry	Call record entry
SPLIT3	0	`00000000` `00000000`
TKGRP	32	`00100000` `00000000`
EQ_LOCID	33	`00100001` `00000000`
ORIG_LOCID	23	`00010111` `00000000`
ANS_LOCID	23	`00010111` `00000000`
OBS_LOCID	42	`00101010` `00000000`
ASSIST	N	0
AUDIO	N	0
CONFERENCE	N	0
DA_QUEUED	N	0
HOLDABN	N	0
MALICIOUS	N	0
OBSERVINGCALL	N	0
TRANSFERRED	N	0
AGT_RELEASED	Y	`10000000`
ACD	1	`00000001`
CALL_DISP	Ans	`00000010`
DISPPRIORITY	5 (HIGH)	`00000101`
HELD	0	`00000000`
SEGMENT	2	`00000010`

Second call segment record table (continued)

Field	Call report entry	Call record entry
ANSREASON	1	`00000001`
ORIGREASON	9	`00001001`
DISPSKLEVEL	0	`00000000`
EVENT1-9	0	`00000000` `00000000` `00000000` `00000000` `00000000` `00000000` `00000000` `00000000` `00000000`

UCID	00063000890 885813486	`00000000`
		`00000000`
		`00000000`
		`00000110`
		`00000011`
		`00000000`
		`00000000`
		`00000000`
		`00001000`
		`00001001`
		`00000000`
		`00001000`
		`00001000`
		`00000101`
		`00001000`
		`00000001`
		`00000011`
		`00000100`
		`00001000`
		`00000110`
		`00000000`

DISPVDN	4371200	`00110100`
		`00110011`
		`00110111`
		`00110001`
		`00110010`
		`00110000`
		`00110000`
		`00000000`

Second call segment record table (continued)

Field	Call report entry	Call record entry
EQLOC	0101A01	00110000 00110001 00110000 00110001 01000001 00110000 00110001 00110000 00000000 00000000
FIRSTVDN	4371200	00110100 00110011 00110111 00110001 00110010 00110000 00110000 00000000
ORIGLOGIN	5018	00110101 00110000 00110001 00111000 00000000 00000000 00000000 00000000 00000000 00000000
ANSLOGIN	2139	00110010 00110001 00110011 00111001 00000000 00000000 00000000 00000000 00000000 00000000

Second call segment record table (continued)

Field	Call report entry	Call record entry
LASTOBSERVER	0	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
DIALED_NUM	43712	00110100 00110011 00110111 00110001 00110010 00000000

Second call segment record table (continued)

Field	Call report entry	Call record entry
CALLING_PTY	0101A0102	00110000 00110001 00110000 00110001 01000001 00110000 00110001 00110000 00110010 00000000 00000000 00000000 00000000
LASTDIGITS	0	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
LASTCWC	67890	00110110 00110111 00111000 00111001 00110000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Second call segment record table (continued)

Field	Call report entry	Call record entry
CALLING_II	0	00000000 00000000 00000000
CWC1	67890	00110110 00110111 00111000 00111001 00110000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
CWC2	67890	00110110 00110111 00111000 00111001 00110000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Second call segment record table (continued)

Field	Call report entry	Call record entry
CWC3	67890	00110110 00110111 00111000 00111001 00110000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Second call segment record table (continued)

Field	Call report entry	Call record entry
CWC4	67890	00110110 00110111 00111000 00111001 00110000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
CWC5	67890	00110110 00110111 00111000 00111001 00110000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Setting up the ECHI package

This section describes how to install the External Call History Interface (ECHI) software, connect the CMS computer to the receiving computer, and select and set up the receiving computer.

> **Note:**
> If you are upgrading your CMS to a newer version, applications associated with ECHI might have to be rewritten because of changes to call record database items in CMS. Make sure the version field is correct. For more information, see About the call record file format on page 16.

This section includes the following sections:

Customer responsibilities

You must perform the following steps to be able to transfer call history data:

- Purchase the ECHI package.
- Contact the Technical Service Center (TSC) to authorize the feature. If you are an international customer, contact your Avaya representative.
- Install the feature on CMS.
- Set up the connection between the CMS computer and the computer that will be receiving the data.
- Provide the receiving computer applications that will:
  - Allow the receiving computer to receive data via uucp or another file-transfer utility of your choosing.

—   Parse the files of data being transferred.

—   Store the data in some usable fashion (For example, in a database on the receiving computer).

—   Convert the data to a usable format.

Prerequisites

Before installation, be sure you have obtained authorization for the ECHI feature package.

Ports

At installation, the ECHI feature will automatically select the port it will use. For further information on setting up ports, see

Installing the ECHI feature

Once the External Call History Interface feature is installed, you will no longer be able to access any call record data from CMS nor will you be able to use the CMS Call Record report. NICE Analyzer is an optional package that allows you to view external call history data. Contact your Avaya representative for more information.

To install and set up ECHI on a Sun system:

1. Log in as **root**.

    The computer must be in run-level 2, and all file systems must be mounted.

2. Enter:

    cmssvc

    The system displays the `Avaya Call Management System Services Menu.`

3. Enter the number associated with the `auth_display` option and verify that the system is authorized to install the External Call History package.

    Note:
    If External Call History is not authorized but needs to be, call the Avaya National Customer Care Center at 1-800-242-2121. International customers should contact their local Avaya distributor or representative.

4. CMS must be turned off before the package can be installed. Enter:

    cmsadm

    The system displays the Avaya Call Management System Administration Menu.

5. Enter the number associated with the run_cms option.

6. Enter the number associated with the Turn off CMS but Leave IDS running option.

    The system displays the following message:

    ```
    *** CMS is now off ***
    ```

7. Enter:

    cmsadm

    The system displays the Avaya Call Management System Administration Menu.

8. Enter the number associated with the pkg_install option.

    The system displays a list of the installed CMS features.

9. Enter the number that corresponds to External Call History.

    The system displays the following message:

    ```
    Enter the name of the computer to which to send call records (up
    to 256 characters):
    ```

10. Enter the name of the receiving computer.

    The system displays the following message:

    ```
    Enter the full path of the program to transfer call history files
    (default: /cms/dc/chr/uucp_copy):
    ```

> **Note:**
> Currently, CMS uses uucp to transfer External Call History files to a designated remote machine and uustat to check that the files were transferred successfully.
>
> You may optionally change these default programs to be customer-specified file transfer and check programs such as:
>
> — Transmission Control Protocol/Internet Protocol (TCP/IP)
>
> — UNIX commands remote copy (rcp) or copy (cp) using a remote file system (RFS)
>
> To use either communications program enter a different full file path instead of the default uucp/uustat interface path.

11. Enter the full path of the default file transfer utility.

    For example, **/cms/dc/chr/uucp_copy** or optionally enter the full path of your specific file-transfer program.

    The system displays the following message:

    ```
    Enter the full path of the program to check the external call
    history file transmission (default: /cms/dc/chr/uucp_check):
    ```

12. Enter the full path of the default file transmission check program.

    For example, **/cms/dc/chr/uucp_check** or optionally enter the full path of your specific transmission-checking program.

    Note:
    > If you have chosen to enter your own file transfer and transmission checking programs, continue with Step 16, entering the number of call records in the buffer.
    >
    > If you have chosen the CMS defaults, continue with Step 13.

    The system displays the following message:

    ```
    Enter password for nuucp login on XXXXXXXX (up to 8
      characters):
    ```

13. Enter the nuucp password for the receiving computer.

    The system displays the following message:

    ```
    Enter CMS port for connection to XXXXXXXX (s-pdeXXXX):
    ```

14. Enter the port on the CMS computer to be used by the receiving computer. (For a Sun system, s_pdevXXXX will be displayed instead.)

    The system displays the following message:

    ```
    Select a speed for this connection
    1) 19200
    2) 38400
    ```

15. Enter the number associated with the speed for the connection between the CMS and the receiving computer.

    The system displays the following message:

    ```
    Number of call segments to buffer for ACD XXXXXXX (0-99999):
    ```

16. Enter the number of call records to be held in the buffer if the receiving machine cannot accept the data (minimum: 20 MB). This reserves disk space; therefore, sufficient disk space must be available.

Note:
> The range (0-99999) specified in the prompt represents the total allowed over all ACDs. If you expend the entire allotment on ACD 1, you will have no more space available to other ACDs.

> ⚠️ **CAUTION:**
> You will chose external call record data if all the buffers on the CMS computer fill up. (Standard CMS call data is not affected.) The buffers could become filled due to a link failure if the storage capacity of the receiving machine is exceeded or if the receiving machine goes down.

If you want to change the buffer size, you need to change the `Number of call records` field in the `System Setup: Data Storage Allocation` window. For more information, see the section on Data Storage Allocation in the appropriate CMS Administration for your CMS release.

Collection of external call records resumes when uucp finishes copying the files from the CMS buffers to the receiving computer.

Note:
> Repeat Step 16 for each administered ACD.

The system displays the following message:

```
Use the extended ECH record format (y/n):(default: n)
```

17. Select whether ECHI will use the extended ECH record format.

    The system displays the following message:

```
Start ECH in the on or off state: (default off)
```

18. Select whether ECH will start in the on or off state (default is off).

19. Verify that the installation completed successfully, enter:

    tail /cms/install/logdir/admin.log

    If the External Call History package is installed successfully, the system displays the following message:

```
External Call History package installed date/time
```

20. Enter:

    cmsadm

    The system displays the `Avaya Call Management System Administration Menu`.

21. Enter the number associated with the `run_cms` option.

22. Enter the number associated with the `Turn on CMS` option.

    The system displays the following message when CMS is on:

    ```
    *** CMS is now up ***
    ```

Data collection with ECHI

When ECHI is off, records are still being collected in files, but the files are not sent to the receiving computer.

This section contains the following procedures:

- Turning ECHI on or off on page 60
- CMS is not running and ECHI if off on page 61
- CMS is running and ECHI is off on page 61
- CMS is either running or not running and ECHI is on on page 61

Turning ECHI on or off

To turn ECHI on or off:

1. Enter:

    cmsadm

    The system displays the `Avaya Call Management System Administration Menu`.

2. Enter the number associated with the `run_pkg` option.

    The system displays a list of the installed CMS features.

3. Enter the number associated with the `external call history` option.

    The system displays the package status (on or off) and asks if you would like to turn it on or off:

    ```
    ECHI is off. Do you wish to turn it on?
    ```

4. Enter **yes** or **no**, depending on what you want to do with the package. If you turn ECHI on, the system displays:

    ```
    Do you wish to send the full call record buffer?
    ```

5. Enter **yes** or **no**, depending on what you want to do with the buffer.

CMS is not running and ECHI if off

If CMS is not running and if ECHI is off, the system displays the following message:

```
External Call History is off.
    Do you wish to turn it on? (default: yes)
```

- To turn ECHI on press **Enter**.

  The system displays the following message:

  ```
  External Call History package turned on successfully
  ```

CMS is running and ECHI is off

If CMS is running and if ECHI is off, the system displays the following message:

```
External Call History is off.
    Do you wish to turn it on? (default: yes)
```

1. To turn ECHI on press **Enter**.

   The system displays the following message:

   ```
   Do you wish to transmit the full buffer of call records?:
   (default: yes)
   ```

2. Press **Enter**.

   The system displays the following message:

   ```
   External Call History package turned on successfully
   ```

CMS is either running or not running and ECHI is on

If CMS is either running or not running and if ECHI is on, the system displays the following message:

```
External Call History is on.
    Do you wish to turn it off? (default: yes)
```

- To turn ECHI off press **Enter**.

  The system displays the following message:

  ```
  External Call History package turned off successfully
  ```

Verifying the current call record format

To verify the current call record format:

1. Log in as **root**.

2. Enter:

   cmssvc

   The system displays the Avaya Call Management System Services Menu.

3. Enter the number associated with the auth_display option.

   The system displays one of the following options in the External Call History authorizations:

   - Standard

     The ECHI package is using the standard record format.

   - Extended

     The ECHI package is using the extended record format.

Changing the call record format

To change the current call record format:

1. Enter:

   cmsadm

   The system displays the Avaya Call Management System Administration Menu.

   Note:
   If you do not need to preserve the buffered call record data you can start this procedure at Step 11.

2. Enter the number associated with the run_pkg option.

   The system displays a list of the installed CMS features.

3. Enter the number associated with the `External Call History` option.

   The system displays one of the following messages depending on the current state of the ECHI software:

   ● If ECHI is on, the system displays the following message:

   ```
   External Call History is on. Do you wish to turn it off? (y/n)
   ```

   ● If ECHI is off, the system displays the following message:

   ```
   External Call History is off. Do you wish to turn it on? (y/n)
   ```

4. Choose one of the following actions:

   ● If External Call History is currently *on*, select: **y**

   ● If External Call History is currently *off*, perform the following procedure:

   a. Select: **n**

   The system displays the following message:

   ```
   You are using the XXXXXXXX record format.
   Do you wish to change the format to the XXXXXXXXX ECH record
   format? (y/n):(default: n)
   ```

   b. Select: **n**

5. Enter:

   cmsadm

   The system displays the `Avaya Call Management System Administration Menu`.

6. Enter the number associated with the `run_pkg` option.

   The system displays a list of the installed CMS features.

7. Enter the number associated with the `external call history` option.

   The system displays the following message:

   ```
   External Call History is off. Do you wish to turn it on? (y/n)
   ```

8. Turn ECHI on by selecting: **y**

   The system displays the following message:

   ```
   Do you wish to transmit the full buffer of call records (y/n)
   ```

9. Enter: **y**

10. Choose one of the following actions:

    ● If External Call History is currently *on*, select: **y**

● If External Call History is currently *off*, perform the following procedure:

    a. Select: **n**

        The system displays the following message:

```
You are using the XXXXXXXX record format.
Do you wish to change the format to the XXXXXXXXX ECH record
format? (y/n):(default: n)
```

    b. Select: **n**

11. Enter the number associated with the `run_cms` option.

12. Enter the number associated with the `Turn CMS off, but leave IDS running` option.

13. Enter:

    cmsadm

    The system displays the `Avaya Call Management System Administration Menu`.

14. Enter the number associated with the `run_pkg` option.

    The system displays a list of the installed CMS features.

15. Enter the number associated with the `external call history` option.

    The system displays the following message:

```
External Call History is on. Do you wish to turn it off? (y/n)
```

16. Select: **y**

17. Enter:

    cmsadm

    The system displays the `Avaya Call Management System Administration Menu`.

18. Enter the number associated with the `run_pkg` option.

    The system displays a list of the installed CMS features.

19. Enter the number associated with the `external call history` option.

    The system displays the following message:

```
External Call History is off. Do you wish to turn it on? (y/n)
```

20. Select: **n**

    The system displays the following message:

    ```
    You are using the XXXXXXXX record format.
    Do you wish to change the format to the XXXXXXXXX ECH record
    format? (y/n):(default: n)
    ```

21. Enter **y** to change the current External Call History record format.

    > ⚠ **Important:**
    > The system might display a message prompting to delete buffer files. If you delete
    > the buffer files some data will be lost. If you want to change the call record format,
    > you must select **y**.

    The system displays the following message:

    ```
    External Call History uses the XXXXXXXXX record format now.
    ```

22. Turn on CMS.

    At this point, you can upgrade your external reporting packages. ECHI is collecting records
    in the new format, but not sending data.

23. Turn on ECHI.

Assigning ports on the CMS computer

Use the Network Administrator (na) program to address and configure the NTS ports.

> **Note:**
> If you are using NICE Analyzer, you can skip this section because connectivity
> will be provided by the LAN/WAN.

To assign ports on the CMS computer:

1. Log in as **root**.

2. At the system prompt, enter:

   na

   The system displays the following message:

   ```
   command:
   ```

3. To associate all subsequent administration with a specific NTS, enter:

   annex *name_of_your_NTS*

   where, ***name_of_your_NTS*** becomes the default NTS until another NTS is selected using the **annex** command. The default setting for the NTS ports is 9600 bps, 8 bits, no parity, and 1 stop bit.

   The system displays the following message:

   ```
   name of your NTS: Annex-3-UXR7, 64 ports

   command:
   ```

4. Perform one of the following steps, depending upon whether you have a direct connection or a modem connection.

   a. For a direct connection, enter the following set port string command. You must include all the parameters you want to change.

      set port=x-y mode slave speed *value* type hardwired location
      ** *value* user_name *value* output_flow_control none**

   b. For a modem connection, enter the following set port string command:

      set port=*x*-y mode slave dedicated_address *Sunaddress*
      ** control_lines modem_control speed *value* type dial_in term_var**
      **   dial_up broadcast_direction network location *value* user_name**
      **   *value* output_flow_control none**

   You must include all the parameters you want to change. The set port string command sets the parameters for the serial line ports. The **x** value is the port number on the NTS. You can specify a range of ports. The speed value must be either 19200 or 38400 bps and must match the speed of the receiving computer. You can choose your own values for ***location*** and ***user_name*** attributes, but they should be port-specific. See the following table for other values.

   Note:
   > Be sure to set the ***location*** and ***user_name*** (and keep them populated) so that any problems can be traced to the correct port.

Parameter	Recommended setting
mode	`slave`
dedicated_address	Set to the ip address of the Sun host. You can find this by searching (use "grep") the file **/etc/host**.
control_lines	Set to `modem_control`.

Parameter	Recommended setting
speed	Set to match the speed of the receiving computer (`19200` or `38400` bps).
type	`dial_in`
term_var	`dial_up`
broadcast_direction	`network`
location	*`<your location>`*
user_name	*`<your user name>`*

5. When you are finished, reset the ports. Enter:

   reset *[port number]*

   The system displays the following message:

   ```
   resetting serial port number of annex your NTS number
   ```

6. To quit na, enter:

   quit

   ⚠️ **CAUTION:**
   The following procedure requires the CMS computer to be rebooted. CMS will be down while the system is rebooting, and you will lose CMS data.

7. To create a character special device, perform the following steps:

   a. At the console terminal, enter:

      cd /etc/rc2.d

   b. Enter:

      ls -l

   c. Enter:

      vi Speripherals

   d. Add the following system command to the file. You must add one line per each modem or direct connection. (If the **Speripherals** file already exists, add the following line to the file.)

      rtelnet -fmrt *name_of_your_NTS xx* /dev/s_pdev*nxx*

      Where ***n*** is the NTS number and ***xx*** is the number of the port on the NTS.

   Note:
   You can choose your own names, but the device name should reflect the terminal server and port names.

      e. To write and quit the file, press **Esc**, and enter:

        :wq!

      f. Enter:

        chmod 755 Speripherals

8. Reboot the CMS computer.

Connecting the CMS to the receiving computer

The port on the CMS computer is set up for outgoing data only during feature installation. The port on the receiving computer must be set up for only incoming data.

The receiving computer must be connected to the CMS computer by an RS-232 connection. This connection uses hardware flow control and runs at 19200 bps or 38400 bps.

> **Note:**
> If you are using NICE Analyzer, you can skip this section because connectivity will be provided by the LAN/WAN.

This section contains the following information:

- Cabling requirements on page 69
- Requirements for the receiving computer on page 69
- Transferred file names on page 70
- Call record transfer on page 71
- Setting up uucp software on page 71

Cabling requirements

Computers: The computers can be connected to a maximum distance of 50 feet with RS-232 cabling and connectors. If the two computers cannot be located within 50 feet of each other, limited-distance modems can be used. Contact your account representative for information about modems.

NTS: If you are using a small NTS (8 or 16 ports), you need either a 10-pin or 8-pin straight-through connector cable. One end of the cable goes to a port on the small NTS, and the other end of the cable goes to the connector. The connector must then attach to a null modem.

Requirements for the receiving computer

Consider the following points before purchasing or connecting the receiving computer:

- Processor speed should be as high as is practically possible.

- The computer must have file transfer capabilities (the default is uucp, but you may choose another file transfer utility).

   Note:
   > If your receiving computer is a UNIX system, refer to your UNIX documentation to set up uucp. If the receiving computer is a Windows based system, see Setting up uucp software on page 71.
   >
   >> If you are also using NICE Analyzer, you must use uucp as the file transfer utility. Select the default settings when installing ECHI.

- The computer must be loaded with software in order to accept, store, and convert the transferred data.

- Transfer speed must be set to the same speed you chose to use on the CMS computer (19200 or 38400 bps).

- Certain files (such as System and Devices) and permissions must be set up on the receiving computer. See the documentation supplied by the computer or software manufacturer for more information.

- Storage capacity should be large enough to meet your needs based on expected call traffic and the length of time you want to store the data. A minimum of 20 MB of available space is required.

   ⚠ CAUTION:
   > If the disk on the receiving computer fills up, a message is sent to the CMS computer and file transfer halts. This condition could cause files to back up in the CMS computer buffers. If files back up, the call record data that has already been collected up to that moment will eventually be transferred to the receiving computer, but no new data will be collected. The standard CMS data is not affected.

The capacity needed to store one standard record in the database is 322 bytes per record. The capacity needed to store one extended record in the database is 493 bytes per record. One call record is generated for each call segment.

Multiply the number of bytes per record by the number of records you want to store (based on estimated call traffic and the length of time you want to store the data) to calculate approximately how much storage capacity you need.

Transferred file names

The transfer process adds an extension of up to three digits to the call history data file name. Therefore, the file name on the receiving computer is *chrxxxx.x, chrxxxx.xx,* or *chrxxxx.xxx.*

Call record transfer

The transferred call history files are placed in the ~/ directory. (The ~/ directory is the default uucp directory that you can administer on the receiving computer.)

The call history files are sent from CMS at the end of each interval or when a file reaches 1 MB in size. You must set up your receiving application to continuously check the ~/ directory for files of the *chrxxxx* format, and then copy these files to your database. The original files in ~/ should then be deleted. If files are not deleted from the ~/ directory after being copied to your database, they will eventually be overwritten.

> ⚠ **CAUTION:**
> If the disk on the receiving computer fills up, a message is sent to the CMS computer, and file transfer halts. This condition could cause files to back up in the CMS computer buffers. If files back up, the call record data that has already been collected will eventually be transferred to the receiving computer, but no new data will be collected. (Standard CMS data is not affected.)
> The purpose of the sequence numbers is to enable the detection of duplicate files. If you receive two files with the same sequence number, those files are duplicates. If duplicated files are received, refer to the sequence number as previously described in File header on page 17.

Setting up uucp software

You must provide the uucp software for the receiving computer. During setup, the site names (which may be identified by other terms such as node names or host names) must match for configuration on both machines. For assistance, call the Avaya National Customer Care Center at 1-800-242-2121. International customers may contact their local Avaya distributor or representative.

> **Note:**
> If you are using NICE Analyzer, you can skip this section because connectivity will be provided by the LAN/WAN.

This section includes the following procedures:

- Setting up uucp software on a Windows based computer on page 72
- Testing the connection and configuration on page 72
- Reading mail on a UNIX system on page 73

Setting up uucp software on a Windows based computer

The following steps describe a typical method for setting up uucp software on a Windows based computer. Specific details will vary from machine to machine. See your uucp software documentation for more information.

To set up the uucp software:

1. Install and configure the uucp software on the receiving computer according to the instructions provided by the manufacturer.

2. Reboot the computer to verify that no configuration or startup procedures conflict.

3. Define the site name for the receiving computer. This must match the name that was entered on the CMS computer for the call history receiving machine during installation of the feature.

4. Define the Spool directory as `C:\uucp`

5. Define the Public directory as `C:\uucp\public`

6. Define the UUXQT path as `C:\bin`

7. Define the default drive as `C:\`

8. Define the receiving port as `COM1`

9. Define the name of the CMS computer that will be sending data. This needs to be the same as the node name of the CMS computer.

   Enter `uname -n`, and press Enter on the CMS at the UNIX prompt and run the name.

   Note:
   For the following step, the login ID is nuucp. The password is the password that you created during the feature installation.

10. Define the nuucp password that will be used by CMS to log in to the receiving computer to send data. This password must match the nuucp password assigned during the External Call History Interface feature installation.

Testing the connection and configuration

Test the connection and configuration by performing the following procedure.

1. Send e-mail from the receiving computer to the CMS computer.

   a. Enter:

      `mail cms_computer_name!cms`

   b. Create a test message and press `Enter`.

   c. Enter:

      . (*a period*)

2. Send e-mail from the CMS computer to the receiving computer.

   a. Enter:

      mail *receiving_computer_name*!user1

   b. Create a test message and press **Enter**.

   c. Enter:

      . (*a period*)

If both machines receive the mail, the connection is configured and operating properly.

> **Note:**
> The UNIX command *uutry machine_name* will test the machine and provide a log of actions for tracing problems.

Reading mail on a UNIX system

To read mail on a UNIX based system, perform the following steps.

1. To display the message, enter: **mail**

2. To delete the message, enter: **?d**

3. To quit reading mail, enter: **?q**

Index

U

V