Videoscape Control Suite Unified Notification Gateway (UNG) User Guide

Overview

Introduction

The Cisco Unified Notification Gateway (UNG) is an application built upon the Cisco Videoscape Control Suite platform. It provides a unified interface for applications to publish notifications to all kinds of devices without regard to device types and their online/offline status.

Document Version

This is the first formal release of this document.

In This Document

- Overview of the UNG 2
- XMPP Interface 4
- UNG Management via the CMC 7
- Application Management 13
- Known Issues and Limitations 14
- UNG Web Services Description Language 15

Overview of the UNG

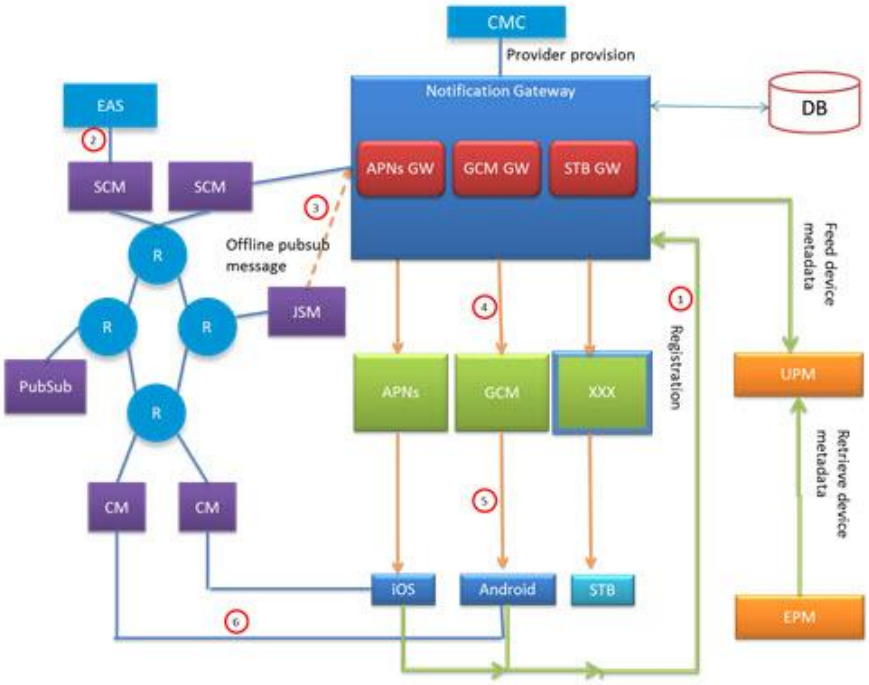
The UNG is a Videoscape Control Suite service that allows the operator to publish real time message notifications to a group of users or devices, or to a specific user or device. Examples of the types of messages include business information, promotions, advertisements, emergency alerts, and bills. The type of devices include managed devices, such as a set-top box (STB), or unmanaged devices, such as phones, tablets, desktop PCs, and so on.

For iOS and Android devices, UNG can leverage the existing Apple Push Notification Service (APN), as well as the Google Cloud Messaging (GCM) service to push notifications to the device. Both the APN and GCM services have well-defined interfaces to allow pushing rich and custom notifications to a device.

The UNG provides the following features:

- A unified service interface for messages that will be sent to applications as notifications
- Well-defined interfaces to extend support to different types of devices easily, such as iOS, Android, or legacy one-way STBs. It also contains the extension to support iOS devices.
- The interface and Conductor Management Console (CMC) UI to customize the payload features of a notification message
- The interface for devices to control the acceptance (register/unregister) of the notifications

Workflow of the UNG Service



XMPP Interface

A Web Services Description Language (WSDL) file, named NotificationService.swdl, is provided so that the client SDK can be used to write client calls to the UNG service APIs. You can use the SDK wsdl2stub.sh script to generate the client stub code.

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

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

Interface Summary

Return	Method	Description
NotificationProviderMetadata	**getNotificationProvider** (String appID)	Retrieve a provider by specifying the application's ID
Void	**provisionProvider** (NotificationProviderMetadata provider)	Provision a provider
Void	**modifyProvider** (NotificationProviderMetadata provider)	Modify a notification provider
String	**removeProvider** (String appID)	Remove a notification provider according to the application's ID
ProviderPagination	**getNotificationProviderForPage** (int currentpage, int perpage)	Paging query of interface of provider
PreferrencePagination	**getProviderPreferenceForPage** (String appId,int currentpage, int perpage)	Paging query for all preferences of a provider by specifying the provider's application ID
Void	**provisionPreference** (ProviderPreferenceMetadata preference)	Provision a preference
ProviderPreference	**getProviderPreference** (String name)	Retrieve a preference by specifying the preference's name
Void	**modifyPreference** (ProviderPreference preference)	Modify a provider preference
Void	**removePreference** (String name)	Remove a provider preference according to the reference's name

Class Summaries

Class NotificationProviderMetadata

Type	Name	Description
String	appId	The application's ID
String	certification	The certification string used to authenticate the provider; it should be encoded with base64
String	type	The type of devices to which this provider intends to send notifications
String	passwd	Password string used to authenticate the provider

Class ProviderPreferenceMetadata

Type	Name	Description
String	element	Element name in the payload which needs to send notification
String	name	The unique name that identifies this preference
String	namespace	Namespace property value of the notification provider element
Map<String, String>	payload	A map contains all the properties used to customize the notification, such as the notification title, sound, and so on. The properties' names are fixed as shown: **1** message The fixed notification title **2** sound The name of the sound file played when the notification is received on a device **3** xpath The xpath string used to parse the notification title from the payload of the pubsub message

Class ProviderPagination

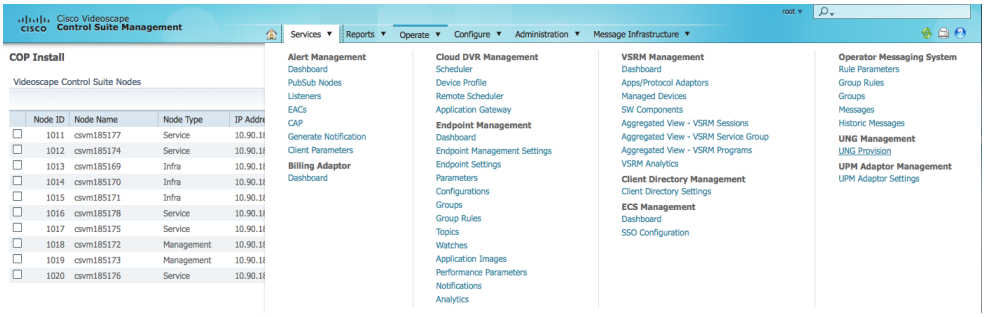
Type	Name	Description
List<NotificationProvider Metadata>	providerList	The list of providers on the current page found by specific criteria
int	totalNum	The number of providers found by specific criteria

PreferencePagination

Type	Name	Description
List<NotificationPrerence Metadata>	preferenceList	The list of preferences on the current page found by specific criteria
int	totalNum	The number of providers found by specific criteria

UNG Management via the CMC

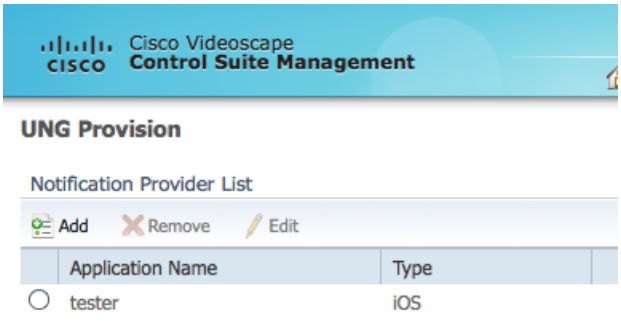
This service of the CMC allows the operator to manage and configure the UNG.



Provider Management

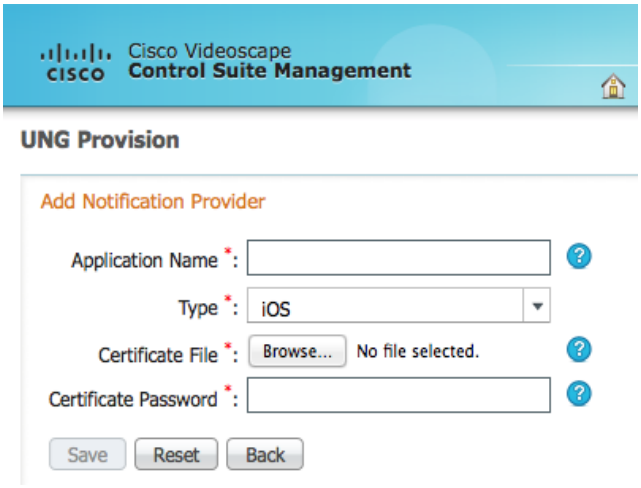
- **List all providers**

  Choose **Service > UNG Management > UNG Provision** to list all providers.



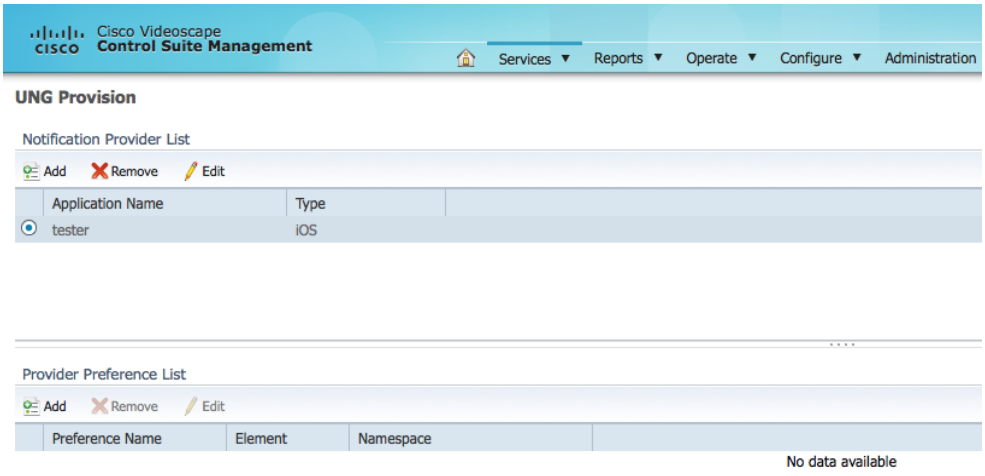
- **Add a provider**

  Click **Add** to add one notification provider for an application. The operator must enter the **Application Name**, choose a **Type** (currently, only iOS is supported), choose a **Certification File** (in .p12 format), and enter the **Certificate Password**. Click **Save**.

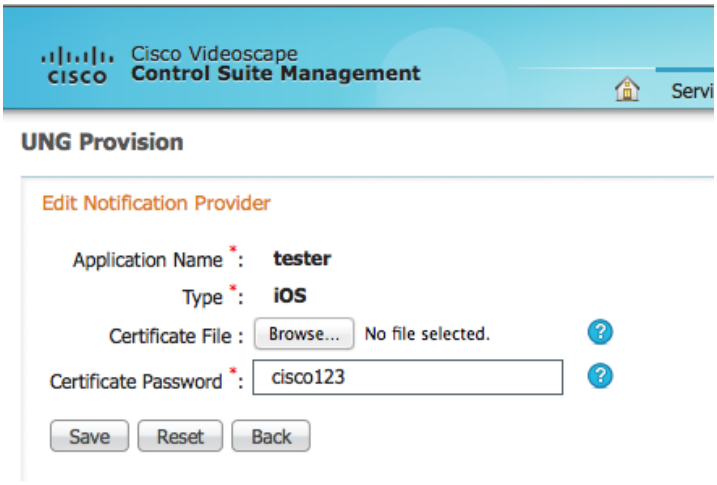


- **Edit a provider**

  Select a provider and click **Edit**.



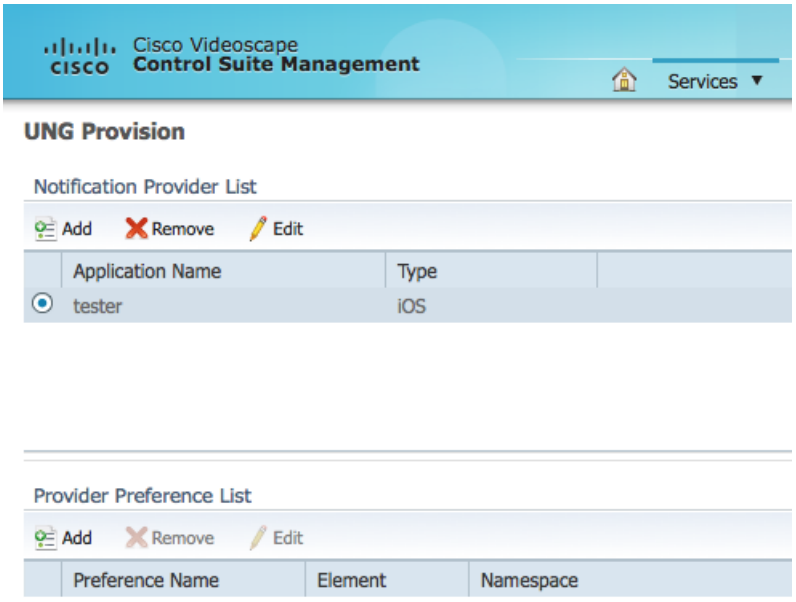
Only the **Certificate File** and the **Certificate Password** can be modified. Note that the **Certificate File** is not mandatory. If not used, the original file is used.



- **Delete a provider**

  Choose one notification provider and click **Remove**. Click **OK** in the confirmation dialog box.

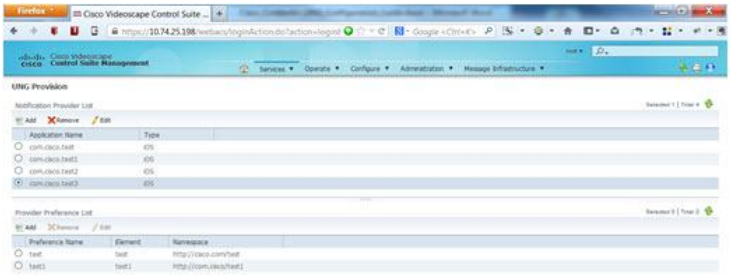
  Note: All preferences belonging to the provider are also deleted.



Preference Management

- **List all preferences**

  Choose a provider and the preferences for that provider are displayed in the Provider Preference List at the bottom of the page.

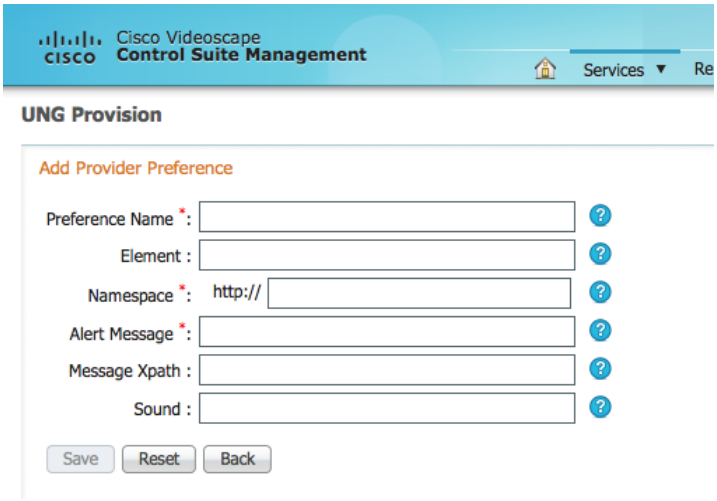


- **Add preferences**

  An operator can add one or many preferences for a provider. Preferences have the following attributes.

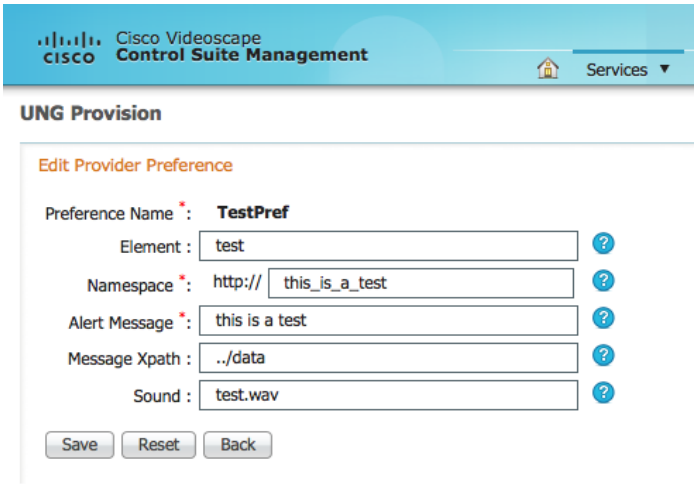
  - **Name** — The preference name is mandatory. It must be globally unique within the system.
  - **Element** — The root element name of the message's payload. It is used to identify the messages with the Namespace. (Optional)
  - **Namespace** — The root element's namespace. (Mandatory) The Element and Namespace combination of one provider should be unique.
  - **Alert Message** — The default message content displayed in the notification to the user. (Mandatory)
  - **Sound** — The name of a sound file in the application bundle. (Optional) If not used, the default sound is used.
  - **Message Xpath** — An Xpath string used to fetch the notification message content from the original message. It is useful when displaying different message texts, rather than a fixed message content. (Optional)

Note: If both the **Alert Message** and the **Message Xpath** are supplied, the system uses the Xpath first to fetch the notification message. If that operation fails, then the **Alert Message** content is used.

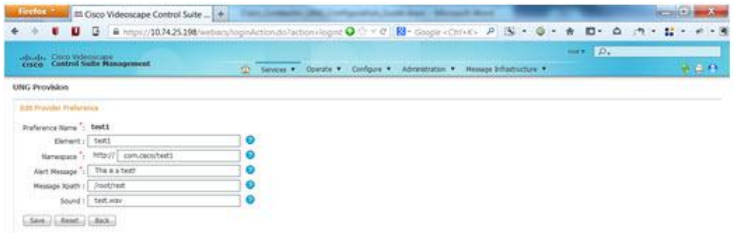


- **Modify a preference**

  Choose a Preference and click **Edit**.

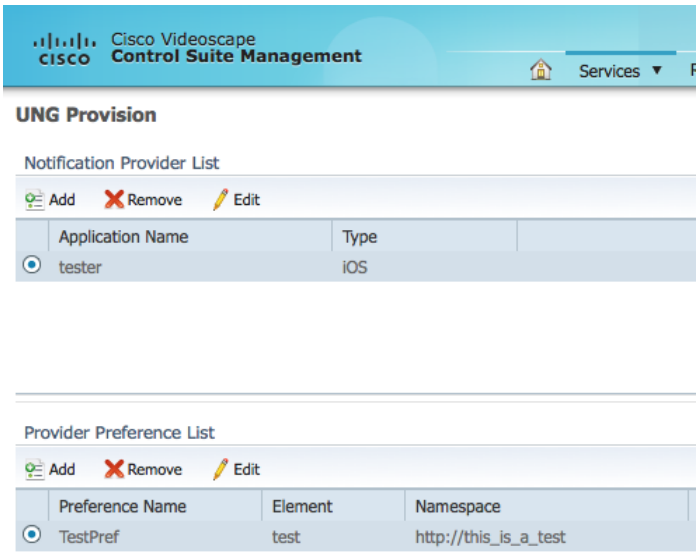


Modify the Preference accordingly and click **Save**.



- **Delete a preference**

  Choose a preference and click **Remove**. Click **OK** in the confirmation dialog box.



Application Management

For application management, two interfaces are provided that enable/disable offline message notifications for your device.

Interface Summary

Return	Method	Description
void	register(AppMetadata metadata)	Register in UNG for enabling notification delivery
void	deregister(AppMetadata metadata)	Deregister an application in UNG for disabling notification delivery

Class AppMetadata

Field Summary

Type	Name	Description
String	jid	The JID that is using this application
String	devToken	The device's token, which is the unique identity of this device
String	devType	The device's type
String	id	The application's ID which must be consistent with the provider's application name

Notes:

- The application developer should call the 'register' interface when the customer wants to book offline notification messages from the Videoscape Control Suite.
- The application developer should call the 'deregister' interface when the customer is uninterested in notification messages.

Known Issues and Limitations

Currently, only iOS devices are supported by the UNG.

UNG Web Services Description Language

```
<?xml version="1.0" encoding="UTF-8"?>

<wsdl:definitions name="NotificationService"
targetNamespace="http://service.ung.conductor.cisco.com/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://service.ung.conductor.cisco.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

  <wsdl:types>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://service.ung.conductor.cisco.com/"
attributeFormDefault="unqualified"
elementFormDefault="unqualified"
targetNamespace="http://service.ung.conductor.cisco.com/">

  <xs:complexType name="providerPagination">

    <xs:sequence>

      <xs:element maxOccurs="unbounded" minOccurs="0"
name="providerList" nillable="true"
type="tns:notificationProviderMetadata"/>

      <xs:element name="totalNum" type="xs:int"/>

    </xs:sequence>

  </xs:complexType>

  <xs:complexType name="notificationProviderMetadata">

    <xs:sequence>

      <xs:element minOccurs="0" name="appId"
type="xs:string"/>

      <xs:element minOccurs="0" name="certification"
type="xs:string"/>

      <xs:element name="index" type="xs:int"/>

      <xs:element minOccurs="0" name="passwd"
type="xs:string"/>

      <xs:element minOccurs="0" name="type" type="xs:string"/>

    </xs:sequence>

  </xs:complexType>

  <xs:complexType name="providerPreferenceMetadata">

    <xs:sequence>

      <xs:element minOccurs="0" name="element"
type="xs:string"/>
```

```
      <xs:element minOccurs="0" name="name" type="xs:string"/>
      <xs:element minOccurs="0" name="namespace"
type="xs:string"/>
      <xs:element minOccurs="0" name="payload"
type="tns:payloadMapType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="payloadMapType">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0"
name="entry" nillable="true" type="tns:payloadMapEntryType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="payloadMapEntryType">
    <xs:sequence>
      <xs:element minOccurs="0" name="key" type="xs:string"/>
      <xs:element minOccurs="0" name="value"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="preferencePagination">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0"
name="preferenceList" nillable="true"
type="tns:providerPreferenceMetadata"/>
      <xs:element name="totalNum" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="appMetadata">
    <xs:sequence>
      <xs:element minOccurs="0" name="appId"
type="xs:string"/>
      <xs:element minOccurs="0" name="deviceToken"
type="xs:string"/>
      <xs:element minOccurs="0" name="deviceType"
type="xs:string"/>
      <xs:element minOccurs="0" name="jid" type="xs:string"/>
    </xs:sequence>
```

```
  </xs:complexType>
  <xs:complexType name="notificationMessage">
    <xs:sequence>
      <xs:element minOccurs="0" name="message"
type="xs:string"/>
      <xs:element minOccurs="0" name="to" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="getNotificationProviderForPage"
type="tns:getNotificationProviderForPage"/>
  <xs:complexType name="getNotificationProviderForPage">
    <xs:sequence>
      <xs:element name="arg0" type="xs:int"/>
      <xs:element name="arg1" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="getNotificationProviderForPageResponse"
type="tns:getNotificationProviderForPageResponse"/>
  <xs:complexType
name="getNotificationProviderForPageResponse">
    <xs:sequence>
      <xs:element minOccurs="0" name="return"
type="tns:providerPagination"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="getNotificationProvider"
type="tns:getNotificationProvider"/>
  <xs:complexType name="getNotificationProvider">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="getNotificationProviderResponse"
type="tns:getNotificationProviderResponse"/>
  <xs:complexType name="getNotificationProviderResponse">
    <xs:sequence>
```

```
      <xs:element minOccurs="0" name="return"
type="tns:notificationProviderMetadata"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="getProviderPreference"
type="tns:getProviderPreference"/>
  <xs:complexType name="getProviderPreference">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="getProviderPreferenceResponse"
type="tns:getProviderPreferenceResponse"/>
  <xs:complexType name="getProviderPreferenceResponse">
    <xs:sequence>
      <xs:element minOccurs="0" name="return"
type="tns:providerPreferenceMetadata"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="modifyProvider"
type="tns:modifyProvider"/>
  <xs:complexType name="modifyProvider">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0"
type="tns:notificationProviderMetadata"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="modifyProviderResponse"
type="tns:modifyProviderResponse"/>
  <xs:complexType name="modifyProviderResponse">
    <xs:sequence/>
  </xs:complexType>
  <xs:element name="getPreference" type="tns:getPreference"/>
  <xs:complexType name="getPreference">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0" type="xs:string"/>
    </xs:sequence>
```

```
  </xs:complexType>
  <xs:element name="getPreferenceResponse"
type="tns:getPreferenceResponse"/>
  <xs:complexType name="getPreferenceResponse">
    <xs:sequence>
      <xs:element minOccurs="0" name="return"
type="tns:providerPreferenceMetadata"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="getProviderPreferenceForPage"
type="tns:getProviderPreferenceForPage"/>
  <xs:complexType name="getProviderPreferenceForPage">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0" type="xs:string"/>
      <xs:element name="arg1" type="xs:int"/>
      <xs:element name="arg2" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="getProviderPreferenceForPageResponse"
type="tns:getProviderPreferenceForPageResponse"/>
  <xs:complexType name="getProviderPreferenceForPageResponse">
    <xs:sequence>
      <xs:element minOccurs="0" name="return"
type="tns:preferencePagination"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="register" type="tns:register"/>
  <xs:complexType name="register">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0"
type="tns:appMetadata"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="registerResponse"
type="tns:registerResponse"/>
  <xs:complexType name="registerResponse">
    <xs:sequence/>
```

```
  </xs:complexType>
  <xs:element name="provisionPreference"
type="tns:provisionPreference"/>
  <xs:complexType name="provisionPreference">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0"
type="tns:providerPreferenceMetadata"/>
      <xs:element minOccurs="0" name="arg1" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="provisionPreferenceResponse"
type="tns:provisionPreferenceResponse"/>
  <xs:complexType name="provisionPreferenceResponse">
    <xs:sequence/>
  </xs:complexType>
  <xs:element name="deRegister" type="tns:deRegister"/>
  <xs:complexType name="deRegister">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0"
type="tns:appMetadata"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="deRegisterResponse"
type="tns:deRegisterResponse"/>
  <xs:complexType name="deRegisterResponse">
    <xs:sequence/>
  </xs:complexType>
  <xs:element name="onMessage" type="tns:onMessage"/>
  <xs:complexType name="onMessage">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0"
type="tns:notificationMessage"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="onMessageResponse"
type="tns:onMessageResponse"/>
  <xs:complexType name="onMessageResponse">
```

```
    <xs:sequence/>
  </xs:complexType>
  <xs:element name="provisionProvider"
type="tns:provisionProvider"/>
  <xs:complexType name="provisionProvider">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0"
type="tns:notificationProviderMetadata"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="provisionProviderResponse"
type="tns:provisionProviderResponse"/>
  <xs:complexType name="provisionProviderResponse">
    <xs:sequence/>
  </xs:complexType>
  <xs:element name="removePreference"
type="tns:removePreference"/>
  <xs:complexType name="removePreference">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="removePreferenceResponse"
type="tns:removePreferenceResponse"/>
  <xs:complexType name="removePreferenceResponse">
    <xs:sequence/>
  </xs:complexType>
  <xs:element name="removeProvider"
type="tns:removeProvider"/>
  <xs:complexType name="removeProvider">
    <xs:sequence>
      <xs:element minOccurs="0" name="arg0" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="removeProviderResponse"
type="tns:removeProviderResponse"/>
  <xs:complexType name="removeProviderResponse">
```

```
      <xs:sequence/>
    </xs:complexType>
    <xs:element name="modifyPreference"
type="tns:modifyPreference"/>
    <xs:complexType name="modifyPreference">
      <xs:sequence>
        <xs:element minOccurs="0" name="arg0"
type="tns:providerPreferenceMetadata"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="modifyPreferenceResponse"
type="tns:modifyPreferenceResponse"/>
    <xs:complexType name="modifyPreferenceResponse">
      <xs:sequence/>
    </xs:complexType>
    <xs:element name="getProviderIndex"
type="tns:getProviderIndex"/>
    <xs:complexType name="getProviderIndex">
      <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="getProviderIndexResponse"
type="tns:getProviderIndexResponse"/>
    <xs:complexType name="getProviderIndexResponse">
      <xs:sequence>
        <xs:element name="return" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
</xs:schema>
    </wsdl:types>
    <wsdl:message name="registerResponse">
      <wsdl:part name="parameters"
element="tns:registerResponse">
      </wsdl:part>
    </wsdl:message>
    <wsdl:message name="provisionProvider">
```

```
    <wsdl:part name="parameters"
element="tns:provisionProvider">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="removePreference">
    <wsdl:part name="parameters"
element="tns:removePreference">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getNotificationProviderForPageResponse">
    <wsdl:part name="parameters"
element="tns:getNotificationProviderForPageResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getProviderPreference">
    <wsdl:part name="parameters"
element="tns:getProviderPreference">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getPreference">
    <wsdl:part name="parameters" element="tns:getPreference">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getPreferenceResponse">
    <wsdl:part name="parameters"
element="tns:getPreferenceResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getProviderIndex">
    <wsdl:part name="parameters"
element="tns:getProviderIndex">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="removeProviderResponse">
    <wsdl:part name="parameters"
element="tns:removeProviderResponse">
    </wsdl:part>
```

```
    </wsdl:message>
    <wsdl:message name="deRegister">
      <wsdl:part name="parameters" element="tns:deRegister">
      </wsdl:part>
    </wsdl:message>
    <wsdl:message name="deRegisterResponse">
      <wsdl:part name="parameters"
element="tns:deRegisterResponse">
      </wsdl:part>
    </wsdl:message>
    <wsdl:message name="removeProvider">
      <wsdl:part name="parameters" element="tns:removeProvider">
      </wsdl:part>
    </wsdl:message>
    <wsdl:message name="onMessage">
      <wsdl:part name="parameters" element="tns:onMessage">
      </wsdl:part>
    </wsdl:message>
    <wsdl:message name="getProviderPreferenceResponse">
      <wsdl:part name="parameters"
element="tns:getProviderPreferenceResponse">
      </wsdl:part>
    </wsdl:message>
    <wsdl:message name="getNotificationProviderForPage">
      <wsdl:part name="parameters"
element="tns:getNotificationProviderForPage">
      </wsdl:part>
    </wsdl:message>
    <wsdl:message name="register">
      <wsdl:part name="parameters" element="tns:register">
      </wsdl:part>
    </wsdl:message>
    <wsdl:message name="modifyProvider">
      <wsdl:part name="parameters" element="tns:modifyProvider">
      </wsdl:part>
    </wsdl:message>
```

```
  <wsdl:message name="modifyPreferenceResponse">
    <wsdl:part name="parameters"
element="tns:modifyPreferenceResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getNotificationProvider">
    <wsdl:part name="parameters"
element="tns:getNotificationProvider">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="provisionProviderResponse">
    <wsdl:part name="parameters"
element="tns:provisionProviderResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getNotificationProviderResponse">
    <wsdl:part name="parameters"
element="tns:getNotificationProviderResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getProviderPreferenceForPageResponse">
    <wsdl:part name="parameters"
element="tns:getProviderPreferenceForPageResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="modifyPreference">
    <wsdl:part name="parameters"
element="tns:modifyPreference">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="onMessageResponse">
    <wsdl:part name="parameters"
element="tns:onMessageResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="provisionPreference">
```

```
    <wsdl:part name="parameters"
element="tns:provisionPreference">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getProviderPreferenceForPage">
    <wsdl:part name="parameters"
element="tns:getProviderPreferenceForPage">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getProviderIndexResponse">
    <wsdl:part name="parameters"
element="tns:getProviderIndexResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="modifyProviderResponse">
    <wsdl:part name="parameters"
element="tns:modifyProviderResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="removePreferenceResponse">
    <wsdl:part name="parameters"
element="tns:removePreferenceResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="provisionPreferenceResponse">
    <wsdl:part name="parameters"
element="tns:provisionPreferenceResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:portType name="NotificationServicePortType">
    <wsdl:operation name="getNotificationProviderForPage">
      <wsdl:input name="getNotificationProviderForPage"
message="tns:getNotificationProviderForPage">
    </wsdl:input>
      <wsdl:output
name="getNotificationProviderForPageResponse"
message="tns:getNotificationProviderForPageResponse">
    </wsdl:output>
```

```
    </wsdl:operation>
    <wsdl:operation name="getNotificationProvider">
      <wsdl:input name="getNotificationProvider"
message="tns:getNotificationProvider">
    </wsdl:input>
      <wsdl:output name="getNotificationProviderResponse"
message="tns:getNotificationProviderResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getProviderPreference">
      <wsdl:input name="getProviderPreference"
message="tns:getProviderPreference">
    </wsdl:input>
      <wsdl:output name="getProviderPreferenceResponse"
message="tns:getProviderPreferenceResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="modifyProvider">
      <wsdl:input name="modifyProvider"
message="tns:modifyProvider">
    </wsdl:input>
      <wsdl:output name="modifyProviderResponse"
message="tns:modifyProviderResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getPreference">
      <wsdl:input name="getPreference"
message="tns:getPreference">
    </wsdl:input>
      <wsdl:output name="getPreferenceResponse"
message="tns:getPreferenceResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getProviderPreferenceForPage">
      <wsdl:input name="getProviderPreferenceForPage"
message="tns:getProviderPreferenceForPage">
    </wsdl:input>
```

```
      <wsdl:output name="getProviderPreferenceForPageResponse"
message="tns:getProviderPreferenceForPageResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="register">
      <wsdl:input name="register" message="tns:register">
    </wsdl:input>
      <wsdl:output name="registerResponse"
message="tns:registerResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="provisionPreference">
      <wsdl:input name="provisionPreference"
message="tns:provisionPreference">
    </wsdl:input>
      <wsdl:output name="provisionPreferenceResponse"
message="tns:provisionPreferenceResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="deRegister">
      <wsdl:input name="deRegister" message="tns:deRegister">
    </wsdl:input>
      <wsdl:output name="deRegisterResponse"
message="tns:deRegisterResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="onMessage">
      <wsdl:input name="onMessage" message="tns:onMessage">
    </wsdl:input>
      <wsdl:output name="onMessageResponse"
message="tns:onMessageResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="provisionProvider">
      <wsdl:input name="provisionProvider"
message="tns:provisionProvider">
    </wsdl:input>
```

```
      <wsdl:output name="provisionProviderResponse"
message="tns:provisionProviderResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="removePreference">
      <wsdl:input name="removePreference"
message="tns:removePreference">
    </wsdl:input>
      <wsdl:output name="removePreferenceResponse"
message="tns:removePreferenceResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="removeProvider">
      <wsdl:input name="removeProvider"
message="tns:removeProvider">
    </wsdl:input>
      <wsdl:output name="removeProviderResponse"
message="tns:removeProviderResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="modifyPreference">
      <wsdl:input name="modifyPreference"
message="tns:modifyPreference">
    </wsdl:input>
      <wsdl:output name="modifyPreferenceResponse"
message="tns:modifyPreferenceResponse">
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getProviderIndex">
      <wsdl:input name="getProviderIndex"
message="tns:getProviderIndex">
    </wsdl:input>
      <wsdl:output name="getProviderIndexResponse"
message="tns:getProviderIndexResponse">
    </wsdl:output>
    </wsdl:operation>
  </wsdl:portType>
```

```
  <wsdl:binding name="NotificationServiceSoapBinding"
type="tns:NotificationServicePortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="getNotificationProvider">
      <soap:operation soapAction="" style="document"/>
      <wsdl:input name="getNotificationProvider">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="getNotificationProviderResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getNotificationProviderForPage">
      <soap:operation soapAction="" style="document"/>
      <wsdl:input name="getNotificationProviderForPage">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output
name="getNotificationProviderForPageResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getProviderPreference">
      <soap:operation soapAction="" style="document"/>
      <wsdl:input name="getProviderPreference">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="getProviderPreferenceResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="modifyProvider">
      <soap:operation soapAction="" style="document"/>
      <wsdl:input name="modifyProvider">
        <soap:body use="literal"/>
```

```
        </wsdl:input>
        <wsdl:output name="modifyProviderResponse">
          <soap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="getPreference">
        <soap:operation soapAction="" style="document"/>
        <wsdl:input name="getPreference">
          <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="getPreferenceResponse">
          <soap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="getProviderPreferenceForPage">
        <soap:operation soapAction="" style="document"/>
        <wsdl:input name="getProviderPreferenceForPage">
          <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output
name="getProviderPreferenceForPageResponse">
          <soap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="register">
        <soap:operation soapAction="" style="document"/>
        <wsdl:input name="register">
          <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="registerResponse">
          <soap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="deRegister">
        <soap:operation soapAction="" style="document"/>
```

```
      <wsdl:input name="deRegister">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="deRegisterResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="provisionPreference">
      <soap:operation soapAction="" style="document"/>
      <wsdl:input name="provisionPreference">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="provisionPreferenceResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="onMessage">
      <soap:operation soapAction="" style="document"/>
      <wsdl:input name="onMessage">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="onMessageResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="provisionProvider">
      <soap:operation soapAction="" style="document"/>
      <wsdl:input name="provisionProvider">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="provisionProviderResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="removePreference">
```

```
    <soap:operation soapAction="" style="document"/>
    <wsdl:input name="removePreference">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="removePreferenceResponse">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="modifyPreference">
    <soap:operation soapAction="" style="document"/>
    <wsdl:input name="modifyPreference">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="modifyPreferenceResponse">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="removeProvider">
    <soap:operation soapAction="" style="document"/>
    <wsdl:input name="removeProvider">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="removeProviderResponse">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getProviderIndex">
    <soap:operation soapAction="" style="document"/>
    <wsdl:input name="getProviderIndex">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="getProviderIndexResponse">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
```

```
  </wsdl:binding>
  <wsdl:service name="NotificationService">
    <wsdl:port name="NotificationServicePort"
binding="tns:NotificationServiceSoapBinding">
      <soap:address
location="http://localhost:9090/NotificationServicePort"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

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.
© 2014 Cisco and/or its affiliates. All rights reserved.

January 2014 Part Number OL-30246-01