Avaya Aura® Contact Center / Avaya Contact Center Select Real-Time Statistic Multicast Programmer's Guide

Release 7.1.1

Issue 0.1

October 2020

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.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYAWEBSITE, 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. “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.

License type(s)

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.

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.

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 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.

Note to Service Provider

The Product or Hosted Service may use Third Party Components subject to Third Party Terms that do not allow hosting and require a Service Provider to be independently licensed for such purpose. It is your responsibility to obtain such licensing.

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.

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 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.

All non-Avaya trademarks are the property of their respective owners, and "Linux" is a registered trademark of Linus Torvalds.

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.

Contents

Chapter 1: Introduction 6

Purpose 6
Intended audience 6
Support 6

Chapter 2: Overview 7

Introduction 7
Assumptions 7

Chapter 3: Obtaining the Real-Time Statistic Multicast Software 8

Introduction 8
Obtaining the Real-Time Statistic Multicast Software Development Kit 8
To install the RSM SDK 8
To uninstall the RSM SDK 9
Installing the SDK on non-WIN32 platforms 9
ZLIB data compression library 9
AACC/ACCS Data Compression 9

Chapter 4: RSM interfaces 11

Introduction 11
Communication between client applications and the RSM server 11
AACC/ACCS RSM CORBA uses mutual TLS to authenticate the identity of CORBA client applications. AACC/ACCS RSM CORBA is configured to use TLS v1.2.Control of access to the RSM data 12
Creating a client application 12
A typical application scenario 12

Chapter 5: Interface Definition 14

Introduction 14
Network data format 14
RTD Statistic Multicast Interface 15
CORBA development environment 19

Chapter 6: How to locate the RSM Service 21

Naming Service 21
To locate the CORBA Naming Service 21

Chapter 7: Statistics 23

Real-time data 23
Skillset statistics 23
Application statistics 25
Agent statistics 28
Nodal statistics 29
IVR statistics 30
Route statistics 31
Splitting of data across multiple packets 31
Checking for data compression 31
Data stream definition 31

Chapter 8: RSM SDK Sample Applications 34

Introduction 34
IP Multicast Sample Application 34
CORBA Sample Application 34

Chapter 9: Building a CORBA application using TAO 37

Introduction to CORBA 37
Introduction to TAO 37
Introduction to TAO Security 37
Basic ORB operation and communication 38
Basic CORBA client operation 38
Obtaining TAO 38
Using the TAO IDL compiler to generate source code from the IDL 38
IOR (Interoperable Object Reference) 39
Naming Service 39
Reference persistence 40
TAO utilities 40
Properties file 40
Client-side settings for TAO 41

Chapter 1: Introduction

Purpose

This document provides information about the Avaya Aura Contact Center (AACC) / Avaya Contact Center Select (ACCS) Real-Time Statistic Multicast (RSM) interface.

Intended audience

This document is intended for people who want to use the Real-Time Statistic Multicast (RSM) interface.

Support

Visit 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, or request an agent to connect you to a support team if an issue requires additional expertise.

Chapter 2: Overview

The Real-Time Statistic Multicast (RSM) interface uses IP multicast technology to provide basic call center status reporting capability to third-party application developers. It provides a data interface between third-party applications and the component in AACC/ACCS responsible for collecting and maintaining real-time display statistics. The third-party application can obtain real-time statistics from AACC/ACCS for use in basic call center status reporting applications, such as reader-boards and agent desktop applications.

The RSM interface is based on the Real-time Data Application Programming Interface (RTD API). The RTD API is used to create Windows-based applications that manipulate and display real-time statistics from AACC/ACCS. However, the RTD API only supports third-party applications on Microsoft Windows operating systems. For non-Windows operating systems, only RSM provides access to the statistical data.

In addition to the IP multicast data stream, the RSM interface provides a CORBA-defined interface for controlling the RSM data stream on the server and performing bidirectional data translations from ID to name.

Introduction

This document includes the following elements:

- definition and description of the statistics data available over the multicast streams
- definition of the Interface Definition Language (IDL) for the CORBA interface to RSM including a full description of the interface methods and parameters
- application development methodology including the definition of the communications protocol between the client and server
- sample code illustrating receiving data from the multicast streams and translating skillset name to ID using the CORBA interface

Assumptions

This document assumes that readers have a solid understanding of software application development using IP multicast and using Common Object Request Broker Architecture (CORBA). Knowledge of AACC/ACCS terminology and operations is required for configuring AACC/ACCS to interact with the developed RSM client application.

Note: RSM must be configured in AACC/ACCS. For more information, see the *Avaya Aura® Contact Center Server Administration*.

Chapter 3: Obtaining the Real-Time Statistic Multicast Software

Introduction

The RSM Software Development Kit (RSMSDK) provides the interface files, documentation, sample projects, and binaries necessary to build applications to access the RSM data in AACC/ACCS. The RSMSDK is delivered as an installer that guides the user through the process of installing the files on the target machine.

The RSMSDK provides the header file defining the data encoding of the RSM data stream. A sample project is provided that illustrates receiving skillset information from the multicast stream. An executable version of this project is provided as a reference for testing the data stream server.

The RSMSDK provides the IDL definition of the CORBA-based interface for stream control and data translation. A sample project is included that illustrates using the CORBA interface to translate skillset name to ID. An executable version of this project is provided as a reference for testing the server. To successfully build the sample project, you require access to a CORBA development and run time environment. The section “Building a CORBA application using TAO” describes obtaining and building an application using The ACE ORB (TAO).

To run the sample application, RSM must be configured and operational on AACC/ACCS. For more information, see the *Server Administration* documentation.

Obtaining the Real-Time Statistic Multicast Software Development Kit

The RSM SDK is available on the developer’s Web site at the following URL:

www.avaya.com/devconnect

Instructions for accessing the RSM SDK are available at this Web site.

To install the RSM SDK

1. Remove any existing RSM SDK installed on your system using the instructions in the programmer’s guide for the version of the RSM SDK.
2. Download the RSM SDK zip file from the Web site onto the development server, and then unzip the files.
3. Run the setup program to install the RSM SDK.
4. If an alternative destination folder is required, browse to the alternative location when the **Choose Destination Location** dialog box appears.

   Notes:

1. The RSM SDK utilites and applications require write access to the install locations. Ensure the logged in user has sufficient privilege to allow write access.
2. The RSM SDK utilites and applications require the Visual Studio 2017 redistributable. The installer allows the user to automatically install the redistributable.

To uninstall the RSM SDK

From the system Control Panel, select **Add/Remove Programs**, and then select **Avaya RSM SDK.**

Installing the SDK on non-WIN32 platforms

The multicast and CORBA functions of the RSM SDK are platform-independent. However, the RSM SDK installer is for Windows only. To use the RSM SDK on a non-Windows platform perform the following steps:

1. Install the RSM SDK on a Win32 platform.
2. Copy the SDK files from the created folders to the target platform.

ZLIB data compression library

ZLIB 1.1.2 (http://www.zlib.net) is a general purpose compression library that is used as part of the Real-time Statistic Multicast (RSM) Service to compress data before it is multicast over the network. ZLIB is an open-source, cross-platform, general purpose data compression library, implementing the DEFLATE compression algorithm, which is supported by many languages.

The ZLIB library is used to provide the capability to compress the multicast data sent from the server.

The ZLIB library is a free, general purpose, and legally unencumbered (that is, not covered by any patents) lossless data-compression library for use on virtually any computer hardware and operating system.

The ZLIB compression library provides in-memory compression and decompression functions, including integrity checks of the uncompressed data.

The ZLIB Web site provides a sufficient FAQ section to answer any development questions:

http://zlib.net/zlib_faq.html

AACC/ACCS Data Compression

AACC/ACCS can be configured to compress the multicast stream data before transmission. The compression reduces the network laoding associated with AACC/ACCS multicast broadcasts. See the *Server Administartion* docuemntion for further details.

Multicast compression is disabled by default. Most AACC/ACCS will be using the operating with no compression. If compression is enabled, the multicast stream receiver must recompress the data using ZLIB.

Chapter 4: RSM interfaces

Introduction

The RSM interfaces are based on standardized and mature standards and technologies. The object-oriented CORBA-defined interface (http://www.corba.org/) provides a mechanism for controlling the real-time data streams and for translating specific data associated with the stream into data more useful to the customer. The IP multicast-based interface (http://www.ietf.org/rfc/rfc1112.txt) provides efficient distribution of real-time data to multiple customers. The IP multicast interface does not mandate a specific programming language or platform.

CORBA provides an interface that is language and platform independent. It allows third-party application developers to use the language (C++, Java, Ada, and others) and the platform of their choice (Linux, Solaris, Windows, and others). The CORBA-based API for application developers is in the CORBA IDL format. The IDL interface is independent of programming language and allows for mappings to the popular programming languages. A large number of tools (both commercial and free) are available to third-party developers for translating the IDL to source code.

IP multicast (RFC1112) defines a mechanism for efficient distribution of data to a group of users. IP multicast enables the deployment of scalable and platform-independent receivers of real-time data from AACC/ACCS. The number of users in the data receiver group is unlimited.

Communication between client applications and the RSM server

Communication is based on the client/server paradigm. The client refers to any software application connected to the RSM server on AACC/ACCS using one or both of the interfaces defined in this document. The RSM server acts as the source of data, and the data consumers are the third-party applications (or clients) connected to the RSM server. Communication between client and server is through IP networks (LANs and WANs), based on a connection-based (point-to-point) protocol for CORBA communication, and IP multicast data for the real-time data.

When security is enabled in AACC/ACCS, the RSM CORBA interface supports secure communication with client applications. Security is enabled in AACC/ACCS using Security Manager. Consult AACC/ACCS documentation for information related to security in AACC/ACCS.

When security is enabled, the RSM CORBA interface supports both secure and unsecured communication. Existing unsecured RSM CORBA client applications continue to work even when security is enabled in AACC/ACCS. When security has been enabled, secure and unsecured RSM CORBA client applications can connect simultaneously to AACC/ACCS.

AACC/ACCS RSM CORBA uses mutual TLS to authenticate the identity of CORBA client applications. AACC/ACCS RSM CORBA is configured to use TLS v1.2.Control of access to the RSM data

The CORBA-based interface enforces a validated logon. To access the statistics stream using the CORBA interface, the client specifies a Unicode user name and password, which are validated by the server against the list of allowable users.

The RSM data stream is version-controlled. The RSM server maintains version information and encodes version information in the stream data. The CORBA-based interface enforces compatibility between the version information of the RSM server and the receiving client application. To access the statistics stream, the client application must include the version information that is defined in the IDL file. The version information has a major and minor release number. If the RSM server version is compatible with the client version, the client can start the multicast data. Validation failure (due to incorrect user name, password, or both) generates an authorization-failed (the CORBA "incompatible version") exception.

Creating a client application

The following example is based on C/C++, but can be adapted for other languages. The example is based on a TAO implementation. It requires access to TAO/ACE include files, libraries, and binaries

1. For CORBA functions, use the TAO IDL compiler to compile the IDL file into source code.
2. Write the C++ client application.
    a. For CORBA functions, retrieve a reference (using the Naming Service or IOR) to the RSM object and access the operations defined by the IDL interface.
    b. For IP multicast, initialize sockets for IP multicast to create the IP multicast handler for the statistical data.
3. Compile and link the client application.
    a. For compilation, define the preprocessor options and include directories.
    b. For linking, define the dependent libraries.
4. Run the client application. Ensure the client application has access to the run-time environment.

A typical application scenario

A client application is required to start and stop the flow of statistic information. Typically, the client application must perform the following steps:

1. Register with the ORB. For a multithreaded client application create threads for handling incoming statistics sent by the RSM server.
2. Request that the server start sending statistics using the StartStatistics() API. User name, password, and version information are passed as part of the API.
3. Receive RSM statistics over the LAN using IP multicast.
4. Translate internal identifiers to their equivalent administrative names. For example, a skillset name is more meaningful than the ID representing the skillset.

5. Continue receiving additional RSM statistics over the LAN using IP multicast.
6. Request the sending of statistics to stop using the StopStatistics() API. Do not request to stop statistics if other applications are receiving statistics.

Chapter 5: Interface Definition

Introduction

The RSM server supports two interfaces—the RSM multicast data stream interface and the CORBA stream control/data translation interface.

The RSM multicast data interface is raw byte stream data. All statistics are packaged into field position data. The client application must retrieve the data packet and parse the appropriate statistics from the packet. IP multicast is not a guaranteed delivery protocol, so the client application must expect packets that may not be intact or in order. The data stream is sent periodically with the period defined on AACC/ACCS. See the AACC/ACCS *Server Administration* documentation. The data packet is sent even if the data is unchanged since the last packet was sent. The update period of all statistical groupings is configured in AACC/ACCS. By default, statistics are sent every 5 seconds for the statistical groupings moving window and interval-to-date.

The CORBA stream control/data translation interface allows the client application to control the RSM server. This interface allows the client application to start and stop the RSM data stream (provided the user and version information is valid), and to translate an ID to a name identifier (and vice versa).

The multicast port information for the statistical data channels is configured through the RSM configuration utility. By default, the following port assignments are used:

- Application statistics—interval-to-date = port 6020
- Application statistics—moving window = port 6030
- Skillset statistics—interval-to-date = port 6040
- Skillset statistics—moving window = port 6050
- Agent statistics—interval-to-date = port 6060
- Agent statistics—moving window = port 6070
- Nodal statistics—interval-to-date = port 6080
- Nodal statistics—moving window = port 6090
- IVR statistics—interval-to-date = port 6100
- IVR statistics—moving window = port 6110
- Route statistic (available only for M1)—interval-to-date = port 6120
- Route statistic (available only for M1)—moving window = port 6130

Note: RSM multicast must be configured on AACC/ACCS. For more information, see AACC/ACCS *Server Administration* documentation.

Network data format

IP multicast data sent from Avaya Aura® Contact Center Server transmits in network byte order (Big-Endian). In Big-Endian, the most significant byte is on the left end of a word. In Little-Endian, the most significant byte is on the right end of a word. For your particular hardware, you may need to convert the network byte order to your hardware platform byte order.

RTD Statistic Multicast Interface

The IDL file **rtd.idl** defines the Request/Response interface. The interface allows the client applications to start and stop the RSM statistic data stream, and to translate an ID to a name and a name to an ID.

rtd.idl is available from the installed RSMSDK.

API descriptions

::StartStatistics ( )

Description

Instruct the RSM server to start the IP multicast sends (for the selected statistics). Note: You must restart the SDP Service for the changes to take effect.

Parameters

clVersion – The client member function version of the interface RSM server.

usrId – The Unicode login user name and password.

::StopStatistics ( )

Description

Instruct the RSM server to stop the IP multicast sends. Note: You must restart the SDP Service for the changes to take effect.

Parameters

usrId – The Unicode login user name and password.

::GetSkillsetIdToName ( )

Description

Given a skillset id, this function will return the corresponding skillset name.

Parameters

SklId – A sequence of skillset id.

SklNm – The returned sequence of skillset name.

::GetSkillsetNameToId ( )

Description

Given a skillset name, this function will return the corresponding skillset id.

Parameters

SklNm – A sequence of skillset name. API descriptions

::StartStatistics ( )

Description

Instruct the RSM server to start the IP multicast sends (for the selected statistics). Note: You must restart the SDP Service for the changes to take effect.

Parameters

clVersion – The client member function version of the interface RSM server.

usrId – The Unicode login user name and password.

::StopStatistics ( )

Description

Instruct the RSM server to stop the IP multicast sends. Note: You must restart the SDP Service for the changes to take effect.

Parameters

usrId – The Unicode login user name and password.

::GetSkillsetIdToName ( )

Description

Given a skillset id, this function will return the corresponding skillset name.

Parameters

SklId – A sequence of skillset id.

SklNm – The returned sequence of skillset name.

::GetSkillsetNameToId ( )

Description

Given a skillset name, this function will return the corresponding skillset id.

Parameters

SklNm – A sequence of skillset name.

SklId – The returned sequence of skillset id.

::GetApplicationIdToName ( )

Description

Given an application id, this function will return the corresponding application name.

Parameters

ApplId – A sequence of application id.

ApplNm – The returned sequence of application name.

::GetApplicationNameToId ( )

Description

Given an application name, this function will return the corresponding application id.

Parameters

ApplNm – A sequence of application name.

ApplId – The returned sequence of application id.

::GetAgentIdToName ( )

Description

Given an agent id, this function will return the corresponding agent name.

Parameters

AgtId – A sequence of agent id.

AgtNm – The returned sequence of agent name.

::GetAgentNameToId ( )

Description

Given an agent name, this function will return the corresponding agent id.

Parameters

AgtNm – A sequence of agent name.

AgtId – The returned sequence of agent id.

::GetCDNNameToId ( )

Description

Given a CDN name, this function will return the corresponding CDN

id.

Parameters

CDNNm – A sequence of CDN name.

CDNId – The returned sequence of CDN id.

::GetCDNIdToName ( )

Description

Given a CDN id, this function will return the corresponding CDN name.

Parameters

CDNId – A sequence of CDN id.

CDNNm – The returned sequence of CDN name.

::GetDNISNameToId ( )

Description

Given a DNIS name, this function will return the corresponding DNIS id.

Parameters

DNISNm – A sequence of DNIS name.

DNISId – The returned sequence of DNIS id.

::GetDNISIdToName ( )

Description

Given a DNIS id, this function will return the corresponding DNIS name.

Parameters

DNISId – A sequence of DNIS id.

DNISNm – The returned sequence of DNIS name.

::newSkillsetRtd()

Description

Returns the skillset RTD object instance.

::newApplicationRtd()

Description

Returns the application RTD object instance.

::newAgentRtd()

Description

Returns the agent RTD object instance.

```
::newCDNRtd()
Description
Returns the CDN RTD object instance.

::newDNISRtd()
Description
Returns the DNIS RTD object instance.

::deleteSkillsetRtd()
Description
Deletes the Skillset RTD object instance.

::deleteApplicationRtd()
Description
Deletes the Application RTD object instance.

::deleteAgentRtd()
Description
Deletes the Agent RTD object instance.

::deleteCDNRtd()
Description
Deletes the CDN RTD object instance.

::deleteDNISRtd()
Description
Deletes the Agent RTD object instance.
```

Note: If the ID in an IdToName command is invalid, all name fields are filled with the text string "NortelNetworksNoMatchName." If the name in a NameToId command is invalid, the ID field is assigned a value of -1.

Exceptions
In addition to the standard exception-generated signals (for example, Communication Failure), the following exceptions are specific to Avaya:

- AuthorizationFailed—This exception is raised when a client attempts an illegal logon (invalid user name or password, or both).
- IncompatibleVersion—This exception is raised when a client has a version that is not supported by the RSM server.
- NotReady—This exception is raised when the RSM server is in the process of initializing itself and is not ready to accept requests.
- InternalError—This exception is raised when the RSM server encounters an internal error. Consult the Contact Center Manager Event Browser for more information.

CORBA development environment

CORBA is an open standard specification that allows compatible ORBs to communicate with each other. AACC/ACCS uses TAO to provide the server end-point for the RSM SDK. The server end-point is referred to as the RSM server. The ORB is configured to use the Internet Inter-ORB

Protocol (IIOP) for communication. The AACC/ACCS RSM CORBA interface adheres to Object Management Group's (OMG) CORBA 2 specification.

A TAO Naming Server is provided by AACC/ACCS to allow client applications locate the RSM server. The default Naming Service port is 4422.

For the client application, any ORB compliant with the CORBA 2 specification can interoperate with the RSM server on AACC/ACCS. The RSM SDK provides a CORBA development and run-time environment based on TAO 2.5.1.

The RSM SDK provides the required runtime and libraries to build and deploy a CORBA RSM application.

Chapter 6: How to locate the RSM Service

Naming Service

The AACC/ACCS RSM server registers the following default CORBA compound name with the AACC/ACCS TAO Naming Service:

```
NortelNetworks\SymposiumCallCenterServer\RSM
```

In addition to the default CORBA compound name, each RSM server registers its Contact Center site name with the Name Service in the manner shown:

```
NortelNetworks\SymposiumCallCenterServer\<Contact Center Manager server name>\RSM
```

By default, all RSM servers attempt to create and log the default CORBA name (NortelNetworks\SymposiumCallCenterServer\RSM). In a network situation, only one RSM server can successfully register with the default CORBA compound name (the first); the other RSM server must register its Contact Center site name. In a non-network environment, the client application can find the RSM server with the default CORBA compound name. Use the following procedure to locate the Naming Service.

Note: The utility `D:\Avaya\Contact Center\Manager Server\TAO17\bin\ tao_nslist.exe` in AACC/ACCS lists the services registered with the Name Service.

When security is enabled in AACC/ACCS, the RSM CORBA interface supports secure communication with client applications. However, the Naming Service is not secured. RSM CORBA client applications must connect unsecured to the Naming Service.

To locate the CORBA Naming Service

In AACC/ACCS, the TAO Naming Service location is defined in the configuration file.

```
D:\Avaya\Contact Center\Manager Server\TAO17\properties\rsm.ini
```

The following is a sample RSM.ini file:

```
# This is a list of command line arguments.  Leave blank to use default
#
# Default IORFile is written out to C:\\rsm_IOR.ref.
#
# NameService setting at
HKEY_LOCAL_MACHINE\\SOFTWARE\\ACE\\TAO\\TaoNamingServiceOptions
```

```
#
[TAO_Setup]
NameServerPort=4422
RSM_Port=0
ORBDebug=true
ORBDebugLevel=0
ORBSvcConf=
IORFile=
```

The RSM Service automatically generates a new persistent IOR file when the service starts up. The IOR file is called `C:\rsm_IOR.ref`.

Chapter 7: Statistics

Real-time data

Each table in this section has a defined data type of Cumulative, State, or Admin.

- Cumulative—The statistics are accumulated over a specified period of time (for example, the number of calls answered during an interval).
- State—The value depends on the instantaneous state of the system (for example, the state of an agent at a given time).
- Admin—The value is entered by a data administrator and is not affected by call events (for example, a skillset ID).

For cumulative statistics, data can be collected in two ways:

- moving window—The data is collected within the fixed size time window of 10 minutes that moves forward as time progresses. The fixed size time window is divided into a number of equal data sampling periods. As each sampling period expires, data collected in the current sampling period is added to the totals of the current time window, while the values from the oldest sampling period within the current time window are subtracted from the totals. Therefore, the totals always represent the last 10 minutes of activity.
- interval-to-date—The data is collected on an interval basis. The interval is in increments of 15 minutes up to a maximum of 24 hours. When the specified interval is reached, all data fields are reset to zero, and collection starts for the next interval.

Statistics are available for multimedia contacts when Open Queue is licensed and enabled. Telephony-specific statistics do not contain meaning for the multimedia contacts.

Skillset statistics

Skillset statistics provide instantaneous state and cumulative performance measurement information on a per-skillset basis.

The following Skillset statistics are supported:

Skillset statistics	Data type	Description
Skillset Id	Admin	A unique number to identify a skillset.
Agents Available	State	The number of agents that are currently waiting for calls and are logged on for this skillset.
Agents In Service	State	The number of agents logged on for this skillset.
Agents On Skillset Calls	State	The number of agents currently on calls that were queued to this skillset.

Skillset statistics	Data type	Description
Agents Not Ready	State	The number of agents logged on for this skillset who are currently in the Not Ready state.
Calls Waiting	State	The number of calls currently waiting for an agent with this skillset.
Longest Waiting Time Since Last Call	State	The longest waiting time of all idle agents who are currently waiting to answer calls for this skillset.
Maximum Waiting Time	State	The maximum waiting time spent by all calls that are currently waiting for an agent with this skillset.
Waiting Time	State	The total waiting time spent by all calls that are currently waiting for an agent with this skillset.
Expected Waiting Time	State	The time that a new call is expected to wait before being answered by an agent with this skillset.
Call Answered After Threshold	Cumulative	The number of calls that were answered after experiencing a delay greater than or equal to the service level threshold for this skillset.
Longest Waiting Time Since Login	State	The longest waiting time of idle agents who are waiting to answer calls for this skillset. Measurement of waiting time begins at logon.
Agents On DN Calls	State	The number of agents logged on for this skillset who are currently handling a DN call.
Skillset State	State	The state of the skillset (In Service or Out of Service).
Agent Unavailable	State	The number of agents who are currently unavailable to take calls. This value is calculated based on ▪ Agents in Service ▪ Agents on Avaya Aura® Contact Center calls ▪ Agents on DN Calls ▪ Agents Available ▪ Agents Not Ready
Network Calls Waiting	State	The number of network calls currently waiting for this skillset.
Network Calls Answered	State	The number of network calls answered by an agent with this skillset.
Total Calls Answered Delay	Cumulative	The delay experienced by all calls that were answered by an agent with this skillset from the time the calls were queued to the skillset until they were answered.
Total Calls Answered	Cumulative	The total number of calls answered by an agent with this skillset.

Skillset statistics	Data type	Description
AgentOnOtherSkillsetCall	State	The number of agents who are assigned to this skillset but are on active calls for other skillsets (local or network).
AgentOnACDDNCall	State	The number of agents who are assigned to this skillset but are also active on ACD calls.
AgentOnNACDDNCall	State	The number of agents who are assigned to this skillset but are also active on NACD calls.
AgentOnNetworkSkillsetCall	State	The number of agents who are handling incoming network calls.
CallsOffered	Cumulative	The number of local and incoming network CDN calls queued to this skillset.
SkillsetAbandoned	Cumulative	The number of local and incoming network CDN calls that were abandoned while being queued to this skillset.
SkillsetAbandonedDelay	Cumulative	The amount of delay experienced by local and incoming network CDN calls that were abandoned while being queued to this skillset.
SkillsetAbandonedAfterThreshold	Cumulative	The number of local and incoming network CDN calls whose Skillset Abandoned Delay values were greater than or equal to the service level threshold.
NetworkCallsOffered	Cumulative	The number of incoming network CDN calls queued to this skillset.
Queued Call Answered	Cumulative	The number of queued calls that were answered for the skillset within the last interval-to-date or moving window.

Application statistics

Application statistics provide current state and cumulative performance measurement information on a per-application basis. An application corresponds to a single primary script that provides call processing for a particular type of call and all its associated secondary scripts. For example, a department store contact center may have a catalog sales application and a credit card inquiry application.

The following Application statistics are supported:

Application statistics	Data type	Description
Application Id	Admin	A unique number to identify an application.

Application statistics	Data type	Description
Calls Abandoned	Cumulative	The number of calls abandoned.
Calls Abandoned After Threshold	Cumulative	The number of calls abandoned after experiencing a delay greater than or equal to the service level threshold for the application.
Calls Abandoned Delay	Cumulative	The total delay experienced by all calls that were abandoned.
Calls Answered	Cumulative	The number of calls answered.
Calls Answered After Threshold	Cumulative	The number of calls answered after experiencing a delay greater than or equal to the service level threshold for the application.
Calls Answered Delay	Cumulative	The total delay experienced by all calls that were answered.
Calls Waiting	State	The number of calls that are currently waiting.
Maximum Waiting Time	State	The amount of time that the oldest unanswered call has been in the system.
	State	The total amount of time that all calls have been waiting to be answered.
	Cumulative	The delay experienced by all calls from the time they are queued against the first skillset until they are answered.
Calls Given Termination Treatment	Cumulative	The number of calls that were terminated with one of the following treatments: ▪ given force busy, force overflow, force disconnect, route to, or default ▪ reached a non-ISDN trunk while being routed to a remote site ▪ transferred in an IVR session ▪ networked out through an NACD queue
Calls Offered	Cumulative	The number of calls that were offered.
NetworkOutCalls	Cumulative	The number of network calls that were sent to another site.
NetworkOutCallsAbandoned	Cumulative	The number of network calls that were abandoned at destination sites.
NetworkOutCallsAbandonedDelay	Cumulative	The total delay experienced by all network calls that were abandoned at destination sites.
NetworkOutCallsAnswered	Cumulative	The number of network calls that were answered at destination sites.
NetworkOutCallsAnsweredDelay	Cumulative	The total delay experienced by all network calls that were answered at destination sites.
NetworkOutCallsWaiting	State	The number of network calls that are currently waiting.
NetworkOutCallsRequested	State	The number of network calls that are currently requesting.

Application statistics	Data type	Description
Delay Before Interflow	Cumulative	The amount of time a call spent in the Master Application before interflowing to the Primary Application. For the Master Application, this is a total delay before interflow. For each Primary Application, this value provides a breakdown of time spent in the Master Application.

Agent statistics

Agent statistics provide instantaneous state information regarding an agent. These statistics provide a supervisor with a means to monitor what the agents are doing at any point in time.

The following Agent statistics are supported:

Agent statistics	Data type	Description	
Agent Id	Admin	A unique number to identify an agent.	
State	State	This is a multistate value indicating the states that the agent is currently in.	
		Contact Center State AGENT_BUSY ACTIVE ONHOLD NOTRDY BRK IDLE RESERVE CALL_PRESENT CONSULTATION EMERGENCY	 0x00000200 0x00000100 0x00000080 0x00000040 0x00000020 0x00000010 0x00000008 0x00000004 0x00000002 0x00000001
		DN Out Call State DN_OUT_ACTIVE DN_OUT_ONHOLD	 0x00002000 0x00001000
		DN In Call State DN_IN_ACTIVE DN_IN_ONHOLD	 0x00020000 0x00010000
		NACD Call State NACD_ACTIVE NACD_ONHOLD	 0x00200000 0x00100000
		ACD Call State ACD_ACTIVE ACD_ONHOLD	 0x02000000 0x01000000
		Walkaway Call State WALKAWAY	 0x10000000
Supervisor Id	Admin	The agent's supervisor ID.	
Time In NGCC State	Cumulative	The length of time that the agent has been in the NGCC state (excluding DN states).	
Answering Skillset	State	The ID of a skillset assigned to the agent answering Avaya Aura® Contact Center calls.	
DN In Time In State	Cumulative	The length of time an agent has been in the DN IN state (that is, answering incoming DN calls).	

Agent statistics	Data type	Description
DN Out Time State	Cumulative	The length of time an agent has been in the DN OUT state (that is, making outgoing DN calls).
Position Id	Admin	The unique identifier of the agent's position ID.
NotReadyReasonCode	State	The Not Ready Reason Code entered by the agent.
DNOutCallNumber	State	The DN number that the agent dialed.
SkillsetCallAnswered	Cumulative	The number of local and incoming network CDN calls answered by an agent.
DNInCallAnswered	Cumulative	The number of DN calls answered by an agent.
DNOutCallAnswered	Cumulative	The number of DN calls made by an agent.
nsweringApplication	State	A unique number to identify an application.
AnsweringCDN	State	A special directory number that allows incoming calls to be queued at a CDN when they arrive at the switch.
AnweringDNIS	State	The phone number dialed by the incoming caller.

Nodal statistics

Nodal statistics provide instantaneous state and cumulative accounting information for a server in Avaya Aura® Contact Center. Usually, there is a single server in a contact center, and the nodal statistics are equal to the contact center statistics.

The following Nodal statistics are supported:

Nodal statistics	Data type	Description
Dummy Key	Admin	An artificial key for use by the application. (It is provided to the application to make the interface consistent, allowing for an easier application of delta, delete, new table values.)
Calls Offered	Cumulative	The number of calls that were offered to this site.
Calls Answered	Cumulative	The number of calls that were answered at this site.
Calls Waiting	State	The number of calls that are currently waiting to be answered.
Network Calls Offered	Cumulative	The number of network calls offered to this site.
Network Calls Being Answered	State	The number of network calls that are currently being answered at this site.
Network Calls Waiting	State	The number of network calls that are currently waiting to be answered.

IVR statistics

IVR statistics provide state and cumulative performance measurement information on a per-IVR queue basis. These statistics provide a means to monitor the usage of the port resources of an IVR queue from a real-time perspective.

The following IVR statistics are supported:

IVR statistics	Data type	Description
IVR Queue ID	Admin	A unique number to identify an IVR queue.
Calls Waiting	State	The number of calls that are currently waiting at this IVR queue.
Calls Answered	Cumulative	The number of calls that were answered by this IVR queue.
Calls Answered Delay	Cumulative	The total delay experienced by all calls that were answered by this IVR queue. The delay begins when a call is queued against this IVR queue.
Calls Answered After Threshold	Cumulative	The number of calls answered by this IVR queue that experienced a delay greater than or equal to the service level threshold for this IVR queue. The delay begins when a call is queued against this IVR queue.
Calls Not Treated	Cumulative	The number of calls that were abandoned or pulled back while waiting in this IVR queue.
Calls Not Treated Delay	Cumulative	The total delay experienced by all calls that were abandoned or pulled back from this IVR queue. The delay begins when a call is queued against this IVR queue.
Calls Not Treated After Threshold	Cumulative	The number of calls abandoned or pulled back while waiting in this IVR queue that experienced a delay greater than or equal to the service level threshold for this IVR queue. The delay begins when a call is queued against this IVR queue.

Route statistics

Route statistics provide instantaneous and cumulative All Trunks Busy (ATB) information on a per-route basis.

The following Route statistics are supported:

Route statistics	Data type	Description
Route Number	Admin	A unique number to identify a route.
All Trunks Busy	State	Indicates whether all trunks in this route are currently busy.
All Trunks Busy	Cumulative	The total time this route has been in the All Trunks Busy state.

Splitting of data across multiple packets

For large skillset or agent numbers, the RSM data may not fit into a single 64K IP packet. Where the server detects that the data cannot fit into a single packet, the server partitions the data across multiple packets. The server uses two of the four reserved fields to allow the receiver to detect that multiple packets are being sent. Reserved field 0 stores the sequence number while reserved field 1 stores the total number of packets in the sequence. The sequence number and the number of packets both start at 1. When the data fits in a single packet, the number of packets is 1.

Checking for data compression

For data compression, the RSM data packet has a header with an Encode reserved field to allow the receiver to detect if the packet received has been compressed. If the Encode field value is 1, the data has been compressed.

Data stream definition

The multicast data stream is defined in a C style header file. The header file is available from the deployed RSM SDK. The location of the header file is:

```
<install locations>\RSMSDK\Multicast\include\nimulticastdef.h
```

The header file provides the information needed to decode the datastream.

stream	***struct***	***Notes***

	NIMultiCastHeader	The header provides message related information. There are a number of fields in the header that are critical to decoding the message content: 1. Release : The major version number. 2. MinorVersion : The minor version number. 3. MessageType : Identifies the type of message. 4. NumRecords : Number of records in the message. 5. Recordsize : Size of data in each record. In particular, the Release and MinorVersion numbers are used to identify the content of the agent stream.
Application	NIMultiCastApplicationRecord	
Skillset	NIMultiCastSkillsetRecord	
Agent	NIMultiCastAgentRecord	NIMultiCastAgentRecord is in effect when Release < 5.
	NIMultiCastAgentRecord_Rls5	Contains all the fields in NIMultiCastAgentRecord plus additional fields identifying the answering application, CDN and DNIS. NIMultiCastAgentRecord_Rls5 is the default operation for all currently supported releases of AACC 6.4 and AACC / ACCS 7.
	NIMultiCastAgentRecordNRRC	Contains all the fields in NIMultiCastAgentRecord_Rls5 plus an additional field identifying the time in NRRC. NIMultiCastAgentRecordNRRC was introduced in AACC 6.4 SP 12 however the default operation for all currently supported releases of AACC 6.4 and AACC / ACCS 7 uses NIMultiCastAgentRecord_Rls5. NIMultiCastAgentRecordNRRC is in effect only when AACC / ACCS has been configured to generate a header with Release = 8 and MinorVersion = 4.
Nodal	NIMultiCastNodalRecord	

IVR	`NIMultiCastIVRRecord`	
Route	`NIMultiCastRouteRecord`	

Chapter 8: RSM SDK Sample Applications

Introduction

The following examples illustrate:

1. how the multicast skillset information from AACC/ACCS can be retrieved and displayed in a console application
2. how the CORBA interface can be used to convert skillset name to its ID and display the ID in a console application in both secure and non-secure mode

These examples are for guidance only. Customers should develop their own applications based on their specific needs.

All sample code is supplied as part of the RSM SDK. The sample code is provided for illustration only. Application developers are free to add to or modify the code as required.

IP Multicast Sample Application

The project **mSklRcv** in the RSM SDK provides an example of C++ application to retrieve the skillset statistics from the skillset multicast stream.

CORBA Sample Application

Third-party application development using the CORBA interface requires access to a CORBA 2 compliant ORB.

The RSM SDK provides a CORBA application development environment based on TAO 2.5.1. The sample client application has been built and tested using TAO 2.5.1 with Visual Studio 2017. It can be modified for other ORBs and other development environments.

The IDL file **RTD.idl** provides the definition of the RSM interface. The IDL must be compiled using an CORBA IDL compiler. The TAO compiler (tao_idl.exe) is supplied by the RSM SDK.

Notes:

- A full client development and deployment environment is provided by the RSM SDK. It provides TAO include files, IDL compiler, libraries and run-time.
- The supplied CORBA IDL compiler (tao_idl.exe) requires access to compiler/linker. Ensure that it is in the path environmental variable.

More information about the TAO compiler can be obtained at
http://www.dre.vanderbilt.edu/~schmidt/DOC_ROOT/TAO/docs/compiler.html

When using TAO, a client application must be compiled and linked with the following definitions.

Compiler	Preprocessor	Required preprocessor definitions: **1.** ACE_HAS_NONSTATIC_OBJECT_MANAGER=0 **2.** TAO_ORBSVCS_HAS_DLL **3.** TAO_NAMING_BUILD_DLL
	Include Directories	Required include file: **1.** RSMSDK\CORBA\TAO\include **2.** RSMSDK\CORBA\TAO\include\tao\orbsvcs **3.** RSMSDK\CORBA\TAO\include\tao\orbsvcs\orbsvcs **4.** RSMSDK\CORBA\idl
Linker	Libraries	Required TAO libraries located in RSMSDK\CORBA\TAO\lib. **1.** ace.lib **2.** TAO.lib **3.** TAO_PortableServer.lib **4.** TAO_CosNaming.lib. **5.** TAO_AnyTypeCode.lib **6.** TAO_Security.lib (required for secure communication) The application requires the following libraries at runtime if secure communication is required: **1.** ACE_SSL.dll **2.** TAO_SSLIOP.dll **3.** TAO_Security.dll **4.** libcrypto-1_1.dll **5.** libssl-1_1.dll
Configuration (if secure commication is used)	TAO conf file	# client.conf dynamic SSLIOP_Factory Service_Object * TAO_SSLIOP:_make_TAO_SSLIOP_Protocol_Factory() "-SSLAuthenticate SERVER_AND_CLIENT -SSLPrivateKey PEM:client_key.pem -SSLCertificate PEM:client_cert.pem -SSLCAfile PEM:cacert.pem -SSLVersionList TLSv1.2"
	Certificates	The certificates (CA and server) have been generated using openssl just as an example to show how SSLIOP can be used. **1.** client_key.pem **2.** client_cert.pem **3.** cacert.pem

The sample project **ID2Name** in the RSM SDK provides an example of C++ application to translate a skillset name to skillset ID using CORBA interface.

The project has been built and tested with the RSM SDK using Visual Studio 2017 in Windows 10.

Chapter 9: Building a CORBA application using TAO

Introduction to CORBA

Common Object Request Broker Architecture (CORBA, http://www.corba.org/) is an architecture and specification for creating, distributing, and managing distributed program objects in a network. It allows programs located in different locations and developed by different vendors to communicate in a network through an interface broker. CORBA was developed under the auspices of the Object Management Group (OMG). It was designed to provide platform- and language-independent, object-oriented distributed computing.

The character set supported by the CORBA interfaces is Unicode (ISO 10646). The standard relies on 16-bit character encoding (instead of the 8-bit encoding defined by ASCII). The character set in the AACC/ACCS CORBA interfaces is defined as type unsigned short (or wchar_t in a Linux or Win32 environment). The type char (8-bit) is not supported on the CORBA interfaces.

Introduction to TAO

TAO (The ACE ORB, http://www.cs.wustl.edu/~schmidt/TAO.html) is an open source, CORBA-compliant, C++ Object Request Broker (ORB). TAO supports IIOP 1.2 enabling a high degree of interoperability with other conforming ORBs. It is implemented on top of ACE, which is infrastructure middleware that implements the core concurrency and distribution patterns for communication software. ACE is a highly portable, multiplatform framework that spans both real-time and general purpose operating systems. TAO uses ACE's high-performance, small footprint operating system adaptation layer for all operating system access, rather than invoking non-portable system calls directly. This allows TAO to be platform independent and easily ported to different operating systems.

Introduction to TAO Security

TAO provides an IIOP over SSL implementation called SSLIOP. SSLIOP can be used to enforce integrity, confidentiality and secure invocation when issuing client requests. Furthermore, it also provides the hooks by which X.509 certificate-based request authorization can be implemented in application code.

TAO's SSLIOP pluggable protocol implementation supports both the standard IIOP transport protocol and the secure IIOP over SSL transport protocol. As SSLIOP is implemeneted as a pluggable protocol, it is dynamically loaded into the ORB.

Basic ORB operation and communication

A client ORB communicates with a server ORB to deliver client request messages to the server and return responses from the server (if any) to the client. On the server, the ORB core delivers the requests to the appropriate Object Adapter and returns a reply message to the client-side ORB. The ORB also actively manages the transport-level communications that are used to transmit the requests and reply messages. As part of the OMG standards, a General Inter ORB Protocol (GIOP) is defined for enabling interoperable communications among disparate ORB implementations.

Basic CORBA client operation

A CORBA client application can access remote objects. To do this, it must obtain object references to the CORBA objects that it wants to access. The client can use a CORBA Naming Service or an IOR to obtain a reference to an object on the server. With a valid reference, the client can invoke operations on the object references. The CORBA client is unaware of how the CORBA object is implemented, and the only operations that are available to the client object are those defined in the objects interface, the IDL file. Note that each CORBA object has a unique identity and interface defined in the IDL.

Obtaining TAO

TAO can be downloaded from http://download.dre.vanderbilt.edu/previous_versions/

The downloaded source must be built for the target platform. Build instructions are located at:

http://www.dre.vanderbilt.edu/~schmidt/DOC_ROOT/TAO/TAO-INSTALL.html

Alternatively, you can obtain supported distribution of TAO from vendors. The current vendor

list is located at http://www.cs.wustl.edu/~schmidt/commercial-support.html

Using the TAO IDL compiler to generate source code from the IDL

To use IDL interfaces, the IDL compiler is used to generate skeleton and stub code so requests can traverse from the client to the server. The code generated by the compiler maps is in accordance to standards set by OMG. The generated output files are not interchangeable between ORB implementations—files generated by the TAO ORB cannot be used by another ORB. The files must be compiled with the IDL compiler for the other ORB. However, the server and client can have two different ORB implementations. Therefore, while AACC/ACCS uses the TAO ORB, a third-party application can be built and deployed with an ORB from another open source or commercial vendor.

The third-party developer uses the TAO IDL compiler to generate source code from the IDL file. The source is compiled and linked to the third-party application.

IOR (Interoperable Object Reference)

Each of the AACC/ACCS services that implements a CORBA interface produces an IOR file. An IOR is a stringified object reference that is written to a file and allows objects to communicate across process boundaries. The IOR file is a data structure specified in the OMG CORBA 2.0 Interoperability specification. The IOR provides platform-independent and vendor-independent object references. The IOR is accessed by the client applications to obtain a reference to the server object. The IOR is useful in shared file systems; for example, the client application has access to the location of the IOR generated by the server.

The IOR files for AACC/ACCS are located at the root of the C: drive.

Naming Service

The Naming Service is a CORBA service that runs on the server. It allows CORBA objects to be named by means of binding a name to an object reference. The name binding is stored in the Naming Service. The client supplies the name to the Naming Service to obtain the reference to the desired object.

When security is enabled in AACC/ACCS, the RSM CORBA interface supports secure communication with client applications. However, the Naming Service is not secured. RSM CORBA client applications must connect unsecured to the Naming Service. See the sample code ID2NameTLS in the RSM SDK.

The TAO Naming Service in AACC/ACCS is configured with the following options:

m1	Multicast enabled. Clients can use IP multicast to query for a Naming Service, and this instance will respond. TAO Naming Server is listening for client multicast requests on a specified port. On the client side, <resolve_initial_references> sends out a multicast request on the network, trying to locate a Naming Service. When a Naming Server receives a multicast request from a client, it replies. The default multicast port is used.
ORBEndPoint iiop://[host]:4422	Specifies that the IIOP protocol is being used. The Naming Service is located on the host and listening on port 4422.
o tao_name_service.ior	Identifies the name of the file used to store the IOR of the root Naming Service context.

Reference persistence

All AACC/ACCS services using CORBA as the underlying architecture provide persistent references. The persistent reference allows the client to continue using a server reference even if the server is restarted.

The TAO Naming Service IOR is stored in the file.
`C:\Windows\SysWOW64\tao_name_service.ior`

This enables the client program running on another machine (not the local host) to copy the file to a directory (for example, D:\Name\) and can thus connect to the name service without searching for it using the connection reference

```
-ORBInitRef NameService=file:///D:\Name\tao_name_service.ior
```

TAO utilities

A number of TAO utilities is provided on AACC/ACCS to allow viewing of the Name Service.
The utilities are located in `D:\Avaya\Contact Center\Manager Server\TAO17\bin`
The utilities are:

tao_nslist	Console Naming Service entries viewer.
NamingViewer	GUI Naming Service entries viewer.
tao_catior	Console IOR viewer.

Properties file

The ORB initialization options are configurable via the properties files

`D:\Avaya\Contact Center\Manager Server\TAO17\properties\*.ini.`

The properties file settings (default values shown) are:

```
NameService=iioploc://<ipaddress>:<NameServerPort>/NameService
NameServerPort=4422
iiop://<ipaddress>:<RSM_Port>
RSM_Port=0
ORBDebug=true
ORBDebugLevel=0
ORBSvcConf=
IORFile=
```

NameServerPort allows for the changing of the Naming Service port. It specifies the location of the Name Service; port is 4422 on the local machine. For more information about RSM.ini file details, see “To locate the CORBA Naming Service”.

When ORBDebug is set to true and the ORBDebugLevel is greater than 0 (max of 10), the TAO logging feature is activated. The log file is located in

`D:\Avaya\Logs\CCMS\*_OrbLog*.log`

The ORBSvcConf option allows the use of default configurations for the services.

TAO configures itself using the ACE Service Configurator framework. Thus, options are specified in the familiar svc.conf file (if you want to use a different file name, use the - ORBSvcConf option).

The IORFile option allows you to change the name of the default *.ior file produced by the service. Read the *.ini file to see what other services will be affected by this change.

Client-side settings for TAO

A TAO client is a CORBA application that actively establishes connections, submits requests, and receives responses from a TAO server. You must be careful when specifying the behavior of clients for multithreaded applications.

In particular, it is important to direct the Service Configurator behavior to provide exclusive access to the Transport so that requests are not multiplexed on a connection. You can use -ORBSvcConfDirective statis client-strategy-Factory "-ORBTransportMuxStrategyEXCLUSIVE" to prevent a multithreaded application from blocking. The default operation is to send and receive information on the same connection.

Note: A new ORB is created only for each thread. If a single-threaded application creates more than one ORB (using ORB_init()), it always references to the first ORB created for that particular thread.

For secure communication, the client must provide a service configurator file client.conf file with the configuration for SSLIOP pluggable protocol.

```
# client.conf
dynamic SSLIOP_Factory Service_Object * TAO_SSLIOP:_make_TAO_SSLIOP_Protocol_Factory()
       "-SSLAuthenticate SERVER_AND_CLIENT -SSLPrivateKey PEM:client_key.pem -
SSLCertificate PEM:client_cert.pem -SSLCAfile PEM:cacert.pem -SSLVersionList TLSv1.2"

static Resource_Factory "-ORBProtocolFactory SSLIOP_Factory"
```