Using Web Services on Avaya Aura® Media Server

Release 7.7
Issue 1
August 2015

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 by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya may generally make available to users of its products and Hosted Services. 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 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: http://support.avaya.com 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 a 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 IF YOU PURCHASE A 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, HTTP://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. YOUR USE OF THE HOSTED SERVICE SHALL BE LIMITED BY THE NUMBER AND TYPE OF LICENSES PURCHASED UNDER YOUR CONTRACT FOR THE HOSTED SERVICE, PROVIDED, HOWEVER, THAT FOR CERTAIN HOSTED SERVICES IF APPLICABLE, YOU MAY HAVE THE OPPORTUNITY TO USE FLEX LICENSES, WHICH WILL BE INVOICED ACCORDING TO ACTUAL USAGE ABOVE THE CONTRACT LICENSE LEVEL. CONTACT AVAYA OR AVAYA’S CHANNEL PARTNER FOR MORE INFORMATION ABOUT THE LICENSES FOR THE APPLICABLE HOSTED SERVICE, THE AVAILABILITY OF ANY FLEX LICENSES (IF APPLICABLE), PRICING AND BILLING INFORMATION, AND OTHER IMPORTANT INFORMATION REGARDING THE HOSTED SERVICE.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, HTTP://SUPPORT.AVAYA.COM/LICENSEINFO 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.

Database License (DL). End User may install and use each copy or an Instance of the Software on one Server or on multiple Servers provided that each of the Servers on which the Software is installed communicates with no more than one Instance of the same database.

CPU License (CP). End User may install and use each copy or Instance of the Software on a number of Servers up to the number indicated in the order provided that the performance capacity of the Server(s) does not exceed the performance capacity specified for the Software. End User may not re-install or operate the Software on Server(s) with a larger performance capacity without Avaya's prior consent and payment of an upgrade fee.

Named User License (NU). You may: (i) install and use the Software on a single Designated Processor or Server per authorized Named User (defined below); or (ii) install and use 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 currently available for license from Avaya is the software contained within the list of Heritage Nortel Products located at http://support.avaya.com/LicenseInfo under the link "Heritage Nortel Products" or such successor site as designated by Avaya. For Heritage Nortel Software, Avaya grants You 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

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: http://support.avaya.com/Copyright or such successor site as designated by Avaya. You agree to the Third Party Terms for any such Third Party Components.

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.

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: http://support.avaya.com or such successor site as designated by Avaya. Suspected security vulnerabilities with Avaya products should be reported to Avaya by sending mail to: securityalerts@avaya.com.

Downloading Documentation

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

Contact Avaya Support

See the Avaya Support website: http://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: http://support.avaya.com (or such successor site as designated by Avaya), scroll to the bottom of the page, and select Contact Avaya Support.

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, 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 and Avaya Aura® are registered trademarks 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.

Contents

Chapter 1: Introduction

Purpose

Avaya Aura® Media Server defines Web Services APIs so that your applications can access the content storage system of Avaya Aura® MS.

Use this document to implement the SOAP-based web services offered by Avaya Aura® MS.

Intended audience

This document is intended for application developers implementing web services to provision and manage Avaya Aura® MS.

Documentation

See the following related documents at http://support.avaya.com.

Title	Use this document to:	Audience
Implementing and Administering		
Deploying and Updating Avaya Aura® Media Server Appliance	Deploy, update, and troubleshoot Avaya Aura® MS appliances deployed in the VMware® virtualized environment or on Avaya Common Servers.	System administrators, implementation engineers, and support personnel
Installing and Updating Avaya Aura® Media Server Application	Deploy, update, upgrade, and patch non-appliance versions of Avaya Aura® Media Server deployed on Platform Vendor Independent (PVI) servers.	System administrators, implementation engineers, and support personnel
Implementing and Administering Avaya Aura® Media Server	Configure and administer Avaya Aura® Media Server.	System administrators and implementation engineers

Table continues…

Title	Use this document to:	Audience
Alarms and Performance Measurements for Avaya Aura® Media Server	Evaluate system performance metrics and troubleshoot events and alarms.	System administrators, implementation engineers, and support personnel

Support

Go to the Avaya Support website at http://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. You can also request an agent to connect you to a support team if a problem requires more expertise.

Warranty

Avaya provides a 90-day limited warranty on the Avaya Aura® MS software. For detailed terms and conditions, see the sales agreement or other applicable documentation. Additionally, for the standard warranty description of Avaya and the details of support, see **Help & Policies** > **Policies & Legal** > **Maintenance and Warranty Information** on the Avaya Support website at http://support.avaya.com.

For additional information, see **Help & Policies** > **Policies & Legal** > **License Terms**.

For more details on the hardware maintenance for supported products, see http://portal.avaya.com/ptlWeb/services/SV0452.

Chapter 2: New in this release

Case-sensitive Content IDs

Content Store Namespaces, Content Groups, and Content IDs are case-insensitive in Release 7.6 and earlier. In Avaya Aura® MS 7.7, Content Store changes to case-sensitive by default for Content IDs only. Namespaces and Content Groups remain case-insensitive.

During major release upgrades from Avaya MS 7.6 to Avaya MS 7.7, the AmsUpgrade tool and automated upgrades preserve the case-insensitive setting. Thus, backward compatibility is maintained with existing applications that can be scripted without regard to case.

Avaya support personnel can use an engineering configuration parameter to change the default case-sensitivity for each installation.

Chapter 3: Content Store Web services

Overview

Content Store is a component of Avaya Aura® MS that provides a reliable, highly available, and persistent storage capability for data and files. Content Store stores content and makes the content accessible to all nodes in an Avaya Aura® MS cluster. Content Store can be used by any application client that has storage requirements that align with the functionality provided by Content Store. In cluster configurations, Content Store mirrors content in real-time with a peer Content Store. A backup capability is also available to protect the stored content.

Content Management

For each item that an application stores in Content Store, a record is entered in a database. If an optional file is associated with the item, that file is stored in a storage root. The storage root is the disk space on the Avaya Aura® MS hard disk, and the root space is managed exclusively by the Content Store.

Content Store stores various system and application accessible metadata fields in the database record for each item in an application. Applications can store and retrieve files of any type in Content Store. Client applications implement functionality using a simple request and response API to access the storage utility web services as shown in the following table:

Content Store Services	
Operation	**Description**
AdministrativeCommand	Storage Space setup and maintenance.
IngestContent	Saves content in storage.
FetchContent	Retrieves previously ingested content.
SearchContent	Finds content matching specified criteria.
QueryContentMetaData	Gets all available detailed metadata attributes on content.
ModifyContentMetaData	Alters the metadata attributes on content.
ModifyXMLContent	Alters XML-based content.
QueryXMLContent	Gets the current value of XML-based content.

Table continues…

Content Store Services	
Operation	**Description**
SearchXMLContent	Searches the content of ingested XML files.
DeliverContent	Constructs an e-mail and sends the specified content to an email user.

Content Store ingests content for storage from URLs provided by clients in IngestContent requests. Content Store organizes the content in Content Groups that are stored under Namespaces. Ingested files are generically handled and catalogued by Content Store, which does not examine or alter the files.

Client applications add, modify, update, and delete content.

To provide high capacity and high availability, Content Store is scaled automatically with the cluster. Content Stores are automatically enabled on each media server in a cluster. The application content in the Content Stores of a cluster is synchronized automatically.

Master Content Stores are configured on both the Primary and Secondary servers of a load sharing cluster. Master Content Stores are also configured on the Primary and Backup servers of a High Availability cluster. The dual master Content Store configuration provides full hardware and functionality redundancy. Standard cluster nodes provide Content Stores that contain synchronized content for local access.

Content Stores communicate with each other when handling requests. A connection to any one Content Store in a cluster is sufficient for any client application. Data integrity and synchronization in a cluster are handled automatically by Content Store peers. However, it is more efficient to provision new media files directly to one of the master Content Stores. When a content modification request is received at a Standard node Content Store, the request is first forwarded to one of the master Content Stores in the cluster for processing.

Using Content Store

Storage organization

Content Stores have an organized storage root that has Namespaces containing Content Groups and the actual content. The Namespace is the top-level container. Content Group containers exist under Namespaces. Namespaces and Content Groups are analogous to folders in a computer operating system. The actual content is identified by a Content ID and stored under a Content Group.

The organization hierarchy is as follows:

```
Namespace_I
      ContentGroup_A
         Content ID1
         Content ID2
      ContentGroup_B
         Content ID1
```

```
         Content ID2
   Namespace_II
      ContentGroup_A
         Content ID1
         Content ID2
```

Content Store maintains multiple Namespaces at the highest level with one level of Content Groups under those namespaces. There is no further depth or nesting of Content Groups. However, the optional Element Manager (EM) logically simulates more levels by using forward slashes in Content Group names. Any application or tool can also simulate logical subgroups using forward slashes or a custom scheme with another character. The use of the forward slash for logical subgroups is compatible with the Media Management tool provided by EM.

Without logical Content Group hierarchy, a Namespace has one level of Content Group under it. The following example shows two Content Groups, `Announcements` and `Music`, in the `Avaya Namespace`:

```
Avaya
    Announcements
    Music
```

By applying logical Content Group hierarchy in the form of the forward slash, the Content Groups can represent many levels of storage organization. For example:

```
Avaya
    Announcements
    Music
      Music/rock
      Music/rock/80s
      Music/rock/90s
      Music/country
      Music/classical
      Music/classical/Bach
      Music/classical/Handel
```

The logical interpretation by your applications of this hierarchy is as follows:

```
Avaya
      Announcements
      Music
         rock
            80s
            90s
         country
         classical
            Bach
            Handel
```

`Music/rock/80s` is a single unique Content Group in the Content Store, but can be parsed by the application using the forward slash delimiter. The delimiter provides additional organizational meaning representing three Content Groups.

Characteristics of Content Store storage:

- Content Store assigns unique system-generated names to all Namespace directories and Content files. Clients are unaware of actual file system names, except the names visible in the URLs used for accessing the content. Files names are subject to change as files are updated by client applications with new versions. Your applications must always query the latest URL or use the FetchContent service to get the latest content.

- Content Store creates its own folders and ID names using unique numbers for actual disk storage.
- You must not navigate the file system of the Content Store. Your applications should not hardcode references to or refer to the stored content directly. Applications must use only the FetchContent operation or the URLs provided in an IngestContent response or a QueryContentMetaData response.
- Content Store stores a record in the database for each file. The record contains system data, such as the time received, and application-managed metadata fields.
- Content Store does not alter the data stored in metadata fields labeled as **Opaque** except as directed by your client application.
- Applications can instruct Content Store to modify simple, flat XML files using the provided XML operations.
- Files of any type can be generically stored in the Content Store. Content Store cannot look into the contents of any file type other than flat XML-type files.
- The management system provides a utility to back up and restore the content of Content Store.

Naming rules for Namespaces and Content Groups:

- Namespaces must be unique.
- Content Group names must be unique within a Namespace.
- Namespace and Content Group names must be less than 128 characters.
- Namespace and Content Group names are case-insensitive.
- Namespace and Content Group names cannot begin with an at (@) symbol.
- Namespace and Content Group names must not contain braces ({ }), apostrophes ('), or backward slashes (\).
- Namespace and Content Group names must not contain spaces.

Naming rules for Content IDs

Naming rules for Content IDs:

- Content ID names must be unique within a Content Group.
- Content ID names must be less than 128 characters.
- Content ID names are case-sensitive.
- Content ID names cannot begin with an at (@) symbol.
- Content ID names must not contain braces ({ }), apostrophes ('), or backward slashes (\).

Note:

If the Content Store receives an Ingest request message that does not contain a Content ID, then the Content Store generates a Content ID and returns the ID to the client in the Ingest response message.

Internal audits

Content deletion

Content Stores periodically audit content for items marked as deleted by client applications. The actual content removal occurs when this audit runs for the periodic garbage collection. Therefore, content previously marked as deleted can still be present for a few minutes. Applications must implement logic to inspect the delete status of any content when it is accessed, to ensure proper functionality. The delayed delete provides your application time to undelete an item, as long as the undelete function is performed within the audit interval. An option is also available to immediately delete a content item. Therefore, avoid the previously mentioned issues related to deletion and the audit.

After each garbage collection is complete, the audit calculates the free disk space. The system raises an alarm if the free space threshold is exceeded.

Content expiration

Applications can provide an expiry date and time for stored data. An audit deletes all expired content. The deletion time of the expired content depends on the configurable audit frequency. If an application accesses content that is expired but not yet removed by audit, then Content Store immediately deletes the content. Content Store responds to the client request with an error response indicating that the content does not exist.

Storage space audit

Content Store periodically audits the entire content that it manages. During the audit, Content Store performs the necessary garbage collection and integrity checks of the database and file system data. The audit removes unmanaged files from the configured storage root because the storage root is a restricted area managed by Content Store.

Content Group real-time access and mirroring

Access to a particular Content Group is serialized by Content Store to protect data integrity. However, Content Store can process different Content Groups at the same time. Content Group access serialization ensures that Content Store:

- Processes requests for a particular Content Group in the order received
- Prevents simultaneous reads or writes on one server
- Prevents simultaneous writes in clusters where two master Content Stores are present.

Content Store peers use the synchronization protocol to acquire write permission from each other. Write permission must be acquired before any content is committed to storage. Content Store grants permission to a requesting peer Content Store after any in-progress local write operations are complete.

After a write operation is complete in Content Store, it mirrors the content to the peer Content Store. Unlike write operations, reads do not require permission from the peer Content Store. Read operations can process simultaneously for the same Content Group on different Content Stores in the cluster.

When either the primary or secondary Content Store is out of service or disconnected from each other, each Content Store raises an alarm.

After restarting or reconnecting with a peer Content Store, the two Content Stores synchronize their content. Each Content Store compares its content with the peer and shares new content or deletes obsolete content.

Choosing a service endpoint node

Client applications must connect to the Content Store SOAP service endpoint on the primary or secondary nodes in an Avaya Aura® MS cluster. The aim is to ensure full functionality and optimal performance. Standard node servers, which have no Content Stores, do not support IngestContent and FetchContent.

E-mail address handling

Content Store does not validate email addresses received from client applications. Email addresses are received by Content Store in client API requests. For example, the DeliverContent request is rejected by downstream processing of the email on the SMTP mail server. However, your client application does not receive notification of the failure from the remote SMTP server. Email addresses are expected to be valid and provisioned on the customer system.

Parameter validation

Mandatory (M), conditional (C), and optional (O) parameters in various combinations provide the required input for the functionality requested by the application. You must carefully include or exclude the parameters to obtain the required functionality. If you specify an operation type in the request then an explicit functionality is requested. Content Store processes the request if all the mandatory parameters pass validation. Content Store ignores any extraneous parameters unrelated to the requested operation. Content Store returns an error message to the client if the inclusion of extra parameters creates an ambiguous request or causes potentially undesirable data impact.

Parameters defined as integers must contain an integer number. Integer parameters must start with a digit from 0 to 9, but the system ignores any characters after the initial integers. For example, the system accepts `123abc` as `123` but rejects `x123` or `hello123abc` because these values do not start with a number.

Any parameters related to naming content are subject to errors based upon the naming rules for Namespaces, Content Groups, and Content IDs.

Accessing Content Store WSDL

About this task

Content Store Web Service Description Language (WSDL) describes Content Store web services that Avaya Aura® MS provides. The Content Store WSDL provides the definitions for the SOAP requests that your client applications can send to Avaya Aura® MS. With these definitions, you can gain access to and manage data in the Content Store.

Procedure

1. Download the Content Store WSDL from an installed Avaya Aura® MS using the following URL, sent from a trusted SOAP node: http://<hostname>:7410/maswscstoreapi.wsdl.
2. You can determine the configured SOAP port by navigating to **Avaya® Aura MS** > **Element Manager** > **System Configuration** > **Network Settings** > **Advanced Settings** > **Port Assignments** > **Management SOAP Server**.

   The default SOAP port is 7410 for http and 7411 for https.
3. You can view other SOAP settings by navigating to **Avaya® Aura MS** > **Element Manager** > **System Configuration** > **Network Settings** > **General Settings** > **SOAP**.

Content Store service endpoint

An installed Avaya Aura® MS application processes web service requests received from trusted SOAP nodes at the following URLs:

- http://<hostname>:7410/soap
- http://<hostname>:7411/soap

Your application uses this address for Avaya Aura® MS Web Services.

Implementing Content Store Web services

This section provides information about the Web Service services provided by the Content Store WSDL API.

AdministrativeCommand

The AdministrativeCommand request provides provisioning and general management services for Content Store. Operations include creating and deleting Namespaces and Content Groups and querying both Content Group and Namespace sizes. The following table describes the AdministrativeCommand request parameters:

AdministrativeCommand Request		
Parameter	**Optionality**	**Description**
Namespace	C	Namespace
ContentGroup	C	Content Group
ContentID	C	The parameter required for Content ID-related operations
Operation	M	The operation to perform
NewName	C	The new name to be used for a rename
CopyToNamespace	C	The target Namespace for a copy
CopyToContentGroup	C	The target ContentGroup for a copy
Gslid	O	Global Session Log Identifier for traceability of the transaction

Depending upon the operation and context, the system ignores received parameters not listed as **Required Parameters**. Otherwise, the system responds to the request with an error. The enumerated values for the Operation parameter are listed in the following table along with the required parameters for each operation:

AdministrativeCommand Operations		
Operation	**Description**	**Required Parameters**
CreateNamespace	Creates a new Namespace with the provided name.	Operation Namespace
CreateContentGroup	Creates a new Content Group with the name provided in the specified Namespace.	Operation Namespace ContentGroup
QueryNamespacePresence	Determines if a Namespace exists. The responses are: 1=found; 0=not found	Operation Namespace
QueryContentGroupPresence	Determines if a Content Group exists. The responses are: 1=found; 0=not found	Operation Namespace ContentGroup
QueryNamespaceSize	Calculates and returns the size of all mailboxes in the specified Namespace.	Operation Namespace
QueryContentGroupSize	Calculates and returns the size of the specified Content Group.	Operation Namespace ContentGroup
CopyContent	Copies one content item from one Namespace/Content Group to the	Operation Namespace

Table continues…

AdministrativeCommand Operations		
Operation	**Description**	**Required Parameters**
	specified Namespace/ ContentGroup. NewName indicates the name of a new content item.	ContentGroup CopyToNamespace CopyToContentGroup ContentID NewName
CopyContentGroup	Copies an entire Content Group and its content to the specified Namespace/ContentGroup	Operation Namespace ContentGroup CopyToNamespace CopyToContentGroup
DeleteNamespace	Deletes the specified Namespace.	Operation Namespace
DeleteContentGroup	Deletes the specified content group from the specified Namespace.	Operation Namespace ContentGroup
GetNamespaces	Returns a list of all Namespaces on the Content Store.	Operation
GetContentGroups	Returns a list of all Content Groups in the specified Namespace.	Operation Namespace
RenameContentGroup	Renames an existing Content Group.	Operation Namespace ContentGroup NewName
RenameContentID	Renames a content item.	Operation Namespace ContentGroup ContentID NewName
RenameNamespace	Renames an existing Namespace.	Operation Namespace NewName

The Result and AdministrativeQueryItem parameters are mutually exclusive and conditional, depending on the request. These two parameters are not included in the same response. The following table provides information about the AdministrativeCommand response parameters:

AdministrativeCommand Response		
Parameter	**Optionality**	**Description**
Result	C	The result of the command.
AdministrativeQueryItem	C	A list of one or more items of information answering the query command.

CreateNamespace Example

The first step to create a managed storage space on Content Store is to create a Namespace for application data. The following example uses the CreateNamespace operation of the AdminstrativeCommand to create a new Namespace called `Music`:

```
<soapenv:Body>
   <v1:AdministrativeCommand>
      <Operation>CreateNamespace</Operation>
      <NameSpace>Music</NameSpace>
   </v1:AdministrativeCommand>
</soapenv:Body>

<SOAP-ENV:Body>
   <ns3:AdministrativeCommandResponse/>
</SOAP-ENV:Body>
```

GetNamespaces Example

To get a list of all Namespaces on Avaya Aura® MS, use the GetNamespaces AdministrativeCommand Operation. The following example shows a response containing a list of the six Namespaces on the system:

```
<soapenv:Body>
   <v1:AdministrativeCommand>
      <Operation>GetNamespaces</Operation>
      <Gslid></Gslid>
   </v1:AdministrativeCommand>
</soapenv:Body>

<SOAP-ENV:Body>
   <ns3:AdministrativeCommandResponse>
      <AdministrativeQueryItem>
         <Item>analytics</Item>
      </AdministrativeQueryItem>
      <AdministrativeQueryItem>
         <Item>mmf</Item>
      </AdministrativeQueryItem>
      <AdministrativeQueryItem>
         <Item>Dallas</Item>
      </AdministrativeQueryItem>
      <AdministrativeQueryItem>
         <Item>LongIsland</Item>
      </AdministrativeQueryItem>
      <AdministrativeQueryItem>
         <Item>Belleville</Item>
      </AdministrativeQueryItem>
      <AdministrativeQueryItem>
         <Item>Avaya</Item>
      </AdministrativeQueryItem>
   </ns3:AdministrativeCommandResponse>
</SOAP-ENV:Body>
```

CreateContentGroup Example

After you create a Namespace, for example, `Avaya`, the next step is to create a ContentGroup, for example, `Music`, so that some content can be ingested.

```
   <soapenv:Body>
      <v1:AdministrativeCommand>
         <NameSpace>Avaya</NameSpace>
         <ContentGroup>Music</ContentGroup>
         <Operation>CreateContentGroup</Operation>
      </v1:AdministrativeCommand>
   </soapenv:Body>

   <SOAP-ENV:Body>
      <ns3:AdministrativeCommandResponse/>
   </SOAP-ENV:Body>
```

CopyContentGroup Example

The AdministrativeCommand CopyContentGroup Operation copies all the content from one Content Group to another. The following example shows an `Avaya` Namespace Content Group named Music duplicated to the Avaya Namespace Content Group named `Music/Rock`.

The storage structure prior to the copy is:

```
Avaya
      Music
         file1
         file2
         file3
         file4
```

After the copy, the storage looks like this:

```
Avaya
      Music
         file1
         file2
         file3
         file4
    Music/Rock
         file1
         file2
         file3
         file4
```

An example request and response is shown as follows:

```
<soapenv:Body>
      <v1:AdministrativeCommand>
         <Operation>CopyContentGroup</Operation>
         <NameSpace>Avaya</NameSpace>
         <ContentGroup>Music</ContentGroup>
         <CopyToNamespace>Avaya</CopyToNamespace>
         <CopyToContentGroup>Music/Rock</CopyToContentGroup>
       </v1:AdministrativeCommand>
   </soapenv:Body>

   <SOAP-ENV:Body>
      <ns3:AdministrativeCommandResponse/>
   </SOAP-ENV:Body>
```

QueryNamespaceSize Example

```
   <soapenv:Body>
      <v1:AdministrativeCommand>
```

```
        <NameSpace>Avaya</NameSpace>
        <Operation>QueryNamespaceSize</Operation>
     </v1:AdministrativeCommand>
  </soapenv:Body>

  <SOAP-ENV:Body>
     <ns3:AdministrativeCommandResponse>
        <Result>152620</Result>
     </ns3:AdministrativeCommandResponse>
  </SOAP-ENV:Body>
```

IngestContent

You can add content to Content Store by using the IngestContent service. The IngestContent request specifies the location of the content that you want to process by using a URL reference to the source file. The supported URL types are `http://`, `https://`, `rtft://` and `file://`. Content Store pulls the file from the specified location and stores the file on Avaya Aura® MS. Alternatively, content can be posted to Content Store by using a URL to provide source file access when a web server is unavailable.

In addition to the normal ingest mode, IngestContent also supports two other ingest modes, pre-ingest and re-ingest. Using the pre-ingest mode, you can reserve a Content ID that can be used for a future content ingest. The re-ingest mode updates existing content by overwriting the current content of an existing Content ID in Content Group.

The `post` IngestType requires using pre-ingest followed by an HTTP post request. Pre-ingest specifies an existing Namespace and Content Group and a unique Content ID. The system reserves the content name and returns an ingest token in the Ingest response. The token must be specified in the HTTP Post request so that the system can complete the ingest operation. The HTTP Post must use the following format:

`http://mediaserverIP:7410/post?ingesttoken=`*`tokenUUID`*

Clients can provide an explicit Content ID when the content is ingested, or the system can assign a generated Content ID to the content. Additional optional parameters can also be included in the ingest request to set metadata associated with the content.

Your application can ingest XML content into Content Store using a URL reference to a file. Another option is by specifying the XML content in the XMLDocument parameter of the IngestContent request.

Note:

Avaya recommends that audio played by Avaya Aura® MS be encoded as 16-bit, 8 kHz, single channel, PCM files. Codecs other than PCM or using higher sampling rates for higher quality recordings can also be used, however, with reduced system performance. Multiple channels, like stereo, are not supported.

IngestContent Request		
Parameter	**Optionality**	**Description**
Namespace	M	Namespace containing the Content Groups to receive the file.
ContentGroup	M	The Content Groups to receive the file specified by the URL parameter.
ContentId	C	The name of the content. This parameter is not included for initial Ingest or Pre-Ingest if content IDs are to be assigned by the Content Store. Required after initial Ingest or Pre-Ingest when ingest is updating content.
IngestType	O	The type of ingest to perform. Valid values are *post* or an empty string. This field is omitted or left empty for normal ingest.
OpaqueFrom	O	Metadata field. Any string can be stored here.
OpaqueType	O	Metadata field. Any integer can be stored here.
OpaqueGroup	O	Metadata field. Any string can be stored here.
OpaqueData	O	Metadata field. Any string can be stored here.
OpaqueStatus	O	Metadata field. Any integer can be stored here.
Priority	O	Metadata field. Any integer can be stored here.
MetaDataMode	O	Client metadata on the Content Store can be controlled by setting the mode. On re-ingests updating existing content, clients might choose to: Purge all existing metadata, resulting in use of metadata included in the re-ingest exclusively. Preserve existing metadata, but overwrite old metadata if new values are provided in the re-ingest for a particular field. The default is to Preserve.
URL	C	URL of the file to be ingested as new content. Valid URLs are `http://`, `https://`, `file://`, `rtft://`. Not included for Pre-Ingest modes because the URL of the file that you want to ingest comes in a subsequent ingest message. This parameter might not be included if you include XMLDocument.
ExpirationTime	O	A future time indicating when this content will be automatically marked deleted by the Content Store. GMT expiration time must be formatted as yyyy-mm-dd hh:mm:ss

Table continues…

IngestContent Request		
Parameter	**Optionality**	**Description**
XMLDocument	C	Entire body of the XML document to be stored on the Content Store. Alternatively, the XML might be specified in the URL parameter. This parameter might not be included if the URL is included.
Gslid	O	Global Session Log Identifier for traceability of the transaction.

The response contains the Content ID assigned to the ingested data. The IngestContent response also contains URLs to the content. However, these URLs are typically the type used by components local to Avaya Aura® MS. Your client applications must use the FetchContent service to retrieve the content if the applications are not running locally on Avaya Aura® MS.

IngestContent Response		
Parameter	**Optionality**	**Description**
Content ID	M	The Content ID assigned to the ingested file.
File URL	M	The file:// URL reference to the content. Useful if the client application is also local on the Avaya Aura® MS.
RTFT URL	M	The rtft:// URL reference to the content. Useful if the client application is not local to the Avaya Aura® MS.
IngestToken	O	The unique token associated with a post IngestType request.

IngestContent Example: Initial File Ingest

The following example shows an initial file ingest with a Content ID specified by the application. The file `jump.wav` is ingested with the Content ID of `Jump` into the Content Group `Music/Rock`, which is in the `Avaya` Namespace.

```
<soapenv:Body>
   <v1:IngestContent>
       <NameSpace>Avaya</NameSpace>
      <ContentGroup>Music/Rock</ContentGroup>
      <ContentId>Jump</ContentId>
      <Url>http://135.102.229.10/jump.wav</Url>
      <OpaqueFrom>MusicStore</OpaqueFrom>
      <OpaqueGroup>New Releases</OpaqueGroup>
   </v1:IngestContent>
</soapenv:Body>

<SOAP-ENV:Body>
   <ns3:IngestContentResponse>
       <ContentId>Jump</ContentId>
      <FILEUrl>file:///D:/Program Files/Avaya/Multimedia_Applications/
               AMS/platdata/CStore/StorageRoot/n-1227146700-1/
               c-1227203568-253694187-9.wav</FILEUrl>
      <RTFTUrl>rtft://47.104.19.62:52007/root0/n-1227146700-1/
               c-1227203568-253694187-9.wav</RTFTUrl>
```

```
      </ns3:IngestContentResponse>
   </SOAP-ENV:Body>
```

IngestContent Example: File Re-Ingest

After Content Store ingests a file, the file can be re-ingested to update the stored file or metadata. When an application specifies `Preserve` for the MetaDataMode in the Ingest Request, the new metadata merges with the existing metadata. In the following example, a new version of the file `jump.wav` is specified for the re-ingest of the content. Notice that the file name that Content Store returns in the URLs is new, but the Content ID used to reference the file is still the same.

As a result of the Content ID staying the same, any application reference to `Avaya/Music/Rock/Jump` using the Content Store Web Services APIs automatically references the new, updated file, and metadata. The updated content has a new file name on disk and the old version of the file is deleted the next time the audit runs.

```
   <soapenv:Body>
      <v1:IngestContent>
          <NameSpace>Avaya</NameSpace>
         <ContentGroup>Music/Rock</ContentGroup>
         <ContentId>Jump</ContentId>
         <MetaDataMode>Preserve</MetaDataMode>
         <Url>http://135.102.229.10/jump.wav</Url>
         <OpaqueFrom>MusicStore</OpaqueFrom>
         <OpaqueGroup>Old Songs</OpaqueGroup>
      </v1:IngestContent>
   </soapenv:Body>

   <SOAP-ENV:Body>
      <ns3:IngestContentResponse>
          <ContentId>Jump</ContentId>
         <FILEUrl>file://D:/Program Files/Avaya/Multimedia_Applications/
                  AMS/platdata/CStore/StorageRoot/n-1227146700-1/
                  c-1227203810-253936984-10.wav</FILEUrl>
         <RTFTUrl>rtft://47.104.19.62:52007/root0/n-1227146700-1/
                  c-1227203810-253936984-10.wav</RTFTUrl>
      </ns3:IngestContentResponse>
   </SOAP-ENV:Body>
```

IngestContent Example: Initial ingest using XMLDocument and no ContentID

This example shows ingest of XML content as specified in the XMLDocument parameter. No ContentID was specified in the IngestContent request. The system generates a ContentID and reports the ID back to the application in the response. The system uses the data in the XMLDocument parameter and creates a file with the XML content. Future queries to access this content must refer to:

`Avaya/Music/Rock/c-1227204303-254429718-11.`

```
   <soapenv:Body>
      <v1:IngestContent>
          <NameSpace>Avaya</NameSpace>
         <ContentGroup>Music/Rock</ContentGroup>
         <OpaqueFrom>MusicStore</OpaqueFrom>
         <OpaqueGroup>Play list</OpaqueGroup>
         <XMLDocument><![CDATA[<?xml version="1.0"?><song>Jump</song> <artist>VanHalen</
artist>]]>
        </XMLDocument>
      </v1:IngestContent>
   </soapenv:Body>

   <SOAP-ENV:Body>
```

```
      <ns3:IngestContentResponse>
          <ContentId>c-1227204303-254429718-11</ContentId>
         <FILEUrl>file:///D:/Program Files/Avaya/Multimedia_Applications/AMS/
                  platdata/CStore/StorageRoot/n-1227146700-1/
                  c-1227204303-254429718-11.xml</FILEUrl>
         <RTFTUrl>rtft://47.104.19.62:52007/root0/n-1227146700-1/
                  c-1227204303-254429718-11.xml</RTFTUrl>
      </ns3:IngestContentResponse>
   </SOAP-ENV:Body>
```

IngestContent Example: Ingest using HTTP post

The following is an example of using the IngestType of post request. The request indicates that this post is a pre-ingest of the Content ID `song.wav`. The pre-ingest response contains the IngestToken required for the HTTP Post request. The HTTP Post follows the pre-ingest response and provides the file that must be ingested.

```
   <soapenv:Body>
      <v1:IngestContent>
          <NameSpace>Avaya</NameSpace>
         <ContentGroup>Music</ContentGroup>
         <ContentID>song.wav</ContentID>
         <IngestType>post</IngestType>
         </v1:IngestContent>
   </soapenv:Body>

   <SOAP-ENV:Body>
      <ns3:IngestContentResponse>
         <ContentID>song.wav</ContentID>
           <IngestToken>f8cc0bd-0805-3305-af52-3823288fcc11</IngestToken>
         </ns3:IngestContentResponse>
   </SOAP-ENV:Body>
```

The following example shows the HTTP Post request URL that includes the IngestToken received from the IngestContentResponse. This HTTP Post pushes the file to Avaya Aura® MS to complete the ingest.

```
POST http://10.60.77.158:7410/post?ingesttoken=f8cc0bd-0805-3305-af52-3823288fcc11 HTTP/
1.1
content-type: audio/x-wav
Transfer-Encoding: chunked
Host: 135.60.77.158:7410
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.7)


HTTP/1.1 200 OK
Server: gSOAP/2.8
Content-Type: audio/x-wav
Transfer-Encoding: chunked
Connection: keep-alive
```

Note:

Pre-ingest and HTTP Post must meet the following requirements:

- The Namespace and Content Group specified in the pre-ingest request must already exist on Avaya Aura® MS.
- The Content ID specified in the pre-ingest request must be unique within the specified Namespace and Content Group.

- Avaya Aura® MS supports chunked and non-chunked data transfer modes for HTTP Post.
- Multiple files or attachments are not supported in a single HTTP Post request.

FetchContent

The FetchContent command retrieves previously ingested file content from the Content Store.

FetchContent Request		
Parameter	**Optionality**	**Description**
Namespace	M	Namespace containing the Content Group.
ContentGroup	M	Content Group containing the content.
ContentId	M	Content ID of the data to Fetch.
Gslid	O	Global Session Log Identifier for traceability of the transaction.

The FetchContent response includes content from the ingested file in the Base64Content parameter when content is less than 1024 bytes. Content Store includes the HTTPUrl parameter for access to content when the ingested content is greater than 1024 bytes.

FetchContent Response		
Parameter	**Optionality**	**Description**
ContentId	M	The Content ID of the data Fetched.
SizeBytes	M	The number of bytes in the data.
MimeType	O	The type of content.
HTTPUrl	C	The HTTP URL to the content when content is more than 1024 bytes.
Base64Content	C	The BASE64 encoded data for content less than 1024 bytes.

FetchContent Example

The following example retrieves the XML data previously stored by an application in Content Store as `Avaya/Music/Rock/SongInfo`:

```
<soapenv:Body>
   <v1:FetchContent>
       <NameSpace>Avaya</NameSpace>
      <ContentGroup>Music/Rock</ContentGroup>
      <ContentId>SongInfo</ContentId>
   </v1:FetchContent>
</soapenv:Body>

<SOAP-ENV:Body>
   <ns3:FetchContentResponse>
       <ContentId>SongInfo</ContentId>
      <SizeBytes>51</SizeBytes>
      <MimeType>text/xml</MimeType>
      <Base64Content>PHhtbD48c29uZz5KdW1wPC9zb25nPj
```

```
                xzb25nPlBob3RvZ3JhcGg8L3Nvbmc+PC94bWw+</Base64Content>
     </ns3:FetchContentResponse>
  </SOAP-ENV:Body>
```

Decoding the Base64Content in the response yields the original XML:

```
<?xml version="1.0"?><song>Jump</song><artist>VanHalen</artist>
```

SearchContent

Using the SearchContent command, you can query the Content Store for data matching the specified criteria within a single Content Group. The response contains a list of Content IDs that match the criteria. Also, an optional piece of metadata of your choice is associated with the matching content.

SearchContent Request		
Parameter	**Optionality**	**Description**
Namespace	M	Namespace containing the Content Group to search.
ContentGroup	M	Content Group to search.
SortKey	O	Allows the user to specify an attribute of the content as the sort key for result set. Valid values are: Group, Type, From, UserData, Status, Deleted, Priority, Size, TimeRcvd, ExpTime, LastAccTime
ResultOrder	O	Determines the order of the results returned. Valid values are: Descending or Ascending. Ascending is the default.
OpaqueFrom	O	Metadata match string. Including this field means the response only includes content which has identical values for this field and any other fields specified in this request.
OpaqueType	O	Metadata match integer. Including this field means the response only includes content which has identical values for this field and any other fields specified in this request.
OpaqueGroup	O	Metadata match string. Including this field means the response only includes content which has identical values for this field and any other fields specified in this request.
OpaqueData	O	Metadata match string. Including this field means the response only includes content which has identical values for this field and any other fields specified in this request.
OpaqueStatus	O	Metadata match integer. Including this field means the response only includes content

Table continues…

SearchContent Request		
Parameter	**Optionality**	**Description**
		which has identical values for this field and any other fields specified in this request..
Deleted	O	Deleted match criteria. A value of 1 indicates deleted. 0 indicates not marked deleted. Including this field means the response only includes content which has identical values for this field and any other fields specified in this request.
Priority	O	Metadata match integer. Including this field means the response only includes content which has identical values for this field and any other fields specified in this request.
MimeType	O	File type to match. Including this field means the response only includes content which has identical values for this field and any other fields specified in this request.
TimeReceived	O	Time to match. GMT time formatted as yyyy-mm-dd hh:mm:ss (Example: 2006-09-01 17:20:03) reflecting the Time Received at the Content Store. Including this field means the response only includes content which has identical values for this field *and* any other fields specified in this request.
LastAccessTime	O	Time when this content was last modified. GMT time formatted as yyyy-mm-dd hh:mm:ss (Example: 2006-09-01 17:20:03). Including this field means the response only includes content which has identical values for this field and any other fields specified in this request.
ExpirationTime	O	Time when this content will be automatically marked deleted by the Content Store. GMT time formatted as yyyy-mm-dd hh:mm:ss. For example, 2006-09-01 17:20:03. Including this field means the response only includes content which has identical values for this field *and* any other fields specified in this request.
SizeBytes	O	File size in bytes. Including this field means the response only includes content which has identical values for this field and any other fields specified in this request.
AssocMetaData	O	Choice of associated metadata to return with each match. Valid values are:

Table continues…

SearchContent Request		
Parameter	**Optionality**	**Description**
		Group, Type, From, UserData, Status, Deleted, Priority, Size, TimeRcvd, ExpTime, LastAccTime
Gslid	O	Global Session Log Identifier for traceability of the transaction.

Content Store reports the results of the query in the Search Content response. Each matching item is listed as a ContentItem element in the result list.

SearchContent Response		
Parameter	**Optionality**	**Description**
ContentItem	C	List of Content Items, each containing a Content ID as well as any optionally requested metadata as specified by the AssocMetaData in the request.

SearchContent Example

In the following example, the search requests a list of all content IDs and associated file sizes in bytes, that have the From metadata value of `MusicStore` in the Content Group `Music/Rock`.

Two matching items are returned. Each item contains the ContentID of the match and the requested metadata, in this case, the size of the file. The results are listed in ascending order by size.

```
<soapenv:Body>
  <v1:SearchContent>
      <SortKey>Size</SortKey>
      <NameSpace>Avaya</NameSpace>
      <ContentGroup>Music/Rock</ContentGroup>
      <OpaqueFrom>MusicStore</OpaqueFrom>
      <AssocMetaData>Size</AssocMetaData>
   </v1:SearchContent>
</soapenv:Body>

<SOAP-ENV:Body>
   <ns3:SearchContentResponse>
      <ContentItem>
         <ContentId>c-1227204303-254429718-11</ContentId>
         <MetaData>51</MetaData>
      </ContentItem>
      <ContentItem>
         <ContentId>Jump</ContentId>
         <MetaData>234</MetaData>
      </ContentItem>
   </ns3:SearchContentResponse>
</SOAP-ENV:Body>
```

ModifyXMLContent

Using the ModifyXMLContent command, an application can modify or delete elements in the XML content.

The XMLTagValues parameter for an Update operation are coded as:

`<![CDATA[<PIN>1234</PIN>]]>` while for a **Delete** it is coded as `<![CDATA[<PIN>]]>`

Multiple tag and value pairs can be included in the same modify request.

ModifyXMLContent Request		
Parameter	**Optionality**	**Description**
Namespace	M	Namespace containing the Content Group to modify.
ContentGroup	M	Content Group containing the content to modify.
ContentId	M	Content ID to modify.
Operation	M	Operation to be performed. The available commands are Update and Delete. Update is also used to add new XML tag/value pairs content.
XMLTagValues	M	XML tag/value pairs to be used for update operation.
Gslid	O	Global Session Log Identifier for traceability of the transaction.

ModifyXMLContent Response		
Parameter	**Optionality**	**Description**
No parameters	—	—

ModifyXMLContent Example

The following example updates only the song and artist values in the existing XML with new values:

```
   <soapenv:Body>
      <v1:ModifyXMLContent>
         <NameSpace>Avaya</NameSpace>
         <ContentGroup>Music/Rock</ContentGroup>
         <ContentId>Favorite</ContentId>
         <Operation>Update</Operation>
         <XMLTagValues><![CDATA[<song>Refugee</song><artist>Tom Petty</artist>]]></
XMLTagValues>
      </v1:ModifyXMLContent>
   </soapenv:Body>

   <SOAP-ENV:Body>
      <ns3:ModifyXMLContentResponse/>
   </SOAP-ENV:Body>
```

QueryXMLContent

Using the QueryXMLContent command, you can retrieve XML content as follows:

QueryXMLContent Request		
Parameter	**Optionality**	**Description**
Namespace	M	Namespace containing the Content Group to query.
ContentGroup	M	Content Group containing the content to query.
ContentId	M	Content Id to query.
Gslid	O	Global Session Log Identifier for traceability of the transaction.

Content Store returns the entire XML file content in the QueryXMLContent response.

QueryXMLContent Response		
Parameter	**Optionality**	**Description**
QueryXMLContentResponse	M	XML document matching request

QueryXMLContent Example

```
   <soapenv:Body>
      <v1:QueryXMLContent>
         <NameSpace>meetme@avaya.com</NameSpace>
         <ContentGroup>4441234</ContentGroup>
         <ContentId>SubscriberData</ContentId>
      </v1:QueryXMLContent>
   </soapenv:Body>

      <SOAP-ENV:Body>
         <ns3:QueryXMLContentResponse>
                  <XMLDocument><![CDATA[<?xml version="1.0"?>
<ACCESS_CODE>4444/ACCESS_CODE>
<SIP_USER_ID>gs444@avaya.com</SIP_USER_ID>
<USER_STATE>ACTIVE</USER_STATE>
<CHAIR_PIN>4444</CHAIR_PIN>
<CHAIR_ENDS>N</CHAIR_ENDS>
<IM_ENABLED>Y</IM_ENABLED>
<FAST_START>N</FAST_START>
<PUBLIC_PIN>8646</PUBLIC_PIN>
<ENTRY_EXIT_INDICATION>tones</ENTRY_EXIT_INDICATION>
<MAX_BRIDGE_SIZE>10</MAX_BRIDGE_SIZE>
<USER_LOCALE>en_US</USER_LOCALE>
<DOMAIN>avaya.com</DOMAIN>
<VERSION>3.2</VERSION>
<AUDIO_EMOTICONS_ENABLED>Y</AUDIO_EMOTICONS_ENABLED>
<PREMIUM_CONF_ENABLED>N</PREMIUM_CONF_ENABLED>
<VIDEO_CONF_ENABLED>N</VIDEO_CONF_ENABLED>
<WEB_COLLAB_ENABLED>N</WEB_COLLAB_ENABLED>
<USER_EMAIL_ADDR></USER_EMAIL_ADDR>
<USER_TIMEZONE>CST</USER_TIMEZONE>
<AUDIO_REC_ENABLED>N</AUDIO_REC_ENABLED>
<AUDIO_REC_EMAIL_ADDR></AUDIO_REC_EMAIL_ADDR>
]]></XMLDocument>
      </ns3:QueryXMLContentResponse>
   </SOAP-ENV:Body>
```

SearchXMLContent

Using the SearchXMLContent command, an application can search XML formatted strings stored in the OpaqueGroup metadata, as opposed to XML content stored in files.

Note:

The SearchXMLContent operation works on a different XML data space than the other XML operations.

Content Store returns a response with a list of Content IDs which contain the search pattern within the specified search scope.

For example, you may use IngestContent or ModifyContentMetaData to put XML data entries in the OpaqueGroup metadata field for many users, which looks like this:

```
<MYAPP>
    <SIP_USER_ID>user1@avaya.com</SIP_USER_ID>
    <PIN>1234</PIN>
</MYAPP>
```

You can then use the SearchXMLContent service to search the XPath of `/MYAPP/PIN` in the OpaqueGroup fields for a match on the Pattern `1234`.

The scope of the search is controlled by an application including or excluding the optional ContentGroup and Content ID parameters:

- If only the Namespace parameter is included in the request then Content Store searches the OpaqueGroup of each content item over the entire namespace.
- If Namespace and ContentGroup are included in the request then Content Store searches the OpaqueGroup of each content item in the specified ContentGroup.
- If Namespace, Content Group, and Content ID are all specified then Content Store searches only that specific content item.

SearchXMLContent Request		
Parameter	**Optionality**	**Description**
Namespace	M	Specifies the Namespace containing the Content Group to query.
ContentGroup	O	Specifies the Content Group containing the content to query.
ContentId	O	Specifies the Content ID to query
Algorithm	O	Specifies the algorithm to use in search. Currently, the only algorithm supported is “substr”. If this parm is not supplied, “substr” is the default.
Pattern	M	The search pattern is any simple text string. Regular expressions are not supported. The

Table continues…

SearchXMLContent Request		
Parameter	**Optionality**	**Description**
		search is only performed in the OpaqueGroup metadata under the specified Namespace, ContentGroup and ContentID scope, and not in XML files.
XPath	M	Specifies the XML path to start in for the search.
LastResult	O	Specifies the content ID of the last result.
Gslid	O	Global Session Log Identifier for traceability of the transaction.

The following table describes the SearchXMLContent response:

SearchXMLContent Response		
Parameter	**Optionality**	**Description**
Complete	M	True if the search result set is complete with this response.
NumberReturned	M	The number of results found which are included in the response.
MatchList	C	List of all the content containing matches. The list is a set of element couples indicating the Content Group and Content ID containing the match. This allows the client to reference the content of the matches found. This parameter is not included if no matches are found. A result list will have braced couples with each member of the couple separated by a pipe delimiter: {cntgrp1|cntid1} {cntgrp2|cntid2} ...

SearchXMLContent Example

The SearchXMLContent example below shows a search for a `PIN` with value `1234` over all the ContentGroups in the Avaya Namespace.

The search results in two matches, as shown below:

```
<soapenv:Body>
   <v1:SearchXMLContent>
      <NameSpace>Avaya</NameSpace>
      <Pattern>1234</Pattern>
      <XPath>/MYAPP/PIN</XPath>
   </v1:SearchXMLContent>
</soapenv:Body>

<SOAP-ENV:Body>
   <ns3:XMLSearchResponse>
      <Complete>true</Complete>
      <NumberReturned>2</NumberReturned>
      <MatchList>{Music|Owner}{Video|Owner}</MatchList>
   </ns3:XMLSearchResponse>
</SOAP-ENV:Body>
```

ModifyContentMetaData

Using the ModifyContentMetaData command, your application can modify metadata for a particular content item. ModifyContentMetaData modifies the same metadata that has been optionally set by the IngestContent request.

ModifyContentMetaData Request		
Parameter	**Optionality**	**Description**
Namespace	M	Namespace containing the Content Group to modify.
ContentGroup	M	Content Group containing the content to modify.
ContentId	M	Content ID to query.
OpaqueFrom	O	Metadata field. Any string may be stored here.
OpaqueType	O	Metadata field. Any integer may be stored here.
OpaqueGroup	O	Metadata field. Any string may be stored here.
OpaqueData	O	Metadata field. Any string may be stored here.
OpaqueStatus	O	Metadata field. Any integer may be stored here.
Deleted*	O	“1” to mark content as deleted and then deleted by configurable audit, “2” to delete immediately
Priority	O	Metadata field. Any string may be stored here.
ExpirationTime	O	A future time indicating when this content will be automatically marked deleted by the Content Store. GMT expiration time must be formatted as yyyy-mm-dd hh:mm:ss
Gslid	O	Global Session Log Identifier for traceability of the transaction.

ModifyContentMetaData Response		
Parameter	**Optionality**	**Description**
No parameters	—	—

ModifyContentMetaData Example

The ModifyContentMetaData example shows how your application can update the **OpaqueData** and **OpaqueStatus** fields of `Avaya/Music/Rock/Favorite`.

```
<soapenv:Body>
   <v1:ModifyContentMetaData>
      <NameSpace>Avaya</NameSpace>
      <ContentGroup>Music/Rock</ContentGroup>
      <ContentId>Favorite</ContentId>
      <OpaqueData>This is any data string</OpaqueData>
      <OpaqueStatus>55</OpaqueStatus>
   </v1:ModifyContentMetaData>
</soapenv:Body>

<SOAP-ENV:Body>
```

```
      <ns3:ModifyContentMetaDataResponse/>
   </SOAP-ENV:Body>
```

QueryContentMetaData

Using the QueryContentMetaData command, you can query the metadata for a specific content item, including the URLs. The URLs will allow you to gain access to the file content.

QueryContentMetaData Request		
Parameter	**Optionality**	**Description**
Namespace	M	Namespace containing the Content Group to query.
ContentGroup	M	Content Group containing the content to query.
ContentId	M	Content Id to query.
Gslid	O	Global Session Log Identifier for traceability of the transaction.

QueryContentMetaData Response		
Parameter	**Optionality**	**Description**
ContentId	M	ContentId of the response.
SizeBytes	M	Number of bytes in the data.
Version	M	Version. Each time a file is re-ingested the version is incremented.
Deleted	M	A value of 1 indicates the files is marked as deleted.
MimeType	O	Type of resource.
TimeReceived	M	GMT time the content was initially received, formatted as yyyy-mm-dd hh:mm:ss
LastAccessTime	M	GMT time the content was last modified, formatted as yyyy-mm-dd hh:mm:ss
FileUrl	O	File URL to content file.
RTFTUrl	O	RTFT URL to content file.
Priority	O	Opaque field.
OpaqueFrom	O	Opaque field.
OpaqueGroup	O	Opaque field.
OpaqueType	O	Opaque field.
OpaqueStatus	O	Opaque field.
OpaqueData	O	Opaque field.
ExpirationTime	O	GMT expiration time, formatted as yyyy-mm-dd hh:mm:ss

QueryContentMetaData Example

Below is an example of a QueryContentMetaData request and response:

```
   <soapenv:Body>
      <v1:QueryContentMetaData>
         <NameSpace>Avaya</NameSpace>
         <ContentGroup>Music/Rock</ContentGroup>
         <ContentId>Favorite</ContentId>
      </v1:QueryContentMetaData>
   </soapenv:Body>

   <soapenv:Body>
     <ns3:QueryContentMetaDataResponse>
         <ContentId>Favorite</ContentId>
         <SizeBytes>49</SizeBytes>
         <Version>8</Version>
         <Deleted>0</Deleted>
         <MimeType>text/xml</MimeType>
         <TimeReceived>2008-11-20 20:39:38</TimeReceived>
         <LastAccessTime>2008-12-09 20:29:55</LastAccessTime>
         <FILEUrl>file:///D:/Program Files/Avaya/Multimedia_Applications/AMS/platdata/
CStore/StorageRoot/n-1228246548-2/c-1228505200-172813044-161.xml</FILEUrl>
         <RTFTUrl>rtft://47.104.19.62:52007/root0/n-1228246548-2/
c-1228505200-172813044-161.xml</RTFTUrl>
         <OpaqueFrom>MusicStore</OpaqueFrom>
         <OpaqueGroup>Play list</OpaqueGroup>
         <OpaqueStatus>55</OpaqueStatus>
         <OpaqueData>This is any data string</OpaqueData>
      </ns3:QueryContentMetaDataResponse>
   </SOAP-ENV:Body>
```

DeliverContent

Using the DeliverContent command, your application can create and deliver an e-mail to a valid e-mail user. Use the various fields in the DeliverContent request to address an e-mail message, specify the e-mail format, and to attach the stored content file to the e-mail.

DeliverContent Request		
Parameter	**Optionality**	**Description**
Namespace	M	Namespace containing the Content Group which has the content to be delivered.
ContentGroup	M	Content Group containing the content to be delivered.
ContentId	M	ID of the content to be attached and delivered.
DeliveryMethod	M	The method used to deliver the content. Valid values are: smtp
DeliveryPreInstructions	O	Operation to be performed on the content prior to delivery. Valid values are: {none, TranscodeToGSM,

Table continues…

DeliverContent Request		
Parameter	**Optionality**	**Description**
		TranscodeToIMAADPCM, TranscodeToWM, TranscodeToMSADPCM, TranscodeToWM8, TranscodeToWM8_, TranscodeToWM9, TranscodeToWM9_, TranscodeToG723, TranscodeToG729 }
MailPriority	O	Priority of mail to send {0-5}
SmtpServer	M	Mail Server hostname (target server of delivery)
MailUserName	O	A valid user for the SMTP Server used for server access, for example, cstoreadmin@avaya.com
MailReplyTo	O	E-mail reply address
MailAttachmentUrl	O	Mail attachment filename or URL as it exists on the network or local disk
MailAttachmentFilename	O	Mail attachment filename encoded with the email
MailAttachmentContentType	O	Mail attachment content type
MailLocale	O	Locale {en_us, etc.}
MailTextmode	O	Rendering the format of the email body (default = html) {plain, html}
MailTemplateDomain	O	Template domain
MailBody	O	The actual body of the email
MailBodyTemplate	O	Template name or load for the email body
MailBodyTemplateParm0 – MailbodyTemplateParm8	O	Specify templates
MailToAddress	M	The full “To” mail address: you@avaya.com
MailFromAddress	M	The full “From” mail address: me@avaya.com
MailSubject	M	Subject line of the email
MailFromDisplay	O	The user-friendly display name: “Steve Mobs” steve@avaya.com
Gslid	O	Global Session Log Identifier for traceability of the transaction.

DeliverContent Response		
Parameter	**Optionality**	**Description**
No parameters	—	—

DeliverContent Example

The following example builds an e-mail with the subject of `Your Message` and the body text of `Hi team, Please see the attached file`. The file `msg.wav` is attached to the email and then it is sent to user@avaya.com:

```
<soapenv:Body>
   <v1:DeliverContent>
      <NameSpace>Avaya</NameSpace>
      <ContentGroup>Music/Rock</ContentGroup>
      <ContentId>Song</ContentId>
      <DeliveryMethod>smtp</DeliveryMethod>
      <DeliveryPreInstructions>TranscodeToWM</DeliveryPreInstructions>
      <SmtpServer>135.1.2.3</SmtpServer>
      <MailReplyTo>name@avaya.com</MailReplyTo>
      <MailToAddress>user@avaya.com</MailToAddress>
      <MailFromAddress>service@avaya.com</MailFromAddress>
      <MailSubject>Your Message</MailSubject>
      <MailAttachmentFilename>msg.wav</MailAttachmentFilename>
      <MailBody>Hi team. Please see the attached file.</MailBody>
   </v1:DeliverContent>
</soapenv:Body>

<SOAP-ENV:Body>
   <ns3:DeliverContentResponse/>
</SOAP-ENV:Body>
```

Index