

# Windows Azure Toolkit For Devices

## 1/ Présentation du toolkit

---

### Windows Azure

Windows Azure est le nom de la plateforme « dans le nuage » de Microsoft. Cette plateforme permet un accès facile à un ensemble de ressources (serveurs, applications, stockage, services) pouvant être rapidement mises à disposition des utilisateurs avec un minimum d'administration. Azure offre un hébergement hautement disponible et met à disposition un ensemble de briques applicatives prêtes à l'emploi pour accélérer le développement d'applications, augmenter la qualité et l'élasticité.

Si vous voulez bien comprendre comment développer des applications pour Windows Azure, je vous invite à passer par le coach Windows Azure que vous trouverez à cette adresse :

<http://msdn.microsoft.com/fr-fr/windowsazure/msdn.coach.azure.aspx>

Parmi les différents services fournis par Azure, il y en a plusieurs qui sont liés au stockage de données. Ce stockage peut se faire de différentes manières :

Type de stockage	Utilisation
Blob	C'est un conteneur simple de données brutes. Idéal pour stocker un ensemble de fichiers
Table	C'est un conteneur d'enregistrements de type dictionnaire (clé/valeur) avec une clé de partition.
Queue	File d'attente contenant des messages en attente de traitement
SQL Azure	Pour stocker des données grâce à un moteur relationnel

Windows Azure propose aussi un service d'identification qui simplifie énormément la gestion des utilisateurs autorisés à accéder aux services offerts par une application Azure.

Coté application, justement, Windows Azure en gère deux grands types qu'on appelle des rôles :

Type de rôle	Utilisation
Web Role	Application web
Compute Role	Unité de traitement construite comme un service

### Philosophie du toolkit Azure

Les applications mobiles ont très souvent besoin d'accéder à des informations distantes et de nombreuses sociétés ont développé des applications pour leurs collaborateurs qui accèdent aux données de l'entreprise en situation de mobilité.

Pour se faire, il existe plusieurs solutions. La plus connue est la mise en œuvre d'une application web qui permet à l'application nomade d'accéder aux données dont elle a besoin sous réserve d'avoir montré patte blanche à la connexion, la sécurité des données de l'entreprise étant, je ne vous apprends rien, cruciale.

Comme Windows Azure propose des services de stockage (Tables, Blobs, Queues), l'accès à ces espaces de stockage depuis une application mobile s'avère très intéressant.

Avec ces services, plus besoin de réinventer la roue, il suffit juste de les utiliser. Mais depuis une application mobile, comment avoir accès en lecture et écriture à ces espaces de stockage ? Comment éviter d'embarquer dans l'application mobile toutes les clés d'accès aux comptes Windows Azure ? Comment assurer la sécurité d'accès aux données ?

Le toolkit Windows Azure est la réponse à toutes ces questions.

Nous devons ce toolkit à Wade Wegner, Technical Evangelist Lead chez Microsoft Corp. Wade dirige une équipe d'évangélistes techniques chargés d'influencer et de conduire la stratégie technique de Microsoft pour la plateforme Windows Azure (pour faire simple, c'est AzureMan et quand on regarde sa manière de coder, c'est aussi CodeMan si vous voyez ce que je veux dire).



Voici quelques liens si vous voulez suivre ses exploits :

Son blog	<a href="http://www.wadewegner.com">http://www.wadewegner.com</a>
Son émission sur Channel 9	<a href="http://channel9.msdn.com/Shows/Cloud+Cover">http://channel9.msdn.com/Shows/Cloud+Cover</a>
Sur Twitter	<a href="http://twitter.com/WadeWegner">http://twitter.com/WadeWegner</a>

## Les plateformes supportées

Microsoft propose gratuitement ce toolkit pour trois plateformes différentes qui sont Windows Phone, c'est la moindre des choses me direz-vous, iOS et Android.

La documentation complète sur le toolkit est disponible sur CodePlex, ainsi que les sources et les bibliothèques pour Windows Phone 7. Pour iOS et Android, les sources et les bibliothèques sont hébergées sur GitHub :

Plateforme	Site d'hébergement	URL
Windows Phone 7	CodePlex	<a href="http://watwp.codeplex.com">http://watwp.codeplex.com</a>
iOS	GitHub	<a href="https://github.com/microsoft-dpe/wa-toolkit-ios">https://github.com/microsoft-dpe/wa-toolkit-ios</a>
Android	GitHub	<a href="https://github.com/microsoft-dpe/wa-toolkit-android">https://github.com/microsoft-dpe/wa-toolkit-android</a>

## Ce que permet le Windows Azure Toolkit (WAT)

A partir de maintenant, nous allons appeler le Windows Azure Toolkit « WAT », ce sera bien plus simple et plus rapide, vous ne croyez pas ?

WAT permet de gérer depuis votre application mobile :

1. Les utilisateurs
2. L'utilisation du stockage Azure (Azure Storage)
3. L'utilisation de SQL Azure
4. L'envoi de notifications (uniquement pour les plateformes Windows Phone et iOS)

L'utilisation de ces fonctionnalités peut se faire de deux manières :

- Directement vers Azure
- En passant par un proxy (fourni par WAT)

### Accès direct vers Azure

Dans ce mode, votre application va communiquer directement avec Azure. Cela sous-entend que votre application mobile va devoir s'identifier auprès d'Azure en fournissant les clés d'accès lors des appels. Cette méthode est donc très risquée et non recommandée.

### Accès en passant par un proxy

Dans ce mode, votre application va communiquer avec un proxy et jamais directement avec la plateforme Azure (sauf pour l'accès aux blobs, mais dans ce cas l'accès se fait via un mode sécurisé qui ne dévoile pas les codes d'accès). Cela signifie que les codes d'accès ne sont plus dans l'application mobile mais dans le proxy. C'est beaucoup plus prudent comme manière de procéder, et c'est la méthode qui est préconisée. WAT propose un proxy prêt à l'emploi qui prend en charge les accès à l'Azure Storage (tables, blobs, queues) et permet de gérer très simplement les notifications.

Le proxy permet de gérer les utilisateurs autorisés à profiter de ces services. On peut donc créer des utilisateurs identifiés par un login et un mot de passe, et ensuite gérer leur connexion aux services. Le proxy possède un mode d'administration qui permet de gérer tous les aspects des services

proposés. Pour faire simple, dans ce mode on a droit de vie et de mort sur tous les utilisateurs enregistrés.

Le proxy dont nous parlons est un service web, hébergé dans Azure, fourni intégralement par WAT. Il est fonctionnel et il ne reste plus, si nécessaire, qu'à l'adapter à vos besoins (mais cela sort du cadre de cette série de tutoriels, je vous préviens tout de suite).

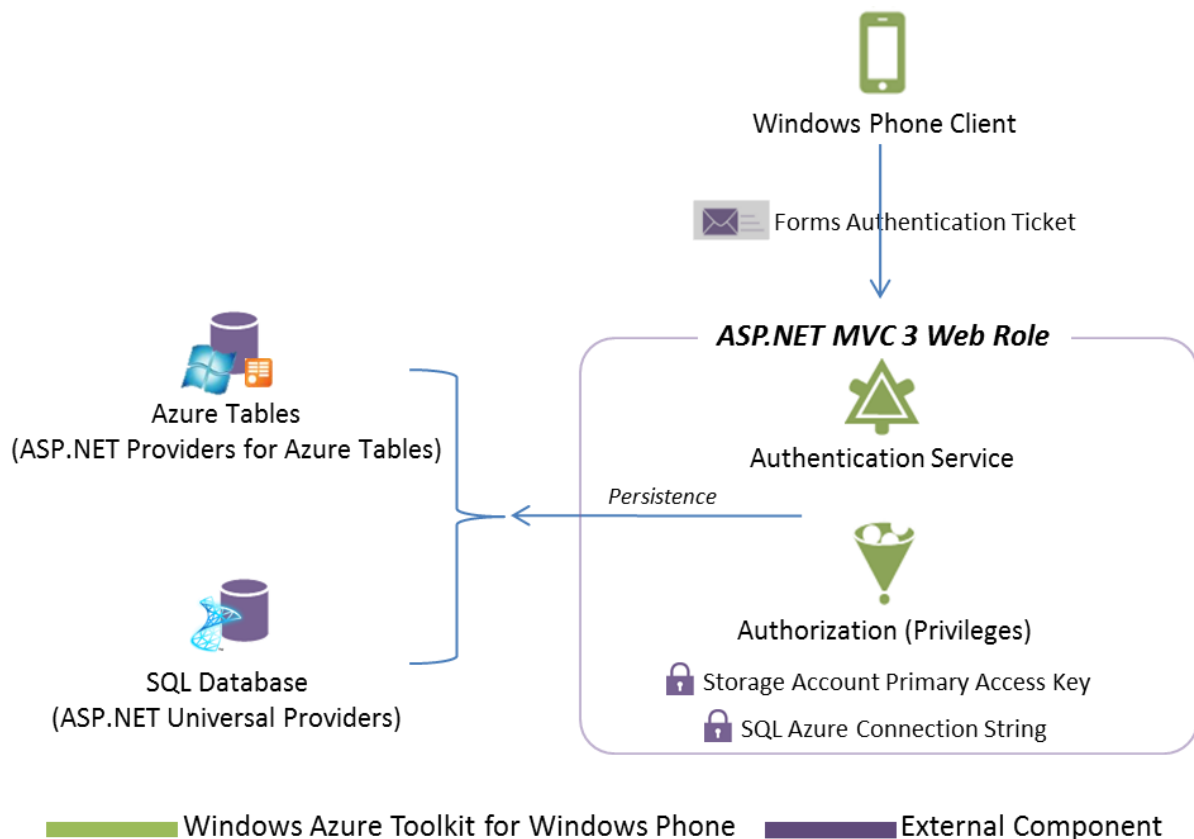
## La gestion des utilisateurs

Pour ne pas avoir à réinventer la roue à chaque fois, WAT propose une gestion des utilisateurs intégrée qui peut être mise en œuvre de deux manières suivant vos besoins :

1. Identification via ASP.NET Membership
2. Identification via Windows Azure Access Control Service (ACS)

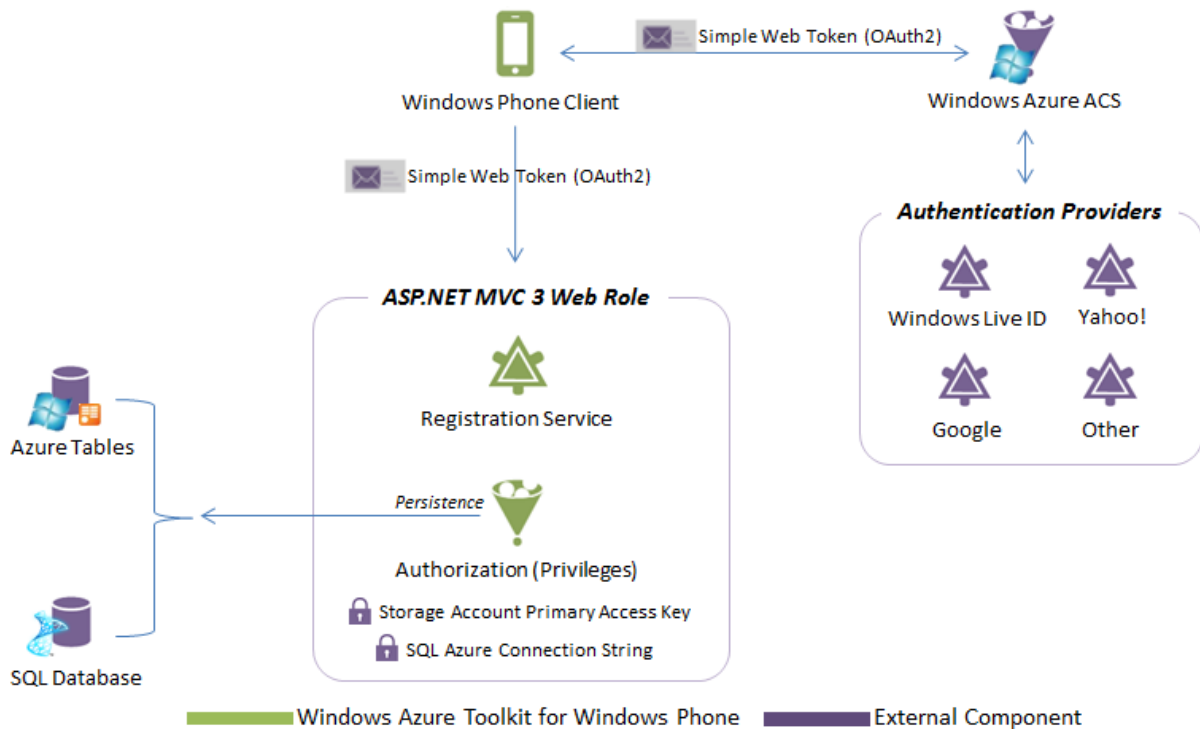
### Identification via ASP.NET Membership

Dans ce mode, les utilisateurs doivent s'enregistrer en fournissant un login et un mot de passe. L'utilisateur crée donc un compte pour bénéficier des services offerts. La création d'un compte peut se faire (c'est plus pratique tout de même) depuis l'application mobile mais aussi depuis le proxy par l'administrateur. Ensuite, l'utilisateur devra s'identifier (login + mot de passe) pour accéder aux services.



## Identification via Windows Azure Access Control Service (ACS)

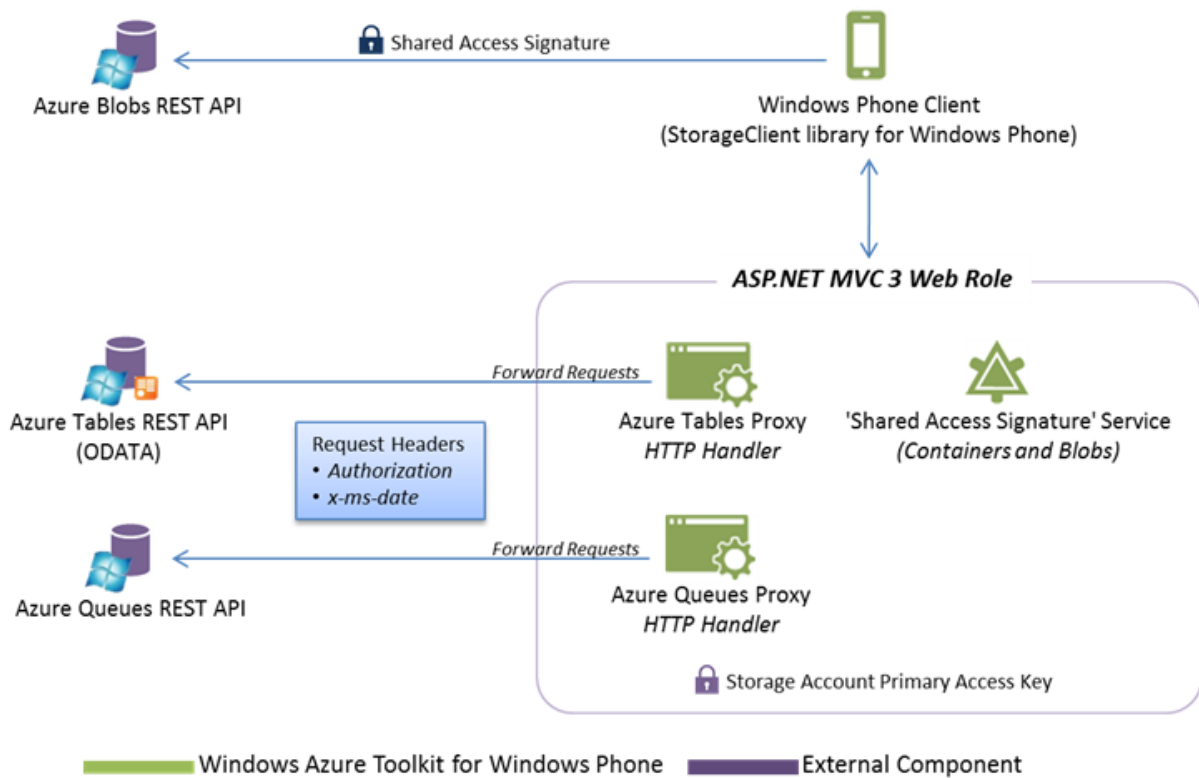
Dans ce mode, l'utilisateur qui souhaite s'enregistrer utilise les services d'un fournisseur d'identité tels que Facebook, Yahoo, Windows Live ID, Google, ou encore Active Directory. Cela lui évite d'avoir encore à créer un nouveau compte. Ce mode est très pratique mais nécessite pas mal de paramètres pour être actif.



## Accès au stockage Azure

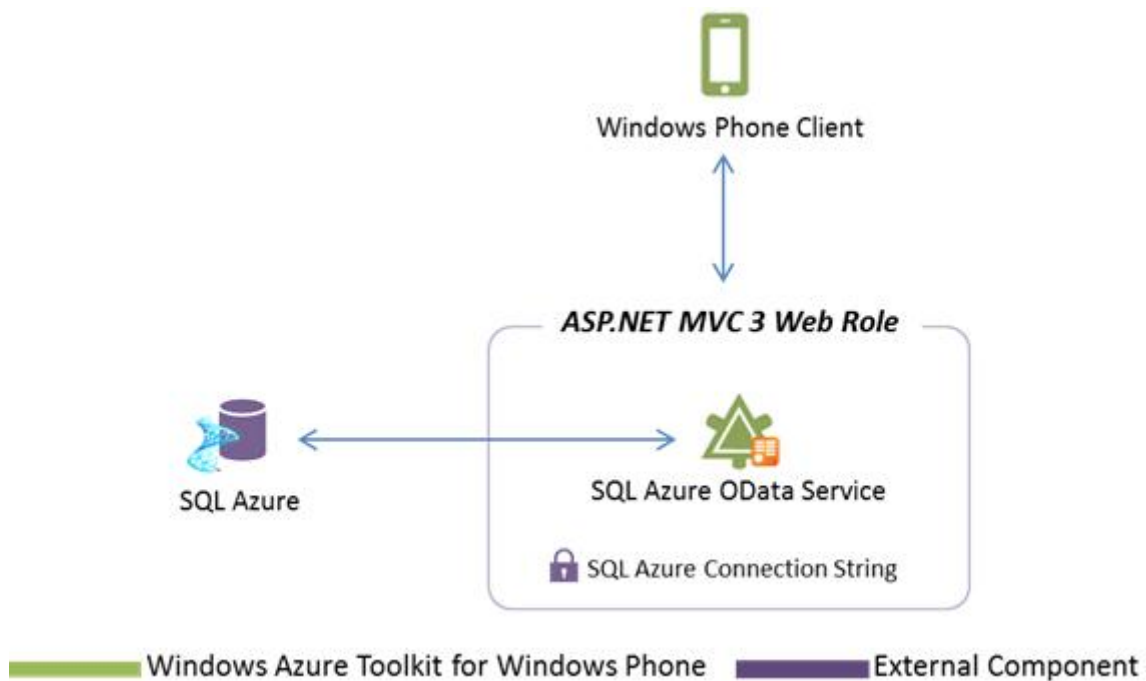
WAT permet l'accès aux blobs, aux tables et aux queues. Comme nous l'avons vu plus haut, il est préférable d'utiliser la méthode sécurisée pour effectuer toutes les opérations souhaitées.

En mode sécurisé, l'application mobile fait ses demandes au proxy. Le proxy vérifie les autorisations et si tout est ok effectue véritablement l'opération vers Azure. C'est ce qui se passe pour les tables et les queues. Dans le cas des blobs, le proxy permet de mettre en place une méthode d'accès directe mais sécurisée (Shared Access Signature). Les codes d'accès ne sont pas dévoilés et les accès se font depuis l'application mobile directement vers Azure.



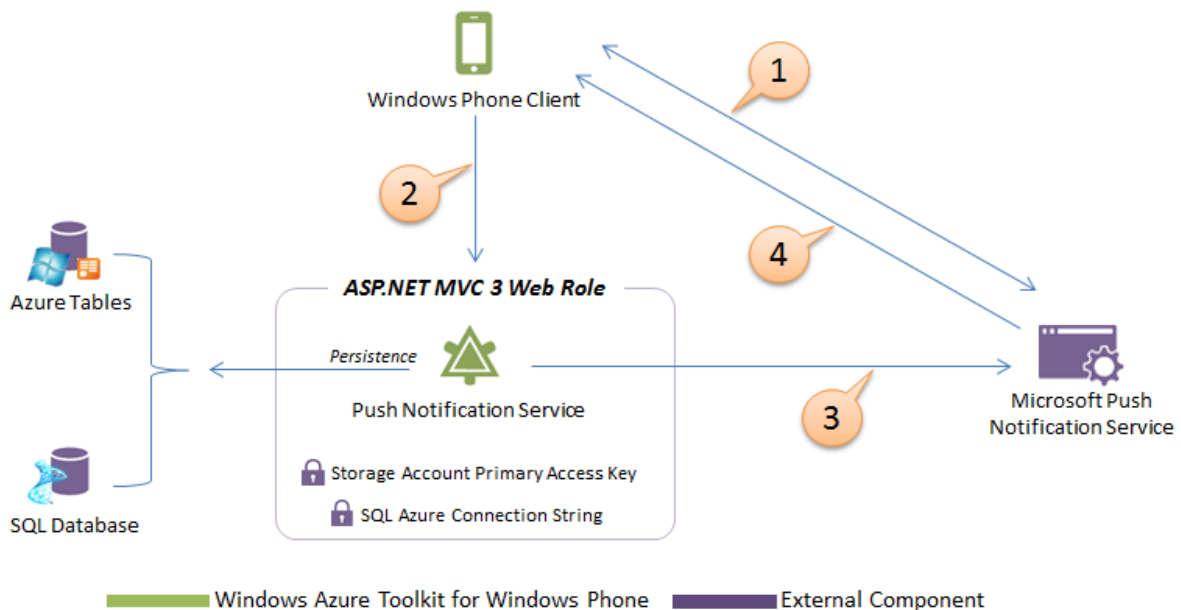
## Accès à SQL Azure

Le service SQL Azure OData est un service WCF construit « au-dessus » de SQL Azure. Il utilise Entity Framework 4.1 Code First. La version actuelle de ce service ne supporte que les opérations de lecture. En plus d'exposer la base de données SQL Azure comme un flux OData, il ajoute une couche de sécurité pour gérer les authentifications et les autorisations.



## Mise en œuvre des notifications

Le service de notification Push de Microsoft (MPN) offre aux développeurs un canal permettant d'envoyer des données à une application Windows Phone à partir d'un service web d'une manière très efficace. WAT fournit un service de notification amélioré construit au-dessus du service MPN qui sauve les notifications envoyées à l'utilisateur alors qu'il est déconnecté, et les envoie dès qu'il se connecte à nouveau. Nous verrons en détails comment mettre en œuvre les notifications dans le tutoriel qui leur sera consacré mais sachez tout de suite que pour les utiliser le proxy fourni par WAT est indispensable.



## Installation de WAT pour le développement Windows Phone

Si vous développez pour Windows Phone vous allez devoir installer WAT sur votre machine de développement pour profiter du proxy et des bibliothèques spécifiques à Windows Phone.

Vous avez le choix entre deux manières de l'installer :

- Téléchargement de l'installateur depuis CodePlex
- Téléchargement en utilisant les packages NuGet

## Utilisation des packages NuGet

NuGet est un gestionnaire de package .NET open source qui automatise toutes les tâches fastidieuses d'installation et de gestion des dépendances d'un package externe. NuGet est intégrable à Visual Studio et permet de tout faire sans quitter l'environnement de développement.

Si vous ne savez pas ce qu'est NuGet et comment l'utiliser je vous invite à lire ce tutoriel de Hinault Romaric DONFACK que vous trouverez sur le très bon site « developpez.com » :

<http://rdonfack.developpez.com/tutoriels/dotnet/presentation-gestionnaire-packages-net-nuget/>

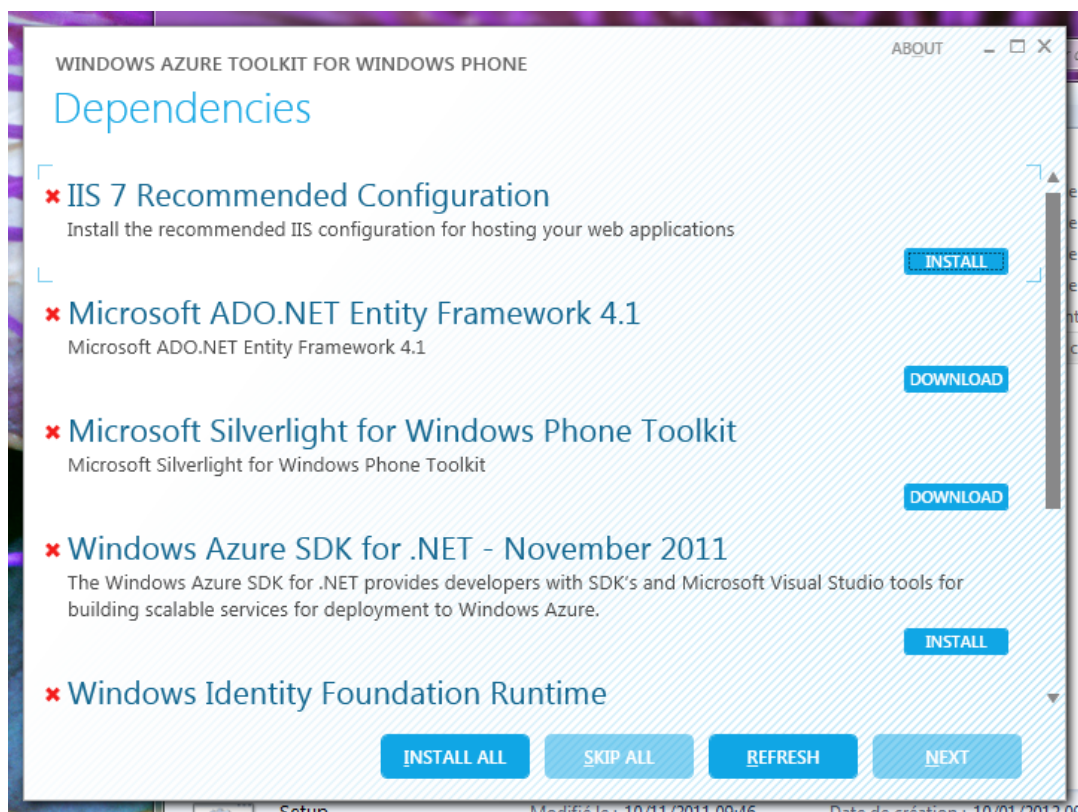
Microsoft a eu la bonne idée de proposer WAT sous la forme de packages NuGet. Vous pouvez ainsi installer et utiliser que les parties qui vous intéressent dans WAT, sans avoir à tout installer d'un bloc.



Phone notifications	<a href="http://www.nuget.org/packages/Phone.Notifications.BasePage">http://www.nuget.org/packages/Phone.Notifications.BasePage</a>
Phone notifications with ACS	<a href="http://www.nuget.org/packages/Phone.Notifications.AccessControl">http://www.nuget.org/packages/Phone.Notifications.AccessControl</a>
Notifications	<a href="http://www.nuget.org/packages/CloudServices.Notifications">http://www.nuget.org/packages/CloudServices.Notifications</a>
Notifications with ACS	<a href="http://www.nuget.org/packages/CloudServices.Notifications.AccessControl">http://www.nuget.org/packages/CloudServices.Notifications.AccessControl</a>
Notifications with SQL Azure	<a href="http://www.nuget.org/packages/CloudServices.Notifications.Sql">http://www.nuget.org/packages/CloudServices.Notifications.Sql</a>

Dans le cadre de cette découverte de WAT, qui est basée sur le toolkit complet, je vous conseille vivement d'installer le package complet (à récupérer sur le site CodePlex de WAT), afin d'avoir sur votre machine de développement tous les éléments que nous allons découvrir au fur et à mesure de cette présentation. Pour rappel, voici l'adresse : <http://watwp.codeplex.com/releases/view/77780>

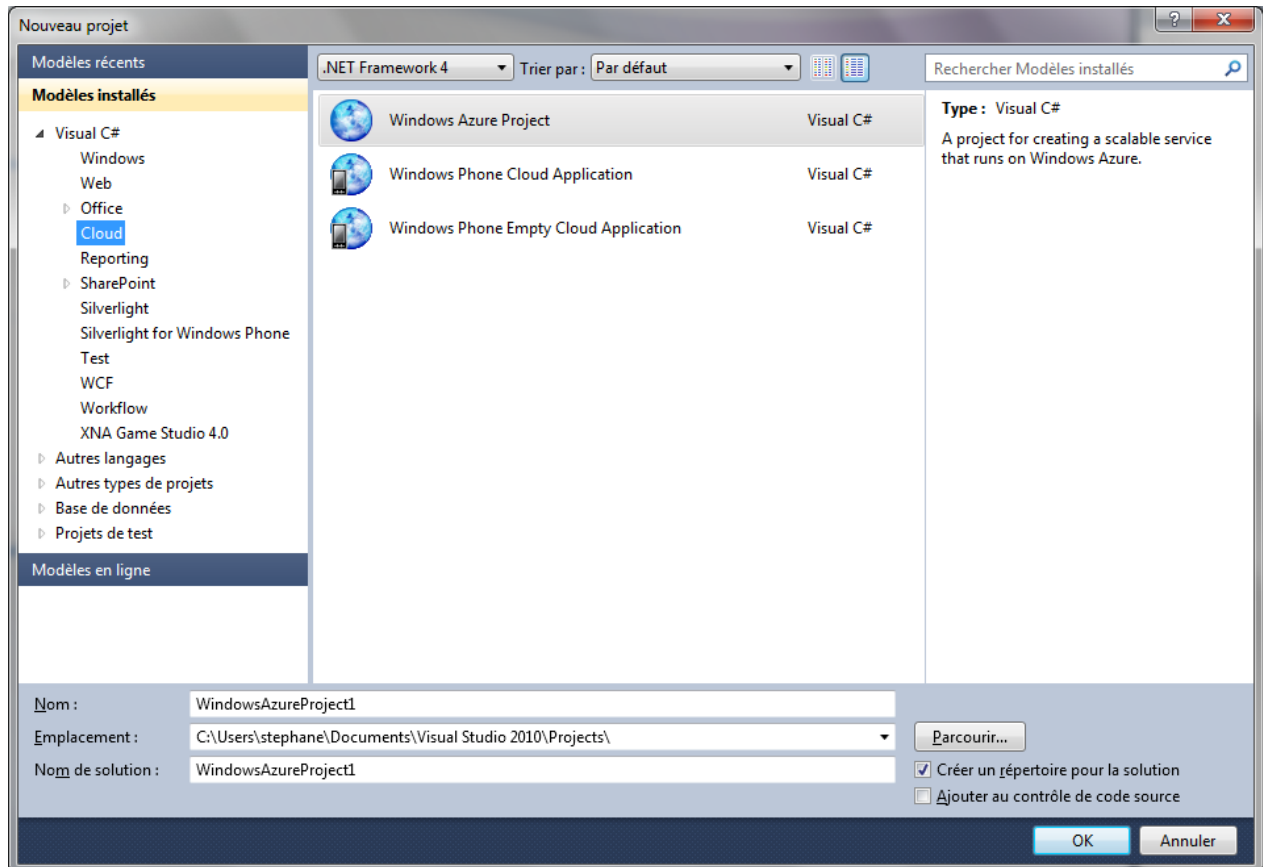
Pour installer le kit complet, il suffit donc de télécharger le fichier « WatWindowsPhone.exe » depuis CodePlex et l'exécuter. Cette opération va installer tous les composants nécessaires pour paramétrer votre PC de développement et installer les Template WAT dans Visual Studio. Les composants sont décompressés par défaut dans le dossier « C:\WindowsAzure\WATWindowsPhone ». Pour lancer l'installation il suffit de lancer « Setup.cmd » depuis ce dossier et suivre les étapes. Votre PC de développement va être analysé et tous les composants manquants seront listés. Vous devez installer tous ceux qui sont marqués « manquants » pour être en mesure d'utiliser pleinement WAT.





Une fois l'installation complète réalisée, vous aurez à votre disposition 2 nouveaux Template de développements dans la catégorie « Cloud » :

- Windows Phone Cloud Application
- Windows Phone Empty Cloud Application



### Windows Phone Cloud Application

Crée une nouvelle solution composée de 3 projets :

- Un projet Windows Azure (pour le déploiement du proxy WAT)
- Un projet web pour le proxy en lui-même
- Un projet Windows Phone qui démontre l'utilisation de WAT

### Windows Phone Empty Cloud Application

Crée une nouvelle solution composée de 3 projets, les mêmes que pour le Template précédent, à la différence que le projet Windows Phone est vide.

WAT est utilisable avec Visual Studio 2010 (version Professional minimum) ou avec Visual Studio 2010 Express for Windows Phone (pour la partie Windows Phone) et Visual Web Developer 2010 Express (pour la partie Azure) qui sont les versions gratuites de Visual Studio 2010.

Je vous invite à visionner cette vidéo qui montre tout ce que WAT permet de réaliser. L'exemple est fait avec Windows Phone mais globalement les fonctionnalités présentées sont aussi accessibles depuis iOS et Android : <http://channel9.msdn.com/posts/Getting-Started-with-the-Windows-Azure-Toolkit-for-Windows-Phone-7-v12>

## Installation de WAT pour le développement iOS ou Android

Pour iOS et Android, si vous n'utilisez pas le proxy, vous n'aurez qu'à télécharger et installer les bibliothèques et sources pour votre plateforme depuis les emplacements GitHub fournis plus haut. Par contre, si vous souhaitez utiliser les services du proxy WAT (c'est plus que recommandé) et que vous n'avez pas de quoi faire pour faire tourner Visual Studio 2010, vous allez devoir télécharger et installer sur votre compte Azure le package qui correspond à vos besoins parmi tous ceux proposés à cette adresse : <https://github.com/microsoft-dpe/wa-toolkit-cloudreadypackages>

Les packages proposés sont différents suivant que vous souhaitez utiliser l'identification classique ou l'ACS.

Il y a aussi un package par type de machine sur lequel vous souhaitez installer (allant de XS à XL soit 5 types) :



Unit of Compute Defined  
[ Equivalent compute capacity of a 1.6GHz 64-bit processor ]



Je vous renvoie au coach Windows Azure pour plus de détails sur les différents types de machines disponibles dans Azure.

## Conclusion

WAT est un toolkit gratuit fourni par Microsoft permettant aux machines fonctionnant sous Windows Phone, iOS et Android d'accéder aux services de stockage de Windows Azure. Un mode sécurisé, très fortement recommandé, permet de ne pas embarquer dans les applications mobiles les clés d'accès au compte Azure ciblé.

Le prochain tutoriel sera consacré à l'identification (ASP.NET Membership et ACS) pour les 3 plateformes.