



Videoscape Control Suite Operator Messaging Service Configuration Guide

Introduction

The Operator Messaging Service (OMS) is a high-availability and high-scalability application in the Cisco Videoscape Control Suite (VCS, formerly Conductor) group of services that enables service providers to easily preview, publish, search, and manage messages to targeted client devices. This guide provides instructions for configuring OMS.

Document Version

This is the fourth formal release of this document, which has been updated to reflect VCS 3.1 changes.

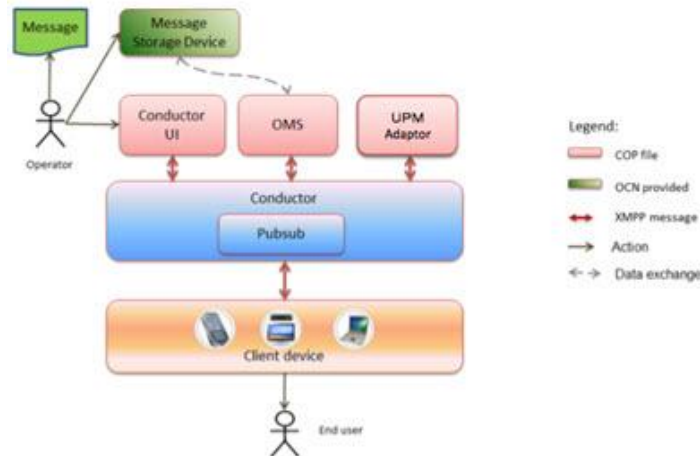
In This Document

■ Overview of OMS	2
■ Processing Messages	3
■ Back Up and Restore OMS Data	13
■ XMPP Interface for OMS	16
■ HTTP/RESTful Interface	23

Overview of OMS

The OMS is a VCS service that allows an operator to publish real time messages to a group of users or devices, or to a specific user or device. Examples of a message are business information updates, promotions, advertisements, emergency alerts, bills, and so on. A device can be a managed device, such as set-top box (STB), or an unmanaged device, such as a phone, tablet, or desktop PC.

OMS leverages the VCS Message Infrastructure to deliver the message encapsulated in the XMPP message stanza to the user or device. The following diagram shows the architecture diagram of the OMS.



Three services are connected with the VCS Message Infrastructure: VCS UI (CMC), Operator Messaging Service (OMS), and User Profile Manager Client Directory Adaptor (UPMCDA). OMS uses a VCS service call to retrieve account, user, and device information from the UPM Adaptor service. It then follows XMPP XEP-0060 protocol to call the Pubsub component to publish the message to a group of users/devices or to a specific user/device.

The OMS subscribes to the Pubsub node of the UPM Adaptor service to listen to the account-, user-, or device-create event to auto-subscribe the user or device into a messaging group, according to pre-defined grouping rules. OMS supports the rule defined by the customer during the OMS group creation.

A Message storage device is a third-party-provided storage that stores the message. The message creation may include several steps before uploading to this storage:

- The message can be created by multiple users and finally delivered to a central system to be submitted for approval.
- Operator A receives the audit request and reviews the message.
- Operator A uploads the message onto storage devices.
- Operator A publishes the message from the OMS UI.

Processing Messages

Import User and User Meta Data

The UPM Adaptor service provides the API to create, update, delete, or query user data and metadata. The UPM Adaptor service will synchronize the operation with the UPM service, which is the real data container. After the user is created, updated, or deleted, OMS receives the user account information, and then auto-subscribes (or unsubscribes) the user to/from the OMS group.

Generate UPM Adaptor Stub Code

A Web Services Description Language (WSDL) file (ClientDirectoryService.wsdl) is provided in the UPM Adaptor package, cisco.conductor-upmcda-3.x-x-xxxx.cop file. You can use the VCS Service SDK wsdl2stub.sh script to generate the client stub code.

Example:

```
/root/conductor-sdk4j-2.0.0-2758/bin/wsdl2stub.sh -p
com.cisco.conductor.clientdirectory.stub
ClientDirectoryService.wsdl
```

Create a Java project and add the stub code into your project to write client code to call these APIs.

Create the Client Proxy Factory

Before you can add a user into the client directory, you need an instance of ClientProxyFactory present to provide the service name, virtual service namespace, version, or service port type for an instance of ServiceRefConfiguration, and set the instance to the ClientProxyFactory instance.

Example:

```
ServiceRefConfiguration sc = new ServiceRefConfiguration();
sc.setName("ClientDirectoryService");
sc.setVns("conductor://clientdirectory");
sc.setVersion("1.0.0");
sc.setType("com.cisco.conductor.oms.stub.ClientDirectoryServicePortType");

ClientProxyFactory factory = new ClientProxyFactory();
factory.setXmppConnector(xc);
factory.setServiceRefConfiguration(sc);

// create a ClientDirectoryServicePortType instance
client = (ClientDirectoryServicePortType) factory.create();
client.initMethod();
```

Add User and User Metadata

Use the client instance to create a user in the client directory database. To create such a user, you need to provide information such as JID, Display Name, Email, and Phone Number. You can use the user ID (uid), as a string type, as a unique identifier to identify the user. The uid parameter is equal to the user name.

Example: `client.createUser(aid, uid, u, jidpassword)`

Notes:

- **aid** – An account belonging to this user
- **uid** – User name
- **u** – Instance of UserCommonData that contains all user information
- **jidpassword** – Used to import the user into the ClientDirectory and to add the user to the Videoscape Control Suite

Note: To add the user, a JID and jidpassword are required. The JID is a unique identifier used by the system to identify the user.

After you create a user, you also need to provide the metadata for this user.

Example: `Client.addMetaData(MetaDataType.USER, uid, namespace, list)`

Notes:

- **namespace** – Namespace of metadata that must equal the namespace configured during OMS installation
- **List** – Key value pair in the metadata that is used in OMS when creating the rule parameter and the group rule

Example:

```
List<MetaDataElement> list = new ArrayList<MetaDataElement>();
MetaDataElement e1 = new MetaDataElement();
e1.setName("city");
e1.setValue("shanghai");
e1.setType("String");
list.add(e1);
```

Result:

	Group Name	Create Time	Rules
1	group01	2012/08/20 17:05:43	city=shanghai;

Implementation:

```

try {
    u = new UserCommonData();
    u.setJid(uid + '@' + this.getUserDomain());
    u.setDisplayName(uid);
    u.setEmail(uid + '@' + this.getUserDomain());
    u.setPhoneNum(String.valueOf(rnd1.nextInt(1000)));
    u.setRole("dummy");
    u.setPriority(3);
    u.setOperateStatus(UserStatus.INACTIVE);

    boolean isExisted = this.isUserExstiedInCD(list, uid);
    if (isExisted != true) {
        client.createUser(aid, uid, u, jidPassword);
        logger.info("user is existed in client directory!");
    }

    if (list != null) {
        client.addMetaData(MetaDataType.USER, uid, namespace, list);
    }
} catch (ConductorException e) {
    logger.error("fail to create user into clientdirectory...", e);
    throw e;
}

```

After you add the user, this user is automatically subscribed into the group. You can use this function to add several users.

Note: If a user is imported into the Client Directory using this "create" interface, the user is added into the database automatically as a JID.

Delete User and User Metadata

Be certain that you delete the user metadata before deleting the existing user.

Example: `Client.deleteMetaData(MetaDataType.USER, uid, namespace, result)`

Notes:

- The first three parameters are identical to the parameters used in the `client.createUser` operation.
- The result is a `List<string>` with a spliced rules name of the OMS group to which the user subscribed.

`List<MetaDataElement>`

`rules=client.getMetaData(MetaDataType.USER, uid, namespace, null)`

Example:

```

List<MetaDataElement> rules = client.getMetaData(MetaDataType.USER,
    uid, namespace, null);

List<String> result = new ArrayList<String>();
for (MetaDataElement meta : rules) {
    result.add(meta.getName());
}

String jid = this.getJid(uid);
logger.info("delete user with iid is " + jid);
client.deleteMetaData(MetaDataType.USER, uid, namespace, result);
client.deleteUser(uid);

```

The Workflow of OMS

Create a Rule Parameter

- 1 Log on to the management console.
- 2 Choose **Services > Operator Messaging System > Rule Parameter**.



- 3 Choose **Add**. The Add Rule Parameter window opens.
- 4 Use the accompanying example to configure the parameter.

A screenshot of the "Add Rule Parameter" configuration window. The window has a title bar and contains several input fields: "Name" with the value "upm.filters.userProfile.metadata.com_cisco_oms.city", "Description" with the value "user's city description", "Type" set to "List", and "Value List" with the value "BEIJING;SHANGHAI;SAN JOSE;ATLANTA;JERUSALEM". There are question mark icons next to the Name, Description, and Value List fields. At the bottom, there are "Save", "Reset", and "Cancel" buttons.

Notes:

- Click the question mark for acceptable entries for each field.
- At any time, you may choose **Reset** or **Cancel**. Reset clears the fields. Cancel returns you to the previous screen without saving changes.
- **Name** – Enter the name of the parameter you are creating.
- **Description** – Enter a brief description of the parameter.
- **Type** – Choose String or List. A string can be defined by a user. A list will force a user to select from a list of options.
 - **Value List** – If the Type field is set to List, enter the values that are acceptable for this parameter.
 - **Default Value** – If the Type field is set to String, enter a default value for this parameter.

5 Choose **Save**.

Note: The name of the new Rule Parameter must match the rule metadata in the corresponding namespace of the Client Directory.

Result: The new Rule Parameter is listed in the Rule Parameter List.

	Parameter Name	Description	Type	Value
1	upm.filters.household.region		List	sh;SH;shanghai;beijing;sichuan;chongqing;hubei;jiangs
2	upm.filters.userProfile.displayName		String	displayName
3	upm.filters.household.metadata.com_cisco_oms.accountkey		String	
4	upm.filters.household.metadata.com_cisco_oms.accountkey		String	
5	upm.filters.userProfile.metadata.com_cisco_oms.userkey1		String	
6	upm.filters.userProfile.metadata.com_cisco_oms.userkey2		String	
7	upm.filters.userProfile.metadata.com_cisco_oms.city	user's city description	List	BEIJING;SHANGHAI;SAN JOSE;ATLANTA;JERUSALEM

Create a Rule

- 1 Log on to the management console.
- 2 Choose **Services > Operator Messaging System > Group Rules**.
- 3 Choose **Add**.
- 4 Use the accompanying example to configure the Add Rule window.

Add Rule

Name * : shanghai.city

Description

Parameter * : upm.filters.userProfile.metadata.com_cisco_oms.city

Operation * : Equal To

Parameter Type * : List

Value * : SHANGHAI

Save Reset Cancel

Notes:

- Click the question mark for acceptable entries for each field.
- At any time, you may choose **Reset** or **Cancel**. Reset clears the fields. Cancel returns you to the previous screen without saving changes.
- **Name** – Enter the name of the parameter you are creating.
- **Description** – Enter a brief description of the parameter.
- **Parameter** – Choose the parameter you want to associate with this group rule.
- **Operation** – Choose the operation that this rule will invoke.
- **Parameter Type** – Choose String or List. A string can be defined by a user. A list will force a user to select from a list of options.
- **Value** – If the Parameter Type field is set to List, choose the value that you want to be the default. If the Parameter Type field is set to String, the predefined default appears.

Processing Messages

- 5 Choose **Save**. The new rule is listed in the Group Rule List.

Name	Parameter	Operation	Value	Parameter Type	Description
shanghai_city	upm.filters.userProfile.metadata.com_cisco_oms.city	EqualTo	SHANGHAI	List	

Create a Group

- 1 Log on to the management console.
- 2 Choose **Services > Operator Messaging System > Groups**.
- 3 Choose **Add**.
- 4 Use the accompanying example to configure the Add OMS Group window.

Name: shanghai_users

Description:

Available Rules:

- name_cisco_oms_filters.metadata.com_cisco_oms.city=SHANGHAI
- hub_region:upm.filters.metadata.region=huber
- user_key1_value2:upm.filters.userProfile.metadata.com_cisco_oms.userkey1=value2
- user_key1_value1:upm.filters.userProfile.metadata.com_cisco_oms.userkey1=value1
- user_value2_key1:upm.filters.userProfile.metadata.com_cisco_oms.userkey2=value1
- user_value2_key2:upm.filters.userProfile.metadata.com_cisco_oms.userkey2=value2

Selected Rules:

- shanghai_city:upm.filters.userProfile.metadata.com_cisco_oms.city=SHANGHAI

Buttons: Save, Reset, Cancel, Add, Remove

Notes:

- Click the question mark for acceptable entries for each field.
 - At any time, you may choose **Reset** or **Cancel**. Reset will clear the fields. Cancel will return you to the previous screen without saving changes.
- **Name** – Enter the name of the group you are creating.
 - **Description** – Enter a brief description of the group.
- 5 Use **Add** and **Remove** to create a list of Selected Rules that you want to apply to this group.
 - 6 Choose **Save** to add the rules parameter. The rule parameter appears in the Rule Parameter List. The group is created. OMS subscribes existing users with matching criteria to the new group.

Group Name	Create Time	Rules	Status	
shanghai_users	2013/09/29 16:43:2	shanghai_city:upm.filters.userProfile.metadata.com_cisco_oms.city=SHANGHAI	Normal	<input type="checkbox"/>

- 7 Choose **List Subscriber** to list the subscribers in this group.

	User ID	User Name	Parameters
1	<input type="radio"/> account_sh4_1_user_1	account_sh4_1_user_1	upm.filters.household.region=shanghai;upm.filters.userProfile.displayName
2	<input type="radio"/> account_sh4_1_user_2	account_sh4_1_user_2	upm.filters.household.region=shanghai;upm.filters.userProfile.displayName

Note: You can also select a specific user for reviewing detailed parameter information.

Subscriber List[Group Name: shanghai_users]

	User ID	User Name	Parameters
1	<input checked="" type="radio"/> account_sh4_1_user_1	account_sh4_1_user_1	upm.filters.household
2	<input type="radio"/> account_sh4_1_user_2	account_sh4_1_user_2	upm.filters.household

Detailed Parameter Information

```

upm.filters.household.region=shanghai;
upm.filters.userProfile.displayName=account_sh4_1_user_1;
upm.filters.userProfile.metadata.com_cisco_oms.city=SHANGHAI;
upm.filters.household.address=address;
upm.filters.household.createTime=2013-09-29T08:48:37Z;
upm.filters.household.lastModifiedTime=2013-09-29T08:49:28Z;
upm.filters.household.operateStatus=INACTIVE;
upm.filters.userProfile.createTime=2013-09-29T08:48:42Z;
upm.filters.userProfile.jid=account_sh4_1_user_1@client.vcs.com;
upm.filters.userProfile.lastLoginTime=-1;
upm.filters.userProfile.lastModifyTime=2013-09-29T08:49:28Z;
upm.filters.userProfile.operateStatus=INACTIVE;
upm.filters.userProfile.presenceStatus=OFFLINE;
upm.filters.userProfile.priority=0;

```

Process Messages

List Messages

- 1 Log on to the management console.
- 2 Choose **Services > Operator Messaging System > Messages**. The system lists all message files on the FTP server.

Message List

	File Name	Last Modify Time	Size(Byte)
1	<input type="radio"/> chinese.xml	2012/08/08 12:01:00	421
2	<input type="radio"/> chinese_msg.xml	2012/08/21 00:00:00	485
3	<input type="radio"/> chinese_msg2.xml	2012/08/08 12:01:00	426
4	<input type="radio"/> chinese_msg3.xml	2012/08/09 15:15:00	517

Processing Messages

Preview Messages

- 1 Click to highlight a message in the Message List.

Message List

Preview Publish to Group Send to User

	File Name	Last Modify Time	Size(Byte)
2	<input checked="" type="radio"/> chinese_msg.xml	2012/08/21 00:00:00	485
3	<input type="radio"/> chinese_msg2.xml	2012/08/08 12:01:00	426
4	<input type="radio"/> chinese_msg3.xml	2012/08/09 15:15:00	517
5	<input type="radio"/> ex am_ple.xml	2012/08/13 21:34:00	410
6	<input type="radio"/> ex_ ample.xml	2012/08/13 21:36:00	410
7	<input type="radio"/> example.xml	2012/08/13 13:12:00	410

- 2 Choose **Preview**.

Message Preview : chinese_msg.xml
• Valid Message Format !

```
<oms xmlns='http://conductor.cisco.com/oms/1.0'>
  <sender>pubsubadmin@oms.ocn.com</sender>
  <senderName>发送者</senderName>
  <expireTime>2012-12-17T14:57:00+08:00</expireTime>
  <priority>3</priority>
  <type>Promotion</type>
  <scope>Public</scope>
  <title>这是一个中文标题</title>
  <description>
    这是一段中文的内容。。。。。。。With English。最后还是中文。
  </description>
  <payload>
    这是一段中文的payload。
  </payload>
</oms>
```

Publish Messages

- 1 Click to highlight a message in the Message List.

Message List


Preview Publish to Group Send to User

	File Name	Last Modify Time	Size(Byte)
2	<input checked="" type="radio"/> chinese_msg.xml	2012/08/21 00:00:00	485
3	<input type="radio"/> chinese_msg2.xml	2012/08/08 12:01:00	426
4	<input type="radio"/> chinese_msg3.xml	2012/08/09 15:15:00	517
5	<input type="radio"/> ex am_ple.xml	2012/08/13 21:34:00	410
6	<input type="radio"/> ex_ ample.xml	2012/08/13 21:36:00	410
7	<input type="radio"/> example.xml	2012/08/13 13:12:00	410

- 2 Choose **Publish to Group**. The Group List window opens.

- Click to highlight a message.

Group List






	Group Name	Create Time	Rules
1	<input checked="" type="radio"/> group01	2012/08/20 17:05:43	city=shanghai;
2	<input type="radio"/> group02	2012/08/20 21:47:47	city=beijing;

- Choose **Send**. A notification window should open that indicates that the message published successfully.

Send Messages

- Click to highlight a message in the Message List.

Message List

	File Name	Last Modify Time	Size(Byte)
2	<input checked="" type="radio"/> chinese_msg.xml	2012/08/21 00:00:00	485
3	<input type="radio"/> chinese_msg2.xml	2012/08/08 12:01:00	426
4	<input type="radio"/> chinese_msg3.xml	2012/08/09 15:15:00	517
5	<input type="radio"/> ex_ample.xml	2012/08/13 21:34:00	410
6	<input type="radio"/> ex_ample.xml	2012/08/13 21:36:00	410
7	<input type="radio"/> example.xml	2012/08/13 13:12:00	410

- Choose **Send to User**. A User ID window opens.

Please input the user ID to send to

User ID: 

- Type the **User ID** of the user to whom you want the message sent and then choose **Send**.

Please input the user ID to send to

User ID: 

Result: A notification window should open that indicates that the message was sent successfully to the user.

Processing Messages

View Historical Messages

- 1 Log on to the management console.
- 2 Choose **Services > Operator Messaging System > Historic Messages**. The system lists all historical messages, sorted by **Send Time**.

Historic Message List

	Send Time	Result	Title	Receiver
<input type="radio"/>	2012/08/22 10:13:30	Success		group01_user1...
<input type="radio"/>	2012/08/22 10:09:07	Success		group01
<input type="radio"/>	2012/08/21 16:05:53	Success	这是一个中文标题	group02
<input type="radio"/>	2012/08/20 22:00:44	Success	A title of message	group01
<input type="radio"/>	2012/08/20 20:03:48	Success		group01
<input type="radio"/>	2012/08/20 16:56:53	Success	这是一个中文标题	group02

- 3 Click to highlight a message. The Message Information window opens for the selected message.

Message Information

File Name:

chinese_msg.xml

Description:

这是一段中文的内容。 。 。 。 。 。 With English。

Payload:

这是一段中文的payload。

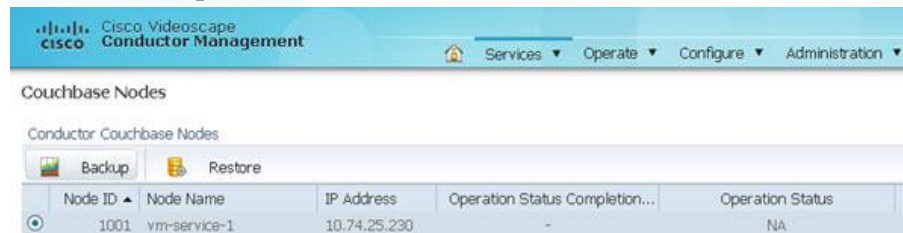
Back Up and Restore OMS Data

OMS stores data in both NOSQLCB and VCS Msginfra Postgres/Oracle format. To back up or restore OMS data, you must complete the following operations in order:

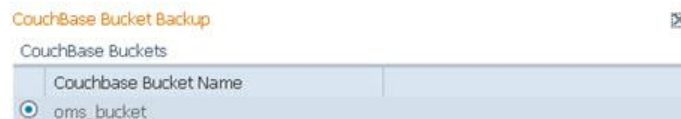
- 1 Back up NOSQLCB data from the CMC.
- 2 Back up Msginfra data stored in the Postgres/Oracle database.
- 3 Restore Msginfra data stored in the Postgres/Oracle database.
- 4 Restore NOSQLCB data from the CMC.

Back Up NOSQLCB Data

- 1 Log on to the management console.
- 2 Choose **Services > nosqlcb > Operations**.
- 3 Select the Couchbase node that you want to back up.
- 4 Choose **Backup**.



- 5 Click **oms_bucket**.
- 6 Choose **Backup**.



Back Up Msginfra Data

Back Up Msginfra Data Stored in Postgres

- 1 Log into the CLI of the Postgres database node.
- 2 From the CLI, enter these commands to back up the Msginfra database (msginfra/platform):

```
database transfer backup msginfra root@host:/root
database transfer backup platform root@host:/root
```

Back Up Msginfra Data Stored in Oracle

An external Oracle utility is used to back up and restore Msginfra data stored in Oracle. Contact Oracle for the procedures.

Restore Msginfra Data

Restore Msginfra Data Stored in Postgres

- 1 Log into the CLI of the Postgres database node.
- 2 From the CLI, enter these commands to restore the Msginfra database (msginfra/platform) from CLI:

```
database transfer restore msginfra
root@host:/root/msginfra_file_name
database transfer restore platform
root@host:/root/platform_file_name
```

Restore Msginfra Data Stored in Oracle

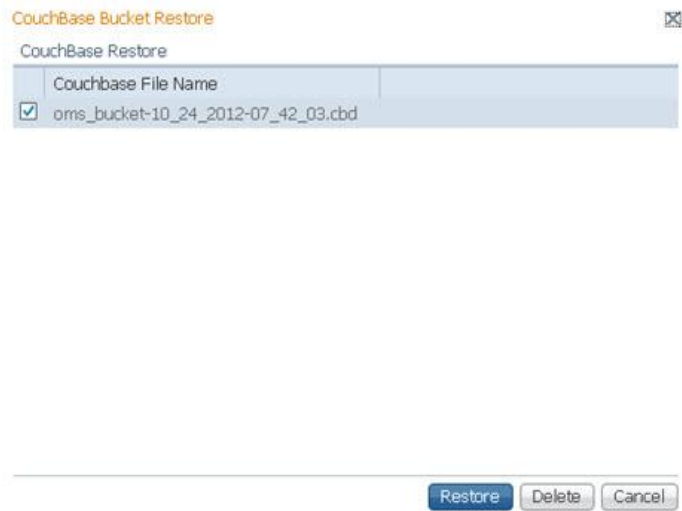
An external Oracle utility is used to back up and restore Msginfra data stored in Oracle. Contact Oracle for the procedures.

Restore NOSQLCB Data

- 1 Log on to the management console.
- 2 Choose **Services > nosqlcb > Operations**.
- 3 Select the Couchbase node that you want to restore.
- 4 Choose **Restore**.

- 5 Select the bucket that you want to use for the restoration.

Note: In the *Back Up NOSQLCB Data* (on page 13) procedure, we used the oms_bucket.



XMPP Interface for OMS

OMS Message Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            elementFormDefault="qualified"

targetNamespace="http://conductor.cisco.com/oms/1.0"
            xmlns:oms="http://conductor.cisco.com/oms/1.0">
  <xs:simpleType name="NameType">
    <xs:restriction base="xs:string">
      <xs:maxLength value="1023" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="DateType">
    <xs:restriction base="xs:dateTime">
      <xs:pattern value="\d\d\d\d-\d\d-\d\dT\d\d:\d\d:\d\d[-
,+]\d\d:\d\d"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="PriorityType">
    <xs:restriction base="xs:integer">
      <xs:pattern value="[0-9]"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="TitleType">
    <xs:restriction base="xs:string">
      <xs:maxLength value="511" />
    </xs:restriction>
  </xs:simpleType>

```



```

<xs:simpleType name="DescriptionType">
  <xs:restriction base="xs:string">
    <xs:maxLength value="4095" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="PayloadType">
  <xs:restriction base="xs:string">
    <xs:maxLength value="4095" />
  </xs:restriction>
</xs:simpleType>

<xs:element name="oms">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="sender" type="oms:NameType"
minOccurs="1" maxOccurs="1"/>
      <xs:element name="senderName" type="oms:NameType"
minOccurs="1" maxOccurs="1"/>
      <xs:element name="expireTime" type="oms:DateType"
minOccurs="1" maxOccurs="1"/>
      <xs:element name="priority" type="oms:PriorityType"
minOccurs="1" maxOccurs="1"/>
      <xs:element name="type" type="xs:NCName" minOccurs="0"
maxOccurs="1"/>
      <xs:element name="scope" type="xs:NCName"
minOccurs="0" maxOccurs="1"/>
      <xs:element name="title" type="oms:TitleType"
minOccurs="1" maxOccurs="1"/>
      <xs:element name="description"
type="oms:DescriptionType" minOccurs="1" maxOccurs="1"/>
      <xs:element name="payload" type="oms:PayloadType"
minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

OMS Sample Message

```
<oms xmlns='http://conductor.cisco.com/oms/1.0'>
  <sender>pubsubadmin@oms.ocn.com</sender>
  <senderName>User1</senderName>
  <expireTime>2012-12-17T14:57:00+08:00</expireTime>
  <priority>3</priority>
  <type>Promotion</type>
  <scope>Public</scope>
  <title>A title of message</title>
  <description>
    XML encoded content.
  </description>
  <payload>
    XML encoded payload.
  </payload>
</oms>
```

Notes:

- **sender: senderJID** – The OMS validates the JID format in the code.
- **senderName** – The name of the sender that the client can display on the screen.
- **expireTime** – ISO 8601 format, UTC + timezone. The client will not display the message if it has expired.
- **priority** – 0 through 9. 0 is the highest; 9 is the lowest.
- **type** – Reserved for the future. The default type is the same as the Pubsub node name.
- **scope** – Reserved for Test, Public, Restricted, Private, and so on.
- **title** – Maximum allowed is 511 bytes.
- **description** – Maximum allowed is 4095 bytes. Binary format is not supported.
- **payload** – Maximum allowed is 4095 bytes. Binary format is not supported.

OMS Service Interface

A WSDL file (OMSService.wsdl) is provided to use Videoscape Control Suite's Client SDK to write a client to call the OMS APIs. You can use Videoscape Control Suite's Service SDK wsdl2stub.sh script to generate the client stub code.

```
/root/conductor-sdk4j-2.0.0-2758/bin/wsdl2stub.sh -p
com.cisco.conductor.oms.stub OMSService.wsdl
```

Create a Java project and add the stub code into your project to write the client to call these APIs.

@Service

```
public interface OMSService {
    List<OmsGroup> listGroups();
    void publishMessage(String groupName, PreviewMessage
message);
    void sendMessage(String userId, PreviewMessage message);
}
```

The declaration of class of PreviewMessage is:

```
public class PreviewMessage {
    public String getFileName() {
        return fileName;
    }

    public void setFileName(String fileName) {
        this.fileName = fileName;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public String getReason() {
        return reason;
    }

    public void setReason(String reason) {
        this.reason = reason;
    }
}
```

XMPP Interface for OMS

```
    }  
    public boolean isValidated() {  
        return isValidated;  
    }  
    public void setValidated(boolean isValidated) {  
        this.isValidated = isValidated;  
    }  
    private String fileName = null;  
    private String message = null;  
    private String reason = null;  
    private boolean isValidated = false;  
};
```

Invoke the OMS API

A sample piece of code that invokes the XMPP interface follows:

```
PreviewMessage message = new PreviewMessage();  
String payload = "<oms  
xmlns='http://conductor.cisco.com/oms/1.0'>"  
    + "<sender>pubsubadmin@oms.ocn.com</sender>"  
    + "<senderName>User1</senderName>"  
    + "<expireTime>2012-12-  
17T14:57:00+08:00</expireTime>"  
    + "<priority>3</priority>" +  
"<type>Promotion</type>"  
    + "<scope>Public</scope>" + "<title>A title of  
message</title>"  
    + "<description>" + "    XML encoded content."  
    + "</description>" + "<payload>" + "    XML encoded  
payload."  
    + "</payload>" + "</oms>";  
message.setFileName("example.xml");  
message.setMessage(payload);  
message.setValidated(true);  
  
ServiceRefConfiguration sc = new ServiceRefConfiguration();  
sc.setName("OMSService");
```

```
sc.setVsns("conductor://com.cisco.conductor.oms");
sc.setType("com.cisco.conductor.oms.stub.OMSServicePort");
```

```
ClientProxyFactory factory = new ClientProxyFactory();
factory.setXmppConnector(xc);
factory.setServiceRefConfiguration(sc);
```

```
client = (OMSServicePortType) factory.create();
oms.publishMessage(grpName, message);
oms.sendMessage(uid, message);
```

Publish Message Example

In the following example, the first parameter is the OMS group name, and the second parameter is the message object that contains the payload of message. The payload can be in raw XML format, as shown in this example.

```
void publishMessage(String groupName, PreviewMessage message);
```

The following is a sample message that can be received by a client:

```
<message from='pubsub.oms.conductor.com'
        to='user1@oms.conductor.com'>
  <event xmlns="http://jabber.org/protocol/pubsub#event">
    <items node="Promotion">
      <item id='ae890ac52d0df67ed7cfd51b644e901'>
        <oms xmlns='http://conductor.cisco.com/oms/1.0'>
          <sender>pubsubadmin@oms.conductor.com</sender>
            <senderName>User1</senderName>
            <expireTime>2012-12-
17T14:57:00+08:00</expireTime>
            <priority>3</priority>
            <type>Promotion</type>
            <scope>Test</scope>
            <title>A title of message</title>
            <description>
              Content.
            </description>
            <payload>
```

```

                Payload.
            </payload>
        </oms>
    </item>
</items>
</event>
</message>

```

Send Message Example

In the following example, the first parameter is the User ID of the Videoscape Control Suite user, and the second parameter is the message object that contains the payload of the message. The payload can be in raw XML format, as shown in this example.

```
void sendMessage(String userId, PreviewMessage message);
```

The following is a sample message that can be received by a client.

```

<message from='oms_10.74.25.231@svc.conductor.com'
        to='user1@oms.conductor.com'>
    <oms xmlns='http://conductor.cisco.com/oms/1.0'>
        <sender>pubsubadmin@oms.conductor.com</sender>
        <senderName>User1</senderName>
        <expireTime>2012-12-17T14:57:00+08:00</expireTime>
        <priority>3</priority>
        <type>Promotion</type>
        <scope>Public</scope>
        <title>A title of message</title>
        <description>
            Content.
        </description>
        <payload>
            Payload.
        </payload>
    </oms>
</message>

```

HTTP/RESTful Interface

Set Up the HTTP Gateway

- 1 Log on to the management console.
- 2 Choose **Message Infrastructure**.
- 3 Select one service node and then choose one service router.

The screenshot shows the Cisco Conductor Management interface. The top navigation bar includes 'Services', 'Operate', 'Configure', 'Administration', and 'Message Infrastructure'. The main content area is titled 'Infrastructure Management' and contains two sections: 'Node List' and 'Router List'.

Node List

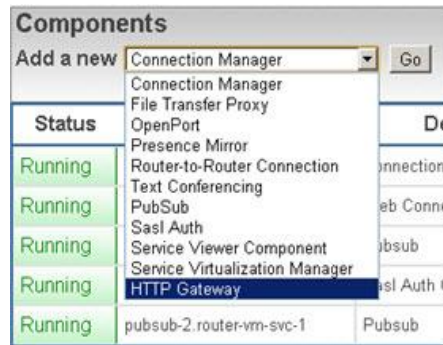
Node Name	Node State	Node Type
<input checked="" type="radio"/> vm-svc-1	Running	Service
<input type="radio"/> vm-mgmt-1	Running	Management

Router List

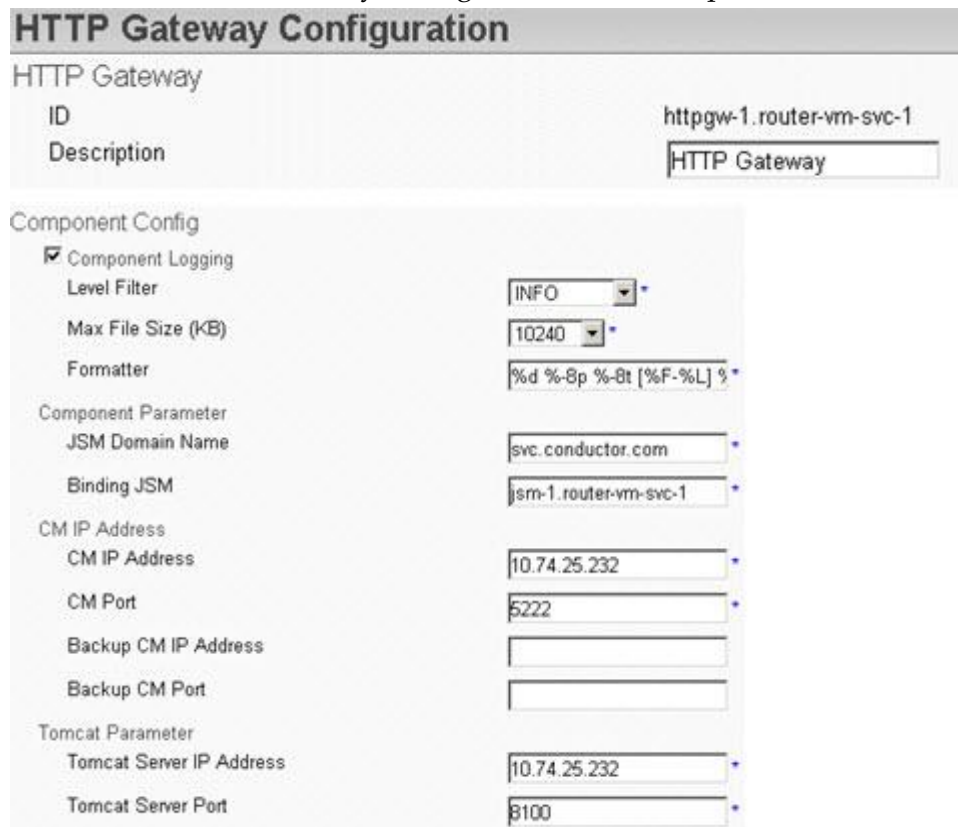
Router Realm	Router State	Router Failure Re...	Router Type	HA State	HA Switch Mode	Router IP	Router Port	Administrat
<input type="radio"/> boot-vm-svc-1	Running		BOOT			10.74.25.232	7400	10.74.25.21
<input checked="" type="radio"/> router-vm-svc-1	Running		NONE-HA			10.74.25.232	7401	10.74.25.21
<input type="radio"/> router-vm-svc-2	Running		NONE-HA			10.74.25.232	7402	10.74.25.21

- 4 Choose **XCP Web Controller**. The XCP Web Controller banner appears.
- 5 Choose **Enter the XCP Controller now**.

- 6 From the Components List that appears, select **HTTP Gateway** and choose **Go**.



Result: The HTTP Gateway Configuration window opens.



- 7 Use the following information to configure the HTTP Gateway Configuration window. What follows are the mandatory fields that the system requires you to populate.

- **JSM Domain Name** – The hostname of the JSM service. This JSM is used for virtual services.
- **Binding JSM** – The ID of the JSM service.
- **CM IP Address** – The IP address of the Connection Manager in your cluster.
- **Tomcat Server IP Address** – The IP address that exposes the RESTful interface. The HTTP client will use this address to call the RESTful interface.

Note: Additional information is available in *Operator Messaging Service (OMS) API Guide* (part number OL-30021).

OMS Restrictions

Message File

- The timezone of the message file server should be identical to the timezone of the server where OMS is installed.
- The name of the message file created by the operator must be in English.
- The encoding format of the message file must be UTF-8. The file header should not contain BOM.
Note: You can use utilities Ultraedit or Notepad++ to convert to 'UTF-8 without BOM header'.
- The **ExpireTime** field in the message file must be in ISO-8601 format. The format should follow the ISO-8601 standard: yyyy-MM-dd'T'HH:mm:ssZ.
- The fields that permit Chinese characters are: **senderName**, **title**, **description**, and **payload**.

Chinese Support

- The fields that permit Chinese characters are: value of rule parameters, rule value, descriptions for rule parameters, rules, and groups.
- The fields that disallow Chinese characters are: name of rule parameters, rules, and groups. The User ID, imported by the Client Directory, also does not support Chinese, such as '中文User01@cisco.com'.

OMS UI

- The OMS UI does not support the back/forward button on the browser. It provides its own button to support the back/forward operation.
- The OMS UI does not support the refresh button on the browser. It provides its own button to support the refresh operation.
- The display time format is UTC. The system can automatically adjust the time display format. For China, the display format is "yyyy/mm/dd hh:mm:ss".
- The listSubscribe operation only supports sorting the subscriber by User ID.
- The OMS UI will display "no data available" in a table when the OMS service returns an error or a request timeout. If the table shows "no data available", confirm whether there is an internal error or if there really is no data to return.
 - Check the OMS service status in the Message Infrastructure page – **Service Instance Management**.
 - Check the NOSQLCB status via the Couchbase Management page.
 - If the Message page shows "no data available", check the FTP server status.

OMS Rules

- An OMS rule cannot be deleted if it is in use by any OMS group.
- An OMS rule parameter cannot be deleted if it is in use by any OMS rule.

OMS Service

- The OMS configures itself to use hash from algorithms for service-load balance. The algorithm must not change after installation.
- The OMS service can process a Pubsub notification, such as a user add/update/delete event from the Client Directory, even if the OMS service instance is disabled.

Parameter Name

- The name of a parameter depends upon the filters defined in the UPM service for OMS. If a name is not defined in the UPM service, groups using the parameter name will display an error status.

Number of Parameters

- The number of parameters supported by OMS depends upon how many dedicated keys of the UPM service are used for OMS. The default number is 2.

UPM Restrictions

OMS depends upon the UPM service for retrieving user data. If you want to upgrade to a newer version of the UPM service, you must make sure that the UPM configuration file has not changed. If your UPM configuration file will change, back up the UPM configuration file before upgrading and then apply the changed items to the UPM configuration file after the upgrade.

NOSQLCB Restrictions

The OMS stores data in the NOSQLCB server. The following is a list of restrictions that pertain to the NOSQLCB server.

- The auto-failover feature is only available when one cluster has at least three NOSQLCB servers.
- The auto-failover feature does not support a double-failure scenario. For example, in a three-node cluster, if one server is down, the two remaining servers are the minimum configuration required for high availability. If an additional node fails, data-loss can occur.
- After auto-failover, the operator must rebalance the cluster from the UI.

Example: Server Nodes >Pending Rebalance

- Key values:
 - The maximum length is 250. Names for each table are limited to 250 characters.
Notes:
 - The Java SDK will throw an exception (report an error) if more than 250 characters are used.
 - Chinese characters are not permitted.
 - The following key characters can be used:
String key = +abcdefghijklmnopqrstuvxyz0123456789-=!@#%&^*();!:'?><
- The NOSQLCB database cannot recover if two nodes fail at the same time.

For Information

If You Have Questions

If you have technical questions, contact Cisco Services for assistance. Follow the menu options to speak with a service engineer.



Americas Headquarters

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

<http://www.cisco.com>

Tel: 408 526-4000

800 553-6387

Fax: 408 527-0883

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL:

www.cisco.com/go/trademarks.

Third party trademarks mentioned are the property of their respective owners.

The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Product and service availability are subject to change without notice.

© 2012, 2013 Cisco and/or its affiliates. All rights reserved.

November 2013

Part Number

OL-28090-04