# Tag and Broadcast Notifications in Native Windows 8 Applications

#### **Overview**

**Prerequisite:** Make sure that you read the Push notifications in native Windows 8 applications (.../) tutorial first

Tag notifications are notification messages that are targeted to all the devices that are subscribed to a particular tag.

Tags represent topics of interest to the user and provide the ability to receive notifications according to the chosen interest.

Broadcast notifications are a form of tag push notifications that are targeted to all subscribed devices. Broadcast notifications are enabled by default for any push-enabled MobileFirst application by a subscription to a reserved Push.all tag (auto-created for every device). This ability can be disabled by by unsubscribing from the reserved Push.all tag.

## **Agenda**

- Notifications configuration
- Notifications API
- Sample application

# **Notifications configuration**

## Tag Notifications configuration

Setting up tags

Tags are defined in the application-descriptor.xml file:

### **Notifications API**

#### API methods for tag notifications

#### Client-side API

- WLPush.subscribeTag(tagName,options) Subscribes the device to the specified tag name.
- WLPush.unsubscribeTag(tagName,options) Unsubscribes the device from the specified tag name.
- WLPush.isTagSubscribed(tagName) Returns whether the device is subscribed to a specified tag

## Common API methods for tag and broadcast notifications

#### Client-side API

- WLNotificationListener
   Defines the callback method to be notified when the notification arrives.
- WLPush.notificationListener = new MyNotificationListener();
   Sets the implementation class of the WLNotificationListener interface.
- WLPush.onReadyToSubscribeListener
  This method registers a listener to be used for push notifications. This listener should implement the onReadyToSubscribe() method.
- The onMessage(props, payload) method of WLNotificationListener is called when a push notification is received by the device.
  - **props** A JSON block that contains the notifications properties of the platform.
  - payload A JSON block that contains other data that is sent from MobileFirst Server. The
    JSON block also contains the tag name for tag-based or broadcast notification. The tag name
    appears in the "tag" element. For broadcast notification, the default tag name is Push.ALL.

#### Server-side API

WL.Server.sendMessage(applicationId,notificationOptions)

This method submits a notification based on the specified target parameters and takes two mandatory parameters:

- applicationId (mandatory) The name of the MobileFirst application
- notificationOptions (mandatory) A JSON block containing message properties

For a full list of message properties, see the WL.Server.sendMessage API in the API reference of the user documentation.

## Sample application

Click to download (https://github.com/MobileFirst-Platform-Developer-Center/TagNotifications/tree/release71) the MobileFirst project.
Click to download (https://github.com/MobileFirst-Platform-Developer-Center/TagNotificationsWin8/tree/release71) the Native project.

 The TagNotifications project contains a MobileFirst native API that you can deploy to your MobileFirst Server instance.

- The TagNotificationsWin8 project contains a native Windows 8 Universal application that uses a
  MobileFirst native API library to subscribe to push notifications and receive notifications from
  Windows Notification Services (WNS).
- Make sure to update the wlclient.properties file in the native project with the relevant server settings.



## Sending a notification

To test the application is able to receive a push notification you can perform one of the following:

- 1. From MobileFirst Studio, right-click the adapter folder, select **Call MobileFirst Adapter** and:
  - If selecting the "sendBroadcastNotification" procedure, provide the application ID and notification text in quotation marks.
  - If selecting the "sendTagNotification" procedure, provide the application ID, notification text and tag name in quotation marks.
  - The application ID can be determined from the id attribute in application-descriptor.xml:

```
<application ... id="NativeWin8TagNotifications" ...>
```

#### 2. If using the CLI:

\$ mfp adapter call

- [?] Which endpoint do you want to use? PushAdapter/sendBroadcastNotification
- [?] Enter the comma-separated parameters: "NativeWin8TagNotifications", "hello"
- [?] How should the procedure be called? GET

#### Or:

\$ mfp adapter call

- [?] Which endpoint do you want to use? PushAdapter/sendTagNotification
- [?] Enter the comma-separated parameters: "NativeWin8TagNotifications", "hello", "sample-tag1"
- [?] How should the procedure be called? GET