How To... Enable User On-boarding using the OData SDK (Windows)

Applicable Releases:

SAP Mobile Platform 3.0

Version 1.0

June 2014

|  |  |
| --- | --- |
| © Copyright 2014 SAP AG. All rights reserved.   1. No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice. 2. Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. 3. Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation. 4. IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation. 5. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. 6. Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries. 7. Oracle is a registered trademark of Oracle Corporation. 8. UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group. 9. Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc. 10. HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology. 11. Java is a registered trademark of Sun Microsystems, Inc. 12. JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape. 13. SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.   Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.   1. Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company. | 1. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary. 2. The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG. 3. This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice. 4. SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. 5. SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence. 6. The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.   SAP “How-to” Guides are intended to simplify the product implement­tation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.  Any software coding and/or code lines / strings (“Code”) included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.  Disclaimer  Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.  Any Java™ Source Code delivered with this product is only to be used by SAP’s Support Services and may not be modified or altered in any way. |

Document History

|  |  |
| --- | --- |
| Document Version | Description |
| 1.00 | First official release of this guide |

Typographic Conventions

|  |  |
| --- | --- |
| Type Style | Description |
| Example Text | Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options.  Cross-references to other documentation |
| **Example text** | Emphasized words or phrases in body text, graphic titles, and table titles |
| Example text | File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools. |
| Example text | User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation. |
| <Example text> | Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system. |
| EXAMPLE TEXT | Keys on the keyboard, for example, F2 or ENTER. |

Icons

|  |  |
| --- | --- |
| Icon | Description |
|  | Caution |
|  | Note or Important |
|  | Example |
|  | Recommendation or Tip |

Table of Contents

[1. Business Scenario 1](#_Toc398077867)

[2. Background Information 1](#_Toc398077868)

[3. Prerequisites 1](#_Toc398077869)

[4. Step-by-Step Procedure 2](#_Toc398077870)

[4.1 Windows Project 2](#_Toc398077871)

[4.2 On-boarding a device 3](#_Toc398077872)

[4.2.1 Steps involved in the On-boarding process 3](#_Toc398077873)

[4.2.2 On-boarding Process Flow 4](#_Toc398077874)

[4.2.3 Screen flow of the application 5](#_Toc398077875)

[4.2.4 Preference Settings 7](#_Toc398077876)

[4.2.5 Registration 7](#_Toc398077877)

[4.3 Running the application 11](#_Toc398077878)

[5. Appendix 13](#_Toc398077879)

[5.1 NuGet Package Manager 13](#_Toc398077880)

[5.1.1 Adding Windows SMP SDK package in Visual Studio 13](#_Toc398077881)

[5.1.2 Adding Windows SMP SDK references to the project 14](#_Toc398077882)

# Business Scenario

Travel Agency X would like to build an online mobile application for its customers, so they can book their flights anywhere, anytime from their devices. The **SAP Mobile Platform** provides a means for them to securely and efficiently access backend flight data exposed by their SAP NetWeaver Gateway system via OData-based REST services.

Before any communication with the SAP backend systems can take place at all, their app needs to on-board users onto the SAP Mobile Platform. Luckily the mobile platform client OData SDK provides easy-to-use libraries that can help app developers perform this task.

# Background Information

The goal of this exercise is not to show how to create a project from scratch and dissect every line of code. Instead, it shows the key pieces of code and information, along with a starter project template, so that developers understand how to leverage the OData SDK to on-board users in their own apps.

# Prerequisites

This exercise targets the Windows Desktop platform and has the following prerequisites:

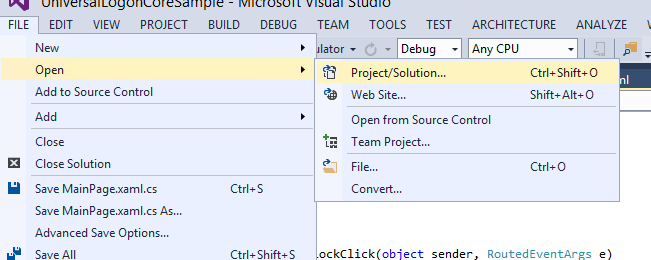
* Visual Studio 2013 with Update 2
* Any Windows machine running .NET 4.5
* To get the most out of this exercise, experience with Windows programming is recommended.

# Step-by-Step Procedure

The following sections provide a detailed step-by-step procedure on how to create a Windows Store application to on-board/register a user onto the SAP Mobile Platform. This is the first exercise of many related exercises in this workshop. Each exercise builds upon the previous exercise, so it is recommended that you complete each exercise before moving to the next.

## Windows Project

1. ...
   1. Open Visual Studio 2013 with Update 2 and open the solution RKT\_WPF\_Onboarding.sln



* 1. SMP SDK ships with 2 separate set of libraries. One set of libraries target Windows Store applications – specifically Windows 8.1 and Windows Phone 8.1. The other set of libraries target Windows Desktop applications – any Windows machine running .NET 4.5 or higher. It is important to understand that the APIs we provide on the 3 platforms (Windows desktop, Windows tablets, Windows Phone 8.1) are all identical.
  2. This sample application targets the Windows Desktop platform. Hence we use the WPF Application template to create the solution.
  3. Windows SMP SDK libraries are packaged as NuGet packages. See appendix on how to add the libraries as reference.

## On-boarding a device

### Steps involved in the On-boarding process

To fully on-board a device in the SMP Server, the developer needs to call these 3 methods. Note that all the 3 methods are asynchronous in nature.

|  |
| --- |
| 1. const string application\_id = “com.sap.windows.flight” ; 2. **// initialize the LogonCore**   var logonCore = await SAP.Logon.Core.LogonCore.InitWithApplicationIdAsync(application\_id);  var logonContext = new SAP.Logon.Core.LogonContext  {  RegistrationContext = new SAP.Logon.Core.RegistrationContext  {  ApplicationId = application\_id,  ServerHost = "<hostname>",  IsHttps = false,  ServerPort = 8080,  }  };   1. **// registers the device** 2. **await logonCore.RegisterWithContextAsync(logonContext);** 3. **// persist locally** 4. **await logonCore.PersistRegistrationAsync(passcode, logonContext);** |

The first method initializes the logonCore variable. This method takes the application id as the parameter.

The second method submits an HTTP POST request from the device to the SMP Server (logonContext variable provides all the necessary information to connect to the SMP Server) with a payload identifying the device and requesting the SMP Server to register the device. Upon successful registration, the SMP Server sends an HTTP response with important registration values that are saved in the logonContext variable.

The third method persists the logonContext variable in the data vault securely using a passcode. The SMP Server administrator has the option of defining client policies that require passcodes on the device. This is optional, but highly recommended.

### On-boarding Process Flow

PersistRegistrationAsync (null, logonContext)

NO

PersistRegistrationAsync (passcode, logonContext)

YES

Is password enabled?

RegisterWithContextAsync

(logonContext)

NO

YES

Unlock data vault (null or passcode)

Is device registered?

### Screen flow of the application

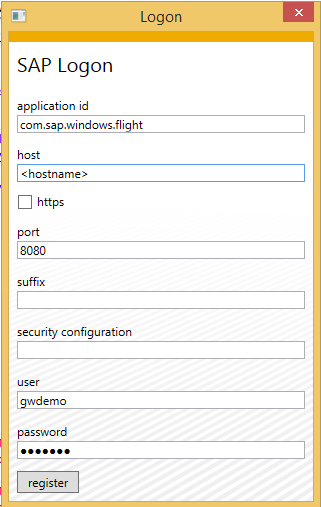
This sample application utilizes the VisualStateManager in XAML to create the on-boarding process flow. A number of VisualState sections are defined that are used to control the on-boarding process. When creating an application from scratch, it is recommended to use the code in this sample application as a template for on-boarding a device.

MainPage.xaml has multiple VisualStates defined :

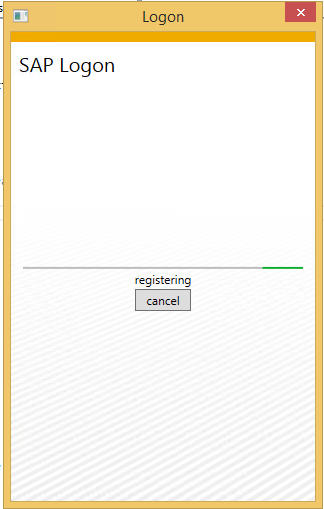
* + - 1. **ShouldRegister VisualState**: Device has not previously registered. This VisualState is shown when the application is launched for the first time. This VisualState is also shown when the registration is deleted.
      2. **Registering VisualState:** This VisualState is shown when the device is in the process of registering.
      3. **NewPassCode VisualState**: Depending on the client policy setup in the SAP Mobile Platform, the user is prompted to enter a client passcode. This VisualState is shown immediately after registration is complete depending on whether user is prompted for client code or not.

1. **EnterPassCode ViewState:** This ViewState is shown when an application is launched and the device is already registered. The SAP Mobile Platform should also be set to prompt for client passcode.
   * + 1. **EnterPassCode VisualState:** This VisualState is shown when an application is launched and the device is already registered. The SAP Mobile Platform should also be set to prompt for client passcode.

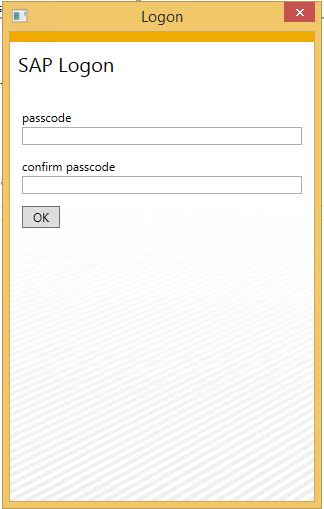
**ShouldRegister VisualState**



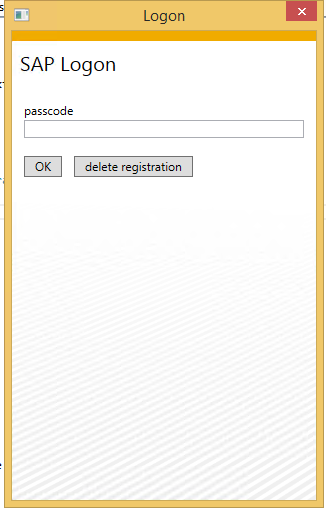
**Registering VisualState**



**NewPasscode VisualState**



**EnterPasscode VisualState**



**Simplified ViewState flow…**

|  |
| --- |
| if (IsRegistered)  {   1. **// if passcode is enabled; Otherwise open application**   GoToState("EnterPasscode");  }  else  {  GoToState("ShouldRegister");  ButtonRegisterClick(object sender, RoutedEventArgs e);   1. **// if passcode is enabled; Otherwise open application**   GoToState("NewPasscode");  } |

### Preference Settings

Any time the app launches, it needs information about connectivity to the mobile platform (e.g. host, port, application\_id etc.). These settings are defined in the LogonPageContext class. Please edit the hostname in the LogonPageContext class to point to the proper SMP Server.

To see how this is implemented in code:

* 1. Go to the Solution Explorer (right pane) and open LogonPageContext.cs file. Since this code can be shared between Windows and Windows Phone platform, this class is implemented in the Shared project.

LogonPageContext class implements the INotifyPropertyChanged interface. This interface is used to notify clients, typically binding clients that a property value has changed. The DataContext of LogonPage is set to an instance of the LogonPageContext class. This allows the UI controls in the XAML page to bind to the properties of the LogonPageContext class.

|  |
| --- |
| public class LogonPageContext : INotifyPropertyChanged |

|  |
| --- |
| private readonly LogonPageContext pageContext = new LogonPageContext();  public LogonPage()  {  this.InitializeComponent();  this.DataContext = this.pageContext;  } |

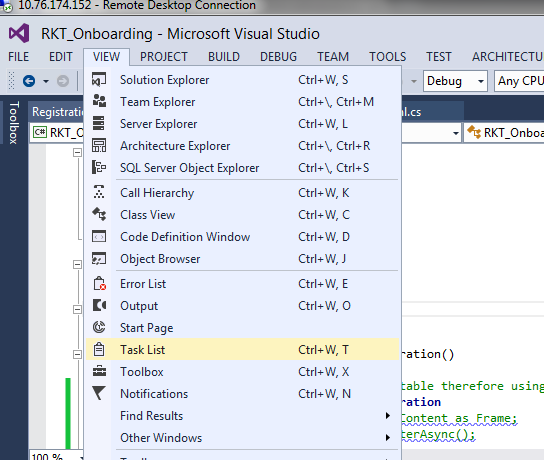
|  |
| --- |
| <StackPanel>  <StackPanel DataContext="{Binding LogonContext.RegistrationContext}">  <StackPanel>  <TextBlock Text="application id" />  <TextBox Text="{Binding ApplicationId, Mode=TwoWay}"/>  </StackPanel>  <StackPanel>  <TextBlock Text="host" />  <TextBox Text="{Binding ServerHost, Mode=TwoWay}"/>  </StackPanel> |

* 1. Because the binding is TwoWay, changes to the UI controls are automatically reflected in the underlying data value.

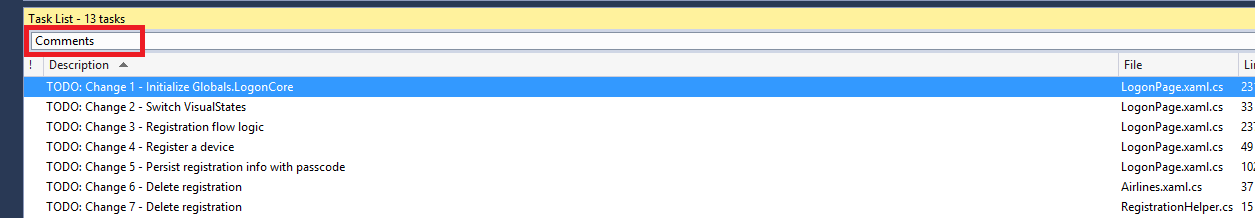
### Registration

On-boarding or registration with the SAP Mobile Platform is performed using the OData SDK **LogonCore** library. Classes in this library consume the services exposed by the SAP Mobile Platform as a REST API.

1. ...
   1. In Visual Studio 2013 + Update 2, click on View -> Task List



* 1. Sort the tasks alphabetically, to view all the tasks associated for this exercise. They should all begin with **// TODO: Change x.** There are 7 tasks for this exercise.



* 1. Open LogonWindow.xaml.cs in the sample application.
  2. Go to the Window\_Loaded event handler. In between the BEGIN and END //TODO Change1: markers enter the following code:

|  |
| --- |
| if (Globals.LogonCore == null)  {  Globals.LogonCore = await SAP.Logon.Core.LogonCore.InitWithApplicationIdAsync("com.sap.windows.flight");  } |

The Window\_Loaded event is raised when the window is initially loaded. This event handler is specified in the LogonWindow.xaml file (Loaded="Window\_Loaded"). An asynchronous call is made to initialize the LogonCore. It is recommended to initialize the LogonCore variable in this event.

* 1. Go to the GoToState method. In between the BEGIN and END //TODO Change2: markers enter the following code:

|  |
| --- |
| if (state == "RegistrationComplete")  {  this.Hide();  (new Airlines()).Show();  }  else  {  if (state == "ShouldRegister")  {  this.PwdBox\_BackendPassword.Password = this.pageContext.LogonContext.RegistrationContext.BackendPassword;  }  VisualStateManager.GoToElementState(this, state, false);  } |

The GoToState method takes care of switching between the various ViewStates that have been defined in the XAML page. If registration is complete, then the application is opened by navigating to the Airlines page. Otherwise, depending on the ViewState that is passed, the appropriate ViewState is opened.

* 1. Go to the Window\_Loaded event handler. In between the BEGIN and END //TODO Change3: markers enter the following code:

|  |
| --- |
| if (Globals.LogonCore.State.IsRegistered)  {  bool shouldEnterPasscode = false;  try  {  await Globals.LogonCore.UnlockSecureStoreAsync(null);  this.pageContext.LogonContext = Globals.LogonCore.LogonContext;  }  catch (Exception)  {  shouldEnterPasscode = true;  }  if ((Globals.LogonCore.LogonContext == null) || (shouldEnterPasscode))  {  this.GoToState("EnterPasscode");  }  else  {  this.GoToState("RegistrationComplete");  }  }  else  {  this.GoToState("ShouldRegister");  } |

Depending on the registration status, the appropriate ViewState is then displayed. Note that an attempt to unlock the data vault without supplying a password is tried first. If that fails, then the user is prompted to enter a password.

* 1. Go to the ButtonRegisterClick event handler. In between the BEGIN and END //TODO Change4: markers enter the following code:

|  |
| --- |
| try  {  await Globals.LogonCore.RegisterWithContextAsync(this.pageContext.LogonContext);  this.pageContext.PasswordToggleEnabled = this.pageContext.LogonContext.PasswordPolicy.IsDefaultPasswordAllowed;  }  catch (Exception ex)  {  message = ex.Message;  }  if (message != null)  {  await this.ShowErrorMessage(message);  }  else  {  if (this.pageContext.LogonContext.PasswordPolicy.IsEnabled)  {  this.GoToState("NewPasscode");  }  else  {  await Globals.LogonCore.PersistRegistrationAsync(null, this.pageContext.LogonContext);  this.GoToState("RegistrationComplete");  }  } |

To fully register a device, the developer needs to call the following 2 methods. RegisterWithContextAsync and PersistRegistrationAsync. In the ButtonRegisterClick event handler the RegisterWithContextAsync is first called. If it succeeds, then depending on whether a passcode is required or not, the PersistRegistrationAsync method is called. If a passcode is required the user is prompted for the password by switching to the NewPasscode ViewState.

* 1. Go to the ButtonNewPasscodeOkClick event handler. In between the BEGIN and END //TODO Change5: markers enter the following code:

|  |
| --- |
| try  {  await Globals.LogonCore.PersistRegistrationAsync(passcodeEnabled ? this.pageContext.Passcode : null, this.pageContext.LogonContext);  this.GoToState("RegistrationComplete");  }  catch (Exception ex)  {  message = ex.Message;  } |

Once the user enters the new passcode, the PersistRegistrationAsync method is called to complete the registration process.

* 1. Open Airlines.xaml.cs in the sample application.
  2. Go to the MenuItemDeleteRegistrationClick event handler. In between the BEGIN and END //TODO Change6: markers enter the following code:

|  |
| --- |
| RegistrationHelper.DeleteRegistration(); |

Once registration is complete, the application is now opened and the user is in the Airlines page. While it is not usually recommended to allow the user to delete the registration, for learning purpose a method is displayed to delete the registration. A helper class is used to delete the registration.

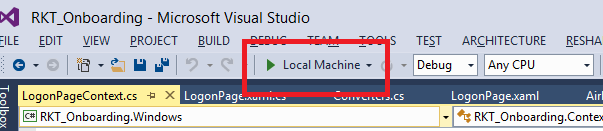
* 1. Open RegistrationHelper.cs in the Shared project
  2. Go to the DeleteRegistration method. In between the BEGIN and END //TODO Change7: markers enter the following code:

|  |
| --- |
| await Globals.LogonCore.UnregisterAsync();  NavigateToLogonPage(); |

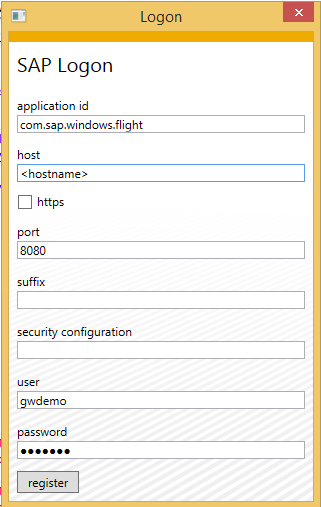
An asynchronous call to the UnregisterAsync method will delete the data vault on the local device. Also, it will unregister the device on the SMP Server. Once registration is deleted, the ShouldRegister ViewState is displayed to the user by navigating back to the LogonPage.

## Running the application

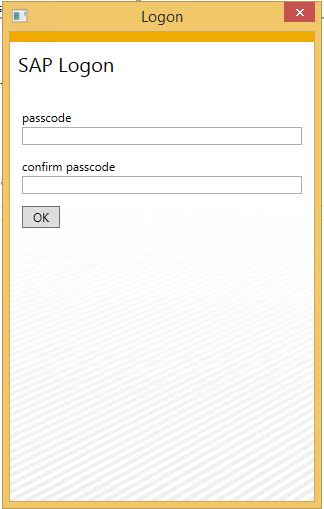
1. Run the application by clicking Run from within Visual Studio 2013 Update 2.



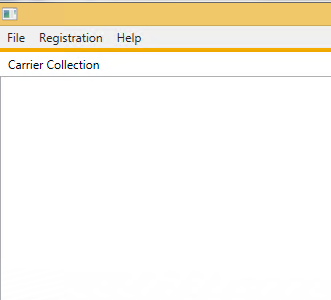
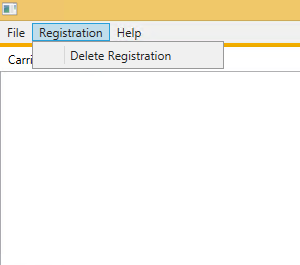
1. Make sure the values are all correct and click register.



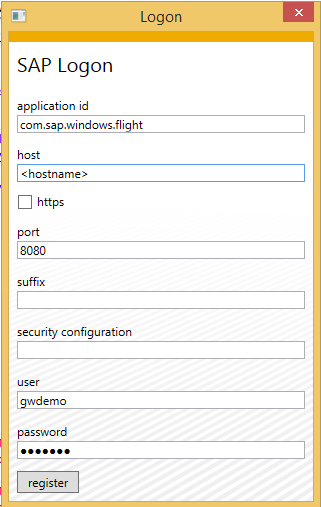
1. Enter the passcode and confirm the passcode. Click Ok



1. The application should now open up. Click on delete registration.

1. You should now be taken back to the LogonPage.



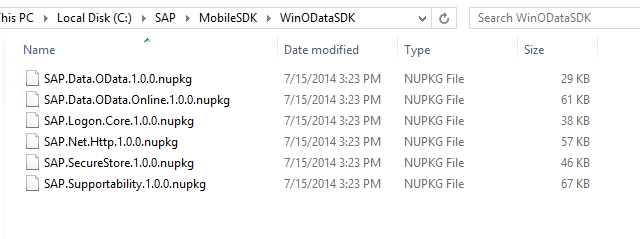
# Appendix

## NuGet Package Manager

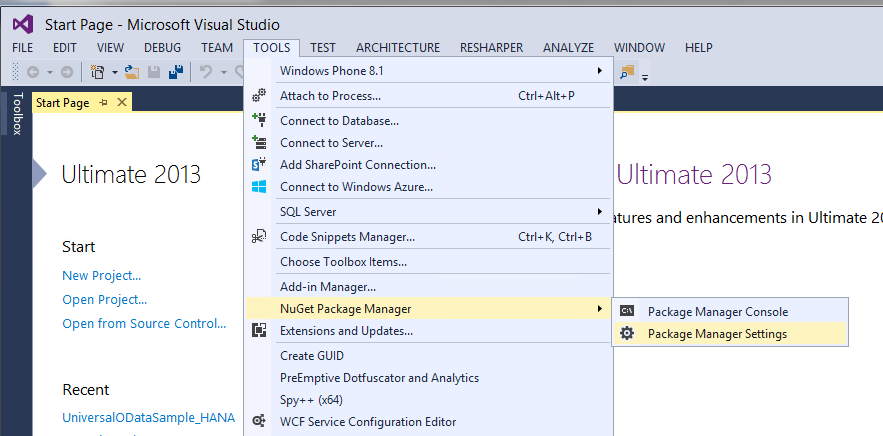
NuGet is the package manager for the Microsoft development platform including .NET. The NuGet client tools provide the ability to produce and consume packages. Starting with Visual Studio 2012, NuGet is included in every edition (except Team Foundation Server) by default. Updates to NuGet can be found through the Extension Manager.

### Adding Windows SMP SDK package in Visual Studio

1. Find the location of the Windows SMP SDK files in your local development machine (default location is C:\SAP\MobileSDK3\NativeSDK\ODataFramework\Windows). (The .nupkg file contains libraries for both Windows Store and Windows Desktop)



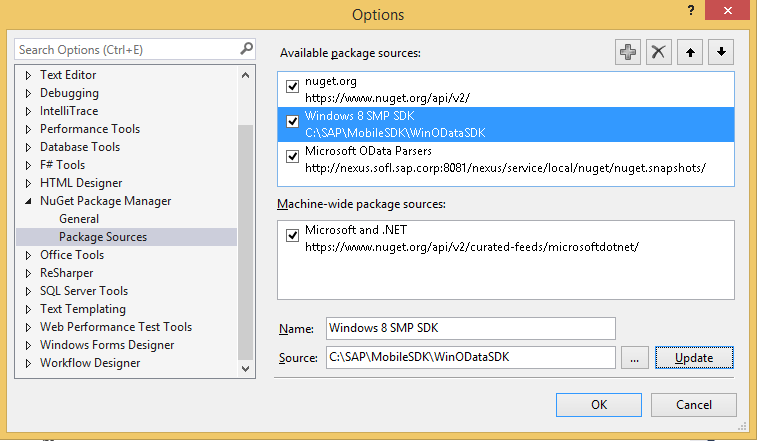
1. Open Visual Studio 2013 with Update 2. Click Tools -> NuGet Package Manager -> Package Manager Settings



1. Click on NuGet Package Manager -> Package Sources on the left pane. On the right pane, click the + sign on the right pane to add a new package source. Enter a name for the package and browse to the source of the unzipped Windows SMP SDK files. Click Update.

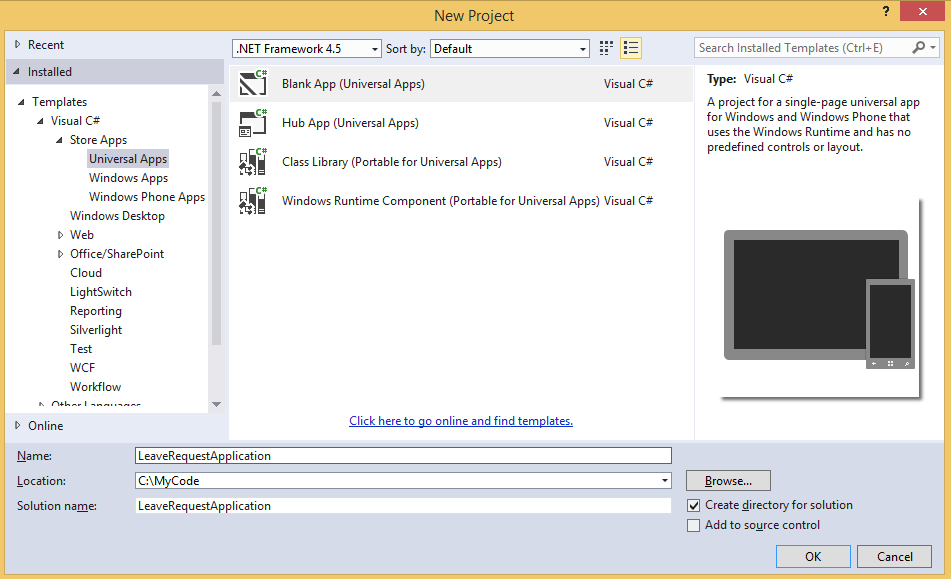
**Name:** Windows SMP SDK

**Source:** C:\SAP\MobileSDK3\NativeSDK\ODataFramework\Windows

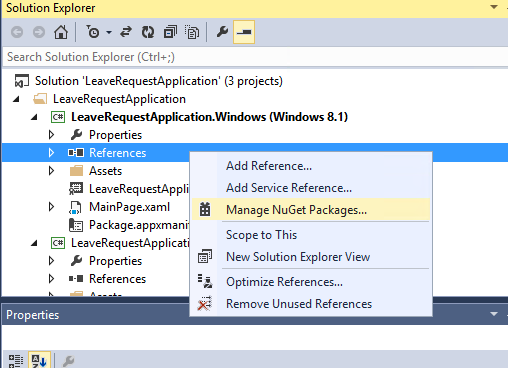


### Adding Windows SMP SDK references to the project

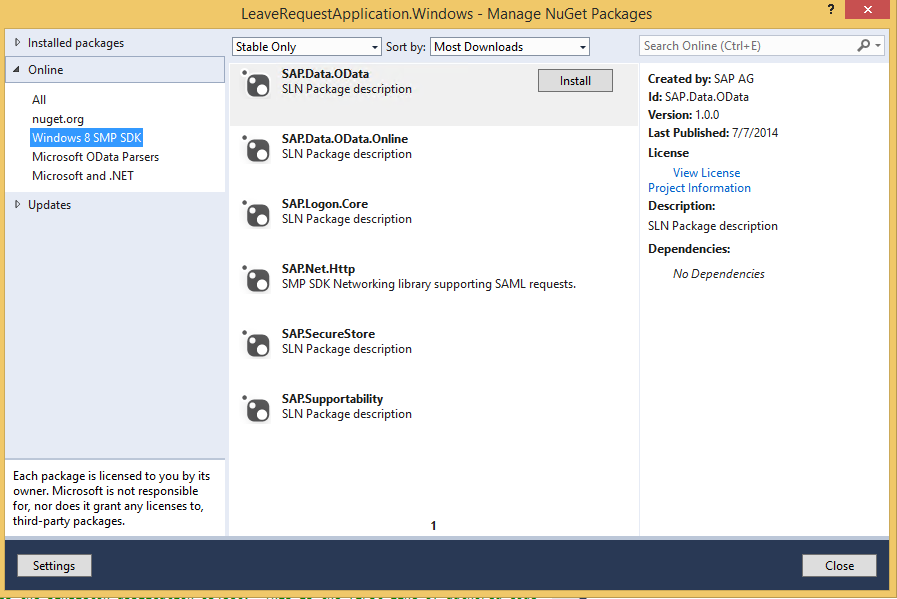
1. Create a new Windows Store Universal project by clicking on the New Project… link. Enter a name for the project.



1. To add references to the Windows project, right click on References and select Manage NuGet Packages…



1. Select the package source on the left pane that you created previously. Select the package that you need to add as a reference and click Install. NuGet Package Manager installs all dependent packages for you automatically. In addition, the proper package for the specified platform is installed.



1. In addition to the SAP NuGet packages, the developer also needs to add references to the Microsoft OData library packages. This can be done directly by adding the packages from the NuGet gallery. Click on nuget.org on the left pane and search for ODataLib. From the Search Results, install the package ODataLib for OData v1-3 (The version is 5.6.2). This will install all the dependent packages. Click I Accept to follow the prompts to install the packages.

Machine generated alternative text: .org
Each package is licensed to you by its
owner. Microsoft is not responsible
for, nor does it grant any licenses to,
third-party packages.
ODatalib for WIndows Phone
Classes to serialize, deserialize and validate OData payloads.
Enables construction of OData producers and consumers.Tar...
ODataLib
Classes to serialize, deserialize and validate OData JSON
payloads. Supports OData v4 only.
EdmLib for OData vi-3
Classes to represent construct, parse, serialize and validate
entity data models. Targets NET 4.0, Silverlight 4.0, or .NET P...
System.Spatial for OData vi-3
Contains classes and methods that facilitate geography and
geometry spatial operations. Targets .NET 4.0, Silverlight 4.0...
WCF Data Services Client for OData vi-3
LINQ-enabled client API for issuing OData queries and
consuming OData payloads. Supports OData v3. Targets .NE...
i 2 b
Created by: Microsoft Corpoi
Id: Microsoft.Data,OData
Version: 5.6.2
Last Published: 8/1/2014
Downloads: 1656106
View License
Project Information
Report Abuse
Description:
Classes to serialize, deserialize and
validate OData payloads. Enables
construction of OData producers and
consumers. Targets .NET 4.0, Silverlight
4.0 or NET Portable Lib with support
for NET 4.0, SL 5.0, Win Phone 8, Win
Phone 8.1, and Win 8. Localized for CHS,
CHT, DEU, ESN, FRA, ITA, JPN, KOR and
RUS.
Tags: wcf data services odata odatalib
edmlib spatial adonet S entity
framework open protocol wcfds
wcfdataservices dataservices
Dependencies:
System.Spatial (t 5.6.2)
Microsoft.DataEdm (= 5.62)
t...a.. :s_..... ...L..... . . LS..... _..L.
settings
L ‘.JL/ULO Parsers
Windows SMP SDK
Microsoft and .NET
ODatalib for OData vi-3
Classes to serialize,
OData JSON payloads.
Install
P Updates
Q
Q
QMachine generated alternative text: License Acceptance fl
The following package(s) require a click-to-accept license:
System.Spatial (Author Microsoft corporation)
I Vie’. LicEnzE
Mkrosoft.Data.Edm (Author Microsoft Corporation)
View License
MicrosoftData.OData (Author Microsoft Corporation)
View License
By clicking “I Accept” you agree to the license terms for the package
(s) listed above. If you do not agree to the license terms, click “I
Decline.”
I ecIine I Accept

1. Follow steps 3 and 4 to add references to the Windows 8.1 Phone project as well.