Windows 8.1 Universal and Windows 10 UWP end-to-end demonstration

Overview

The purpose of this demonstration is to experience an end-to-end flow:

1. A scaffold application - an application that is pre-bundled with the MobileFirst client SDK, is registered and downloaded from the MobileFirst Operations Console.
2. An new or provided adapter is deployed to the MobileFirst Operations Console.
3. The application logic is changed to make a resource request.

End result:

- Successfully pinging the MobileFirst Server.
- Successfully retrieving data using a MobileFirst Adapter.

Prerequisites:

- Configured Visual Studio 2013/5
- *Optional*. MobileFirst Developer CLI (download (file:////home/travis/build/MFPSamples/DevCenter/_site/downloads))
- *Optional*. Stand-alone MobileFirst Server (download (file:////home/travis/build/MFPSamples/DevCenter/_site/downloads))

1. Starting the MobileFirst Server

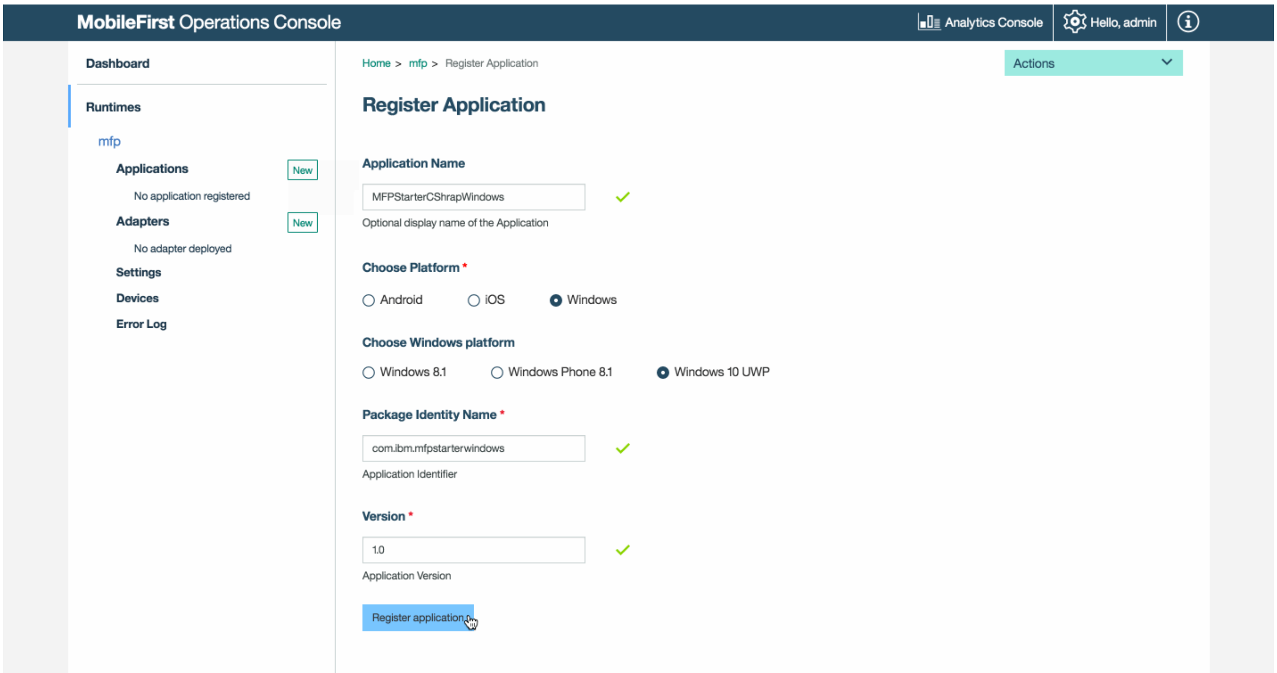
> If a remote server was already set-up, skip this step.

From a **Command-line** window, navigate to the server's folder and run the command: `run.bat`.

2. Creating an application

In a browser window, open the MobileFirst Operations Console by loading the URL: `http://your-server-host:server-port/mfpconsole`. If running locally, use: http://localhost:9080/mfpconsole (http://localhost:9080/mfpconsole). The username/password are *admin/admin*.

1. Click the **New** button next to **Applications**
    - Select a **Windows** platform
    - Enter **MFPStarterCSharp.Windows** as the **application identifier** for Windows, or **MFPStarterCSharp.WindowsPhone** for Windows Phone
    - Enter **1.0.0** as the **version** value
    - Click on **Register application**



Register Application

No application registered

No adapter deployed

Application Name

MFPStarterCShrapWindows ✓

Optional display name of the Application

Choose Platform *

○ Android ○ iOS ● Windows

Choose Windows platform

○ Windows 8.1 ○ Windows Phone 8.1 ● Windows 10 UWP

Package Identity Name *

com.ibm.mfpstarterwindows ✓

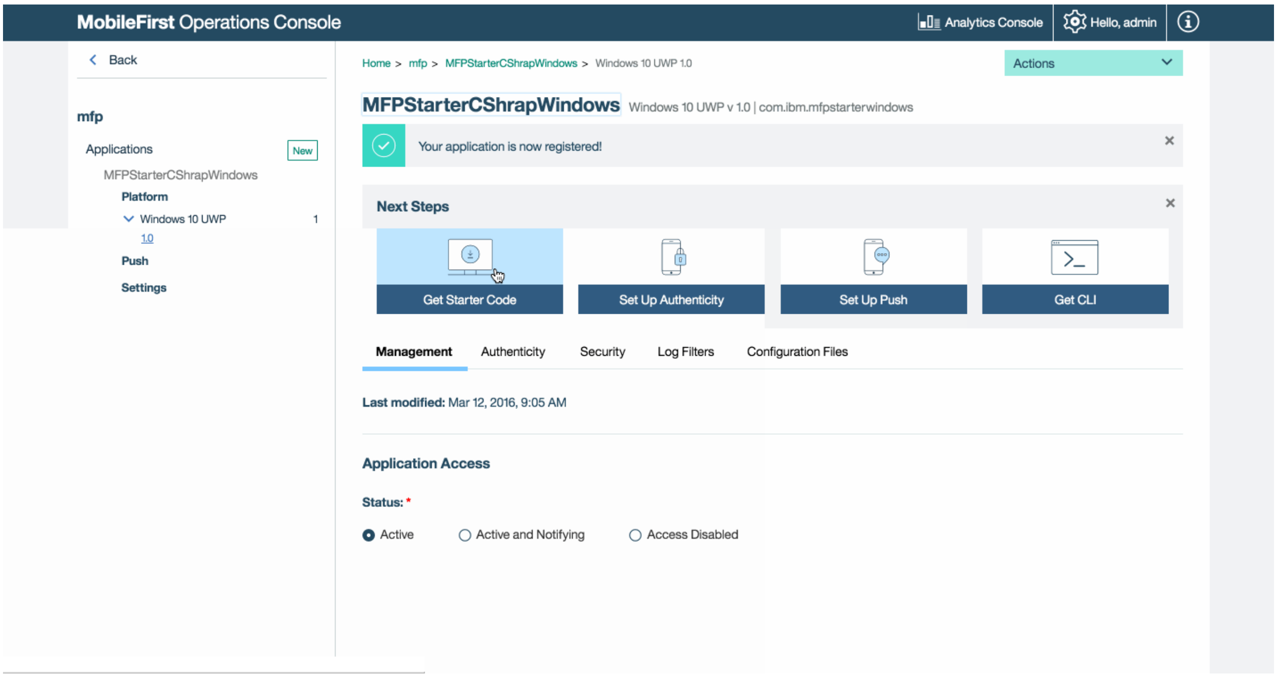
Application Identifier

Version *

1.0 ✓

Application Version

2. Click on the **Get Starter Code** tile and select to download the Windows 8.1 or Windows 10 mobile app scaffold.



MFPStarterCShrapWindows Windows 10 UWP v 1.0 | com.ibm.mfpstarterwindows

Your application is now registered!

Last modified: Mar 12, 2016, 9:05 AM

Application Access

Status: *

● Active ○ Active and Notifying ○ Access Disabled

3. Editing application logic

1. Open the Visual Studio project.
2. Select the solution's **MainPage.xaml.cs** file and paste the following code snippet:

```
WorklightAccessToken accessToken = await Worklight.WorklightClient.CreateInstance().AuthorizationManager.ObtainAccessToken("");

if(accessToken.IsValidToken && accessToken.Value != null && accessToken.Value != "")
{
  try
  {
    IWorklightClient _newClient = WorklightClient.CreateInstance();

    StringBuilder uriBuilder = new StringBuilder().Append("/adapters/JavaAdapter/users/world");

    WorklightResourceRequest rr = _newClient.ResourceRequest(uriBuilder.ToString(), "GET");

    WorklightResponse resp= await rr.send();

    if (resp.success)
    {
      Debug.WriteLine("Success: " + resp.ResponseText);
    } else
    {
      Debug.WriteLine("Failure: " + resp.error);
    }
  }catch(Exception e)
  {
    Debug.WriteLine(e.StackTrace);
  }
}
```

4. Creating an adapter

Download this prepared .adapter artifact (../javaAdapter.adapter) and deploy it from the MobileFirst Operations Console using the **Actions → Deploy adapter** action.

Alternatively, click the **New** button next to **Adapters**.

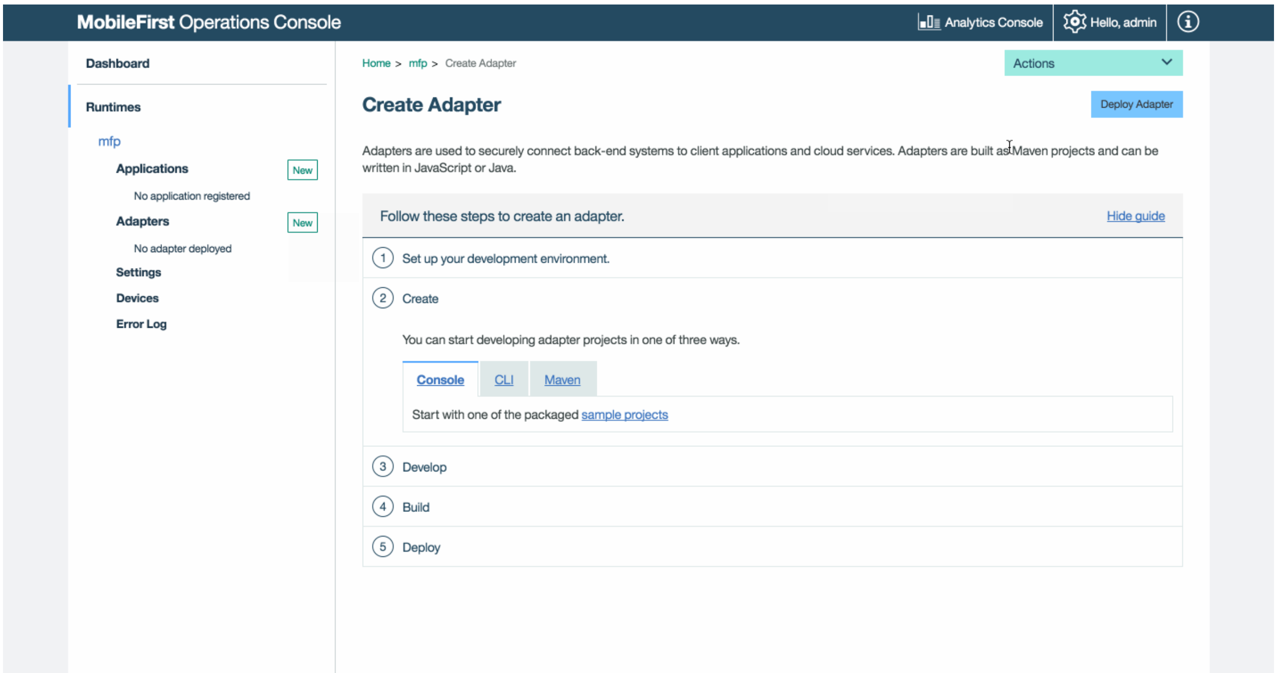
1. Select the **Actions → Download sample** option. Download the "Hello World" **Java** adapter sample.

   > If Maven and MobileFirst Developer CLI are not installed, follow the on-screen **Set up your development environment** instructions.

2. From a **Command-line** window, navigate to the adapter's Maven project root folder and run the command:

   ```
   mfpdev adapter build
   ```

3. When the build finishes, deploy it from the MobileFirst Operations Console using the **Actions → Deploy adapter** action. The adapter can be found in the **[adapter]/target** folder.



MobileFirst Operations Console

Analytics Console | Hello, admin

Dashboard

Runtimes

mfp

Applications New

No application registered

Adapters New

No adapter deployed

Settings

Devices

Error Log

Home > mfp > Create Adapter

Actions

Create Adapter

Deploy Adapter

Adapters are used to securely connect back-end systems to client applications and cloud services. Adapters are built as Maven projects and can be written in JavaScript or Java.

Follow these steps to create an adapter. Hide guide

1 Set up your development environment.

2 Create

You can start developing adapter projects in one of three ways.

Console | CLI | Maven

Start with one of the packaged sample projects

3 Develop

4 Build

5 Deploy

5. Testing the application

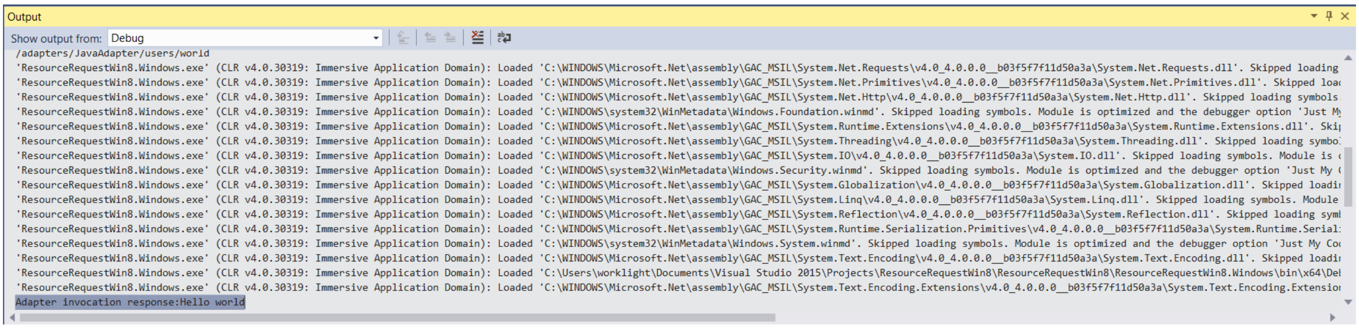
1. In Visual Studio, select the **mfpclient.resw** file and edit the **host** property with the IP address of the MobileFirst Server.

   Alternatively, if you have installed the MobileFirst Develper CLI then navigate to the project root folder and run the command `mfpdev app register`. If a remote server is used instead of a local server, first use the command `mfpdev server add` to add it.
2. Press the **Run App** button.

Results

- Clicking the **Ping MobileFirst Server** button will display **Connected to MobileFirst Server**.
- If the application was able to connect to the MobileFirst Server, a resource request call using the deployed Java adapter will take place.

The adapter response is then printed in Visual Studio's Outpout console.



Output

Show output from: Debug

```
/adapters/JavaAdapter/users/world
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Net.Requests\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Net.Requests.dll'. Skipped loading
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Net.Primitives\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Net.Primitives.dll'. Skipped loa
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Net.Http\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Net.Http.dll'. Skipped loading symbols
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\system32\WinMetadata\Windows.Foundation.winmd'. Skipped loading symbols. Module is optimized and the debugger option 'Just My
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Runtime.Extensions\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Runtime.Extensions.dll'. Ski
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Threading\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Threading.dll'. Skipped loading symbo
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.IO\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.IO.dll'. Skipped loading symbols. Module is
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\system32\WinMetadata\Windows.Security.winmd'. Skipped loading symbols. Module is optimized and the debugger option 'Just My
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Globalization\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Globalization.dll'. Skipped loadi
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Linq\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Linq.dll'. Skipped loading symbols. Module
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Reflection\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Reflection.dll'. Skipped loading sym
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Runtime.Serialization.Primitives\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Runtime.Serial
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\system32\WinMetadata\Windows.System.winmd'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Co
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Text.Encoding\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Text.Encoding.dll'. Skipped loadi
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\Users\worklight\Documents\Visual Studio 2015\Projects\ResourceRequestWin8\ResourceRequestWin8\ResourceRequestWin8.Windows\bin\x64\De
'ResourceRequestWin8.Windows.exe' (CLR v4.0.30319: Immersive Application Domain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Text.Encoding.Extensions\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Text.Encoding.Extensio
Adapter invocation response:Hello world
```

Next steps

Learn more on using adapters in applications, and how to integrate additional services such as Push Notifications, using the MobileFirst security framework and more:

- Review the Using the MobileFirst Platform Foundation (../../using-the-mfpf-sdk/) tutorials
- Review the Adapters development (../../adapters/) tutorials
- Review the Authentication and security tutorials (../../authentication-and-security/)

- Review the Notifications tutorials (../../notifications/)
- Review All Tutorials (../../all-tutorials)