UCMA Cellular Service Walkthrough

# Overview

The UCMA Cellular Service sample demonstrates how the Unified Communications Managed API (UCMA) v2.0 can be used to publish and subscribe to the presence of users, as well as publish and subscribe to custom activities and custom presence. The UCMA Cellular Service demonstrates each of these two types of data; it publishes cell phone availability as a custom activity and GPS location as custom presence information. The sample also provides an example of how to leverage WCF to make remote presence calls using HTTP.

## Custom Presence

Office Communications Server (OCS) employs an enhanced presence model that aggregates a number of built-in categories of presence data. This presence data is published by client endpoints (e.g. Office Communicator), which can also subscribe to the presence of other users. The OCS presence publish/subscribe solution also supports custom presence categories that can be leveraged by future custom endpoints for more extensible solutions. UCMA implements custom presence through the <[rawCategory> element](http://msdn.microsoft.com/en-us/library/bb969569.aspx). This element has attributes which define name, schema location, version, container membership, lifetime, and other properties. It also contains a child element with an XML string that conforms to the defined schema location. This design empowers a custom UCMA client with the ability to publish and subscribe to any information that can be serialized into XML.

## Custom Activity

Some of the built in presence categories have associated token strings (e.g., Available, Busy), which are known as activity messages. OCS supports publishing custom token strings (known as “custom activities”) in conjunction with presence information.

## Solution Components

This sample solution has two components. The PubSub service exposes methods for publishing presence and subscribing to presence over HTTP. The PubSub client can be used to demonstrate and test the PubSub service.



## PubSub Service

The PubSub service is a Windows Communication Foundation (WCF) service that exposes methods for publishing presence and subscribing to presence. On startup, it registers a list of user endpoints representing a defined number of users.

Functionally, the solution exposes an external interface which allows cellular companies to publish the cellular activity of a target user, along with their GPS location, as custom presence information. The service also subscribes to the presence of all target users. If the user is away or offline and he or she receives an instant message, the service will intercept that the message and forward it as a text message to the target user (by sending an email to the user’s SMS gateway address).

### Presence Containers

The enhanced presence model in Office Communications Server uses “presence containers” to organize the information that is available to users with varying levels of access. Containers are essentially objects into which clients can publish presence data. Each container holds the information that is provided to users with a certain level of access. Each user can assign other users or groups of users to specific containers, to specify the presence data to which they have access.

The PubSub service publishes its custom presence information into the container with the ID of 200. As a result, all users who are members of that container will be able to see the target user’s presence. In this case, 200 is the “company” level container.

### Application Trusting

Instead of storing users’ credentials, which would create a security problem, this solution uses OCS application trusting. Through a server-side configuration, this security feature allows the service to start user endpoints without user authentication assuming the connection is coming from a known fully qualified domain name (FQDN).

**Important note**: This solution poses an obvious security risk in its current state assuming no additional protection is provided. A real implementation of this service would need to use proper security measures to prevent malicious or accidental use of another user’s account. Providing these security measures is beyond the scope of this sample.

### Publishing Cellular Presence

A cellular company can invoke the PubSub service to publish a user’s cell presence as “On Hook” or “Off Hook.” The service will validate the target user’s SIP URI by comparing the SIP URI to the one stored in the Config.xml. (In a real implementation of this sample, this validation would be performed by an external system, such as Active Directory.) The PubSub service then publishes the phone state of the user as Available or Busy based on the cellular provider’s OnHook or OffHook value. If a custom message is provided, then the service will publish it as a custom activity.

## PubSub Client

The PubSub client is a WinForms test application that allows manual entry of the cell presence and GPS location information that cellular companies would theoretically publish to the service. The PubSub client takes a GPS statement in NMEA format and the SIP URI of a target user, and publishes the GPS location of the user as custom presence information. (For more details on the NMEA format visit <http://www.gpsinformation.org/dale/nmea.htm>.)

The client publishes this information to a custom presence category, which needs to be registered on the OCS server ahead of time (instructions for doing this are below). Once the custom presence category is registered, OCS publishes the custom presence category just as it does for the built-in presence categories.

Because Office Communicator does not subscribe to custom presence categories, the PubSub client demonstrates this functionality by subscribing to the GPS custom presence information of all target users and displaying it in a list.

**NOTE!**

To implement a proper client application to subscribe to these custom presence categories, you would not ordinarily use UCMA, which is a server-side product. We have used it to implement this test client for simplicity, but other approaches would be more appropriate for a production environment.

# Prerequisites

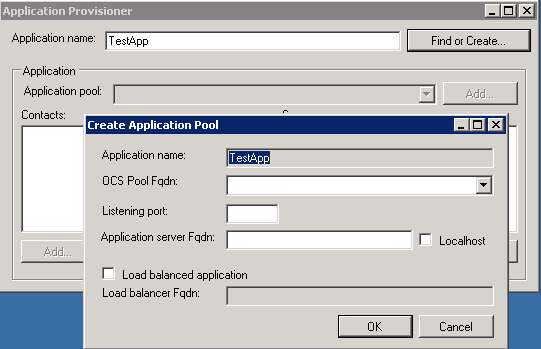
* UCMA v2.0 Core API
* Office Communicator 2007
* Visual Studio 2008

# Getting started

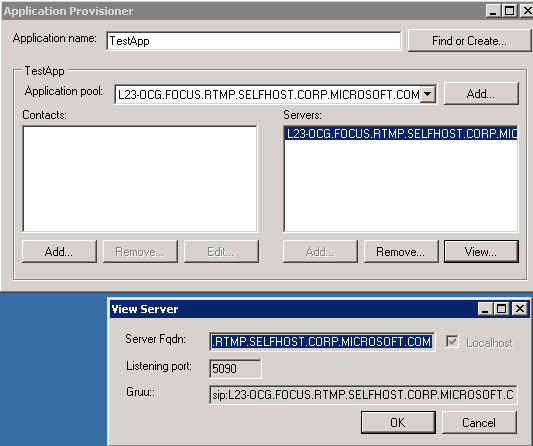
## Create a Trusted Service Entry

The UCMA SDK includes a provisioning tool, *ApplicationProvisioner .exe,* which can be used to create a trusted service entry and contact object(s) for a trusted application that you want to deploy in your production environment. This step is typically performed by a system administrator. For more details about the terminology used below, please refer to the UCMA SDK help file, located at UCMA SDK 2.0\UCMACore\Documentation\.

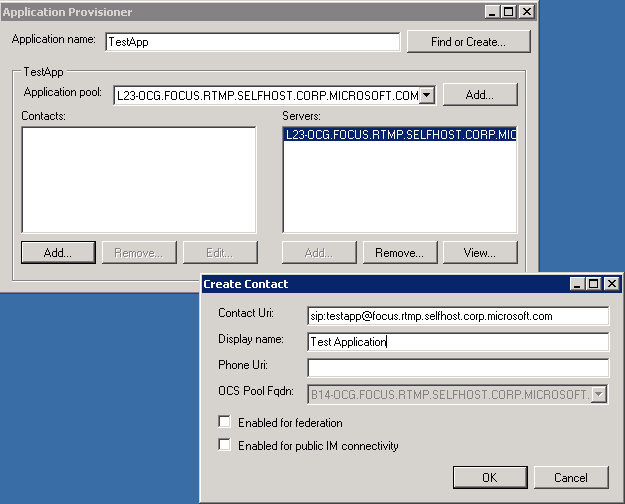
1. Run the ApplicationProvisioner tool, which is located in the UCMA SDK at UCMA SDK 2.0\UCMACore\Sample Applications\Collaboration\ApplicationProvisioner
2. Enter an application name in the text box tagged as **Application Name**.
3. Press **Find** **or** **Create.**
4. A window will pop up with the application name provided.
5. Choose the appropriate OCS pool.
6. Provide a listening port for the application.
7. Click on **Localhost** if this is the machine where you want your application to be deployed.
8. Click on **Load balanced application** if you want your application to be behind a load balancer. You will need to provide the load balancer’s FQDN in that case.
9. Click **OK**. This will create a trusted server entry (as shown below) for your application.



1. Click on **View** to view the trusted server entry.
2. Click on **Remove** to remove the trusted server entry (if needed).



1. Now we have to create the contact object for this trusted server. At least one contact object instance is necessary. Click **Add** in the **Contacts** section.
2. Provide a contact URI for your application (don’t forget to begin the URI with **sip:**).
3. Provide a display name for your application.
4. Click **OK**.



## ­Register Custom Presence Category (gpsLocation)

To allow clients to publish custom presence of users, you will need to register the custom presence category on the OCS server. In order to do that, execute the following script on your Office Communications Server’s SQL Server instance.

DECLARE @RC int

DECLARE @\_CategoryName nvarchar(4000)

DECLARE @\_Private bit

DECLARE @\_QuotaTotalSize int

SET @\_CategoryName ='gpsLocation'

EXECUTE @RC = [rtc].[dbo].[RtcRegisterCategoryDef]

   @\_CategoryName

  ,@\_Private

  ,@\_QuotaTotalSize

In this script, @\_CategoryName is the name of the custom category that the client is trying to publish. You can use this same script to publish other custom presence categories.

# Configuring the sample

1. **Run Visual Studio 2008 as an Administrator -** Navigate to Start | All Programs | Microsoft Visual Studio 2008, right-click on the Microsoft Visual Studio 2008 icon, and select Run as administrator from the context menu.
2. **Open the PubSub Solution -** From the menu, select File | Project/Solution. In the Open Project dialog, navigate to the PubSub.sln file and click Open.
3. **Configure the PubSub Service** - Open the web.config file under PubSubService Project and set all the configuration settings. ([Refer to configuration setting section below](#_Configuration_Settings))
4. **Configure the PubSub Client –** Open the app.config file under PubSubClient Project and set all the configuration settings. ([Refer to configuration setting section below](#_Configuration_Settings))
5. **Set Multiple Startup Projects –** Right click on the PubSub solution, and select Set StartUp Projects from the context menu. In the dialog box that appears, select the Multiple Startup Projects option. Select Start as the action for the PubSubClient and PubSubService projects.

# Configuration Settings

Both the PubSub service and the PubSub client require configuration. Both projects include .config files (web.config for the PubSubService and app.config for the PubSub client) which store these configuration settings in the <appSettings> element. The following settings are required:

## ****UCMA Settings****

* **applicationUserAgent** – The application name for the trusted service you provisioned (see the instructions above).
* **ocsServerFqdn** – The address of your Office Communications Server.
* **ocsServerTlsPort** – The TLS port for your Office Communications Server, if different from the default of 5061.
* **boturi** – The SIP URI of the contact you created for your trusted service (see the instructions above). Use one SIP URI for the service and another for the client.

## ****MapPoint Settings****

The PubSub service uses the Microsoft MapPoint web service to get the gpsLocation of the target user based on a provided GPS statement. It is assumed that a cellular provider will provide this information from the users’ cellular phone. The following settings are required to configure the proxy for the MapPoint web service.

* **liveAccountID** – The MapPoint web service account ID. To create a new developer account, go to <https://mappoint-css.live.com/MwsSignUp/Default.aspx>. After you create your account, a unique account ID will be assigned to you, and you can enter it here.
* **liveAccountPwd** – The password for your MapPoint web service account.
* **clientIP** – The public-facing IP address of the machine which is hosting the PubSub service.
* **tokenDuration** – How long tokens will remain valid.

The following settings are also required in order for the application to run:

* **smtpServer** –The SMTP server address. The service will use this to send emails to SMS gateways.
* **smtpPort** – The port on which the SMTP server is running.

## ****Config.xml****

The Config.xml file contains a list of target users for the PubSub service and the PubSub client. If the client passes into the service a SIP URI that does not exist in the service’s configuration file, the service will throw an InvalidDataException. The following settings are required for each target user:

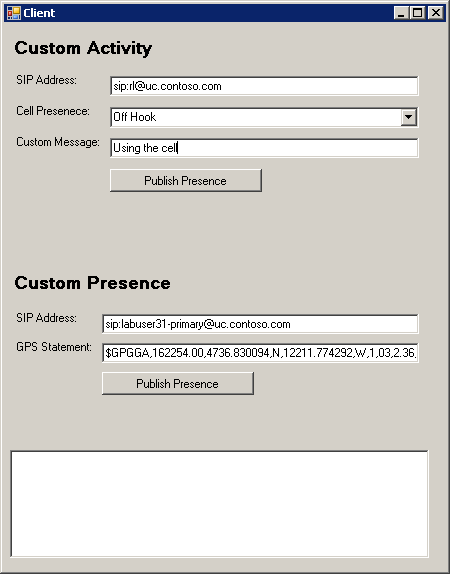
* **Uri** – SIP URI of the user
* **SMSAddress** – SMS email address where the user’s messages will be routed, so that they will be delivered to the user as text messages.

# Running the Sample

Press F5 to run the application. Navigate to <http://localhost/PubSub/Pubsub.svc> in the browser window. The PubSub client application will also start.

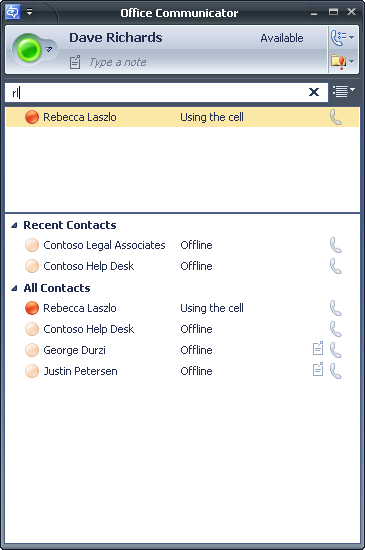
To publish cellular presence, do the following:

* In the client window, enter the target user’s SIP URI, cell presence and custom message (optional).
* Click the Publish Presence button.



* Open Office Communicator.
* Confirm that the user’s presence reflects the cellular state.

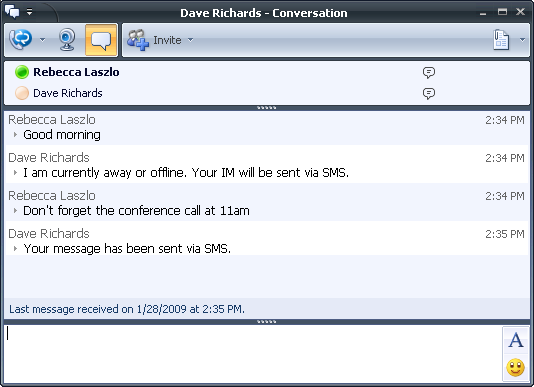
.



## Routing IM messages via SMS

The PubSub service subscribes to the presence of all the target users in the config.xml file. Instant messages sent to enrolled users who are away or offline are re-routed as text messages. The sender is notified that his or her instant messages will be routed to target user via SMS. To test this functionality, do the following:

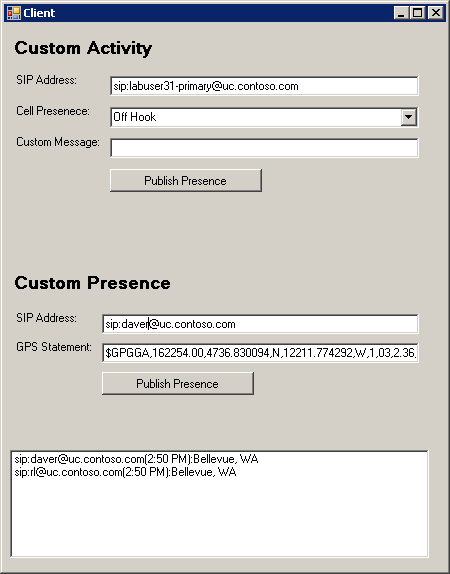
* Open Office Communicator.
* Select a user who is offline or away.
* Send an instant message to the user.
* The PubSub service will respond on behalf of the user with “I am currently away or offline. Your IM will be sent via SMS.”
* The PubSub service will respond to subsequent messages with “Your message has been sent via SMS.”



## Publish Custom Presence (gpsLocation)

To see how the client publishes custom presence information and receives information from subscriptions to other users’ custom presence, do the following:

* In the client window, enter SIP address and a valid GPS statement in the NMEA format (<http://www.gpsinformation.org/dale/nmea.htm>).
* Press the Publish Presence button.
* The client will receive the notification of newly-published custom presence. It will then parse the GPS location from the published presence XML schema and will display it in a list box along with the SIP URI of the target user and the time at which the presence was published.



# Conclusion

In this sample, you have seen the following features implemented using UCMA and OCS’s enhanced presence model:

* Publishing presence with custom activity messages
* Subscribing to the presence of the target users
* Intercepting instant messages if the target user is away or offline and forwarding them as text messages via email
* Publishing custom presence such as GPS location
* Subscribing to custom presence

These features are only a small subset of the many custom OCS solutions that can be implemented with minimal time and effort using the straightforward, concise syntax of UCMA v2.0.

# Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

2004 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Outlook, Windows, Windows NT, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.