Cisco TelePresence Manager Reporting API Developer's Guide for Release 1.8

November 2011

Cisco TelePresence Manager (CTS-Manager) provides programmatic access to an organization's Cisco TelePresence meeting details via a secure application programming interface (API). This API is known as the CTS-Manager Reporting API. The following sections provide information about this guide and the Reporting API:

- Who Should Use this Guide, page 1
- Overview, page 2
- Requirements, page 4
- Standards Compliance, page 5
- Development Platforms, page 5
- Implementation Overview, page 5
- Implementation Details, page 8
- Creating and Using the Reporting API Client, page 8
- Sample Code, page 9
- API Details, page 20
- Troubleshooting, page 54
- Service and Support, page 59
- References, page 59
- Glossary, page 59

Who Should Use this Guide

The primary intent of this document is to provide information to the software developer who will implement the Reporting API. Developers who use this document should have a basic familiarity with software development, web services, and the functionality offered by CTS-Manager.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

This guide also provides a Reporting API overview, which is appropriate for an audience with a less technical background than software developers.

Although the Reporting API works in conjunction with CTS-Manager, this document is not intended for the CTS-Manager administrator who is responsible for some prerequisites that ensure the Reporting API works, for example, uploading a valid Metrics Dashboard and Reporting API license to CTS-Manager.

For complete information intended for the CTS-Manager administrator, see the technical documents at the following location:

http://www.cisco.com/en/US/products/ps7074/tsd_products_support_series_home.html

Accessing Additional Reporting API Information

The Cisco Developer Network (CDN) portal provides access to multiple Cisco technology developer interfaces and collaborative support communities. The CDN also provides formalized support services for these interfaces to enable developers, customers, and partners to accelerate their development.

Developers can access additional Reporting API documentation, forums, blogs, and a Wiki on the CDN. A Cisco.com login is required to access this information.

To access the CDN, go to this location:

http://developer.cisco.com

For more information about the formalized support service, see the "Service and Support" section on page 59.

Overview

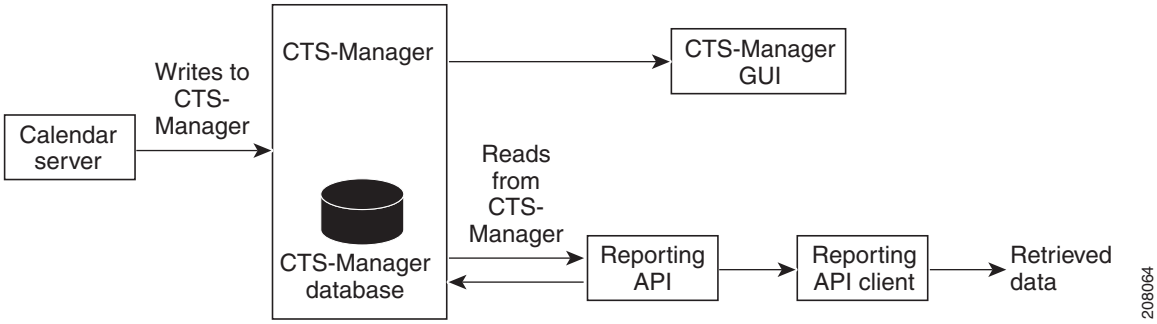
This section provides the following overview information for the Reporting API:

- Functional Overview, page 2
- Benefits of Using the Reporting API, page 3
- Reporting API Output, page 4

Functional Overview

Figure 1 shows a simple Cisco TelePresence System (CTS) topology with components that illustrate how the Reporting API works.

Figure 1 Sample Reporting API Topology



When end users schedule meetings, the calendar server, as shown in Figure 1, handles the scheduling, then writes the meeting data to CTS-Manager. CTS-Manager stores this data, some of which can be viewed using the CTS-Manager graphical user interface (GUI).

An organization can optionally implement the Reporting API, which enables a developer to retrieve additional detailed meeting and associated user survey data from the CTS-Manager database. For information on how the organization can use the retrieved data, see the "Benefits of Using the Reporting API" section on page 3.

Benefits of Using the Reporting API

CTS-Manager provides the Metrics Dashboard, which graphically displays pre-defined reports based on meeting data stored in the CTS-Manager database. For example, the Metrics Dashboard displays a graphical representation of the number of meetings scheduled during a specified timeframe.

For complete information on the Metrics Dashboard, see the *CTS Manager Administration and Installation Guide*, which you can access at this location:

http://www.cisco.com/en/US/products/ps7074/prod_maintenance_guides_list.html

In addition to the data provided in the Metrics Dashboard, the Reporting API is available. When implemented, the Reporting API can provide the following benefits:

- Retrieves More Detailed Meeting Data—In general, the data retrieved by the Reporting API is more detailed than the data presented in the Metrics Dashboard. For example, the Metrics Dashboard reports the number of scheduled meetings that occurred during a specified timeframe, while the Reporting API can collect additional meeting details such as start and end times, features used during the meetings, and so on.

  Using this detailed data in conjunction with the higher-level Metrics Dashboard data, an organization can get a comprehensive view of their Cisco TelePresence return on investment (ROI) and usage statistics.

- Retrieves Scheduled and Ad Hoc Meeting Data—The Reporting API can retrieve both scheduled and ad hoc meeting data.

- Provides the Ability to Create Customized Reports—Using the meeting and associated user survey data retrieved by the Reporting API, an organization can create reports suited for their particular needs and present the data in a variety of ways, for example, in gadgets shown in third-party applications.

- Retrieves Customized Usage Survey Results—If an organization adds customized questions to their usage survey, the Reporting API can retrieve these questions and the corresponding responses for each meeting.

- Retrieves Older Meeting Data—By default, CTS-Manager purges meeting data after it is 1 month old. (This setting is configurable.) The Reporting API enables an organization to retrieve meeting data for archival purposes before CTS-Manager purges it.

Reporting API Output

In general, the Reporting API can capture the following data:

- TelePresence scheduled and ad hoc meeting data, which includes the following:
  - Scheduled start and end times (for scheduled meetings only)
  - Scheduled meeting room names (for scheduled meetings only)
  - Actual start and end times (for scheduled and ad hoc meetings)
  - Actual attended meeting room names (for scheduled and ad hoc meetings)
- Meeting state, which can be scheduled, ad hoc, in progress, and so on
- Usage survey questions and responses associated with the retrieved meeting data
- Time that the usage surveys were last modified
- All features that were enabled during the meeting, for example, Cisco WebEx, recording, interoperability, and so on
- The meeting organizer ID and multipoint conference units (MCUs) used for the meeting (for completed meetings only)

For a complete list of what the Reporting API can retrieve, see the "API Details" section on page 20.

The Reporting API provides meeting objects for the specified start and end dates in XML format. The Reporting API developer can transform this output into the format that best suits the needs of their organization.

Requirements

Before using the Reporting API, you must ensure that the following requirements are met:

- A minimum release of CTS-Manager version 1.8 is being used.
- The Reporting API requires the Metrics Dashboard and Reporting API license. This license must be purchased, then the CTS-Manager administrator must upload it to CTS-Manager via the Licenses page. If a valid license is not in CTS-Manager, the functionality of the Reporting API is not available.
- The Reporting API requires the configuration of a user account for the Reporting API client in these two locations:
  - Lightweight Directory Access Protocol (LDAP) server—The LDAP administrator must create a user group, which does not have a required name. Within this user group, the administrator must create one or more user accounts, including passwords, for the Reporting API client.
  - CTS-Manager—In the Access Management page, the CTS-Manager administrator must map the "Reporting API User" role to the LDAP user group.

  After these user accounts are created, any user who belongs to this LDAP group is authorized to access CTS-Manager data using the Reporting API's web services. If no user accounts exist, a user who tries to access the data receives an unauthorized error.

- The CTS-Manager administrator can enable the optional Meeting Organizer Usage Survey and Benefits Report feature in CTS-Manager via the Application Setting page/Usage Survey tab. If this feature is not enabled, using the MeetingUsageSurveyQuestionnaire API does not return any results.

For complete information on the CTS-Manager Licenses, Access Management, and Application Setting pages, see the *CTS Manager Administration and Installation Guide*, which you can access at this location:

http://www.cisco.com/en/US/products/ps7074/prod_maintenance_guides_list.html

Standards Compliance

Table 1 outlines the specifications with which the Reporting API implementation complies.

Table 1 Reporting API-Supported Specifications

Standard	Where to Find More Information
Simple Object Access Protocol (SOAP) 1.2	http://www.w3.org/TR/soap12-part1/
Web Services Description Language (WSDL) 2.0	http://www.w3.org/TR/wsdl20/
Web Services Base Notification 1.3	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn

Note Compliance with these standards is on limited basis. Where applicable, compliance details are called out in the subsequent sections of this guide.

Development Platforms

The Reporting API works in most SOAP development environments but has been tested with CTS-Manager release 1.8 and with Apache Axis 1.3.

This document provides examples in the Java programming language. The Java examples are based on Apache Axis 1.3 and JDK 5.0 (Java 2 Platform Standard Edition Development Kit 5.0). For more information about Apache Axis 1.3, go to the following location:

http://ws.apache.org/axis/

Note Different development platforms vary in their specification implementations. Implementation differences might prevent access to some or all of the features in the API.

Implementation Overview

API calls represent specific operations that an application can invoke at runtime to perform tasks. The Reporting API enables you to run a query on the following:

- Cisco TelePresence meetings between specified start and end dates
- Meeting organizer usage survey associated with the Cisco TelePresence meetings

This section provides the following overview information on the Reporting API:

- Reporting API Host URL Format, page 6
- Data Model, page 6
- Reporting APIs, page 6

Reporting API Host URL Format

The following URL provides a means for the Reporting API client to access the web service, which runs on CTS-Manager:

http://*hostname*:*port*/rsoapis/services/ReportingMgmt

Where:

hostname is the CTS-Manager hostname.

port should be 8080.

A sample URL is as follows:

http://myCTSManager.mydomain.com:8080/rsoapis/services/ReportingMgmt

Data Model

The Reporting API uses the following class hierarchy for the java.lang.Object:

- **CTMSecuritySubject** (implements java.io.Serializable)
- **UsageSurveyItem** (implements java.io.Serializable)
  - **UsageSurveyQuestion** (implements java.io.Serializable)
  - **UsageSurveyResponseOption** (implements java.io.Serializable)
- **UsageSurveyQuestionnaire** (implements java.io.Serializable)
- **UsageSurveyResponse** (implements java.io.Serializable)
- **MeetingRecord** (implements java.io.Serializable)

Reporting APIs

The CTS-Manager Reporting API supports the following APIs.

Method Summary	
`UsageSurveyQuestionnaire`	`getMeetingUsageSurveyQuestionaire()`
`MeetingRecord[]`	`searchMeetingRecordsByMeetingTime (Calendar start, Calendar end)`

These APIs are read-only and do not provide the capability of writing data to CTS-Manager. For more information about these APIs, see the "API Details" section on page 20.

Query Survey Questionnaire

Description

This API returns the survey questions that meeting organizers can access via the CTS-Manager GUI.

API

getMeetingUsageSurveyQuestionaire

Input

None.

Output:

UsageSurveyQuestionnaire object, which contains an array of UsageSurveyQuestion objects and LastModifedDate. If the request itself is successfully completed, the response code is set to OK.

Possible Errors

- UNAUTHORIZED_ERROR: User is not authorized to run the Reporting API.
- TOO_MANY_ATTEMPT_ERROR: Too many data query attempts for the same session.
- INTERNAL_ERROR: CTS-Manager is experiencing an internal error.

For information on what to do if you receive these errors, see the "Troubleshooting" section on page 54.

Query Meetings

Description

This API returns all scheduled and ad hoc meetings between the specified start and end times and the usage survey responses for each meeting record.

API

searchMeetingRecordsByMeetingTime

Input

Start Date and End Date (UTC time-zone)

Note The maximum allowable range between start and end dates is 6 months.

Output

Array of MeetingRecord object.

Possible Errors

- UNAUTHORIZED_ERROR: User is not authorized to run the Reporting API.
- TOO_MANY_ATTEMPT_ERROR: Too many data query attempts for the same session.
- EXCESSIVE_DATE_RANGE_ERROR: Date range specified may be more than allowed (range of 6 months).
- INTERNAL_ERROR: CTS-Manager is experiencing internal error.

For information on what to do if you receive these errors, see the "Troubleshooting" section on page 54.

Implementation Details

Before implementing the Reporting API, you must be aware of these implementation details:

- CTS-Manager includes a denial-of-service (DoS) handler, which prevents spikes in its CPU usage. As a result, these limitations exist for sending requests to CTS-Manager using the Reporting API:
  - When performing a meeting query, you must specify a start and end date. The maximum allowable range between start and end dates is 6 months.
  - You must wait a minimum interval of 5 minutes between performing two consecutive queries.
- Data in the CTS-Manager database tends to be dynamic. For example, a meeting organizer can respond to usage survey questions initially, then change their responses later. Therefore, when retrieving meeting and usage survey data, carefully consider the timing of your retrieval to ensure that the data you are retrieving is the latest.
- In a TelePresence network topology where Cisco Unified Video Conferencing (CUVC) is used to enable interoperability calls, the meetings retrieved by the Reporting API do not display a list of video conferencing (VC) room participants in the meeting. Instead, the CUVC conference number is shown as a participant. This anomaly is caused by a CUVC limitation and is outside the scope of CTS Manager.

Creating and Using the Reporting API Client

Table 2 provides a high-level task flow that describes how to implement the Reporting API client, who is responsible for performing each task, and where to find additional information if applicable.

Table 2 Reporting API Client Creation and Usage Task Flow

Task No.	Task	Who Is Responsible for Performing this Task?	Where to Find Information
1	Create a user account for the Reporting API client on the LDAP server, which entails performing these two sub-tasks: • Create a user group, which does not have a required name. • Within this user group, create one or more user accounts for the Reporting API client.	LDAP administrator	–
2	Designate a machine as the host of the Reporting API client.	CTS-Manager administrator	For information on the URL that enables the Reporting API client to access the web service on this host, see "Reporting API Host URL Format" section on page 6.
3	If not already done, install CTS-Manager.	CTS-Manager administrator	*CTS Manager Administration and Installation Guide*, which is available at this location: http://www.cisco.com/en/US/products/ps7074/prod_maintenance_guides_list.html

Table 2 Reporting API Client Creation and Usage Task Flow (continued)

Task No.	Task	Who Is Responsible for Performing this Task?	Where to Find Information
4	Upload a valid Metrics Dashboard and Reporting API license to CTS-Manager.	CTS-Manager administrator	*CTS Manager Administration and Installation Guide*, which is available at this location: http://www.cisco.com/en/US/products/ps7074/prod_maintenance_guides_list.html
5	Map the "Reporting API User" role in CTS-Manager to the LDAP user group.	CTS-Manager administrator	*CTS Manager Administration and Installation Guide*, which is available at this location: http://www.cisco.com/en/US/products/ps7074/prod_maintenance_guides_list.html
6	Enable the optional Meeting Organizer Usage Survey and Benefits Report feature.	CTS-Manager administrator	*CTS Manager Administration and Installation Guide*, which is available at this location: http://www.cisco.com/en/US/products/ps7074/prod_maintenance_guides_list.html
7	On the Reporting API client host, download Apache Ant and Apache Axis version 1.3 or later. Alternatively, you can use Eclipse, NetBeans, or another integrated development environment (IDE).	Reporting API developer	–
8	Generate stubs to be used by the Reporting API client code.	Reporting API developer	For more information about generating the stub code, see the "Stub Code Generation" section on page 15.
9	Deploy the Reporting API client as a separate web application on the designated machine.	Reporting API developer	–

After finishing these tasks, the Reporting API client should be able to invoke web services from CTS-Manager.

Sample Code

This section provides sample code. It also describes the following aspects of implementing the Reporting API.

Note The intent of the sample code is to provide examples. These samples should not be used for production systems.

- Development Prerequisites, page 10
- WDSL File, page 10
- Runtime Prerequisites, page 15

- Stub Code Generation, page 15
- Initialization, page 16
- Query Survey Questionnaire, page 19
- Query Meetings, page 19

Development Prerequisites

The sample code assumes that you use Apache Ant for building Java code. For more information about Apache Ant, go to this location:

http://ant.apache.org/

You can also use the Eclipse software development kit (SDK) when working on the Reporting API client Java code.

WDSL File

You can use the following sample web service definition language (WDSL) file content to generate automated client code stubs for the service:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://rsoapis.soap.scheduler.ts.cisco.com"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://rsoapis.soap.scheduler.ts.cisco.com"
xmlns:intf="http://rsoapis.soap.scheduler.ts.cisco.com"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns10="http://lang.java"
xmlns:tns2="http://object.api.rsoapis.soap.scheduler.ts.cisco.com"
xmlns:tns3="http://exception.rsoapis.soap.scheduler.ts.cisco.com"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.3
Built on Oct 05, 2005 (05:23:37 EDT)-->
 <wsdl:types>
  <schema targetNamespace="http://object.api.rsoapis.soap.scheduler.ts.cisco.com"
xmlns="http://www.w3.org/2001/XMLSchema">
   <import namespace="http://exception.rsoapis.soap.scheduler.ts.cisco.com"/>
   <import namespace="http://xml.apache.org/xml-soap"/>
   <import namespace="http://rsoapis.soap.scheduler.ts.cisco.com"/>
   <import namespace="http://lang.java"/>
   <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
   <complexType abstract="true" name="UsageSurveyItem">
    <sequence>
     <element name="ID" nillable="true" type="soapenc:string"/>
     <element name="deleted" type="xsd:boolean"/>
    </sequence>
   </complexType>
   <complexType name="UsageSurveyResponseOption">
    <complexContent>
     <extension base="tns2:UsageSurveyItem">
      <sequence>
       <element name="displayText" nillable="true" type="soapenc:string"/>
      </sequence>
     </extension>
    </complexContent>
   </complexType>
   <complexType name="UsageSurveyQuestion">
```

```
    <complexContent>
     <extension base="tns2:UsageSurveyItem">
      <sequence>
       <element name="displayText" nillable="true" type="soapenc:string"/>
       <element name="required" type="xsd:boolean"/>
       <element name="responseDataType" nillable="true" type="soapenc:string"/>
       <element name="responseOptionType" nillable="true" type="soapenc:string"/>
       <element name="responseOptions" nillable="true"
type="impl:ArrayOf_tns2_UsageSurveyResponseOption"/>
       <element name="responseValueType" nillable="true" type="soapenc:string"/>
      </sequence>
     </extension>
    </complexContent>
   </complexType>
   <complexType name="UsageSurveyQuestionnaire">
    <sequence>
     <element name="lastModified" nillable="true" type="xsd:dateTime"/>
     <element name="questions" nillable="true"
type="impl:ArrayOf_tns2_UsageSurveyQuestion"/>
    </sequence>
   </complexType>
   <complexType name="UsageSurveyResponse">
    <sequence>
     <element name="owningQuestionID" nillable="true" type="soapenc:string"/>
     <element name="value" nillable="true" type="xsd:anyType"/>
     <element name="valueType" nillable="true" type="soapenc:string"/>
    </sequence>
   </complexType>
   <complexType name="MeetingRecord">
    <sequence>
     <element name="MCUNames" nillable="true" type="impl:ArrayOf_soapenc_string"/>
     <element name="meetingEndTime" nillable="true" type="xsd:dateTime"/>
     <element name="meetingActualEndTime" nillable="true" type="xsd:dateTime"/>
     <element name="meetingFeatures" nillable="true" type="impl:ArrayOf_soapenc_int"/>
     <element name="meetingStartTime" nillable="true" type="xsd:dateTime"/>
     <element name="meetingActualStartTime" nillable="true" type="xsd:dateTime"/>
     <element name="meetingState" nillable="true" type="soapenc:int"/>
     <element name="meetingSubject" nillable="true" type="soapenc:string"/>
     <element name="meetingSystemID" nillable="true" type="soapenc:string"/>
     <element name="normal" type="xsd:boolean"/>
     <element name="organizerID" nillable="true" type="soapenc:string"/>
     <element name="organizerTimezone" nillable="true" type="soapenc:string"/>
     <element name="recurringMeeting" type="xsd:boolean"/>
     <element name="roomNames" nillable="true" type="impl:ArrayOf_soapenc_string"/>
     <element name="participatedRoomNames" nillable="true"
type="impl:ArrayOf_soapenc_string"/>
     <element name="surveyLastModifiedTime" nillable="true" type="xsd:dateTime"/>
     <element name="surveyResponses" nillable="true"
type="impl:ArrayOf_tns2_UsageSurveyResponse"/>
    </sequence>
   </complexType>
  </schema>
  <schema targetNamespace="http://rsoapis.soap.scheduler.ts.cisco.com"
xmlns="http://www.w3.org/2001/XMLSchema">
   <import namespace="http://exception.rsoapis.soap.scheduler.ts.cisco.com"/>
   <import namespace="http://object.api.rsoapis.soap.scheduler.ts.cisco.com"/>
   <import namespace="http://xml.apache.org/xml-soap"/>
   <import namespace="http://lang.java"/>
   <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
   <complexType name="ArrayOf_tns2_UsageSurveyResponseOption">
    <complexContent>
     <restriction base="soapenc:Array">
      <attribute ref="soapenc:arrayType"
wsdl:arrayType="tns2:UsageSurveyResponseOption[]"/>
```

```
      </restriction>
     </complexContent>
    </complexType>
    <complexType name="ArrayOf_tns2_UsageSurveyQuestion">
     <complexContent>
      <restriction base="soapenc:Array">
       <attribute ref="soapenc:arrayType" wsdl:arrayType="tns2:UsageSurveyQuestion[]"/>
      </restriction>
     </complexContent>
    </complexType>
    <complexType name="ArrayOf_xsd_anyType">
     <complexContent>
      <restriction base="soapenc:Array">
       <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:anyType[]"/>
      </restriction>
     </complexContent>
    </complexType>
    <complexType name="ArrayOf_soapenc_string">
     <complexContent>
      <restriction base="soapenc:Array">
       <attribute ref="soapenc:arrayType" wsdl:arrayType="soapenc:string[]"/>
      </restriction>
     </complexContent>
    </complexType>
    <complexType name="ArrayOf_soapenc_int">
     <complexContent>
      <restriction base="soapenc:Array">
       <attribute ref="soapenc:arrayType" wsdl:arrayType="soapenc:int[]"/>
      </restriction>
     </complexContent>
    </complexType>
    <complexType name="ArrayOf_tns2_UsageSurveyResponse">
     <complexContent>
      <restriction base="soapenc:Array">
       <attribute ref="soapenc:arrayType" wsdl:arrayType="tns2:UsageSurveyResponse[]"/>
      </restriction>
     </complexContent>
    </complexType>
    <complexType name="ArrayOf_tns2_MeetingRecord">
     <complexContent>
      <restriction base="soapenc:Array">
       <attribute ref="soapenc:arrayType" wsdl:arrayType="tns2:MeetingRecord[]"/>
      </restriction>
     </complexContent>
    </complexType>
   </schema>
   <schema targetNamespace="http://exception.rsoapis.soap.scheduler.ts.cisco.com"
xmlns="http://www.w3.org/2001/XMLSchema">
    <import namespace="http://object.api.rsoapis.soap.scheduler.ts.cisco.com"/>
    <import namespace="http://xml.apache.org/xml-soap"/>
    <import namespace="http://rsoapis.soap.scheduler.ts.cisco.com"/>
    <import namespace="http://lang.java"/>
    <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
    <complexType name="ErrorCode">
     <sequence>
      <element name="ID" nillable="true" type="soapenc:string"/>
      <element name="code" type="xsd:int"/>
      <element name="correctiveAction" nillable="true" type="soapenc:string"/>
      <element name="description" nillable="true" type="soapenc:string"/>
      <element name="module" nillable="true" type="soapenc:string"/>
      <element name="rawMessage" nillable="true" type="soapenc:string"/>
      <element name="severity" nillable="true" type="soapenc:string"/>
     </sequence>
    </complexType>
```

```
    <complexType abstract="true" name="TSException">
     <sequence>
      <element name="ID" nillable="true" type="soapenc:string"/>
      <element name="cause" nillable="true" type="xsd:anyType"/>
      <element name="code" type="xsd:int"/>
      <element name="errorCode" nillable="true" type="tns3:ErrorCode"/>
      <element name="erroredObjects" nillable="true" type="impl:ArrayOf_xsd_anyType"/>
      <element name="message" nillable="true" type="soapenc:string"/>
      <element name="module" nillable="true" type="soapenc:string"/>
     </sequence>
    </complexType>
    <complexType name="APIDispatchException">
     <complexContent>
      <extension base="tns3:TSException">
       <sequence/>
      </extension>
     </complexContent>
    </complexType>
    <complexType name="SearchException">
     <complexContent>
      <extension base="tns3:TSException">
       <sequence/>
      </extension>
     </complexContent>
    </complexType>
   </schema>
   <schema targetNamespace="http://xml.apache.org/xml-soap"
xmlns="http://www.w3.org/2001/XMLSchema">
    <import namespace="http://exception.rsoapis.soap.scheduler.ts.cisco.com"/>
    <import namespace="http://object.api.rsoapis.soap.scheduler.ts.cisco.com"/>
    <import namespace="http://rsoapis.soap.scheduler.ts.cisco.com"/>
    <import namespace="http://lang.java"/>
    <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
    <complexType name="Vector">
     <sequence>
      <element maxOccurs="unbounded" minOccurs="0" name="item" type="xsd:anyType"/>
     </sequence>
    </complexType>
   </schema>
  </wsdl:types>
    <wsdl:message name="getMeetingUsageSurveyQuestionaireResponse">
       <wsdl:part name="getMeetingUsageSurveyQuestionaireReturn"
type="tns2:UsageSurveyQuestionnaire">
       </wsdl:part>
    </wsdl:message>
    <wsdl:message name="SearchException">
       <wsdl:part name="fault" type="tns3:SearchException">
       </wsdl:part>
    </wsdl:message>
    <wsdl:message name="searchMeetingRecordsByMeetingTimeResponse">
       <wsdl:part name="searchMeetingRecordsByMeetingTimeReturn"
type="impl:ArrayOf_tns2_MeetingRecord">
       </wsdl:part>
    </wsdl:message>
    <wsdl:message name="searchMeetingRecordsByMeetingTimeRequest">
       <wsdl:part name="in0" type="xsd:dateTime">
       </wsdl:part>
       <wsdl:part name="in1" type="xsd:dateTime">
       </wsdl:part>
    </wsdl:message>
    <wsdl:message name="APIDispatchException">
       <wsdl:part name="fault" type="tns3:APIDispatchException">
       </wsdl:part>
    </wsdl:message>
```

```
   <wsdl:message name="getMeetingUsageSurveyQuestionaireRequest">
   </wsdl:message>
   <wsdl:portType name="ReportingMgmt">
      <wsdl:operation name="getMeetingUsageSurveyQuestionaire">
         <wsdl:input message="impl:getMeetingUsageSurveyQuestionaireRequest"
name="getMeetingUsageSurveyQuestionaireRequest">
       </wsdl:input>
         <wsdl:output message="impl:getMeetingUsageSurveyQuestionaireResponse"
name="getMeetingUsageSurveyQuestionaireResponse">
       </wsdl:output>
         <wsdl:fault message="impl:APIDispatchException" name="APIDispatchException">
       </wsdl:fault>
         <wsdl:fault message="impl:SearchException" name="SearchException">
       </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="searchMeetingRecordsByMeetingTime" parameterOrder="in0 in1">
         <wsdl:input message="impl:searchMeetingRecordsByMeetingTimeRequest"
name="searchMeetingRecordsByMeetingTimeRequest">
       </wsdl:input>
         <wsdl:output message="impl:searchMeetingRecordsByMeetingTimeResponse"
name="searchMeetingRecordsByMeetingTimeResponse">
       </wsdl:output>
         <wsdl:fault message="impl:APIDispatchException" name="APIDispatchException">
       </wsdl:fault>
         <wsdl:fault message="impl:SearchException" name="SearchException">
       </wsdl:fault>
      </wsdl:operation>
   </wsdl:portType>
   <wsdl:binding name="ReportingMgmtSoapBinding" type="impl:ReportingMgmt">
      <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="getMeetingUsageSurveyQuestionaire">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="getMeetingUsageSurveyQuestionaireRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://rsoapis.soap.scheduler.ts.cisco.com" use="encoded"/>
         </wsdl:input>
         <wsdl:output name="getMeetingUsageSurveyQuestionaireResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://rsoapis.soap.scheduler.ts.cisco.com" use="encoded"/>
         </wsdl:output>
         <wsdl:fault name="APIDispatchException">
            <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="APIDispatchException" namespace="http://rsoapis.soap.scheduler.ts.cisco.com"
use="encoded"/>
         </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="searchMeetingRecordsByMeetingTime">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="searchMeetingRecordsByMeetingTimeRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://rsoapis.soap.scheduler.ts.cisco.com" use="encoded"/>
         </wsdl:input>
         <wsdl:output name="searchMeetingRecordsByMeetingTimeResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://rsoapis.soap.scheduler.ts.cisco.com" use="encoded"/>
         </wsdl:output>
         <wsdl:fault name="APIDispatchException">
            <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="APIDispatchException" namespace="http://rsoapis.soap.scheduler.ts.cisco.com"
use="encoded"/>
         </wsdl:fault>
         <wsdl:fault name="SearchException">
```

```
            <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
name="SearchException" namespace="http://rsoapis.soap.scheduler.ts.cisco.com"
use="encoded"/>
         </wsdl:fault>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="ReportingMgmtService">
      <wsdl:port binding="impl:ReportingMgmtSoapBinding" name="ReportingMgmt">
         <wsdlsoap:address
location="http://localhost:8080/rsoapis/services/ReportingMgmt"/>
      </wsdl:port>
   </wsdl:service>
</wsdl:definitions>
```

Runtime Prerequisites

For the sample code to run successfully, the following prerequisites must be met:

- For information on CTS-Manager requirements that must be met for the Reporting API to function properly, see the "Requirements" section on page 4.
- A deployment descriptor that the client code uses exists and is named client-config.wsdd. This descriptor is included in the class path and includes the following content:

```
<deployment xmlns="http://xml.apache.org/axis/wsdd/"
xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
 <transport name="http" pivot="java:org.apache.axis.transport.http.HTTPSender"/>
<service name="ReportingMgmt">
   <requestFlow >
    <handler type="java:org.apache.ws.axis.security.WSDoAllSender" >
     <parameter name="action" value="UsernameToken"/>
    </handler>
   </requestFlow >
 </service>
</deployment>
```

Stub Code Generation

Use the Axis1.3 WSDL2Java tool to generate stub code, a sample of which is shown in the "Sample Ant Code" section on page 16. Your working directories will be different than the sample code and as a result, will require the appropriate changes for the code to work. Alternatively, you can use the Eclipse SDK or another IDE to automatically create the client stub from the WDSL file provided in the "WDSL File" section on page 10.

The sample code shows an ant target to generate stub code from API WSDL. This code generates stub code and the following supporting classes:

- ReportingMgmt.java
- ReportingMgmtService.java
- ReportingMgmtServiceLocator.java
- ReportingMgmtSoapBindingStub.java
- UsageSurveyItem.java
- UsageSurveyQuestion.java
- UsageSurveyQuestionnaire.java

- UsageSurveyResponse.java
- UsageSurveyResponseOption.java

Sample Ant Code

```
<project name="ReportingAPI_Project" default="reportrun" basedir="c:\">
<description> Generate Java code</description>
<!-- set global properties for this build -->
<property name="devdir" location="Report_TestClient"/>
<path id="myaxis.classpath">
        <fileset dir="C:\API\axis-1.3\lib">
            <include name="*.jar"/>
        </fileset>
    </path>

    <target name="reportrun"
description="Converts AXIS1.3 WSDL files to Java">
        <java classname="org.apache.axis.wsdl.WSDL2Java"
fork="true" maxmemory="512m">
            <classpath refid="myaxis.classpath"/>
            <arg line ="--output ${devdir} -v ${devdir}/ReportingMgmt.wsdl"/>
        </java>
    </target>
</project>
```

Initialization

Each API invocation must pass along the username (login name) and password for the Reporting API User account, which exists on the LDAP server. CTS-Manager authenticates and authorizes the Reporting API user before it proceeds with the API invocation.

Sample Code for Initializing the API

```
package com.cisco.ts.scheduler.server.security;

import java.io.IOException;

import javax.security.auth.callback.Callback;
import javax.security.auth.callback.CallbackHandler;
import javax.security.auth.callback.NameCallback;
import javax.security.auth.callback.PasswordCallback;
import javax.security.auth.callback.UnsupportedCallbackException;

import org.apache.ws.security.WSPasswordCallback;

/**
 * <p>
 * ReportingCallBackHandler has constructor that takes a username and password so
 * its handle() method does not have to prompt the user for input. Useful for
 * server-side applications.
 */
public final class ReportingCallBackHandler implements CallbackHandler {

    private String username;

    private char[] password;

    /**
```

```
     * <p>
     * Creates a callback handler with the give username and password.
     * @param user String
     * @param pass String
     */
    public ReportingCallBackHandler(String user, String pass) {
        this.username = user;
        this.password = pass.toCharArray();
    }

    /**
     * Handles the specified set of Callbacks. Uses the username and password
     * that were supplied to our constructor to popluate the Callbacks. This
     * class supports NameCallback and PasswordCallback.
     *
     * @param callbacks
     *            the callbacks to handle
     * @throws java.io.IOException
     * @throws UnsupportedCallbackException
     *             if the callback is not an instance of NameCallback or
     *             PasswordCallback
     * @throws IOException
     *             if an input or output error occurs.
     * @see javax.security.auth.callback.CallbackHandler#handle(Callback[])
     */
    public void handle(Callback[] callbacks) throws java.io.IOException,
            UnsupportedCallbackException {
        final int theLength = callbacks.length;
        for (int i = 0; i < theLength; i++) {
            if (callbacks[i] instanceof NameCallback) {
                ((NameCallback) callbacks[i]).setName(username);
            } else if (callbacks[i] instanceof PasswordCallback) {
                ((PasswordCallback) callbacks[i]).setPassword(password);
            } else if (callbacks[i] instanceof WSPasswordCallback) {
                ((WSPasswordCallback) callbacks[i]).setPassword(new String(password));
            } else {
                throw new UnsupportedCallbackException(callbacks[i],
                        "Callback class not supported");
            }
        }
    }

    /**
     * Clears out password state.
     */
    public void clearPassword() {
        if (password != null) {
            final int theLength = password.length;
            for (int i=0; i<theLength; i++) {
                password[i] = ' ';
            }
            password = null;
        }
    }

}


import javax.xml.rpc.ServiceException;

import org.apache.commons.httpclient.NameValuePair;
import org.apache.log4j.Logger;
import org.apache.ws.security.WSConstants;
import org.apache.ws.security.handler.WSHandlerConstants;
```

```
import com.cisco.ts.scheduler.server.security.ReportingCallBackHandler;
import com.cisco.ts.scheduler.soap.rsoapis.ReportingMgmtServiceLocator;
import com.cisco.ts.scheduler.soap.rsoapis.ReportingMgmtSoapBindingStub;
import com.cisco.ts.scheduler.soap.rsoapis.api.object.UsageSurveyResponse;
import com.cisco.ts.scheduler.soap.rsoapis.api.object.MeetingRecord;

import com.cisco.ts.scheduler.exception.ErrorCodes;
import com.cisco.ts.scheduler.exception.ServerInternalException;
import com.cisco.ts.scheduler.util.GlobalConstants;
import com.cisco.ts.scheduler.util.XMLParser;

public class ReportingMgmtClient {

    private static final int TIME_OUT = 60000;

    private ReportingMgmtServiceLocator itsServiceLocator = null;

    private String userName = null;
    private String userPassword = null;

    String HTTP_URI = "http://";

   public ReportingMgmtClient(String aHostname, int aPort) {
        this();
        itsServiceLocator.setReportingMgmtEndpointAddress(formAddress(aHostname, aPort));
    }

    public void setUserName(String name) {
        userName = name;
    }

    public void setUserPassword(String psw) {
        userPassword = psw;
    }

    private ReportingMgmtSoapBindingStub getRsoapisBinding()
    throws ServiceException {
    ReportingMgmtSoapBindingStub theBinding = null;
    theBinding = (ReportingMgmtSoapBindingStub) itsServiceLocator.getReportingMgmt();
    theBinding.setTimeout(TIME_OUT);
    return theBinding;
    }

    private String formAddress(String aHostname, int aPort) {
    return HTTP_URI + aHostname + ":" + aPort + "/rsoapis/services/ReportingMgmt";
    }

   private ReportingMgmtClient() {
        itsServiceLocator = new ReportingMgmtServiceLocator();
    }

// Use this stub for all API calls as shown later
```

Query Survey Questionnaire

The Reporting API client uses the getMeetingUsageSurveyQuestionaire API to retrieve the survey questions.

Sample Client Code

The following sample reference code shows how you can retrieve the usage survey questions:

```
// Initialize the stub first as shown earlier
public com.cisco.ts.scheduler.soap.rsoapis.api.object.UsageSurveyQuestionnaire
querySurveyQuestions()
    throws Exception {

        ReportingMgmtSoapBindingStub theStub = getRsoapisBinding();

        if(userName != null && userPassword != null) {
            theStub._setProperty(WSHandlerConstants.PASSWORD_TYPE, WSConstants.PW_TEXT);
            theStub._setProperty(WSHandlerConstants.USER, userName);

            theStub._setProperty(WSHandlerConstants.PW_CALLBACK_REF,
                    new ReportingCallBackHandler(userName, userPassword));
        }
        return theStub.getMeetingUsageSurveyQuestionaire();
    }
```

Query Meetings

The Reporting API client uses the searchMeetingRecordsByMeetingTime API to retrieve scheduled and and ad hoc meetings between the specified start and end times and the usage survey responses for each meeting record.

Sample Client Code

```
// Initialize the stub first as shown earlier
public Collection<MeetingRecord> querySurveyRecords(Date aStartTime, Date anEndTime)
    throws Exception {
        Calendar theStart = Calendar.getInstance();
        theStart.setTime(aStartTime);
        Calendar theEnd = Calendar.getInstance();
        theEnd.setTime(anEndTime);

        ReportingMgmtSoapBindingStub theStub = getRsoapisBinding();

        if(userName != null && userPassword != null) {
            theStub._setProperty(WSHandlerConstants.PASSWORD_TYPE, WSConstants.PW_TEXT);
            theStub._setProperty(WSHandlerConstants.USER, userName);

            theStub._setProperty(WSHandlerConstants.PW_CALLBACK_REF,
                    new ReportingCallBackHandler(userName, userPassword));
        }
        MeetingRecord[] theRecords = theStub.searchMeetingRecordsByMeetingTime(theStart,
theEnd);
return theRecords;
    }
```

API Details

This section provides details about the following Reporting APIs:

- Reporting API, page 21
- MeetingRecord, page 22
- UsageSurveyItem, page 29
- UsageSurveyQuestion, page 32
- UsageSurveyQuestionnaire, page 36
- UsageSurveyResponse, page 38
- UsageSurveyResponseOption, page 41
- SearchException, page 43
- TSException, page 46
- ErrorCode, page 50

Reporting API

com.cisco.ts.scheduler.soap.rsoapis

Interface ReportingMgmt

All Superinterfaces

java.rmi.Remote

All Known Implementing Classes

ReportingMgmtSoapBindingStub

```
public interface ReportingMgmt
extends java.rmi.Remote
```

Method Summary

UsageSurveyQuestionnaire	getMeetingUsageSurveyQuestionaire()
MeetingRecord[]	searchMeetingRecordsByMeetingTime (java.util.Calendar in0, java.util.Calendar in1)

Method Detail

getMeetingUsageSurveyQuestionaire

```
UsageSurveyQuestionnaire getMeetingUsageSurveyQuestionaire()
```

Throws

```
java.rmi.RemoteException
APIDispatchException
SearchException
```

searchMeetingRecordsByMeetingTime

```
MeetingRecord[] searchMeetingRecordsByMeetingTime(java.util.Calendar in0,
                                                  java.util.Calendar in1)
```

Throws

```
java.rmi.RemoteException
APIDispatchException
SearchException
```

MeetingRecord

com.cisco.ts.scheduler.soap.rsoapis.api.object

Class MeetingRecord

```
java.lang.Object
  com.cisco.ts.scheduler.soap.rsoapis.api.object.MeetingRecord
```

All Implemented Interfaces

java.io.Serializable

```
public class MeetingRecord
extends java.lang.Object
implements java.io.Serializable
```

Method Summary

java.lang.String[]	getMCUNames() Gets the MCUNames value for this MeetingRecord.
java.util.Calendar	getMeetingActualEndTime() Gets the meetingActualEndTime value for this MeetingRecord.
java.util.Calendar	getMeetingActualStartTime() Gets the meetingActualStartTime value for this MeetingRecord.
java.util.Calendar	getMeetingEndTime() Gets the meetingEndTime value for this MeetingRecord.
java.lang.Integer[]	getMeetingFeatures() Gets the meetingFeatures value for this MeetingRecord.
java.util.Calendar	getMeetingStartTime() Gets the meetingStartTime value for this MeetingRecord.
java.lang.Integer	getMeetingState() Gets the meetingState value for this MeetingRecord.
java.lang.String	getMeetingSubject() Gets the meetingSubject value for this MeetingRecord.
java.lang.String	getMeetingSystemID() Gets the meetingSystemID value for this MeetingRecord.
java.lang.String	getOrganizerID() Gets the organizerID value for this MeetingRecord.
java.lang.String	getOrganizerTimezone() Gets the organizerTimezone value for this MeetingRecord.
java.lang.String[]	getParticipatedRoomNames() Gets the participatedRoomNames value for this MeetingRecord.
java.lang.String[]	getRoomNames() Gets the roomNames value for this MeetingRecord.
java.util.Calendar	getSurveyLastModifiedTime() Gets the surveyLastModifiedTime value for this MeetingRecord.
UsageSurveyResponse[]	getSurveyResponses() Gets the surveyResponses value for this MeetingRecord.

boolean	isNormal() Gets the normal value for this MeetingRecord.
boolean	isRecurringMeeting() Gets the recurringMeeting value for this MeetingRecord.

Methods Inherited From class java.lang.Object

getClass, notify, notifyAll, toString, wait, wait, wait

Note When using the CTS-Manager Reporting API to retrieve information about meetings scheduled with a TelePresence Server, Call Detail Record (CDR) information is not available.

Constructor Detail

MeetingRecord

```
public MeetingRecord()
```

MeetingRecord

```
public MeetingRecord(java.lang.String[] MCUNames,
                     java.util.Calendar meetingEndTime,
                     java.util.Calendar meetingActualEndTime,
 java.lang.Integer[] meetingFeatures,
                     java.util.Calendar meetingStartTime,
                     java.util.Calendar meetingActualStartTime,
                     java.lang.Integer meetingState,
                     java.lang.String meetingSubject,
                     java.lang.String meetingSystemID,
                     boolean normal,
                     java.lang.String organizerID,
                     java.lang.String organizerTimezone,
                     boolean recurringMeeting,
                     java.lang.String[] roomNames,
                     java.lang.String[] participatedRoomNames,
                     java.util.Calendar surveyLastModifiedTime,
                     UsageSurveyResponse[] surveyResponses)
```

Method Detail

getMCUNames

```
public java.lang.String[] getMCUNames()
```

Gets the MCUNames value for this MeetingRecord. The MCUNames value is the Conference Bridges configured for this meeting.

Returns

MCUNames/Conference Bridges

getMeetingEndTime

```
public java.util.Calendar getMeetingEndTime()
```

Gets the meetingEndTime value for this MeetingRecord.

Returns

meetingEndTime

getMeetingActualEndTime

```
public java.util.Calendar getMeetingActualEndTime()
```

Gets the meetingActualEndTime value for this MeetingRecord. This value is the time the meeting actually ended (only for past meetings).

Returns

meetingActualEndTime

getMeetingFeatures

```
public java.lang.Integer[] getMeetingFeatures()
```

Gets the Integer array of possible Meeting features for this MeetingRecord. Meetings can have more than one feature enabled simultaneously. If this array contains any of the possible Integer values, the corresponding feature was enabled for the meeting.

Possible Values

Meeting Feature	Integer Value	Description
INTEROP ENABLED	1	Interoperability with video conferencing is enabled for this meeting.
B2B MEETING AS HOST	2	Hosting Intercompany meeting.
B2B MEETING AS PARTICIPANT	4	Participating in Intercompany meeting.
RECORDING ENABLED	8	Studio Mode Recording is enabled for the meeting.
WEBEX ENABLED	16	Meeting includes WebEx.
WEBEX_PLUGIN	1024	Meeting scheduled through WebEx Productivity Tools plug-in for Microsoft Outlook.
MANUAL_CALLIN	4096	Meeting includes TelePresence Call-In Number for participants not included in the meeting invitation.

Example

To determine if a Cisco WebEx feature was enabled for a meeting, you can search the returned Integer array to see if the Integer Value of 16 is included. If this value is included, you can conclude that Cisco WebEx was enabled. If this value is not included, you can conclude that Cisco WebEx was not enabled.

Returns

meetingFeatures

getMeetingStartTime

```
public java.util.Calendar getMeetingStartTime()
```

Gets the meetingStartTime value for this MeetingRecord. This value is the scheduled start time of this meeting.

Returns

meetingStartTime

getMeetingActualStartTime

```
public java.util.Calendar getMeetingActualStartTime()
```

Gets the meetingActualStartTime value for this MeetingRecord. This value is the actual start time of this meeting (only for past meetings)

Returns

meetingActualStartTime

getMeetingState

```
public java.lang.Integer getMeetingState()
```

Gets the meetingState value for this MeetingRecord. This value indicates the meeting state when the request is made.

Possible Values

Meeting State	Integer Value	Description
SCHEDULED	0	Meeting is successfully scheduled.
ERRORED	1	There is a problem with the meeting.
COMPLETED	3	Meeting is completed.
INPROGRESS	4	Meeting is currently in progress.
NOT_OCCURRED	6	Meeting has not taken place yet.
PENDING	7	Meeting is not scheduled yet.
NOT_TELEPRESENCE	8	TelePresence is not used for this meeting.
ADHOC	9	Meeting is not a scheduled meeting.

Returns

meetingState

getMeetingSubject

```
public java.lang.String getMeetingSubject()
```

Gets the meetingSubject value for this MeetingRecord. This value is the meeting subject, which is set by the meeting organizer.

Returns

meetingSubject

getMeetingSystemID

```
public java.lang.String getMeetingSystemID()
```

Gets the meetingSystemID value for this MeetingRecord. This value is a globally unique identifier that is assigned to this meeting.

Returns

meetingSystemID

isNormal

```
public boolean isNormal()
```

A “normal meeting” is a meeting wherein the following explicit features are not used:

- INTEROP ENABLED,
- B2B_HOST,
- B2B_PARTICIPANT,
- RECORDING,
- WEBEX

This call returns a boolean value that indicates whether or not the meeting is normal.

Returns

normal

getOrganizerID

```
public java.lang.String getOrganizerID()
```

Gets the organizerID value for this MeetingRecord.

Returns

organizerID

getOrganizerTimezone

```
public java.lang.String getOrganizerTimezone()
```

Gets the organizerTimezone value for this MeetingRecord.

Returns

organizerTimezone

isRecurringMeeting

```
public boolean isRecurringMeeting()
```

Gets the recurringMeeting value for this MeetingRecord. This value indicates whether or not the meeting is set up as recurring.

Returns

recurringMeeting

getRoomNames

```
public java.lang.String[] getRoomNames()
```

Gets the roomNames value for this MeetingRecord. This value indicates the room names that were specified when the meeting was scheduled.

Returns

roomNames

getParticipatedRoomNames

```
public java.lang.String[] getParticipatedRoomNames()
```

Gets the participatedRoomNames value for this MeetingRecord. This value indicates the room names that actually participated in the meeting.

In the case of VC endpoints (non-TelePresence endpoints), the value returned by the system depends on how the end points were configured using CTS-Manager. If configured correctly, the call returns the room names. However, if configured incorrectly, the call returns only the dial-in number for the end points.

For information on how to correctly configure VC endpoints, see the *CTS Manager Administration and Installation Guide*, which you can access at the following location:

http://www.cisco.com/en/US/products/ps7074/prod_maintenance_guides_list.html

Returns

participatedRoomNames

getSurveyLastModifiedTime

```
public java.util.Calendar getSurveyLastModifiedTime()
```

Gets the surveyLastModifiedTime value for this MeetingRecord.

Returns

surveyLastModifiedTime

getSurveyResponses

```
public UsageSurveyResponse[] getSurveyResponses()
```

Gets the surveyResponses value for this MeetingRecord.

Returns

surveyResponses

UsageSurveyItem

com.cisco.ts.scheduler.soap.rsoapis.api.object

Class UsageSurveyItem

```
java.lang.Object
  com.cisco.ts.scheduler.soap.rsoapis.api.object.UsageSurveyItem
```

All Implemented Interfaces

java.io.Serializable

Direct Known Subclasses:

UsageSurveyQuestion, UsageSurveyResponseOption

```
public abstract class UsageSurveyItem
extends java.lang.Object
implements java.io.Serializable
```

Constructor Summary

UsageSurveyItem()
UsageSurveyItem(java.lang.String ID, boolean deleted)

Method Summary

static org.apache.axis.encoding. Deserializer	getDeserializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Deserializer.
java.lang.String	getID() Gets the ID value for the UsageSurveyItem.
static org.apache.axis.encoding. Serializer	getSerializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Serializer.
static org.apache.axis.description. TypeDesc	getTypeDesc() Gets the type metadata object.
int	hashCode()
boolean	isDeleted() Gets the deleted value for this UsageSurveyItem.

Methods Inherited From Class java.lang.Object

getClass, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

UsageSurveyItem

```
public UsageSurveyItem()
```

UsageSurveyItem

```
public UsageSurveyItem(java.lang.String ID,
                       boolean deleted)
```

Method Detail

getID

```
public java.lang.String getID()
```

Gets the ID value for this UsageSurveyItem.

Returns

ID

isDeleted

```
public boolean isDeleted()
```

Gets the deleted value for this UsageSurveyItem. The value indicates whether or not this survey item was deleted.

Returns

deleted

getTypeDesc

```
public static org.apache.axis.description.TypeDesc getTypeDesc()
```

Return the type metadata object.

getSerializer

```
public static org.apache.axis.encoding.Serializer
getSerializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Serializer.

getDeserializer

```
public static org.apache.axis.encoding.Deserializer
getDeserializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Deserializer.

UsageSurveyQuestion

com.cisco.ts.scheduler.soap.rsoapis.api.object

Class UsageSurveyQuestion

```
java.lang.Object
  com.cisco.ts.scheduler.soap.rsoapis.api.object.UsageSurveyItem
      com.cisco.ts.scheduler.soap.rsoapis.api.object.UsageSurveyQuestion
```

All Implemented Interfaces:

java.io.Serializable

```
public class UsageSurveyQuestion
extends UsageSurveyItem
implements java.io.Serializable
```

Constructor Summary

UsageSurveyQuestion()
UsageSurveyQuestion(java.lang.String ID, boolean deleted, java.lang.String displayText, boolean required, java.lang.String responseDataType, java.lang.String responseOptionType, UsageSurveyResponseOption[] responseOptions, java.lang.String responseValueType)

Method Summary

static org.apache.axis.encoding.Deserializer	getDeserializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Deserializer.
java.lang.String	getDisplayText() Gets the displayText value for the UsageSurveyQuestion.
java.lang.String	getResponseDataType() Gets the responseDataType value for the UsageSurveyQuestion.
UsageSurveyResponseOption[]	getResponseOptions() Gets the responseOptions value for the UsageSurveyQuestion.
java.lang.String	getResponseOptionType() Gets the responseOptionType value for the UsageSurveyQuestion.
java.lang.String	getResponseValueType() Gets the responseValueType value for this UsageSurveyQuestion.

static org.apache.axis.encoding.Serializer	getSerializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Serializer.
static org.apache.axis.description.TypeDesc	getTypeDesc() Gets the type metadata object.
boolean	isRequired() Gets the required value for the UsageSurveyQuestion.

Methods Inherited From Class com.cisco.ts.scheduler.soap.rsoapis.api.object.UsageSurveyItem

getID, isDeleted

Constructor Detail

UsageSurveyQuestion

```
public UsageSurveyQuestion()
```

UsageSurveyQuestion

```
public UsageSurveyQuestion(java.lang.String ID,
                           boolean deleted,
                           java.lang.String displayText,
                           boolean required,
                           java.lang.String responseDataType,
                           java.lang.String responseOptionType,
                           UsageSurveyResponseOption[] responseOptions,
                           java.lang.String responseValueType)
```

Method Detail

getDisplayText

```
public java.lang.String getDisplayText()
```

Gets the displayText value for the UsageSurveyQuestion.

Returns

displayText

isRequired

```
public boolean isRequired()
```

Gets the required value for the UsageSurveyQuestion. The system returns a response if the question is marked as required in the Usage Survey.

Returns

required

getResponseDataType

```
public java.lang.String getResponseDataType()
```

Gets the responseDataType value for the UsageSurveyQuestion.

Returns

responseDataType

getResponseOptionType

```
public java.lang.String getResponseOptionType()
```

Gets the responseOptionType value for the UsageSurveyQuestion.

Returns

responseOptionType

getResponseOptions

```
public UsageSurveyResponseOption[] getResponseOptions()
```

Gets the responseOptions value for the UsageSurveyQuestion.

Returns

responseOptions

getResponseValueType

```
public java.lang.String getResponseValueType()
```

Gets the responseValueType value for the UsageSurveyQuestion.

Returns

responseValueType

getTypeDesc

```
public static org.apache.axis.description.TypeDesc getTypeDesc()
```

Gets the type metadata object.

getSerializer

```
public static org.apache.axis.encoding.Serializer
getSerializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Serializer.

getDeserializer

```
public static org.apache.axis.encoding.Deserializer
getDeserializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Deserializer.

UsageSurveyQuestionnaire

com.cisco.ts.scheduler.soap.rsoapis.api.object

Class UsageSurveyQuestionnaire

```
java.lang.Object
  com.cisco.ts.scheduler.soap.rsoapis.api.object.UsageSurveyQuestionnaire
```

All Implemented Interfaces

java.io.Serializable

```
public class UsageSurveyQuestionnaire
extends java.lang.Object
implements java.io.Serializable
```

Constructor Summary

UsageSurveyQuestionnaire()
UsageSurveyQuestionnaire(java.util.Calendar lastModified, UsageSurveyQuestion[] questions)

Method Summary

static org.apache.axis.encoding.Deserializer	getDeserializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Deserializer.
java.util.Calendar	getLastModified() Gets the lastModified value for the UsageSurveyQuestionnaire.
UsageSurveyQuestion[]	getQuestions() Gets the questions value for the UsageSurveyQuestionnaire.
static org.apache.axis.encoding.Serializer	getSerializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Serializer.
static org.apache.axis.description.TypeDesc	getTypeDesc() Gets the type metadata object.

Constructor Detail

UsageSurveyQuestionnaire

```
public UsageSurveyQuestionnaire()
```

UsageSurveyQuestionnaire

```
public UsageSurveyQuestionnaire(java.util.Calendar lastModified,
                                UsageSurveyQuestion[] questions)
```

Method Details

getLastModified

```
public java.util.Calendar getLastModified()
```

Gets the lastModified value for the UsageSurveyQuestionnaire.

Returns

lastModified

getQuestions

```
public UsageSurveyQuestion[] getQuestions()
```

Gets the questions value for the UsageSurveyQuestionnaire.

Returns

questions

getTypeDesc

```
public static org.apache.axis.description.TypeDesc getTypeDesc()
```

Gets the type metadata object.

getSerializer

```
public static org.apache.axis.encoding.Serializer
getSerializer(java.lang.String mechType,

java.lang.Class _javaType,
javax.xml.namespace.QName _xmlType)
```

Gets the Custom Serializer.

getDeserializer

```
public static org.apache.axis.encoding.Deserializer
getDeserializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Deserializer.

UsageSurveyResponse

com.cisco.ts.scheduler.soap.rsoapis.api.object

Class UsageSurveyResponse

```
java.lang.Object
  com.cisco.ts.scheduler.soap.rsoapis.api.object.UsageSurveyResponse
```

All Implemented Interfaces

java.io.Serializable

```
public class UsageSurveyResponse
extends java.lang.Object
implements java.io.Serializable
```

Constructor Summary

UsageSurveyResponse()
UsageSurveyResponse(java.lang.String owningQuestionID, java.lang.Object value, java.lang.String valueType)

Method Summary

static org.apache.axis.encoding.Deserializer	getDeserializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Deserializer.
java.lang.String	getOwningQuestionID() Gets the owningQuestionID value for the UsageSurveyResponse.
static org.apache.axis.encoding.Serializer	getSerializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Serializer.
static org.apache.axis.description.TypeDesc	getTypeDesc() Gets the type metadata object.
java.lang.Object	getValue() Gets the value value for the UsageSurveyResponse.
java.lang.String	getValueType() Gets the valueType value for the UsageSurveyResponse.

Constructor Detail

UsageSurveyResponse

```
public UsageSurveyResponse()
```

UsageSurveyResponse

```
public UsageSurveyResponse(java.lang.String owningQuestionID,
                           java.lang.Object value,
                           java.lang.String valueType)
```

Method Detail

getOwningQuestionID

```
public java.lang.String getOwningQuestionID()
```

Gets the owningQuestionID value for the UsageSurveyResponse.

Returns

owningQuestionID

getValue

```
public java.lang.Object getValue()
```

Gets the value value for the UsageSurveyResponse.

Returns

value

getValueType

```
public java.lang.String getValueType()
```

Gets the valueType value for the UsageSurveyResponse.

Returns

valueType

getTypeDesc

```
public static org.apache.axis.description.TypeDesc getTypeDesc()
```

Gets the type metadata object.

getSerializer

```
public static org.apache.axis.encoding.Serializer
getSerializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Serializer.

getDeserializer

```
public static org.apache.axis.encoding.Deserializer
getDeserializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Deserializer.

UsageSurveyResponseOption

com.cisco.ts.scheduler.soap.rsoapis.api.object

Class UsageSurveyResponseOption

```
java.lang.Object
  com.cisco.ts.scheduler.soap.rsoapis.api.object.UsageSurveyItem
      com.cisco.ts.scheduler.soap.rsoapis.api.object.UsageSurveyResponseOption
```

All Implemented Interfaces

java.io.Serializable

```
public class UsageSurveyResponseOption
extends UsageSurveyItem
implements java.io.Serializable
```

Constructor Summary

UsageSurveyResponseOption()
UsageSurveyResponseOption(java.lang.String ID, boolean deleted, java.lang.String displayText)

Method Summary

static org.apache.axis.encoding.Deserializer	getDeserializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Deserializer.
java.lang.String	getDisplayText() Gets the displayText value for the UsageSurveyResponseOption.
static org.apache.axis.encoding.Serializer	getSerializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Serializer.
static org.apache.axis.description.TypeDesc	getTypeDesc() Gets the type metadata object.

Methods Inherited From Class com.cisco.ts.scheduler.soap.rsoapis.api.object.UsageSurveyItem

getID, isDeleted

Constructor Detail

UsageSurveyResponseOption

```
public UsageSurveyResponseOption()
```

UsageSurveyResponseOption

```
public UsageSurveyResponseOption(java.lang.String ID,
                                 boolean deleted,
                                 java.lang.String displayText)
```

Method Detail

getDisplayText

```
public java.lang.String getDisplayText()
```

Gets the displayText value for the UsageSurveyResponseOption.

Returns

displayText

getTypeDesc

```
public static org.apache.axis.description.TypeDesc getTypeDesc()
```

Gets the type metadata object.

getSerializer

```
public static org.apache.axis.encoding.Serializer
getSerializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Serializer.

getDeserializer

```
public static org.apache.axis.encoding.Deserializer
getDeserializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Deserializer.

SearchException

com.cisco.ts.scheduler.soap.rsoapis.exception

Class SearchException

```
java.lang.Object
  java.lang.Throwable
      java.lang.Exception
          java.io.IOException
              java.rmi.RemoteException
                  org.apache.axis.AxisFault
                      com.cisco.ts.scheduler.soap.rsoapis.exception.TSException
                          com.cisco.ts.scheduler.soap.rsoapis.exception.SearchException
```

All Implemented Interfaces

java.io.Serializable

```
public class SearchException
extends TSException
implements java.io.Serializable
```

Field Summary

Fields Inherited From class java.rmi.RemoteException

detail

Constructor Summary

SearchException()
SearchException(java.lang.String ID, java.lang.Object cause1, int code, ErrorCode errorCode, java.lang.Object[] erroredObjects, java.lang.String message1, java.lang.String module)

Method Summary

static org.apache.axis.encoding.Deserializer	getDeserializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Deserializer.
static org.apache.axis.encoding.Serializer	getSerializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Serializer.

static org.apache.axis.description.TypeDesc	getTypeDesc() Gets the type metadata object.
void	writeDetails(javax.xml.namespace.QName qname, org.apache.axis.encoding.SerializationContext context) Writes the exception data to the faultDetails.

Methods Inherited From Class com.cisco.ts.scheduler.soap.rsoapis.exception.TSException

getCause1, getCode, getErrorCode, getErroredObjects, getID, getMessage1, getModule

Constructor Detail

SearchException

```
public SearchException()
```

SearchException

```
public SearchException(java.lang.String ID,
                       java.lang.Object cause1,
                       int code,
                       ErrorCode errorCode,
                       java.lang.Object[] erroredObjects,
                       java.lang.String message1,
                       java.lang.String module)
```

Method Detail

getTypeDesc

```
public static org.apache.axis.description.TypeDesc getTypeDesc()
```

Gets the type metadata object.

getSerializer

```
public static org.apache.axis.encoding.Serializer
getSerializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Serializer.

getDeserializer

```
public static org.apache.axis.encoding.Deserializer
getDeserializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Deserializer.

writeDetails

```
public void writeDetails(javax.xml.namespace.QName qname,
                         org.apache.axis.encoding.SerializationContext context)
                  throws java.io.IOException
```

Writes the exception data to the faultDetails.

Overrides

`writeDetails` in class `TSException`

Throws

`java.io.IOException`

TSException

com.cisco.ts.scheduler.soap.rsoapis.exception

Class TSException

```
java.lang.Object
  java.lang.Throwable
      java.lang.Exception
          java.io.IOException
              java.rmi.RemoteException
                  org.apache.axis.AxisFault
                      com.cisco.ts.scheduler.soap.rsoapis.exception.TSException
```

All Implemented Interfaces

java.io.Serializable

Direct Known Subclasses

APIDispatchException, AuthenticationException, AuthorizationException, SearchException

```
public abstract class TSException
extends org.apache.axis.AxisFault
implements java.io.Serializable
```

Field Summary

Fields Inherited From Class java.rmi.RemoteException

detail

Constructor Summary

TSException()
TSException(java.lang.String ID, java.lang.Object cause1, int code, ErrorCode errorCode, java.lang.Object[] erroredObjects, java.lang.String message1, java.lang.String module)

Method Summary

java.lang.Object	getCause1() Gets the cause1 value for the TSException.
int	getCode() Gets the code value for the TSException.
static org.apache.axis.encoding.Deserializer	getDeserializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Deserializer.

ErrorCode	getErrorCode() Gets the errorCode value for the TSException.
java.lang.Object[]	getErroredObjects() Gets the erroredObjects value for the TSException.
java.lang.String	getID() Gets the ID value for the TSException.
java.lang.String	getMessage1() Gets the message1 value for the TSException.
java.lang.String	getModule() Gets the module value for the TSException.
static org.apache.axis.encoding.Serializer	getSerializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Serializer.
static org.apache.axis.description.TypeDesc	getTypeDesc() Gets the type metadata object.

Constructor Detail

TSException

```
public TSException()
```

TSException

```
public TSException(java.lang.String ID,
                   java.lang.Object cause1,
                   int code,
                   ErrorCode errorCode,
                   java.lang.Object[] erroredObjects,
                   java.lang.String message1,
                   java.lang.String module)
```

Method Detail

getID

```
public java.lang.String getID()
```

Gets the ID value for the TSException.

Returns

ID

getCause1

```
public java.lang.Object getCause1()
```

Gets the cause1 value for the TSException.

Returns

cause1

getCode

```
public int getCode()
```

Gets the code value for the TSException.

Returns

code

getErrorCode

```
public ErrorCode getErrorCode()
```

Gets the errorCode value for the TSException.

Returns

errorCode

getErroredObjects

```
public java.lang.Object[] getErroredObjects()
```

Gets the erroredObjects value for the TSException.

Returns

erroredObjects

getMessage1

```
public java.lang.String getMessage1()
```

Gets the message1 value for the TSException.

Returns

message1

getModule

```
public java.lang.String getModule()
```

Gets the module value for the TSException.

Returns

module

getTypeDesc

```
public static org.apache.axis.description.TypeDesc getTypeDesc()
```

Gets the type metadata object.

getSerializer

```
public static org.apache.axis.encoding.Serializer
getSerializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Serializer.

getDeserializer

```
public static org.apache.axis.encoding.Deserializer
getDeserializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Deserializer.

writeDetails

```
public void writeDetails(javax.xml.namespace.QName qname,
                         org.apache.axis.encoding.SerializationContext context)
                  throws java.io.IOException
```

Writes the exception data to the faultDetails.

Overrides

`writeDetails` in class `org.apache.axis.AxisFault`

Throws

`java.io.IOException`

ErrorCode

com.cisco.ts.scheduler.soap.rsoapis.exception

Class ErrorCode

```
java.lang.Object
  com.cisco.ts.scheduler.soap.rsoapis.exception.ErrorCode
```

All Implemented Interfaces

java.io.Serializable

```
public class ErrorCode
extends java.lang.Object
implements java.io.Serializable
```

Constructor Summary

ErrorCode()
ErrorCode(java.lang.String ID, int code, java.lang.String correctiveAction, java.lang.String description, java.lang.String module, java.lang.String rawMessage, java.lang.String severity)

Method Summary

int	getCode() Gets the code value for the ErrorCode.
java.lang.String	getCorrectiveAction() Gets the correctiveAction value for the ErrorCode.
java.lang.String	getDescription() Gets the description value for the ErrorCode.
static org.apache.axis.encoding.Deserializer	getDeserializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Gets the Custom Deserializer.
java.lang.String	getID() Gets the ID value for the ErrorCode.
java.lang.String	getModule() Gets the module value for the ErrorCode.
java.lang.String	getRawMessage() Gets the rawMessage value for the ErrorCode.
static org.apache.axis.encoding.Serializer	getSerializer(java.lang.String mechType, java.lang.Class _javaType, javax.xml.namespace.QName _xmlType) Get the Custom Serializer.

java.lang.String	getSeverity() Gets the severity value for the ErrorCode.
static org.apache.axis.description.TypeDesc	getTypeDesc() Gets the type metadata object.

Methods Inherited From Class java.lang.Object

getClass, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ErrorCode

```
public ErrorCode()
```

ErrorCode

```
public ErrorCode(java.lang.String ID,
                 int code,
                 java.lang.String correctiveAction,
                 java.lang.String description,
                 java.lang.String module,
                 java.lang.String rawMessage,
                 java.lang.String severity)
```

Method Detail

getID

```
public java.lang.String getID()
```

Gets the ID value for the ErrorCode.

Returns

ID

getCode

```
public int getCode()
```

Gets the code value for the ErrorCode.

Returns

code

getCorrectiveAction

```
public java.lang.String getCorrectiveAction()
```

Gets the correctiveAction value for the ErrorCode.

Returns
correctiveAction

getDescription

```
public java.lang.String getDescription()
```

Gets the description value for the ErrorCode.

Returns
description

getModule

```
public java.lang.String getModule()
```

Gets the module value for the ErrorCode.

Returns
module

getRawMessage

```
public java.lang.String getRawMessage()
```

Gets the rawMessage value for the ErrorCode.

Returns
rawMessage

getSeverity

```
public java.lang.String getSeverity()
```

Gets the severity value for the ErrorCode.

Returns
severity

getTypeDesc

```
public static org.apache.axis.description.TypeDesc getTypeDesc()
```

Gets the type metadata object.

getSerializer

```
public static org.apache.axis.encoding.Serializer
getSerializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Serializer.

getDeserializer

```
public static org.apache.axis.encoding.Deserializer
getDeserializer(java.lang.String mechType,

java.lang.Class _javaType,

javax.xml.namespace.QName _xmlType)
```

Gets the Custom Deserializer.

Troubleshooting

Table 3 provides a summary of the error codes you might receive when using the Reporting API. This table describes the errors and where to find more information about them.

Table 3 Error Code Summary

Error Code	Error ID	Description	Where to Find More Information
501615	UNAUTHORIZED_ERROR	User is not authorized to use the Reporting API.	Unauthorized Error, page 54
504100	EXCESSIVE_DATE_RANGE_ERROR	Date range for the requested meetings exceeds the supported value of 6 months.	Excessive Date Range Error, page 55
504101	TOO_MANY_ATTEMPT_ERROR	Too many APIs calls made in a given session.	Too Many Attempts Error, page 55
501000	INTERNAL_ERROR	CTS-Manager is unable to handle the request.	Internal Error, page 55

This section also describes some common problems that can occur when you are setting up the Reporting API client or using the Reporting API. For more information about these problems, including solutions, see the "Troubleshooting Common Problems" section on page 56.

Unauthorized Error

Symptom

Reporting API invocation fails and throws a search exception with the UNAUTHORIZED_ERROR error code.

Possible Cause

- A license required for the Reporting API is not detected in CTS-Manager.
- The user account for the Reporting API client is not authorized to use the Reporting API.
- The password for the Reporting API client user account is incorrect.

Solution

Depending on the suspected cause of the error, take the appropriate action:

- Work with the CTS-Manager administrator to ensure that the required license was purchased and uploaded. If not already uploaded, the CTS-Manager administrator must upload it to CTS-Manager via the Configure > Licenses page.
- Work with the LDAP administrator to ensure that a user group exists and that within this group, one or more user accounts for the Reporting API client exists. If these LDAP mechanisms are already in place, then work with the CTS-Manager administrator to ensure that the "Reporting API User" role is mapped to the LDAP user group that contains the user accounts. If not yet mapped, the CTS-Manager administrator must map the role to the group in CTS-Manager via the Configure > Access Management page.

- Work with the LDAP administrator to ensure that the LDAP group mapped to the Reporting API User role contains the user account that cannot be authorized and that this user has the proper password.

For complete information on the Configure > Licenses and Configure > Access Management pages, see the *CTS Manager Administration and Installation Guide*, which you can access at this location:

http://www.cisco.com/en/US/products/ps7074/prod_maintenance_guides_list.html

Excessive Date Range Error

Symptom

API invocation fails and throws a search exception with the EXCESSIVE_DATE_RANGE_ERROR error code.

Possible Cause

CTS-Manager refuses the client request if the start and end date range provided for a meeting query exceeds 6 months. A maximum range of 6 months is established to prevent spikes in the CPU usage of CTS-Manager.

Solution

Reduce the date range to less than 6 months, then re-submit your query.

Too Many Attempts Error

Symptom

API invocation fails and throws a search exception with the TOO_MANY_ATTEMPT_ERROR error code.

Possible Cause

CTS-Manager refuses a request if the Reporting API client is making calls too frequently to the Reporting API web service. A minimum interval of 5 minutes between two consecutive calls is established to prevent spikes in the CPU usage of CTS-Manager.

Solution

Wait 10 to 15 minutes, then re-submit your query.

Internal Error

Symptom

API invocation fails and throws a search exception with the INTERNAL_ERROR error code.

Possible Cause

There is an internal problem with the CTS-Manager server.

Solution

Wait 10 to 15 minutes, then re-submit your query. If the problem persists, ask the CTS-Manager administrator to restart the CTS-Manager server.

Troubleshooting Common Problems

Table 4 presents common problems that can occur when setting up the Reporting API client and using the Reporting API. It also presents solutions for each of the problems.

Note Before you open a support case, we strongly recommend reviewing Table 4 to determine if the problem you are experiencing appears in this table.

Table 4 Common Problems and Solutions

Problem	Who Is Responsible	Solution
Reporting API Client Setup		
URL for Reporting API client to access web service is incorrect.	Reporting API developer CTS-Manager administrator	Check to make sure the URL adheres to this format: http://*hostname*:*port*/rsoapis/services/ReportingMgmt For example: http://myCTSManager.mydomain.com:8080/rsoapis/services/ReportingMgmt
The ReportingCallbackHandler is not defined.	Reporting API developer	When using secure web services, this handler, which authenticates and authorizes the Reporting API client, is invoked. If not already done, define this handler. For information on implementing this handler, see the "Sample Code for Initializing the API" section on page 16.
A license required to access the Reporting API functionality does not exist, or the license is invalid.	CTS-Manager administrator	Work with the CTS-Manager administrator to ensure that this license is uploaded in CTS-Manager and that the license has the status of valid. (In the left navigation panel of the CTS-Manager GUI, the administrator can click Configure > Licenses to access the Licenses page.) For complete information on uploading and checking the status of the license, see the *CTS Manager Administration and Installation Guide* at this location: http://www.cisco.com/en/US/products/ps7074/prod_maintenance_guides_list.html

Table 4 Common Problems and Solutions (continued)

Problem	Who Is Responsible	Solution
The LDAP server does not include one or both of these authentication mechanisms for the Reporting API client: • A user group for the Reporting API client user accounts. • Within this user group, one or more user accounts for the Reporting API client.	LDAP administrator	Work with the LDAP administrator to ensure that these mechanisms are set up.
In CTS-Manager, the "Reporting API User" role is not mapped to the LDAP user group that contains the Reporting API client user accounts.	CTS-Manager administrator	Work with the CTS-Manager administrator to ensure that this mapping is created in the Access Management page. (In the left navigation panel of the CTS-Manager GUI, the administrator can click Configure > Access Management to access this page.) For complete information on mapping the role to the LDAP user group, see the *CTS Manager Administration and Installation Guide* at this location: http://www.cisco.com/en/US/products/ps7074/prod_maintenance_guides_list.html
The Reporting API client runs, but you notice that exceptions are being thrown.	Reporting API developer	To successfully access the Reporting API's web services, we recommend the use of the following libraries: • axis.jar (1.3 or higher) • commons-discovery-0.2.jar • commons-httpclient.jar • javax.wsdl_1.5.1 • jaxrpc.jar • log4j.jar • opensaml.jar • org.apache.commons.logging_1.0.4 • saaj.jar • wss4.jar • xmlsec.jar • JDK 1.5

Table 4 Common Problems and Solutions (continued)

Problem	Who Is Responsible	Solution
Reporting API Usage		
You notice discrepancies between the usage survey responses and the results captured by the Reporting API.	Reporting API developer	Typically, there is a delay between a meeting organizer responding to the usage survey and the running of the Reporting API. During this delay, the meeting organizer can change their responses. To avoid these discrepancies, you can do the following: • Impose a deadline by which meeting organizers must respond to the survey. • Run the Reporting API after the deadline for the last meeting.
Running the MeetingUsageSurvey-Questionnaire API does not return any results.	CTS-Manager administrator	Work with the CTS-Manager administrator to ensure that the optional Meeting Organizer Usage Survey and Benefits Report feature is enabled. (In the left navigation panel of the CTS-Manager GUI, the administrator can click Configure > Application Setting, then click the Usage Survey tab. In the page that appears, the administrator must select Yes for the Enable Meeting Organizer Usage Survey and Benefits Report feature.) For complete information on enabling this feature, see the *CTS Manager Administration and Installation Guide* at this location: http://www.cisco.com/en/US/products/ps7074/prod_maintenance_guides_list.html
The searchMeetingRecordsByMeetingTime (*calendar-start*, *calendar-end*) API returns an array of MeetingRecord objects.	Reporting API developer	Returning an array of MeetingRecord objects instead of a list is expected behavior for the searchMeetingRecordsByMeetingTime API. No action is required.
For interoperability calls enabled by Cisco Unified Video Conferencing (CUVC), the Reporting API does not retrieve a list of video conferencing (VC) room participants in the meeting. Instead, a number is shown as the participant.	Reporting API developer	In this situation, a CUVC conference number is shown as a participant. This anomaly is caused by a CUVC limitation and is outside the scope of CTS Manager.

Service and Support

Technical support for the Reporting API is available through the CDN to developers with a CDN support contract. Before implementing the Reporting API, we recommend that you join the CDN as a technology developer or a registered developer.

For information on available CDN support services, go to this location:

http://developer.cisco.com

At this location, developers with a CDN support contract can open a case when needed.

When opening a case, we recommend having the following information available:

- CTS-Manager version number
- Details about the issue you are experiencing
- Log files to help troubleshoot the issue
- A PC with Internet connectivity that can connect to the CTS-Manager server so that the technical support person can view your system

References

- *RFC 2445*, http://tools.ietf.org/html/rfc2445
- *RFC 2446*, http://tools.ietf.org/html/rfc2446

Glossary

The following acronyms and terms are used throughout this document:

- **API** application programming interface
- **CDN** Cisco Developer Network
- **CTS** Cisco TelePresence System
- **CTS-Manager** Cisco TelePresence Systems Manager
- **DoS** denial of service
- **GUI** graphical user interface
- **IDE** integrated development environment
- **JDK** Java Development Kit
- **LDAP** Lightweight Directory Access Protocol
- **SDK** software development kit
- **SOAP** Simple Object Access Protocol
- **WSDL** Web Services Description Language