MATRIX42
SMARTER WORKSPACE – BETTER LIFE

Silverback Best Practices

Windows (Phone) 8.1 PFN Substitution

Version 3.0.0

07/05/2015

Copyright © 2000 - 2015 Matrix42 AG

This documentation is copyright protected. All rights are reserved by Matrix42 AG.

Any other use, in particular the disclosure to third parties, storage in a data system, dissemination, processing, presentation, performance and demonstration are prohibited. This applies to the entire document, as well as parts thereof.

Subject to change. Reprint, also in excerpts, is permitted only with the written consent of Matrix42 AG.

The software described in this document is subject to a permanent development due to which there may be differences in the documentation and the actual software. This documentation is not entitled to the actual functionality of the software.

Apple and **Mac OS X** are registered trademarks of Apple Inc.
Citrix® software or **Citrix® server** are Trademarks and Registered Trademarks of Citrix Systems, Inc. in the United States and other countries.
cygwin is copyrighted by Red Hat Inc. 1996-2003.
expat is copyrighted by Thai Open Source Software Center Ltd.
gSOAP is copyrighted by Robert A. van Engelen, Genivia, Inc. All rights reserved.
Iconv is copyrighted by 1999-2003 Free Software Foundation, Inc.
Iperf is copyrighted by the University of Illinois, except for the gnu_getopt.c, gnu_getopt_long.c, gnu_getopt.h files, and inet_aton.c, which are under the GNU General Public License.
Libmspack (C) 2003-2004 by Stuart Caie <kyzer@4u.net>.
OpenSSL This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit.
PuTTY is copyrighted by Simon Tatham. Portions copyright Robert de Bath, Joris van Rantwijk, Delian Delchev, Andreas Schultz, Jeroen Massar, Wez Furlong, Nicolas Barry, Justin Bradford, Ben Harris, Malcolm Smith, Ahmad Khalifa, Markus Kuhn, and CORE SDI S.A.
RSA Data Security, Inc. MD5 Message-Digest Algorithm is copyrighted by RSA Data Security Inc. Created 1991. All rights reserved.
rsync is an open source utility that provides fast incremental file transfer. rsync is freely available under the GNU General Public License version 2.
runcontrol The Initial Developer of the Original Code is James Clark. Portions created by James Clark are Copyright (c) 1998 James Clark. All rights reserved.
SNMP++ Copyright (c) 1996 Hewlett-Packard Company.
VMware, the **VMware "boxes" logo and design, Virtual SMP, VMotion vSphere, vSphere Hypervisor (ESXi), ESX, View, ThinApp, vCenter** and **vCloud** are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.
Windows, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, and **Windows Server 2012** are registered trademarks of Microsoft Corporation.

Others, at this point not explicitly listed, company, brand and product names are trademarks or registered trademarks of their respective owners and are subject to trademark protection.

Table of Contents

1. Overview 4
2. Microsoft Push Network Settings Substitution Process 5
2.1. Before you start 5
2.2. Process Overview 5
2.3. Create a Windows Store dummy enterprise app 5
2.4. Create a Microsoft Package Family Name (PFN) 7
2.5. Configure Silverback for Windows (Phone) 8.1 Management 11
2.6. Prepare MDM queue for Windows devices 13
3. Appendix 15
3.1. I forgot to record my dummy enterprise settings 15
3.1.1. How to get my enterprise app details 15
3.1.2. How to get the PFN name from within Visual Studio 16

Windows (Phone) 8.1 PFN Substitution

1. Overview

Silverback by Matrix42 uses the Microsoft Push Network Service (MPNS) mechanism to remotely trigger Windows (Phone) 8.1 devices. Once triggered the MDM registered device checks into the Silverback server and receives the MDM commands that are queued for the device (e.g. lock the device or install an Exchange profile).

In order to enable the Microsoft Push Network Service function, current Silverback by Matrix42 customers need to register a Windows Store dummy enterprise application with their Microsoft Application Developer Account.

After the dummy enterprise app creation process the administrator needs to apply a couple of settings – including the Microsoft PFN (Package Family Name) - in the Silverback solution to enable the push network function. This is either done in the Silverback Configurator tool (up to Silverback version 4.6) or in the Silverback Web Settings starting with version 4.7.

Why substitute the Microsoft Push Network Services settings?

There are a couple of situations where you need to substitute your MPNS settings:

1 The customer wants to merge multiple Microsoft Application Developer Accounts.

2 The Microsoft Application Developer Account expires or may get deleted because someone forgot to extend the account.

3 The Windows dummy enterprise app settings used in Silverback become invalid because the dummy app was not prolonged within 12 months and Microsoft deleted the reserved dummy application name.

What are the risks

While situation #1 above can be treated in a planned change management manner without impacting the quality of device management, situations #2 and #3 can happen unplanned. If unplanned the impact is that Windows device cannot be remotely triggered to check-in to the Silverback server for ad-hoc MDM actions initiated manually by the administrator or automatically by regular Silverback server routines. Even though Windows devices frequently check-in regularly (default is every 120 minutes) this might be too long if the customer needs to wipe corporate data from a lost device immediately.

Important: The registered devices within Silverback by Matrix42 will remain under MDM management control and do not need to be re-enrolled after executing the MPNS settings substitution process.

How does MPNS substitution affect end-users?

The substitution for end-users with MDM-registered devices is transparent. Users will not be impacted and can continue to use their Windows devices.

2. Microsoft Push Network Settings Substitution Process

2.1. Before you start

To complete all steps below you will need to meet the following pre-requisites:

- **Access to your Microsoft Developer Account:**
  Required to create a new Windows Store app
- **Visual Studio 2013 SP2 and higher:**
  Required for app and PFN creation process.
  You can use and download the latest and cost free Express 2013 version
  (https://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx)
- **Enable Java Script**
  During Visual Studio process a web browser based wizard will open that requires Java script

2.2. Process Overview

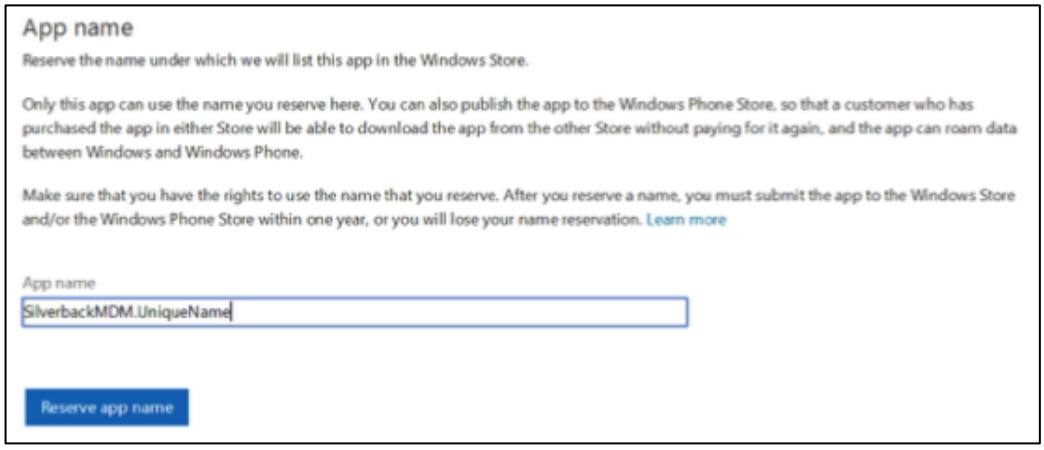
The main process steps are the following:

- Create a new Windows Store dummy enterprise app
- Change Silverback server settings
- Re-start IIS and Silverback Windows services
- Execute an SQL script against your Silverback SQL database

2.3. Create a Windows Store dummy enterprise app

1. Go to the Windows Dev Center
   1. https://appdev.microsoft.com/StorePortals/en-US/Home/Index
2. On the top right log in with your Microsoft developer account
3. Once logged in, navigate to **Windows Store Apps**
4. Click **Submit an app**
5. Click on **App Name** and enter a unique name for the app.
6. Note: This should be named in an obvious way, so that future users can easily tell what this is for. You may want to number your push app. For example: ***CompanyName.SilverbackPush02***

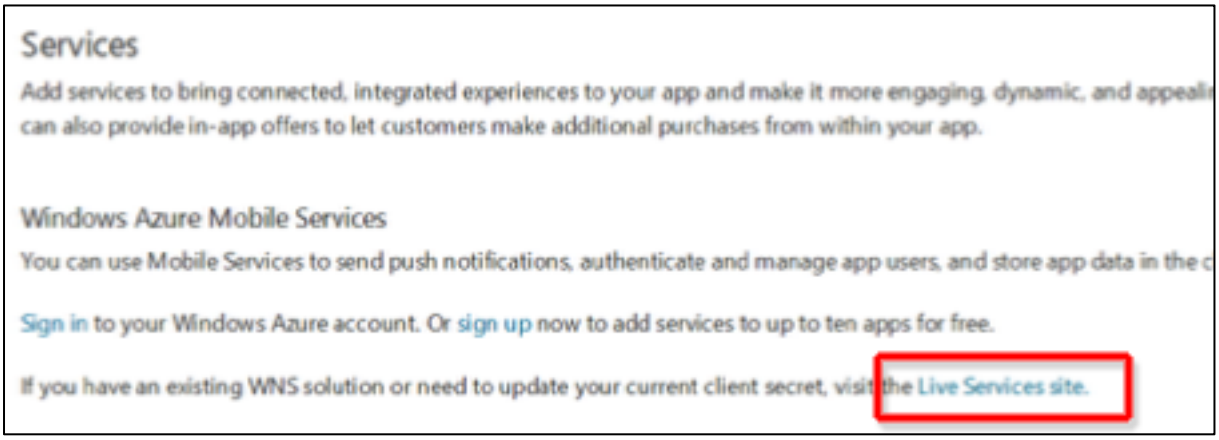
Windows (Phone) 8.1 PFN Substitution



7 Click **Save** on the next screen

8 From the next screen click on **Services**

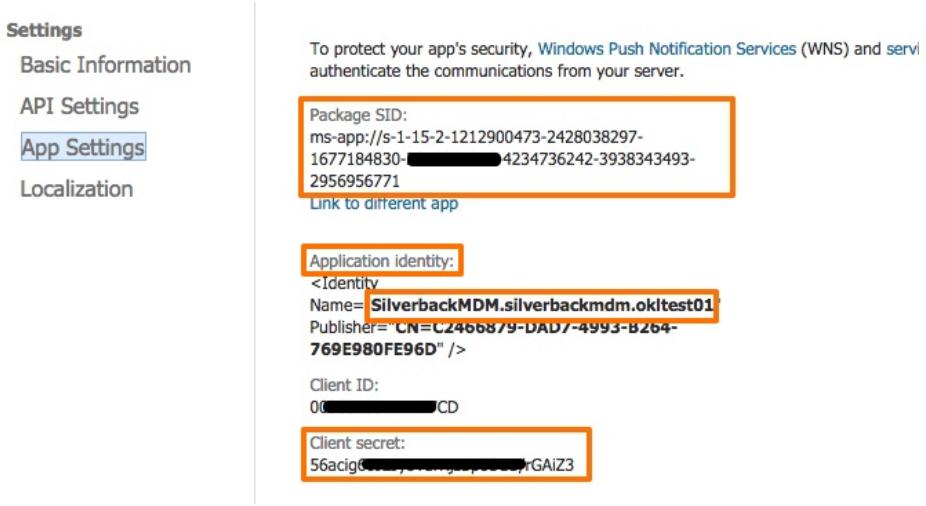
9 From the Services page, click on the "**Live Services Site**" link in the Azure Mobile Services section.



- The screen will show several pieces of information that will be needed later. **Be sure to record the following information** for when you need to configure the Silverback server later in this document.
  - Package SID
  - Application Identity
  - Client secret

See example screenshot below:

Windows (Phone) 8.1 PFN Substitution



2.4. Create a Microsoft Package Family Name (PFN)

This section of the document requires the use of Visual Studio 2013 to create the dummy app and register it for WNS Push notifications.

On the computer running Visual Studio 2013 SP2 and higher, perform the following steps:

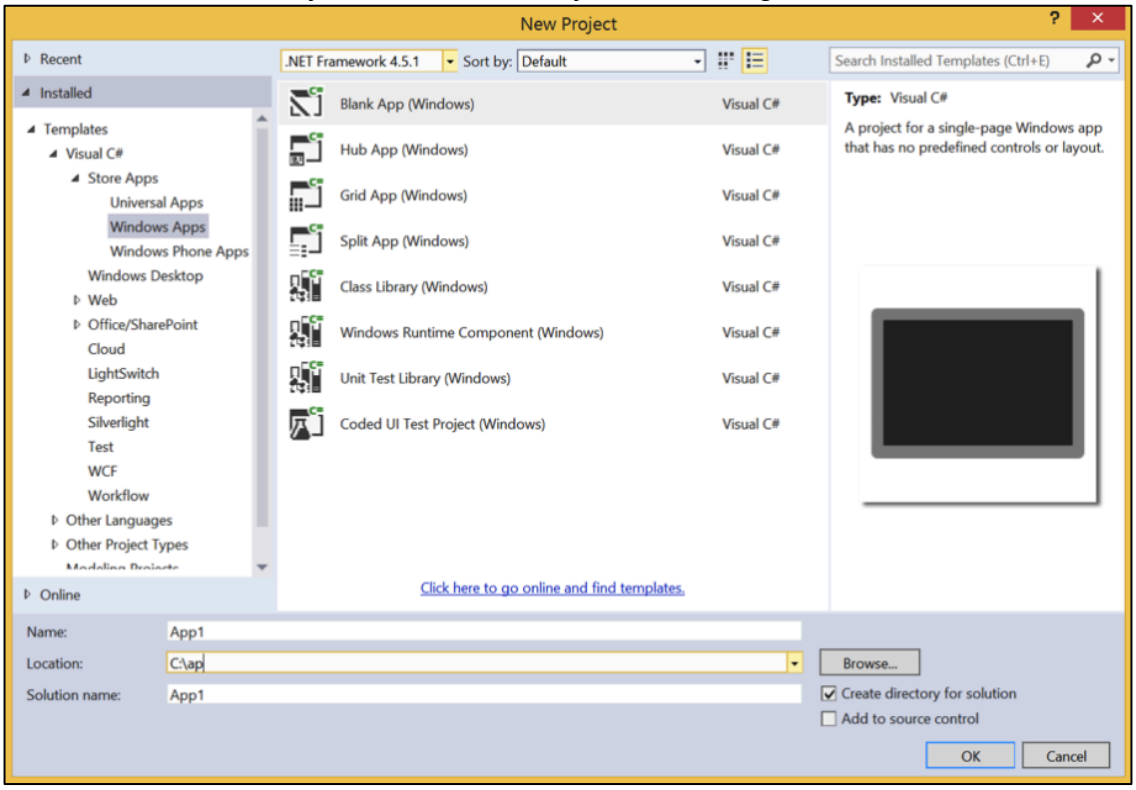
1 Launch Visual Studio 2013

2 Create a **New Project**

3 Select **Templates -> Visual C# -> Store Apps -> Windows Apps**

4 Create a "Blank Store App (Windows)" project and fill out the Name Location and Solution Name

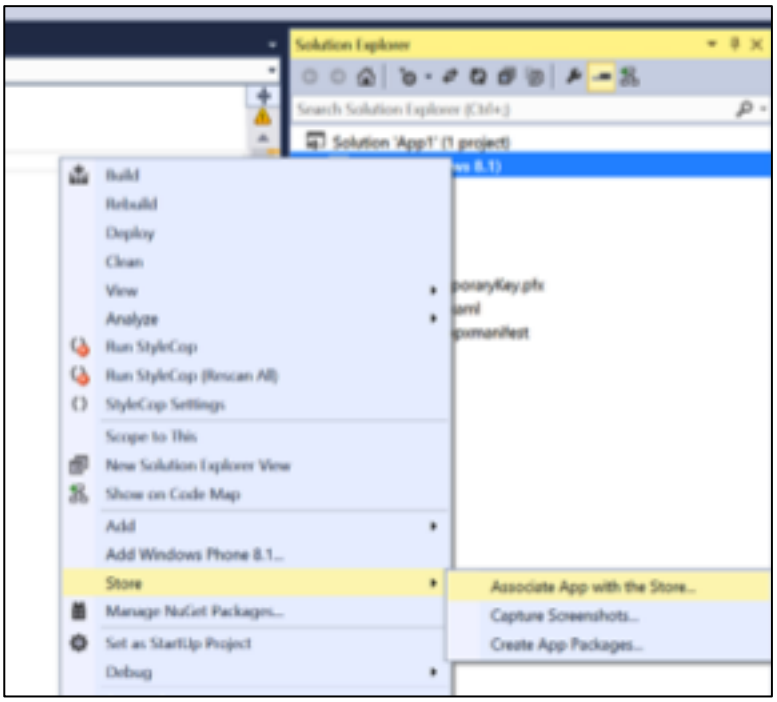
   1 This Name and location information is not critical, but the project should be saved in a location that is easily accessible for any future changes



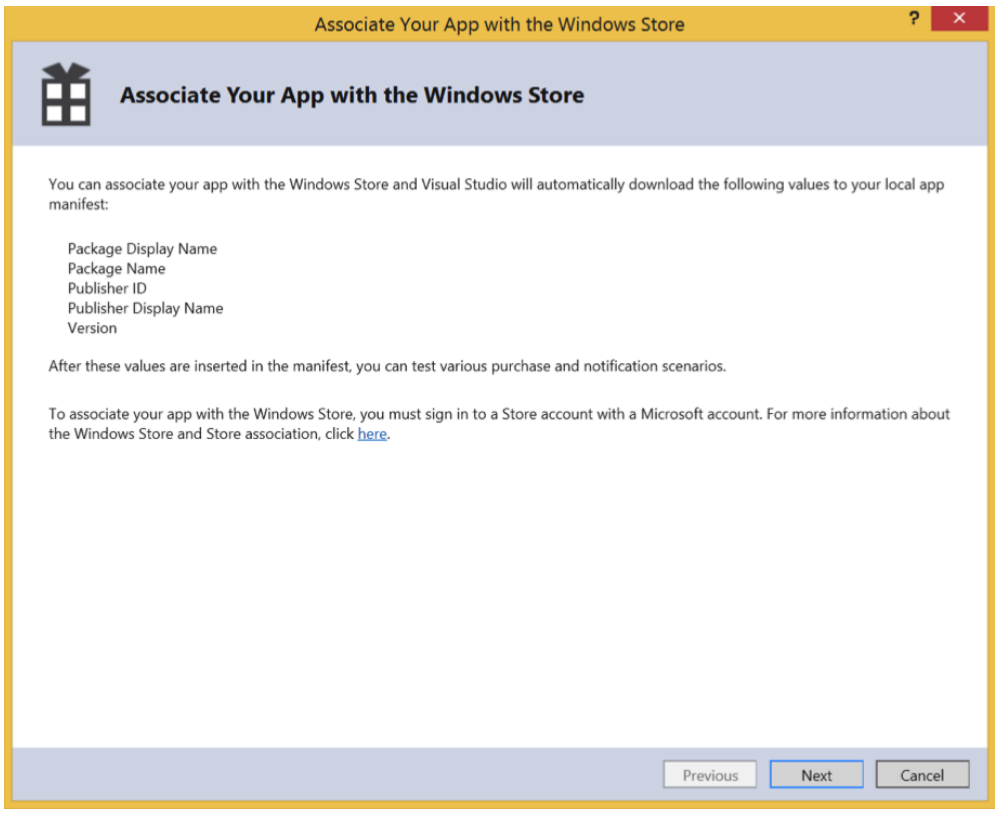
Windows (Phone) 8.1 PFN Substitution

5 When the project opens, **right click on the project in the Solution Explorer**

6 Select **Store > Associate App with the Store**...

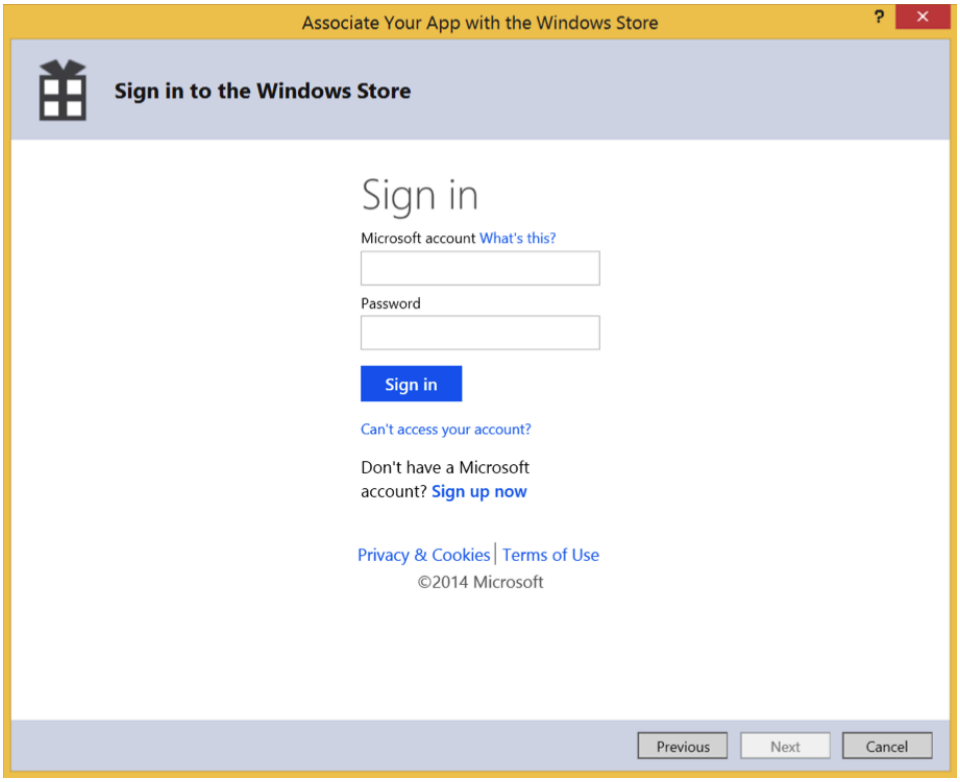


7 Click Next

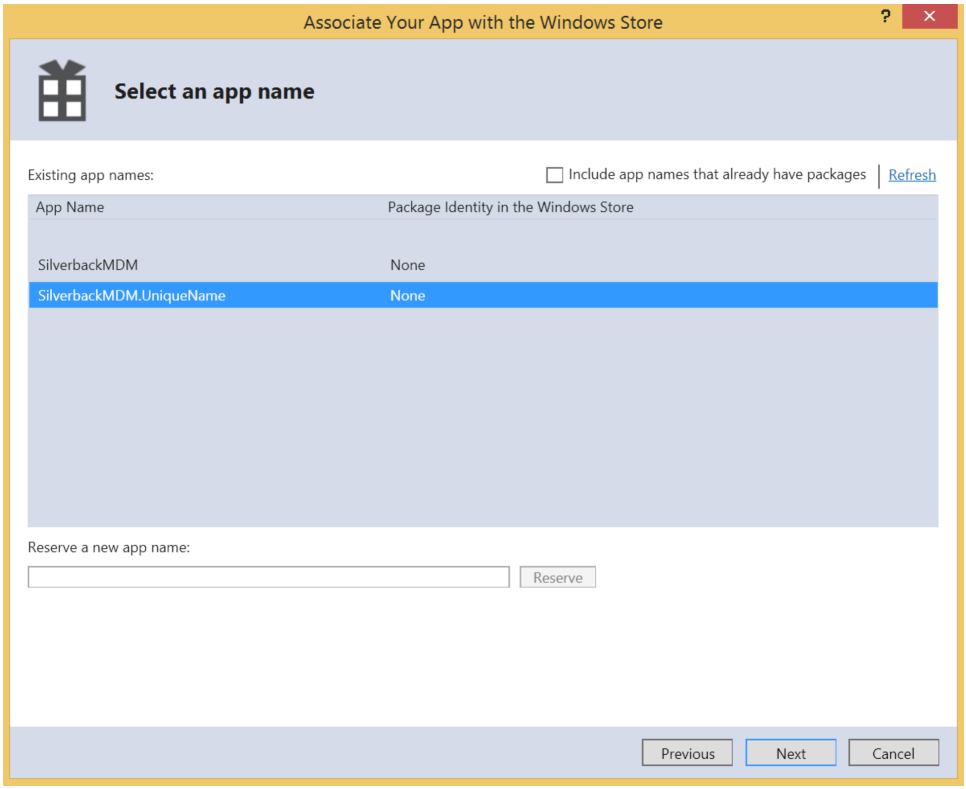


Windows (Phone) 8.1 PFN Substitution

8 Sign in with the Microsoft Account used to create the App in the previous section

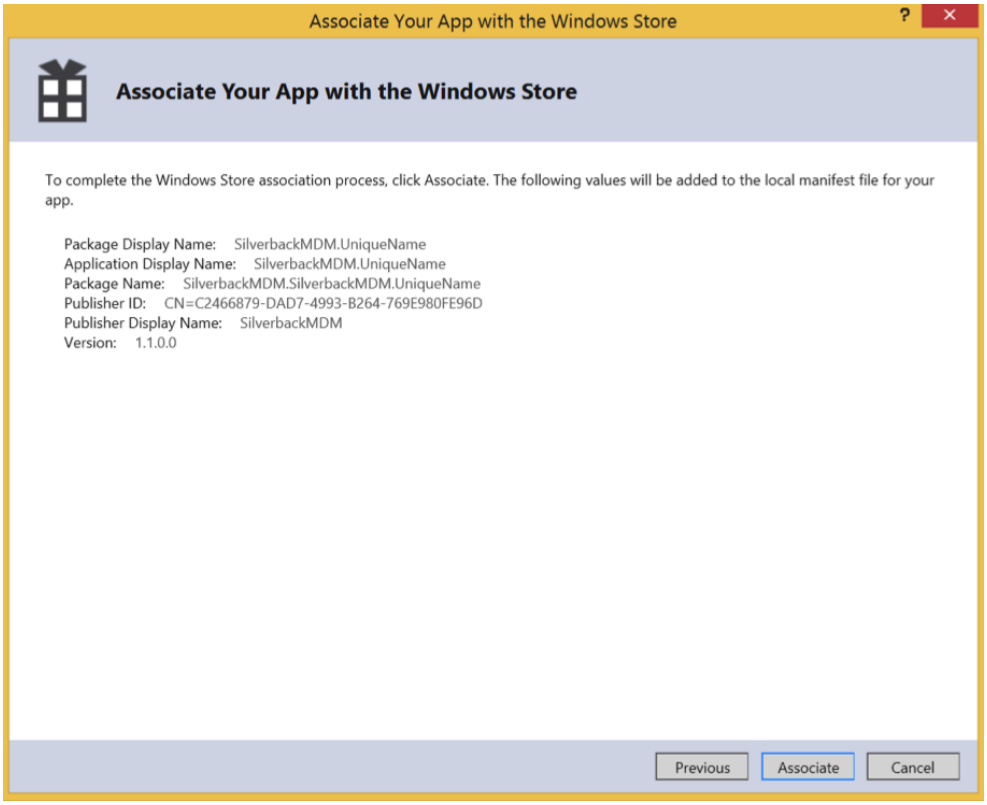


9 Choose the Created App Name from the list, Click Next



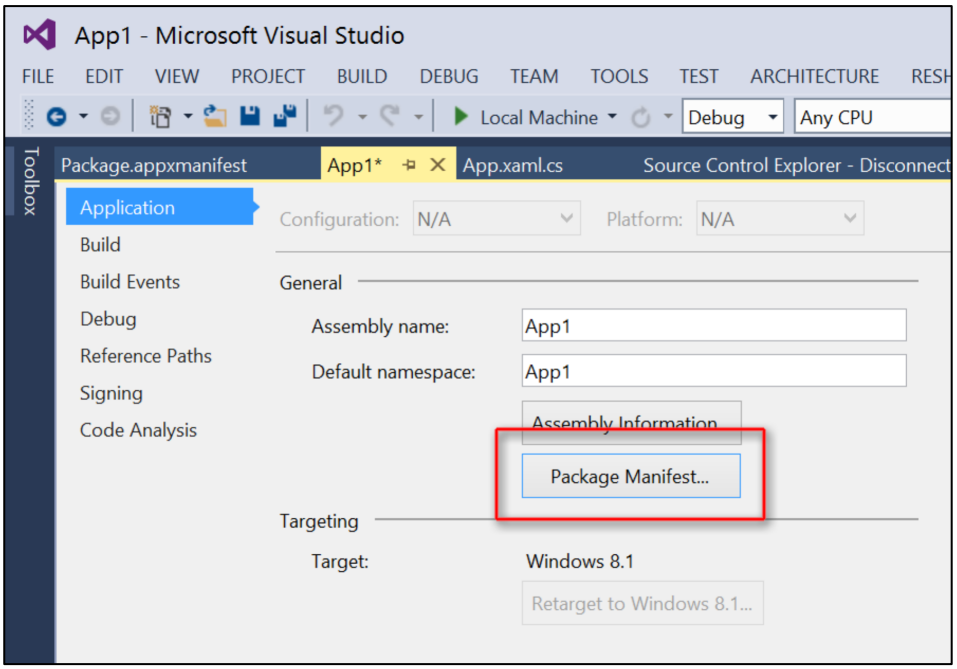
Windows (Phone) 8.1 PFN Substitution

10 Click Associate



11 Right Click the project again and select Properties

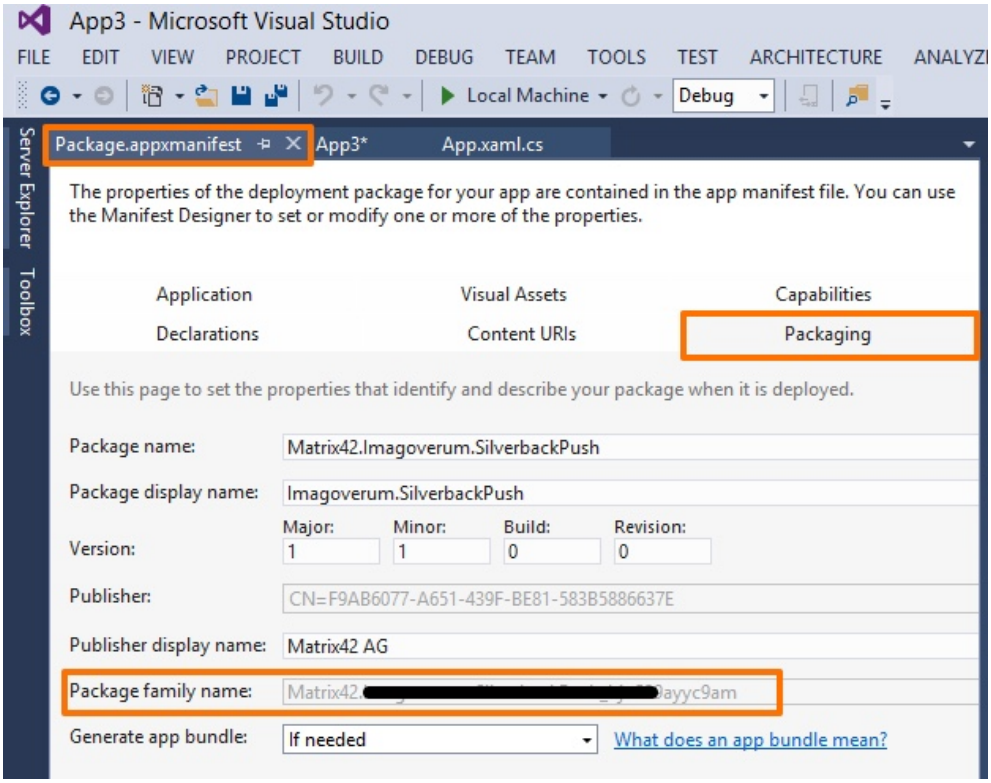
12 In the pop up, Click Package Manifest from the Application section



Windows (Phone) 8.1 PFN Substitution

13 Click on the **Packaging** tab on the middle right side

14 **Record the "Package family name" (PFN)** for use in the Silverback server configuration in the next section below.



2.5. Configure Silverback for Windows (Phone) 8.1 Management

Using the information generated from the above steps, you are now ready to configure the Silverback server for Windows 8.1 Microsoft Push Network management. The following settings needs to be mapped into the Silverback server settings:

Silverback (SB) Value	Microsoft (MS) Value
WNS App ID	Name part of 'Application identity'
WNS Package Family Name (PFN)	Package Family Name
WNS App Store URI	Package SID
WNS App Secret	Client secret

Using the information recorded from the above steps, you are now ready to configure the Silverback server settings with updated Windows 8.1 Microsoft Push Network values. The following settings needs to be mapped into the Silverback server settings:

Important:
Please change all required Silverback server settings before you save them!

Windows (Phone) 8.1 PFN Substitution

Following these steps will configure your Silverback server:

1. If you have Silverback 4.7 and higher: Login to the Silverback admin console (e.g. https://silverbackserver.company.com/admin) with a console user account with the role of “Settings Administrator”
2. If you have Silverback 4.6 and lower: Start the Silverback Configurator tool on your Silverback server via RDP.
3. Click the **“WP Settings” section**, and populate the values into the ‘’Windows 8.1 Family device settings” section.
   1. WNS App ID – Use the **Name** part within the “quotes” of the MS application ID
   2. WNS Package Family Name (PFN) – Use the Microsoft PFN
   3. WNS App Store URI – Use the Microsoft Package SID
   4. WNS App Secret – Use the Microsoft Client Secret
4. If you have Silverback 4.7 and higher: Saving the web settings
5. If you have Silverback 4.6 and lower: Save the settings and export the new configs
6. RDP to the Silverback server and re-start IIS and all Silverback Windows services. Alternatively you can restart the entire Silverback server.

2.6. Prepare MDM queue for Windows devices

Finally you execute an SQL script against your Silverback SQL database. The purpose of the script is to create MDM queues for all Windows devices to get their new PFN information.

Create MDM queue commands for Windows Phone devices

Log into the Microsoft SQL Server Management Studio and run the following script against the Silverback MDM database:

```
INSERT INTO SyncMlCommandQueue(
       [DeviceId]
      ,[SyncMlCommandTypeId]
      ,[Priority]
      ,[SyncMlCommandStatusId]
      ,[QueuedTime]
      ,[Active])
  SELECT DeviceId,
  50,
  15,
  1,
  GETDATE(),
  1
  FROM Devices WHERE DeviceTypeName IN ('WP81') AND DeviceStatusId
IN (2,3)
```

When executed successfully you will get the amount of rows (devices) affected.

Create MDM queue commands for Windows desktop devices

Log into the Microsoft SQL Server Management Studio and run the following script against the Silverback MDM database:

```
INSERT INTO SyncMlCommandQueue(
       [DeviceId]
      ,[SyncMlCommandTypeId]
      ,[Priority]
      ,[SyncMlCommandStatusId]
      ,[QueuedTime]
      ,[Active])
  SELECT DeviceId,
  250,
  15,
  1,
  GETDATE(),
  1
  FROM Devices WHERE DeviceTypeName IN ('Windows81') AND
DeviceStatusId IN (2,3)
```

When executed successfully you will get the amount of rows (devices) affected.

Windows (Phone) 8.1 PFN Substitution

Monitor MDM queue commands for Windows Phone devices

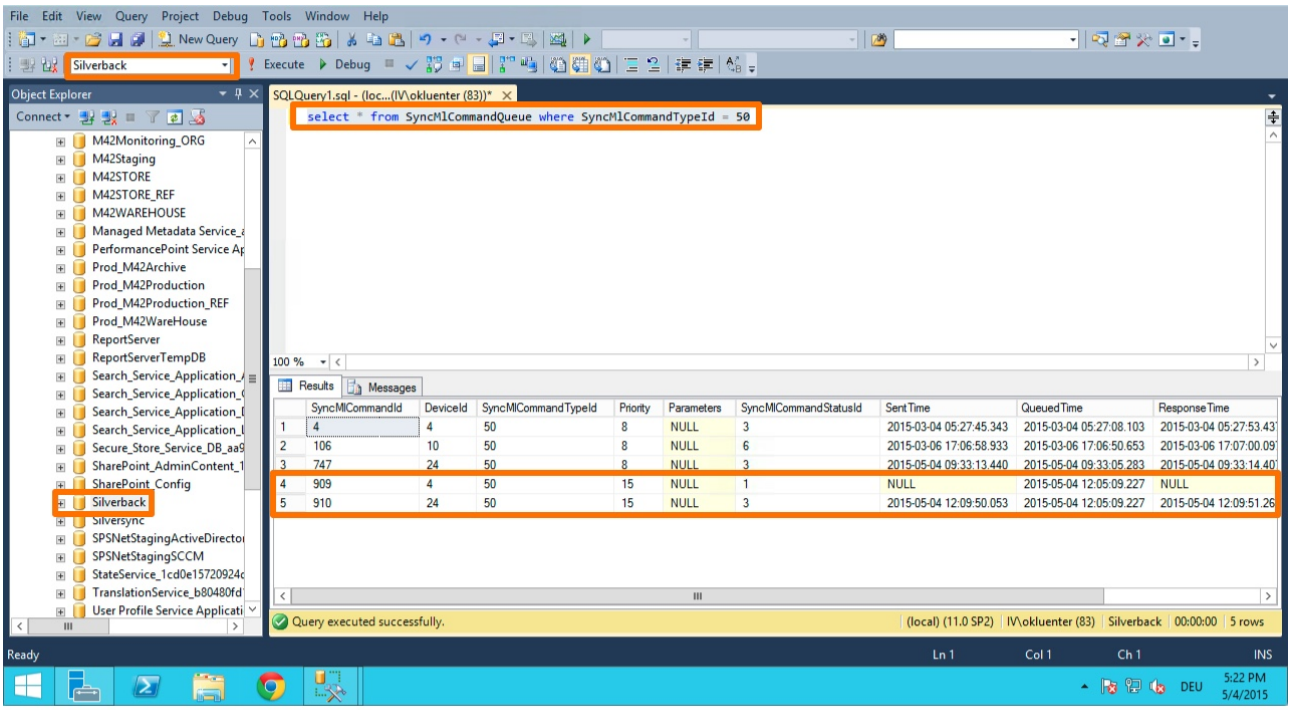
Run the following SQL command against your Silverback MDM database to monitor the outstanding and migrated Windows Phone devices:

```
select * from SyncMlCommandQueue where SyncMlCommandTypeId = 50
```

All created queues will have the actual date & time in the ‘QueueTime’ column.

Migrated devices will have a ‘SentTime’ and ‘RespondTime’ value when the PFN has been changed.

Outstanding devices will have ‘SentTime’ and ‘RespondTime’ value equals ‘NULL’.



Monitor MDM queue commands for Windows desktop devices

Run the following SQL command against your Silverback MDM database to monitor the outstanding and migrated Windows desktop devices:

```
select * from SyncMlCommandQueue where SyncMlCommandTypeId = 250
```

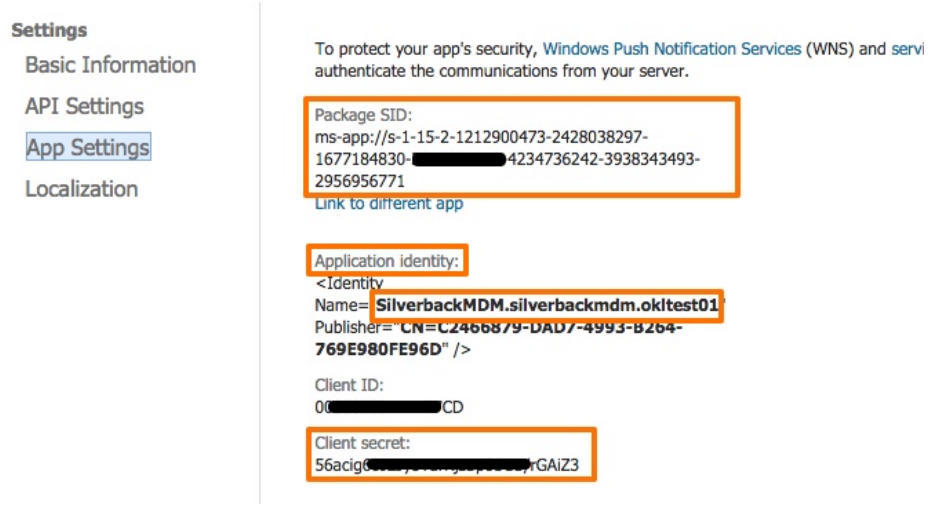
3. Appendix

3.1. I forgot to record my dummy enterprise settings

3.1.1. How to get my enterprise app details

Log into your Microsoft Application Development Center (https://appdev.microsoft.com/StorePortals/en-US/Home/Index). Follow the ***Dashboard -> Windows Store Apps*** to get an overview of you apps. You will find registered but unpublished apps in the ***Apps in progress*** section.

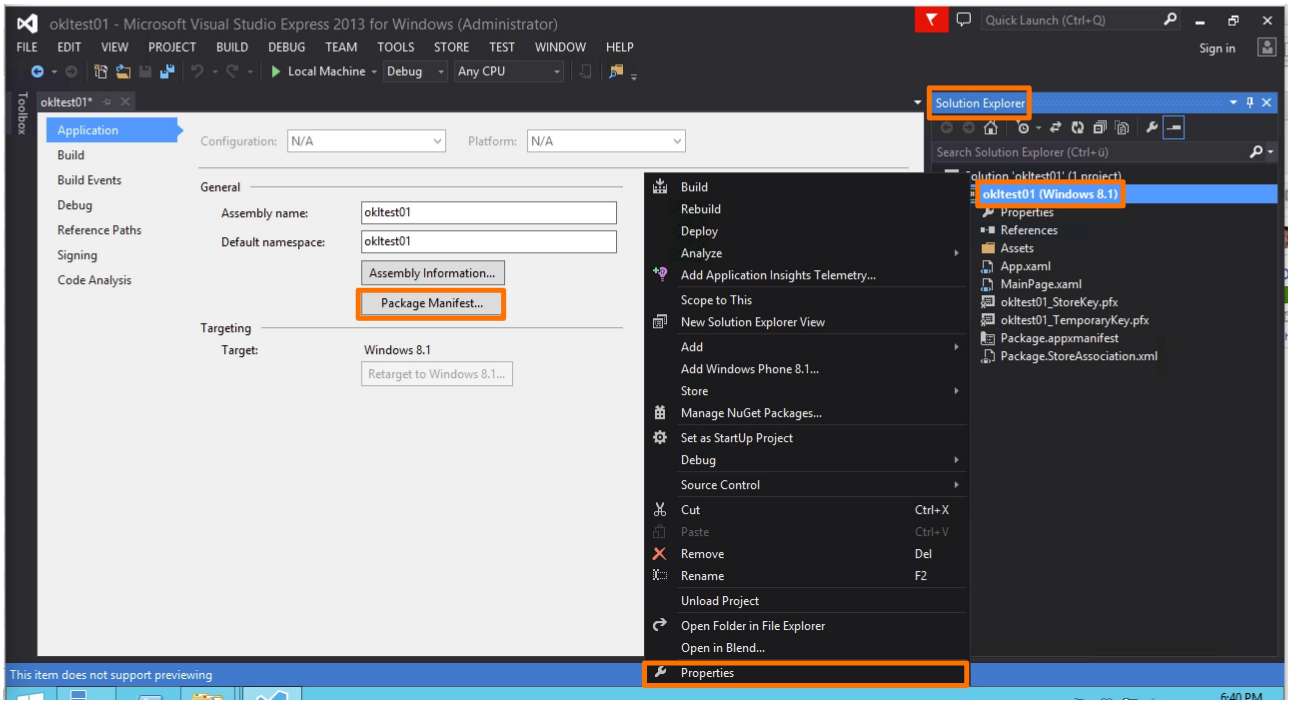
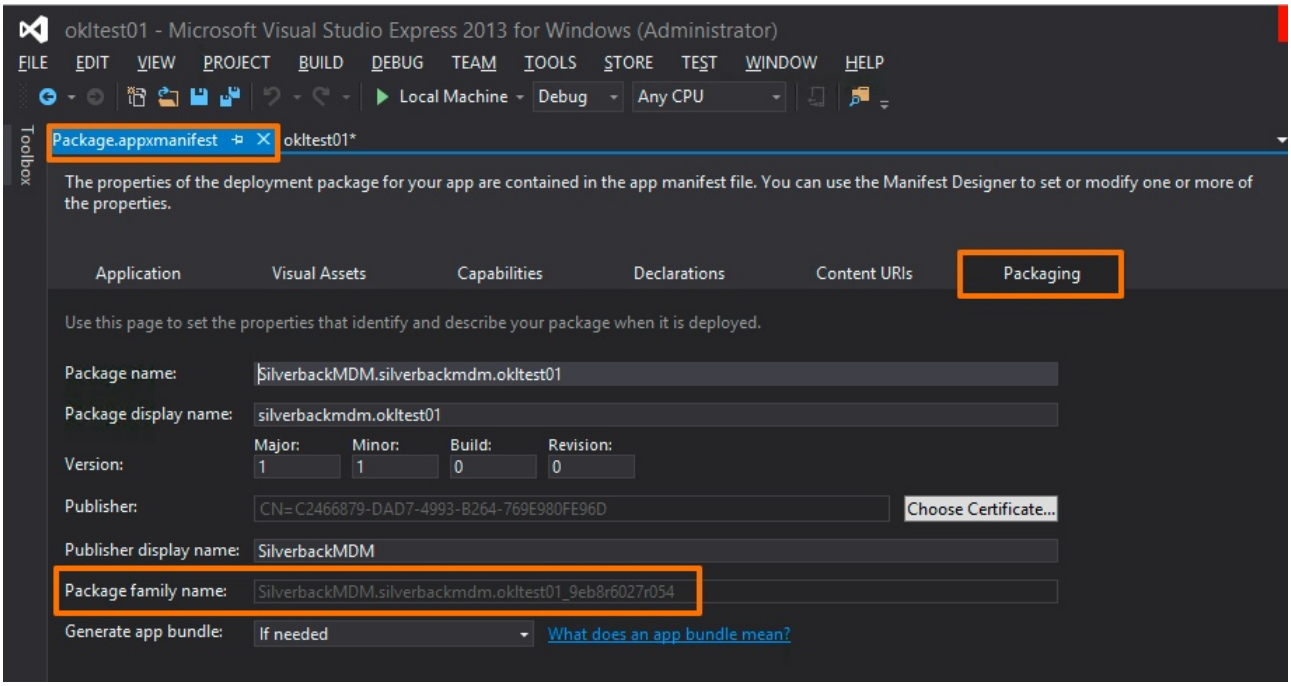
Click on EDIT for the dummy app you created. Click on **Services -> Live Services Site**. To obtain the required values – see example below.



Windows (Phone) 8.1 PFN Substitution

3.1.2. How to get the PFN name from within Visual Studio

In Visual Studio open your dummy app project. Switch to the 'Solution Explorer' overview. Right click on the project in the Solution Explorer and select 'Properties'. Afterwards click on 'Package Manifest' in the main window.



Finally switch to the 'Packaging' tab. Now you will see the Package family name at the bottom in the main window.