



# **Avaya Call Management System**

## **Call History Interface**

Release 18  
February 2017

#### Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

#### Documentation disclaimer

"Documentation" means information published in varying mediums which may include product information, operating instructions and performance specifications that are generally made available to users of products. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of Documentation unless such modifications, additions, or deletions were performed by or on the express behalf of Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

#### Link disclaimer

Avaya is not responsible for the contents or reliability of any linked websites referenced within this site or Documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

#### Warranty

Avaya provides a limited warranty on Avaya hardware and software. Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya's standard warranty language, as well as information regarding support for this product while under warranty is available to Avaya customers and other parties through the Avaya Support website:

<https://support.avaya.com/helpcenter/getGenericDetails?detailId=C20091120112456651010> under the link "Warranty & Product Lifecycle" or such successor site as designated by Avaya. Please note that if You acquired the product(s) from an authorized Avaya Channel Partner outside of the United States and Canada, the warranty is provided to You by said Avaya Channel Partner and not by Avaya.

"Hosted Service" means an Avaya hosted service subscription that You acquire from either Avaya or an authorized Avaya Channel Partner (as applicable) and which is described further in Hosted SAS or other service description documentation regarding the applicable hosted service. If You purchase a Hosted Service subscription, the foregoing limited warranty may not apply but You may be entitled to support services in connection with the Hosted Service as described further in your service description documents for the applicable Hosted Service. Contact Avaya or Avaya Channel Partner (as applicable) for more information.

#### Hosted Service

THE FOLLOWING APPLIES ONLY IF YOU PURCHASE AN AVAYA HOSTED SERVICE SUBSCRIPTION FROM AVAYA OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE), THE TERMS OF USE FOR HOSTED SERVICES ARE AVAILABLE ON THE AVAYA WEBSITE, [HTTPS://SUPPORT.AVAYA.COM/LICENSEINFO](https://support.avaya.com/licenseinfo) UNDER THE LINK "Avaya Terms of Use for Hosted Services" OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, AND ARE APPLICABLE TO ANYONE WHO ACCESSES OR USES THE HOSTED SERVICE. BY ACCESSING OR USING THE HOSTED SERVICE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE DOING SO (HEREINAFTER REFERRED TO INTERCHANGEABLY AS "YOU" AND "END USER"), AGREE TO THE TERMS OF USE. IF YOU ARE ACCEPTING THE TERMS OF USE ON BEHALF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE AUTHORITY TO BIND SUCH ENTITY TO THESE TERMS OF USE. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT WISH TO ACCEPT THESE TERMS OF USE, YOU MUST NOT ACCESS OR USE THE HOSTED SERVICE OR AUTHORIZE ANYONE TO ACCESS OR USE THE HOSTED SERVICE.

#### Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, [HTTPS://SUPPORT.AVAYA.COM/LICENSEINFO](https://support.avaya.com/licenseinfo), UNDER THE LINK "AVAYA SOFTWARE LICENSE TERMS (Avaya Products)" OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AVAYA CHANNEL PARTNER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA

CHANNEL PARTNER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS "YOU" AND "END USER"), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE AVAYA AFFILIATE ("AVAYA").

Avaya grants You a license within the scope of the license types described below, with the exception of Heritage Nortel Software, for which the scope of the license is detailed below. Where the order documentation does not expressly identify a license type, the applicable license will be a Designated System License. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the documentation or other materials available to You. "Software" means computer programs in object code, provided by Avaya or an Avaya Channel Partner, whether as stand-alone products, pre-installed on hardware products, and any upgrades, updates, patches, bug fixes, or modified versions thereto. "Designated Processor" means a single stand-alone computing device. "Server" means a Designated Processor that hosts a software application to be accessed by multiple users. "Instance" means a single copy of the Software executing at a particular time: (i) on one physical machine; or (ii) on one deployed software virtual machine ("VM") or similar deployment.

#### License type(s)

**Designated System(s) License (DS).** End User may install and use each copy or an Instance of the Software only on a number of Designated Processors up to the number indicated in the order. Avaya may require the Designated Processor(s) to be identified in the order by type, serial number, feature key, Instance, location or other specific designation, or to be provided by End User to Avaya through electronic means established by Avaya specifically for this purpose.

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

**Named User License (NU).** You may: (i) install and use each copy or Instance of the Software on a single Designated Processor or Server per authorized Named User (defined below); or (ii) install and use each copy or Instance of the Software on a Server so long as only authorized Named Users access and use the Software. "Named User," means a user or device that has been expressly authorized by Avaya to access and use the Software. At Avaya's sole discretion, a "Named User" may be, without limitation, designated by name, corporate function (e.g., webmaster or helpdesk), an e-mail or voice mail account in the name of a person or corporate function, or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software.

**Shrinkwrap License (SR).** You may install and use the Software in accordance with the terms and conditions of the applicable license agreements, such as "shrinkwrap" or "clickthrough" license accompanying or applicable to the Software ("Shrinkwrap License").

#### Heritage Nortel Software

"Heritage Nortel Software" means the software that was acquired by Avaya as part of its purchase of the Nortel Enterprise Solutions Business in December 2009. The Heritage Nortel Software is the software contained within the list of Heritage Nortel Products located at <https://support.avaya.com/LicenseInfo/> under the link "Heritage Nortel Products," or such successor site as designated by Avaya. For Heritage Nortel Software, Avaya grants Customer a license to use Heritage Nortel Software provided hereunder solely to the extent of the authorized activation or authorized usage level, solely for the purpose specified in the Documentation, and solely as embedded in, for execution on, or for communication with Avaya equipment. Charges for Heritage Nortel Software may be based on extent of activation or use authorized as specified in an order or invoice.

#### Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, Hosted Service, or hardware provided by Avaya. All content on this site, the documentation, Hosted Service, and the product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole

or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

#### Virtualization

The following applies if the product is deployed on a virtual machine. Each product has its own ordering code and license types. Note that each Instance of a product must be separately licensed and ordered. For example, if the end user customer or Avaya Channel Partner would like to install two Instances of the same type of products, then two products of that type must be ordered.

#### Third party components

"Third Party Components" mean certain software programs or portions thereof included in the Software or Hosted Service may contain software (including open source software) distributed under third party agreements ("Third Party Components"), which contain terms regarding the rights to use certain portions of the Software ("Third Party Terms"). As required, information regarding distributed Linux OS source code (for those products that have distributed Linux OS source code) and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply is available in the products, Documentation or on Avaya's website at:

<https://support.avaya.com/Copyright> or such successor site as designated by Avaya. The open source software license terms provided as Third Party Terms are consistent with the license rights granted in these Software License Terms, and may contain additional rights benefiting You, such as modification and distribution of the open source software. The Third Party Terms shall take precedence over these Software License Terms, solely with respect to the applicable Third Party Components, to the extent that these Software License Terms impose greater restrictions on You than the applicable Third Party Terms.

The following applies if the H.264 (AVC) codec is distributed with the product. THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO (I) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (II) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE <HTTP://WWW.MPEGLA.COM>.

#### Service Provider

THE FOLLOWING APPLIES TO AVAYA CHANNEL PARTNER'S HOSTING OF AVAYA PRODUCTS OR SERVICES. THE PRODUCT OR HOSTED SERVICE MAY USE THIRD PARTY COMPONENTS SUBJECT TO THIRD PARTY TERMS AND REQUIRE A SERVICE PROVIDER TO BE INDEPENDENTLY LICENSED DIRECTLY FROM THE THIRD PARTY SUPPLIER. AN AVAYA CHANNEL PARTNER'S HOSTING OF AVAYA PRODUCTS MUST BE AUTHORIZED IN WRITING BY AVAYA AND IF THOSE HOSTED PRODUCTS USE OR EMBED CERTAIN THIRD PARTY SOFTWARE, INCLUDING BUT NOT LIMITED TO MICROSOFT SOFTWARE OR CODECS, THE AVAYA CHANNEL PARTNER IS REQUIRED TO INDEPENDENTLY OBTAIN ANY APPLICABLE LICENSE AGREEMENTS, AT THE AVAYA CHANNEL PARTNER'S EXPENSE, DIRECTLY FROM THE APPLICABLE THIRD PARTY SUPPLIER.

WITH RESPECT TO CODECS, IF THE AVAYA CHANNEL PARTNER IS HOSTING ANY PRODUCTS THAT USE OR EMBED THE G.729 CODEC, H.264 CODEC, OR H.265 CODEC, THE AVAYA CHANNEL PARTNER ACKNOWLEDGES AND AGREES THE AVAYA CHANNEL PARTNER IS RESPONSIBLE FOR ANY AND ALL RELATED FEES AND/OR ROYALTIES. THE G.729 CODEC IS LICENSED BY SIPRO LAB TELECOM INC. SEE <WWW.SIPRO.COM/CONTACT.HTML>. THE H.264 (AVC) CODEC IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO: (I) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (II) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION FOR H.264 (AVC) AND H.265 (HEVC) CODECS MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE <HTTP://WWW.MPEGLA.COM>.

#### Compliance with Laws

You acknowledge and agree that it is Your responsibility for complying with any applicable laws and regulations, including, but not limited to laws and regulations related to call recording, data privacy, intellectual property, trade secret, fraud, and music performance rights, in the country or territory where

the Avaya product is used.

#### 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 is not working on your company's behalf). Be aware that there can 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 Toll 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 Support website: <https://support.avaya.com>, or such successor site as designated by Avaya.

#### Security Vulnerabilities

Information about Avaya's security support policies can be found in the Security Policies and Support section of <https://support.avaya.com/security>. Suspected Avaya product security vulnerabilities are handled per the Avaya Product Security Support Flow (<https://support.avaya.com/css/P8/documents/100161515>).

#### Trademarks

The trademarks, logos and service marks ("Marks") displayed in this site, the Documentation, Hosted Service(s), and product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, its licensors, its suppliers, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the Documentation, Hosted Service(s) and product(s) should be construed as granting, by implication, estoppel, or otherwise, any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

Avaya is a registered trademark of Avaya Inc.

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

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

#### Downloading Documentation

For the most current versions of Documentation, see the Avaya Support website: <https://support.avaya.com>, or such successor site as designated by Avaya.

#### Contact Avaya Support

See the Avaya Support website: <https://support.avaya.com> for product or Hosted Service notices and articles, or to report a problem with your Avaya product or Hosted Service. For a list of support telephone numbers and contact addresses, go to the Avaya Support website: <https://support.avaya.com> (or such successor site as designated by Avaya), scroll to the bottom of the page, and select Contact Avaya Support.



# Contents

|   |           |
|---|-----------|
| <b>Chapter 1: Introduction</b> . . . . .                                | <b>9</b>  |
| <b>Purpose.</b> . . . . .   | <b>9</b>  |
| <b>Intended audience.</b> . . . . .                                     | <b>9</b>  |
| <b>Document changes since last issue</b> . . . . .                      | <b>9</b>  |
| <b>Related resources</b> . . . . .                                      | <b>10</b> |
| <b>Documentation.</b> . . . . .   | <b>10</b> |
| <b>Avaya Mentor videos</b> . . . . .                                    | <b>11</b> |
| <b>Documentation websites</b> . . . . .                                 | <b>11</b> |
| <b>Support.</b> . . . . .   | <b>11</b> |
| <b>Chapter 2: Overview of Internal Call History Interface</b> . . . . . | <b>13</b> |
| <b>Space allocation</b> . . . . .                                       | <b>13</b> |
| <b>Call record processing</b> . . . . .                                 | <b>14</b> |
| <b>About the call record format</b> . . . . .                           | <b>14</b> |
| <b>Chapter 3: Overview of External Call History Interface</b> . . . . . | <b>15</b> |
| <b>Space allocation</b> . . . . .                                       | <b>15</b> |
| <b>Call record processing</b> . . . . .                                 | <b>15</b> |
| <b>About the call record file format</b> . . . . .                      | <b>16</b> |
| <b>Chapter 4: Call record formats</b> . . . . .                         | <b>17</b> |
| <b>Data characteristics</b> . . . . .                                   | <b>17</b> |
| <b>File name format.</b> . . . . .                                      | <b>17</b> |
| <b>File header</b> . . . . .  | <b>17</b> |
| <b>About the VERSION field</b> . . . . .                                | <b>18</b> |
| <b>About the SEQUENCE_NUMBER</b> . . . . .                              | <b>18</b> |
| <b>Example call record file header.</b> . . . . .                       | <b>18</b> |
| <b>Call record field descriptions</b> . . . . .                         | <b>18</b> |
| <b>ACD.</b> . . . . .   | <b>18</b> |
| <b>ACWTIME.</b> . . . . .   | <b>19</b> |
| <b>AGENTSKILLLEVEL.</b> . . . . .                                       | <b>19</b> |
| <b>AGENTSURPLUS</b> . . . . .   | <b>19</b> |
| <b>AGT_RELEASED.</b> . . . . .  | <b>19</b> |
| <b>ANS_ATTRIB_ID</b> . . . . .  | <b>20</b> |
| <b>ANS_LOCID</b> . . . . .  | <b>20</b> |
| <b>ANSHOLDTIME.</b> . . . . .   | <b>20</b> |
| <b>ANSLOGIN</b> . . . . .   | <b>20</b> |
| <b>ANSREASON.</b> . . . . .   | <b>21</b> |
| <b>ASAI_UUI.</b> . . . . .  | <b>21</b> |
| <b>ASSIST</b> . . . . .   | <b>21</b> |

## Contents

|                             |    |
|-----------------------------|----|
| AUDIO . . . . .             | 21 |
| CALL_DISP . . . . .         | 21 |
| CALLID . . . . .            | 22 |
| CALLING_II . . . . .        | 23 |
| CALLING_PTY . . . . .       | 23 |
| CONFERENCE . . . . .        | 23 |
| CONSULTTIME . . . . .       | 23 |
| CWC1 through CWC5 . . . . . | 24 |
| DA_QUEUED . . . . .         | 24 |
| DIALED_NUM. . . . .         | 24 |
| DISPIVECTOR . . . . .       | 24 |
| DISPPRIORITY . . . . .      | 25 |
| DISPSKLEVEL . . . . .       | 25 |
| DISPSPLIT . . . . .         | 25 |
| DISPTIME . . . . .          | 25 |
| DISPVDN . . . . .           | 26 |
| DURATION . . . . .          | 26 |
| EQ_LOCID . . . . .          | 26 |
| EQLOC . . . . .             | 26 |
| EVENT1-9. . . . .           | 26 |
| FIRSTVDN . . . . .          | 27 |
| FIRSTVECTOR . . . . .       | 27 |
| HELD . . . . .              | 27 |
| HOLDABN . . . . .           | 27 |
| INTERRUPTDEL . . . . .      | 27 |
| ICRRESENT . . . . .         | 28 |
| ICRPULLREASON . . . . .     | 28 |
| LASTCWC . . . . .           | 29 |
| LASTDIGITS . . . . .        | 29 |
| LASTOBSERVER. . . . .       | 29 |
| MALICIOUS. . . . .          | 29 |
| NETINTIME . . . . .         | 29 |
| OBS_ATTRIB_ID . . . . .     | 30 |
| OBS_LOCID . . . . .         | 30 |
| OBSERVINGCALL . . . . .     | 30 |
| ORIG_ATTRIB_ID . . . . .    | 30 |
| ORIG_LOCID . . . . .        | 30 |
| ORIGHOLDTIME . . . . .      | 31 |
| ORIGLOGIN. . . . .          | 31 |
| ORIGREASON . . . . .        | 31 |
| PREFSKILLLEVEL . . . . .    | 31 |

|   |           |
|---|-----------|
| <b>QUEUETIME</b> . . . . .  | <b>32</b> |
| <b>RINGTIME.</b> . . . . .  | <b>32</b> |
| <b>SEGMENT</b> . . . . .  | <b>32</b> |
| <b>SEGSTART</b> . . . . .   | <b>33</b> |
| <b>SEGSTART_UTC.</b> . . . . .  | <b>33</b> |
| <b>SEGSTOP.</b> . . . . .   | <b>33</b> |
| <b>SEGSTOP_UTC</b> . . . . .  | <b>33</b> |
| <b>SPLIT1</b> . . . . .   | <b>33</b> |
| <b>SPLIT2</b> . . . . .   | <b>34</b> |
| <b>SPLIT3</b> . . . . .   | <b>34</b> |
| <b>TALKTIME</b> . . . . .   | <b>34</b> |
| <b>TENANT.</b> . . . . .  | <b>34</b> |
| <b>TKGRP</b> . . . . .  | <b>34</b> |
| <b>TRANSFERRED</b> . . . . .  | <b>35</b> |
| <b>UCID</b> . . . . .   | <b>35</b> |
| <b>UUI_LEN</b> . . . . .  | <b>35</b> |
| <b>VDN2 through VDN9</b> . . . . .                                    | <b>35</b> |
| <b>Example call scenario and call records</b> . . . . .               | <b>36</b> |
| <b>First call segment</b> . . . . .                                   | <b>37</b> |
| <b>First call segment record</b> . . . . .                            | <b>37</b> |
| <b>Second call segment</b> . . . . .                                  | <b>52</b> |
| <b>Second call segment record</b> . . . . .                           | <b>52</b> |
| <b>Chapter 5: Setting up the ECHI package</b> . . . . .               | <b>67</b> |
| <b>Customer responsibilities.</b> . . . . .                           | <b>67</b> |
| <b>Prerequisites</b> . . . . .  | <b>68</b> |
| <b>Ports</b> . . . . .  | <b>68</b> |
| <b>Installing the ECHI feature</b> . . . . .                          | <b>68</b> |
| <b>Data collection with ECHI</b> . . . . .                            | <b>70</b> |
| <b>Turning ECHI on or off</b> . . . . .                               | <b>71</b> |
| <b>CMS is not running and ECHI is off</b> . . . . .                   | <b>71</b> |
| <b>CMS is running and ECHI is off.</b> . . . . .                      | <b>72</b> |
| <b>CMS is either running or not running and ECHI is on.</b> . . . . . | <b>72</b> |
| <b>Appendix A: Changes in releases R12.x through R18</b> . . . . .    | <b>73</b> |
| <b>ECH fields added in CMS R12.x.</b> . . . . .                       | <b>73</b> |
| <b>ECH fields added in R16.0.</b> . . . . .                           | <b>73</b> |
| <b>ECH fields populated in R16.1</b> . . . . .                        | <b>74</b> |
| <b>ECH fields added in R16.3.</b> . . . . .                           | <b>74</b> |
| <b>ECH fields added in R17.</b> . . . . .                             | <b>74</b> |

**Contents**

|  |    |
|--|----|
| ECH fields added in R18. . . . .   | 74 |
| Releases with no ECH field changes . . . . .   | 75 |
| Appendix B: Calculation changes for ECH data . . . . .                                       | 77 |
| Service Level/Call Profile reporting . . . . .   | 77 |
| VDN return destination and the transfer bit . . . . .  | 77 |
| Agent state time for consultative transfers when agent reconnects with caller after transfer | 78 |
| Index . . . . .  | 81 |



# Chapter 1: Introduction

---

## Purpose

The document describes the format of the Call History data files and how to transfer these files to another computer.

---

## Intended audience

This document is intended for contact center administrators and support personnel. Users of this document must be familiar with Avaya Call Management System (CMS).

---

## Document changes since last issue

The following change has been made to this document since the last issue:

- Changed the value of the version field to 180 in the call record file header.
- Modified the description of ANS\_ATTRIB\_ID, OBS\_ATTRIB\_ID, and ORIG\_ATTRIB\_ID.
- Updated Appendix A with the changes for R18.
- Added TENANT as a new field and described the TENANT field.
- The length of the VDN and login fields was changed to 17 characters.
- Updated the first and second call segment record tables with the updated type and length of ANS\_ATTRIB\_ID, OBS\_ATTRIB\_ID, and ORIG\_ATTRIB\_ID. Added TENANT as a field in the first and second call segment record tables.

March 2016

- Removed all instances of NICE Analyzer as it is no longer supported.

July 2016

- Explained how the last VDN call is handled.

February 2017

- Changed the DISPTIME description from the wait time (in the vector, in queue, and ringing) to the wait time from the beginning of the call (including, but not limited to, the time in the vector, in queue, and ringing).

## Related resources

## Documentation

See the following documents.

**Table 1: Related documents**

| Title   | Use this document to:  | Audience  |
|---|--|---|
| Implementing  |  |   |
| <i>Avaya Call Management System Software Installation, Maintenance, and Troubleshooting for Solaris</i> | Install, maintain, and troubleshoot CMS on the Solaris operating system.   | Implementation engineers and system administrators  |
| <i>Avaya Call Management System Overview and Specification</i>  | Describes tested product characteristics and capabilities including product overview and feature descriptions, interoperability, performance specifications, security, and licensing requirements. | Intended for anyone who wants to gain a high-level understanding of the product features, functions, capacities, and limitations within the context of solutions and verified reference configurations. |

---

## Avaya Mentor videos

Avaya Mentor is an Avaya-run channel on YouTube that includes technical content on how to install, configure, and troubleshoot Avaya products.

Go to <http://www.youtube.com/AvayaMentor> and perform one of the following actions:

- Enter a key word in the Search Channel to search for a specific product or topic.
- Scroll down Playlists, and click the name of a topic to see the available list of videos posted on the site.

---

## Documentation websites

All CMS documentation can be found at <http://www.support.avaya.com>. New issues of CMS documentation will be placed on this website when available.

Use the following websites to view related support documentation:

- Information about Avaya products and service  
<http://www.avaya.com>
- Sun hardware documentation  
<http://docs.sun.com>
- Dell hardware documentation  
<http://www.dell.com>

---

## Support

Visit the Avaya website at <http://www.support.avaya.com> for the most up-to-date documentation, product notices, and knowledge articles. You can also search for release notes, downloads, and resolutions to issues. Use the online service request system to create a service request. Chat with live agents to get answers to questions, or request an agent to connect you to a support team if an issue requires additional expertise.



# Chapter 2: Overview of Internal Call History Interface

The Internal Call History Interface (ICHI) feature is the default call history recording application that is delivered with CMS. If you do not want to use the External Call History Interface (ECHI) feature package, you can use ICHI to collect and process call records directly from the Informix database using applications such as ODBC. As with ECHI, you must allocate space to store the records.

**Note:**

You cannot use both ICHI recording and ECHI recording at the same time.

This section includes the following topics:

- [Space allocation](#) on page 13
- [Call record processing](#) on page 14
- [About the call record format](#) on page 14

---

## Space allocation

You need to allocate space with Data Storage Allocation (DSA). You must use DSA to administer the number of call records stored for each ACD. The call rate, which determines the rate at which ICH records are written to the database tables without degrading CMS performance, is documented in *Avaya Call Management System Overview and Specification*. The sum total of call records which can be stored is shown in the CMS limits documented in *Avaya Aura™ Communication Manager System Capacities Table*. After the call record limit is reached, CMS automatically begins deleting the oldest records in the table to make room for newer records. The CMS Maintenance Backup will back up the **call\_rec** table, which can be restored or migrated. You may also create custom reports from the **call\_rec** table using CMS custom reports or CMS Supervisor Report Designer.

**Note:**

In addition to using custom and designer reports, you can also use ODBC/JDBC to pull the **call\_rec** records to an external client for reporting.

## Call record processing

Call records are stored in the call\_rec table if the following conditions are true:

- The master ACD link must be operational.
- The value for the **Number of call records** field in Data Storage Allocation must be greater than zero.

---

## About the call record format

The call record field descriptions are the same for ICH and ECH. For more information please refer to [Call record field descriptions](#) on page 18.

# Chapter 3: Overview of External Call History Interface

The External Call History Interface (ECHI) feature 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.

**Note:**

You cannot use both ICHI recording and ECHI recording at the same time.

This section includes the following topics:

- [Space allocation](#) on page 15
- [Call record processing](#) on page 15
- [About the call record file format](#) on page 16

---

## Space allocation

You need to allocate space with Data Storage Allocation (DSA). You must use DSA to administer the number of call records stored for each ACD. The call rate, which determines the rate at which ECH records are written to the call history files without degrading CMS performance, is documented in *Avaya Call Management System Overview and Specification*. The sum total of call records which can be stored is shown in the CMS limits documented in *Avaya Aura™ Communication Manager System Capacities Table*. After the call record limit is reached, CMS begins deleting the oldest files not extracted from the `/cms/cmstables` directory.

---

## Call record processing

As calls are processed, call records are recorded in a file (one record per call segment) until either the file size reaches approximately 1 MB or at the end of an interval. Then the file of call record segments will be transferred to another computer via uucp or other designated software.

The following requirements must be met for ECHI to transfer the files at the end of an interval:

- The master ACD link must be operational.

## Chapter 3: Overview of External Call History Interface

- For the master ACD, the value for the **Number of call records** field in Data Storage Allocation must be greater than zero.

If the above conditions are not met, the files will not be transferred until they reach the maximum size.

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.

---

## About the call record file format

The Call record field descriptions are transferred by the Call History Interface feature. For more information please refer to [Call record formats](#) on page 17.



# Chapter 4: Call record formats

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 17
- [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.

**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.

## Chapter 4: Call record formats

This section includes the following topics:

- [About the VERSION field](#) on page 18
- [About the SEQUENCE\\_NUMBER](#) on page 18
- [Example call record file header](#) on page 18

### About the VERSION field

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

The **VERSION** field will contain the value 180 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.

|         | <b>VERSION</b>                      | <b>SEQUENCE_NUMBER</b>              |
|---------|-------------------------------------|-------------------------------------|
| Decimal | 180                                 | 1                                   |
| Binary  | 10110100 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. Call History Interface does not use CMS database items. The 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".

### 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.

## AGENTSKILLLEVEL

**Type:** integer

**Length:** 1 byte

**Description:** Level assigned to the agent for the skill the call is delivered to.

**Note:**

This call record field is actively used in CMS R16.1 and later.

## AGENTSURPLUS

**Type:** integer

**Length:** 1 byte

**Description:** Whether the call is delivered under agent surplus or call surplus condition.

- 0 = NA
- 1 = Call surplus: ACD call was routed to the agent after waiting in the queue.
- 2 = Agent surplus: ACD call was routed to the agent without waiting in the queue.

**Note:**

This call record field is actively used in CMS R16.1 and later.

## 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\_ATTRIB\_ID

**Type:** char

**Length:** 21 bytes

**Description:** An alphanumeric value set in the Communication Manager server which the call center customer enters as a character string. This string represents a combination of characteristics of an agent defined by the call center management for use in reporting.

## 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:** 17 bytes

**Description:** With EAS, the ANSLOGIN field has the agent login ID of the agent logged into the phone extension. This is true whether the skill involved is measured or not. If no agent is logged into the phone extension, the ANSLOGIN field is blank.

With non-EAS, the ANSLOGIN field is populated with the agent's login ID only if the associated ACD split hunt group is measured. If the split is unmeasured, the ANSLOGIN field is blank.

## 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.

## ASAI\_UII

**Type:** unsigned char

**Length:** 97 bytes

**Description:** The last ASAI user-to-user information associated with a call segment. If an **ASAI\_UII** is not sent, then this field will be NULL.

## 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.

## CALL\_DISP

**Type:** integer

**Length:** 1 byte

## Chapter 4: Call record formats

**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**.
- **8=icr-pulled** (ICRPULLED)  
ICR pulled calls are calls pulled back from a CM by ICR. Starting with ICR 2.0, a call can be pulled back from the CM at any point in the call progression before an agent answers.

## CALLID

**Type:** integer

**Length:** 4 bytes

**Description:** 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 are unique for calls over a day per ACD. 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:** 25 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 24 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.

## 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), 7=REDIRECT. 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.

Some direct agent call scenarios will result in DISPSPLIT being set to the agent's top skill.

## DISPTIME

**Type:** integer

**Length:** 4 bytes

**Description:** The wait time from the beginning of the call (including, but not limited to, the 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:** 17 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:** 17 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.

## INTERRUPTDEL

**Type:** Integer

**Length:** 1 byte

**Description:** Flag indicating that agent became available when agent was interrupted from an interruptible aux state with an Interruptible AUX reason code. Valid values are:

- **0=N/A**  
Agent was not interrupted while in Aux.
- **1=auto-in-interrupt**  
The agent has automatically been made available (forced) to handle a contact. Once the contact is completed, the agents are automatically made available for the next call.
- **2=manual-in-interrupt**  
The agent has automatically been made available (forced) to handle a contact. In this work-mode, the agent is placed in the wrap-up state at the end of the call.
- **3=notify-interrupt**  
The agent has been notified but remains in Aux until the agent manually becomes available using an auto-in or manual-in or dial code.

**Note:**

This call record field is actively used in CMS R16.1 and later.

### ICRRESENT

**Type:** integer

**Length:** 1byte

**Description:** Whether the call was re-sent to the CM by ICR. Valid values are 0=NO, 1=YES.

### ICRPULLREASON

**Type:** integer

**Length:** 1 byte

**Description:** The reason a call was pulled back by ICR. Valid values are:

- **0=Call not pulled back**
- **1=Resources not available**
- **2=Wait time increased drastically**
- **3=Caller receiving treatment**
- **4=Network failure recovery**
- **5=VP failure recovery**
- **6=Caller interaction**

## 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:** 17 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\_ATTRIB\_ID

**Type:** char

**Length:** 21 bytes

**Description:** An alphanumeric value set in the Communication Manager server which the call center customer enters as a character string. This string represents a combination of characteristics of an agent defined by the call center management for use in reporting.

## 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\_ATTRIB\_ID

**Type:** char

**Length:** 21 bytes

**Description:** An alphanumeric value set in the Communication Manager server which the call center customer enters as a character string. This string represents a combination of characteristics of an agent defined by the call center management for use in reporting.

## 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:** 17 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.

## PREFSKILLLEVEL

**Type:** integer

**Length:** 1 byte

**Description:** Whether the call is delivered via the preferred skill level check vector command.

- 0 = NA: Call was not delivered via the preferred skill level command or no skill preference was specified.
- 1 = not preferred: Agent's level for the skill does not match the level specified in the check vector command.
- 2 = preferred: Agent's level for the skill matches the preferred skill level specified by the check vector command.

**Note:**

This call record field is actively used in CMS R16.1 and later.

## 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.

## SEGMENT

**Type:** integer

**Length:** 1 byte

**Description:** A numeric value representing a segment. Segment numbers are from 1 up to the number of segments in the call.

The value that appears in this field does not always follow the order in which the segment occurs in the progression of the call. In most cases, the value of 1 indicates the first segment, but it is not possible to rely on this to determine the order of segments. The only reliable method to determine the segment order is to sort the segments on the SEGSTART value. However, the SEGMENT field imposes the limitation that two segments in a call can have the same SEGSTART value. If this is the case, it is not possible to determine which segment occurred first. But this scenario is rare for most call centers.



## 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.)

## SEGSTART\_UTC

**Type:** integer

**Length:** 4 bytes

**Description:** SEGSTART time (UNIX time and date) adjusted to be Coordinated Universal Time. See SEGSTART for details.

## 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.

## SEGSTOP\_UTC

**Type:** integer

**Length:** 4 bytes

**Description:** SEGSTOP time (UNIX time and date) adjusted to be Coordinated Universal Time. See SEGSTOP for details.

## 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.

## TENANT

**Type:** integer

**Length:** 4 bytes

**Description:** The ID of the tenant partition to which the resource belongs.

## 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:** Indicates that a call transfer was attempted on this segment. It does not indicate that the transfer was successful or completed. For example, if an agent initiates a consultative transfer to a second agent, but decides to cancel the transfer after the second agent answers and talks to the first agent, the TRANSFERRED field remains set even though the first agent retains control of the call.

TRANSFERRED also indicates transfers that are not initiated by an agent. For example, when a call is transferred after a Q Signal (QSIG) path replacement request, this flag is set to 1 in one of the segments, but no agent is involved.

Possible values are 0=NO and 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.

## UUI\_LEN

**Type:** short integer

**Length:** 2 bytes

**Description:** The length of the UUI information in bytes

If an **ASAI\_UUI** is not provided by the communication manager server, the default length is 0.

## VDN2 through VDN9

**Type:** char

**Length:** 17 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 more than 9 VDNs associated with a call, the last VDN is tracked as **DISPVDN**.

---

## Example call scenario and call records

This section presents an example of a record scenario in 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.

**Note:**

The tables in the following examples do not include any of the fields which were used in the extended ECH or the fields added with R16.



**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](#) on page 37
- [First call segment record](#) on page 37
- [Second call segment](#) on page 52
- [Second call segment record](#) on page 52

## First call segment

The following occurs during the first call segment:

1. A call is resent by ICR to 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 37 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 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                            |
|---------------------------------|-------------------|--|
| <b>VERSION</b><br>(header)      | 180               | 10110100<br>00000000<br>00000000<br>00000000 |
| <b>SEQUENCE_NUM</b><br>(header) | 1                 | 00000001<br>00000000<br>00000000<br>00000000 |
| <b>CALLID</b>                   | 212               | 11010100<br>00000000<br>00000000<br>00000000 |

First call segment record table (continued)

| Field               | Call report entry    | Call record entry                            |
|---------------------|----------------------|--|
| <b>ACWTIME</b>      | 00:42                | 00101010<br>00000000<br>00000000<br>00000000 |
| <b>ANSHOLDTIME</b>  | 00:04                | 00000100<br>00000000<br>00000000<br>00000000 |
| <b>CONSULTTIME</b>  | 00:00                | 00000000<br>00000000<br>00000000<br>00000000 |
| <b>DISPTIME</b>     | 00:15                | 00001111<br>00000000<br>00000000<br>00000000 |
| <b>DURATION</b>     | 04:25                | 00001001<br>00000001<br>00000000<br>00000000 |
| <b>SEGSTART</b>     | 07:37:10<br>04/16/02 | 10100110<br>11010100<br>10111011<br>00111100 |
| <b>SEGSTART_UTC</b> | 14:37:10<br>04/16/02 | 00010110<br>00110111<br>10111010<br>00111100 |
| <b>SEGSTOP</b>      | 07:42:35<br>04/16/02 | 11101011<br>11010101<br>10111100<br>00111100 |
| <b>SEGSTOP_UTC</b>  | 14:42:35<br>04/16/02 | 01011011<br>00111000<br>10111010<br>00111100 |
| <b>TALKTIME</b>     | 00:44                | 00101100<br>00000000<br>00000000<br>00000000 |

First call segment record table (continued)

| Field        | Call report entry | Call record entry                            |
|--------------|-------------------|--|
| NETINTIME    | 98                | 01100010<br>00000000<br>00000000<br>00000000 |
| ORIGHOLDTIME | 55                | 00110111<br>00000000<br>00000000<br>00000000 |
| QUEUETIME    | 10                | 00001010<br>00000000<br>00000000<br>00000000 |
| RINGTIME     | 5                 | 00000101<br>00000000<br>00000000<br>00000000 |
| TENANT       | 1                 | 00000001<br>00000000<br>00000000<br>00000000 |
| DISPIVECTOR  | 37                | 00100101<br>00000000                         |
| DISPSPLIT    | 1                 | 00000001<br>00000000                         |
| FIRSTIVECTOR | 37                | 00100101<br>00000000                         |
| SPLIT1       | 1                 | 00000001<br>00000000                         |
| SPLIT2       | 2                 | 00000010<br>00000000                         |
| SPLIT3       | 0                 | 00000000<br>00000000                         |
| TKGRP        | 32                | 00100000<br>00000000                         |
| EQ_LOCID     | 33                | 00100001<br>00000000                         |

First call segment record table (continued)

| Field         | Call report entry | Call record entry    |
|---------------|-------------------|----------------------|
| ORIG_LOCID    | 13                | 00001101<br>00000000 |
| ANS_LOCID     | 23                | 00010111<br>00000000 |
| OBS_LOCID     | 42                | 00101010<br>00000000 |
| UUI_LEN       | 0                 | 00000000<br>00000000 |
| ASSIST        | N                 | 0                    |
| AUDIO         | N                 | 0                    |
| CONFERENCE    | N                 | 0                    |
| DA_QUEUED     | N                 | 0                    |
| HOLDABN       | N                 | 0                    |
| MALICIOUS     | N                 | 0                    |
| OBSERVINGCALL | Y                 | 1                    |
| 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   | 3                 | 00000011             |



First call segment record table (continued)

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

First call segment record table (continued)

| Field     | Call report entry | Call record entry  |
|-----------|-------------------|--|
| EQLOC     | 0101A01           | 00110000<br>00110001<br>00110000<br>00110001<br>01000001<br>00110000<br>00110001<br>00110000<br>00000000<br>00000000   |
| FIRSTVDN  | 43211             | 00110100<br>00110011<br>00110010<br>00110001<br>00110001<br>00110000<br>00110000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000 |
| ORIGLOGIN | 5011              | 00110101<br>00110000<br>00110001<br>00110001<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000 |















First call segment record table (continued)

| Field           | Call report entry | Call record entry  |
|-----------------|-------------------|--|
| VDN2-9          | 0                 | 00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000 |
| ASAI_UUI        | 0                 | 97 bytes<br>of<br>00000000   |
| INTERRUPTDEL    | 0                 | 00000000   |
| AGENTSURPLUS    | 0                 | 00000000   |
| AGENTSKILLLEVEL | 3                 | 00000011   |
| PREFSKILLLEVEL  | 0                 | 00000000   |
| ICRRESENT       | 1                 | 00000001   |
| ICRPULLREASON   | 0                 | 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 [Second call segment record](#) on page 52 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 52 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 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                            |
|----------------|-------------------|--|
| <b>CALLID</b>  | 212               | 11010100<br>00000000<br>00000000<br>00000000 |
| <b>ACWTIME</b> | 01:00             | 00111100<br>00000000<br>00000000<br>00000000 |

Second call segment record table (continued)

| Field               | Call report entry    | Call record entry                            |
|---------------------|----------------------|--|
| <b>ANSHOLDTIME</b>  | 00:00                | 00000000<br>00000000<br>00000000<br>00000000 |
| <b>CONSULTTIME</b>  | 00:00                | 00000000<br>00000000<br>00000000<br>00000000 |
| <b>DISPTIME</b>     | 00:03                | 00000011<br>00000000<br>00000000<br>00000000 |
| <b>DURATION</b>     | 03:22                | 11001010<br>00000000<br>00000000<br>00000000 |
| <b>SEGSTART</b>     | 07:38:13<br>04/16/02 | 11100101<br>11010100<br>10111011<br>00111100 |
| <b>SEGSTART_UTC</b> | 14:38:13<br>04/16/02 | 01010101<br>00110111<br>10111100<br>00111100 |
| <b>SEGSTOP</b>      | 07:42:35<br>04/16/02 | 01010101<br>11010101<br>10111011<br>00111100 |
| <b>SEGSTOP_UTC</b>  | 14:42:35<br>04/16/02 | 01011011<br>00111000<br>10111100<br>00111100 |
| <b>TALKTIME</b>     | 03:19                | 11000111<br>00000000<br>00000000<br>00000000 |
| <b>NETINTIME</b>    | 98                   | 01100010<br>00000000<br>00000000<br>00000000 |

Second call segment record table (continued)

| Field               | Call report entry | Call record entry                            |
|---------------------|-------------------|--|
| <b>ORIGHOLDTIME</b> | 55                | 00110111<br>00000000<br>00000000<br>00000000 |
| <b>QUEUETIME</b>    | 1                 | 0000010<br>00000000<br>00000000<br>00000000  |
| <b>RINGTIME</b>     | 2                 | 00000010<br>00000000<br>00000000<br>00000000 |
| <b>TENANT</b>       | 1                 | 00000001<br>00000000<br>00000000<br>00000000 |
| <b>DISPIVECTOR</b>  | 37                | 00100101<br>00000000                         |
| <b>DISPSPLIT</b>    | 5                 | 00000101<br>00000000                         |
| <b>FIRSTIVECTOR</b> | 37                | 00100101<br>00000000                         |
| <b>SPLIT1</b>       | 4                 | 00000100<br>00000000                         |
| <b>SPLIT2</b>       | 5                 | 00000101<br>00000000                         |
| <b>SPLIT3</b>       | 0                 | 00000000<br>00000000                         |
| <b>TKGRP</b>        | 32                | 00100000<br>00000000                         |
| <b>EQ_LOCID</b>     | 33                | 00100001<br>00000000                         |
| <b>ORIG_LOCID</b>   | 23                | 00010111<br>00000000                         |
| <b>ANS_LOCID</b>    | 23                | 00010111<br>00000000                         |

Second call segment record table (continued)

| Field         | Call report entry | Call record entry  |
|---------------|-------------------|--|
| OBS_LOCID     | 0                 | 00000000<br>00000000   |
| UUI_LEN       | 0                 | 00000000<br>00000000   |
| ASSIST        | N                 | 0  |
| AUDIO         | N                 | 0  |
| CONFERENCE    | N                 | 0  |
| DA_QUEUED     | N                 | 0  |
| HOLDABN       | N                 | 0  |
| MALICIOUS     | N                 | 0  |
| OBSERVINGCALL | Y                 | 1  |
| TRANSFERRED   | Y                 | 1  |
| AGT_RELEASED  | Y                 | 10000000   |
| ACD           | 1                 | 00000001   |
| CALL_DISP     | Ans               | 00000010   |
| DISPPRIORITY  | 5 (HIGH)          | 00000101   |
| HELD          | 0                 | 00000000   |
| SEGMENT       | 2                 | 00000010   |
| ANSREASON     | 1                 | 00000001   |
| ORIGREASON    | 9                 | 00001001   |
| DISPSKLEVEL   | 3                 | 00000011   |
| EVENT1-9      | 0                 | 00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000 |

Second call segment record table (continued)

| Field          | Call report entry        | Call record entry  |
|----------------|--------------------------|--|
| <b>UCID</b>    | 00063000890<br>885813486 | 00000000<br>00000000<br>00000000<br>00000110<br>00000011<br>00000000<br>00000000<br>00000000<br>00001000<br>00001001<br>00000000<br>00001000<br>00001000<br>00000101<br>00001000<br>00000001<br>00000011<br>00000100<br>00001000<br>00000110<br>00000000 |
| <b>DISPVDN</b> | 43712                    | 00110100<br>00110011<br>00110111<br>00110001<br>00110010<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000                                     |



Second call segment record table (continued)

| Field     | Call report entry | Call record entry  |
|-----------|-------------------|--|
| EQLOC     | 0101A01           | 00110000<br>00110001<br>00110000<br>00110001<br>01000001<br>00110000<br>00110001<br>00110000<br>00000000<br>00000000   |
| FIRSTVDN  | 43712             | 00110100<br>00110011<br>00110111<br>00110001<br>00110010<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000 |
| ORIGLOGIN | 5018              | 00110101<br>00110000<br>00110001<br>00111000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000 |













Second call segment record table (continued)

| Field           | Call report entry | Call record entry  |
|-----------------|-------------------|--|
| VDN2-9          | 0                 | 00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000<br>00000000 |
| ASAI_UUI        | 0                 | 97 bytes<br>of<br>00000000   |
| INTERRUPTDEL    | 0                 | 00000000   |
| AGENTSURPLUS    | 0                 | 00000000   |
| AGENTSKILLLEVEL | 3                 | 00000011   |
| PREFSKILLLEVEL  | 0                 | 00000000   |
| ICRRESENT       | 0                 | 00000000   |
| ICRPULLREASON   | 0                 | 00000000   |







# Chapter 5: 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 [Call record formats](#) on page 17.

This section includes the following sections:

- [Customer responsibilities](#) on page 67
- [Prerequisites](#) on page 68
- [Ports](#) on page 68
- [Installing the ECHI feature](#) on page 68
- [Data collection with ECHI](#) on page 70

---

## 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.

---

## Installing the ECHI feature

CMS can only support Internal Call History or External Call History at one time. The two packages can not be used simultaneously. Once the External Call History Interface feature is installed, call data will no longer be inserted into the call\_rec table and you will not be able to access the CMS Call Record report.

To install and set up ECHI on a Sun system:

1. Verify that:
  - A separate computer is available for the storage and reporting of call records.
  - The Avaya CMS software is off and the IDS software is on.
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 the current authorizations. The system can display different authorizations depending on the version of Avaya CMS on your system.
4. Verify that the system is authorized for the ECHI package. If ECHI is not authorized but should be, see section *Configuring Avaya CMS authorizations* in *Avaya CMS Software Installation, Maintenance, and Troubleshooting for Solaris*.

5. Enter:  
**cmsadm**  
The system displays the Avaya Call Management System Administration Menu.

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

The system displays the following message:

```
The CMS Features that can be installed are
 1) forecasting
 2) external call history
Enter choice (1-2) or q to quit:
```

**Note:**

The system only displays feature packages that are authorized but not yet installed.

7. Enter the number that corresponds to the ECHI package (in this example, 2).

The system displays the following message:

```
Enter full path of the program to transmit the external call
history files: (default: /cms/dc/chr/no_op.sh)
```

8. Press **Enter**.

The system displays the following message:

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

9. Press **Enter**.

The system displays the following message:

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

10. Enter the number of call records to be held in the buffer if the Call History machine cannot accept the data. Repeat this step for each administered ACD.

The system displays the following message:

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

11. Select whether ECHI will start in the on or off state (default is off). If the receiving system has not yet been set up, the recommended state is off. ECHI can be turned on at a later date with the `run_pkg` option in the Avaya Call Management System Administration Menu.

If the setup determines that you do not have enough file space, you get the following warning message:

```
Failed to find sufficient file space for CMS data.

WARNING: You do not currently have sufficient file space for your
existing CMS data. At this point you should turn on CMS, go to the
"Data Storage Allocation" screen, and verify/modify the
administration, or go to the "Free Allocation" screen and verify/
modify your existing free space.

External call history package installed with warnings.
```

12. Verify that the installation completed successfully by entering:

```
tail /cms/install/logdir/admin.log
```

If the ECHI package was installed successfully, the system displays the following message:

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

You can edit this file in order to add comments about the packages that were installed or authorized.

13. If you are not installing any other feature packages, do the following to turn on the Avaya CMS software:
  - a. Enter:

```
cmssvc
```

The system displays the Avaya Call Management System Services Menu.
  - b. Enter the number associated with the `run_cms` option.
  - c. Enter the number associated with the `Turn on CMS` option.

---

## 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 71
- [CMS is not running and ECHI is off](#) on page 71
- [CMS is running and ECHI is off](#) on page 72
- [CMS is either running or not running and ECHI is on](#) on page 72

---

## 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 is off

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

- 1.

```
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
```



# Appendix A: Changes in releases R12.x through R18

---

## ECH fields added in CMS R12.x

In this release, the customer was provided with the choices of a basic format and an extended format. The following set of extended fields was added:

- ringtime
- queuetime
- vdn2-vdn9
- uui\_len
- asai\_uui

---

## ECH fields added in R16.0

In this release, the choice of basic format was abandoned and only the extended format was supported. The following set of extended fields was added:

- segstart\_utc
- segstop\_utc
- interruptdel
- agentsurplus
- agentskilllevel
- prefskilllevel

**Note:**

The interruptdel, agentsurplus, agentskilllevel and prefskilllevel fields were not populated in this release.

## ECH fields populated in R16.1

CMS populated the following fields in this release as a result of added support for CM 5.2 features:

- interruptdel
- agentsurplus
- agentskilllevel
- prefskilllevel

---

## ECH fields added in R16.3

The following fields were added in this release for providing ICR support:

- ICRRESENT
- ICRPULLREASON

---

## ECH fields added in R17

The following fields were added in this release and will be populated in a later release:

- ORIG\_ATTRIB\_ID
- ANS\_ATTRIB\_ID
- OBS\_ATTRIB\_ID

---

## ECH fields added in R18

The following fields were added in this release:

- TENANT

The following fields were populated in this release:

- ORIG\_ATTRIB\_ID
- ANS\_ATTRIB\_ID

- OBS\_ATTRIB\_ID

**Note:**

The field type of the ORIG\_ATTRIB\_ID, ANS\_ATTRIB\_ID, and OBS\_ATTRIB\_ID fields was changed to alphanumeric and each of these fields is 21 characters long. ORIG\_ATTRIB\_ID, ANS\_ATTRIB\_ID, and OBS\_ATTRIB\_ID are moved to the end of the **call\_rec** record.

The length of the VDN and login fields is changed to 17 characters.

---

## Releases with no ECH field changes

There were no changes to the ECH fields in the CMS R13, R14, R15 and R16.2 releases.

## Appendix A: Changes in releases R12.x through R18

# Appendix B: Calculation changes for ECH data

---

## Service Level/Call Profile reporting

In CMS R14, the *ring time* calculation of a requeued call was reset when the following occurred:

- if a call to a skill was not answered by the agent
- if RONA was activated
- if the call was re-queued to the same skill

In CMS R17, when this sequence of events occurs, the *ring time* calculation accrues the entire duration of the ring time. The *ring time* is not reset if the call is requeued or RONA is activated. This provides an accurate computation of the total *ring time* for the caller.

---

## VDN return destination and the transfer bit

### VDN return destination feature

The normal VDN return destination feature operation works as follows:

If the VDN return destination is set on the CM switch for a particular VDN, and an agent completes a call from that VDN, the caller is automatically routed to a different VDN for further processing.

When the call is routed to this different VDN, CMS creates a new call record in the ECH file to gather the information for the portion of the call processed after the VDN return destination operation. Though the call is transferred, this transfer is not an agent initiated transfer. Therefore, no transfer bit should be set in the ECH record after the VDN return destination operation.

### Bug related to the VDN return destination feature

There was a bug in the R15 and earlier versions of CMS for a particular scenario using VDN return destination and another agent transfer.

## Appendix B: Calculation changes for ECH data

The scenario was as follows:

- An ACD call comes into a VDN and is answered by an agent. This call creates an ECH record.
- The agent then transfers the call to a second agent. This creates a second ECH record.
- The second agent handles the call. After the call completes, the caller is routed to a different VDN by VDN return destination. This creates a third ECH call record.

The proper recording of the transfer bit information should have set the transfer bit only in the first ECH call record when the actual agent transfer occurred. However, when the third call record was created for the VDN return destination operation, the transfer bit for the first ECH record was removed and a transfer bit was set on the second ECH record.

In addition, the transfer bit was set for any VDN return destination, even if no other transfers occurred.

### Resolution

This bug was corrected in R16. In CMS R16, no transfer bits are set in the ECH records for VDN return destination. In case of this scenario, the transfer bit is correctly set for the agent transfers.

#### Note:

Customers using the releases of CMS that contain this bug see more transfer bits set in the ECH records.

---

## Agent state time for consultative transfers when agent reconnects with caller after transfer

### Blind and consultative transfers

There are two approaches for creating a transfer. They are as follows:

- Blind transfer: In this case, an agent pushes the transfer button, dials a number, and presses the transfer button a second time.
- Consultative transfer: In this case, an agent pushes the hold button, then dials a number, places the second call on hold, rejoins the first caller, then pushes the transfer button, and joins the two calls together.

### Bug related to consultative transfers

In CMS R16.2 and earlier releases, for a consultative transfer, after the agent rejoined the original call, the talk time for the original call was no longer tracked. In addition, the ACW and AUX time for the agent was not computed properly.

**Resolution**

This bug was corrected in CMS R16.3. In R16.3 and later, the entire duration of the talk time for the original caller is tracked and counted. The ACW and AUX time for the agent are also computed properly.

## Appendix B: Calculation changes for ECH data



# Index

---

## C

- call record
  - field descriptions . . . . . [18](#)
  - processing . . . . . [14](#), [15](#)
- CMS space allocation . . . . . [13](#), [15](#)

---

## D

- data collection
  - turn on or off . . . . . [70](#)
- disk pairs . . . . . [73](#), [77](#)

---

## E

- ECHI
  - call record file format . . . . . [17](#)
  - data characteristics . . . . . [17](#)
  - External Call History Interface . . . . . [13](#), [15](#)
  - file header . . . . . [17](#)
  - file name format . . . . . [17](#)
  - overview . . . . . [13](#), [15](#)
  - port set up . . . . . [68](#)
- ECHI software
  - set up . . . . . [67](#)

---

## H

- helplines . . . . . [11](#)

---

## I

- installing ECHI . . . . . [68](#)

---

## M

- mirrored system disk pairs . . . . . [73](#), [77](#)

---

## S

- set up
  - ECHI ports . . . . . [68](#)
  - ECHI software . . . . . [67](#)

---

## T

- turn on or off
  - data collection. . . . . [70](#)

---

## V

- videos . . . . . [11](#)

