Déployer l’installeur automatisé de solutions Open Data ODPI dans Azure

# Détails du document

Version : 1.0

Publication : septembre 2012

Auteurs : Rémi Olivier (Microsoft France)

Contenu

[Détails du document 1](#_Toc335402681)

[Résumé 3](#_Toc335402682)

[Prérequis 3](#_Toc335402683)

[Connaissances et attendu 4](#_Toc335402684)

[Présentation rapide d’ODPI 5](#_Toc335402685)

[Organisation du code source d’ODPI 6](#_Toc335402686)

[ODI (Open Data Installer) 6](#_Toc335402687)

[ODI.Azure 7](#_Toc335402688)

[Mise en place de la configuration du compte Windows Azure 8](#_Toc335402689)

[Etape 1 – Création du compte de stockage 8](#_Toc335402690)

[Etape 2 – Configuration du compte de stockage 11](#_Toc335402691)

[Publication d’ODPI dans Windows Azure 14](#_Toc335402692)

[Etape 1 – Configuration des paramètres 14](#_Toc335402693)

[Etape 2 – Déploiement d’ODPI 15](#_Toc335402694)

[Mise à jour des solutions Open Data dans ODPI 17](#_Toc335402695)

[Etape 1 – Génération du package de services Azure d’une la solution Open Data 17](#_Toc335402696)

[Pour OGDI DataLab… 17](#_Toc335402697)

[Pour ODAF Openturf… 18](#_Toc335402698)

[Pour le Portail Citoyen Open Data… 18](#_Toc335402699)

[Etape 2 – Mise à jour des fichiers dans ODPI 19](#_Toc335402700)

[Etape 3 – Ajout dans le compte de stockage 20](#_Toc335402701)

[Ajouter une solution Open Data dans ODPI 21](#_Toc335402702)

[Etape 1 – Ajout dans le compte de stockage blob 21](#_Toc335402703)

[Création du template de configuration 21](#_Toc335402704)

[Envoi des fichiers vers le compte de stockage blobs 22](#_Toc335402705)

[Etape 2 – Création des nouveaux fichiers dans ODPI 22](#_Toc335402706)

[Etape 3 – Mise à jour des fichiers existants 25](#_Toc335402707)

[Etape 4 – Déploiement d’ODPI dans Windows Azure 26](#_Toc335402708)

[Pour aller plus loin 27](#_Toc335402709)

[Restons en contact 29](#_Toc335402710)

# Résumé

Ce document détaille l’ensemble du processus de mise en œuvre et de publication sur la [plateforme Windows Azure](http://www.microsoft.com/france/windows-azure/)[[1]](#footnote-1) d’une solution personnalisée de l’installer automatisé de solution Open Data ODPI (Open Data Platform Installer). Un accompagnement pas-à-pas vous est ainsi proposé pour toutes les étapes du processus vis-à-vis de la plateforme Windows Azure.

Ainsi il sera abordé comment configurer et publier la solution, à savoir ODPI, dans votre compte Windows Azure. Ceci correspond à l’étape ultime avant d’avoir une solution entièrement fonctionnelle et opérationnelle.

De même il sera décrit comment mettre à jour les solutions Open Data fournies par ODPI et commet ajouter une solution personnalisée dans ODPI afin de faire évoluer le produit en même temps que les solutions qu’il propose.

# Prérequis

ODPI à destination de l’environnement d’exécution Windows Azure nécessite un ordinateur sous Windows Server 2003 (R2) SP1, Windows 7, Windows 8, Windows Server (R2) 2008 ou ultérieur.

Le lecteur doit connaître les technologies des services Web RESTful, le langage C# et .NET Framework afin de pouvoir facilement comprendre les éléments mis ainsi à disposition.

# Connaissances et attendu

La mise en œuvre d’ODPI requiert une expérience préalable :

* De la plateforme Microsoft Windows Azure,
* De l’environnement de développement Microsoft Visual Studio 2010,
* De la programmation avec le Microsoft Framework .NET,
* Des dernières technologies Web comme ASP.NET MVC ou encore jQuery.

Microsoft met à disposition des formations gratuites téléchargeables sur ces environnements et technologies. Nous vous invitons à considérer le cas échéant les kits de formation suivants Microsoft Web Camps Training Kit

note_ddRemarque :

Microsoft met à disposition des formations gratuites téléchargeables sur ces environnements et technologies. Nous vous invitons à considérer le cas échéant les kits de formation suivants qui comprennent des présentations, des vidéos de formation, des démonstrations accompagnées de leur script ainsi que des ateliers didactiques complets :

* [Microsoft Windows Azure Training Kit (mise à jour de mai 2011)](http://www.microsoft.com/downloads/en/details.aspx?FamilyID=413E88F8-5966-4A83-B309-53B7B77EDF78&displaylang=en)[[2]](#footnote-2) couvrant la plateforme Windows Azure et l’environnement de développement Microsoft Visual Studio 2010.
* [Microsoft Web Camps Training Kit](http://trainingkit.webcamps.ms/Default.htm)[[3]](#footnote-3) couvrant notamment dans le contexte de ce kit de démarrage les technologies ASP.NET MVC, OData et jQuery.

note_ddRemarque :

Le [blog du groupe Produit de la plateforme Windows Azure](http://blogs.msdn.com/b/windowsazure/)[[4]](#footnote-4) propose également de nombreuses ressources en termes de formation sur cet environnement d’exécution et d’hébergement.

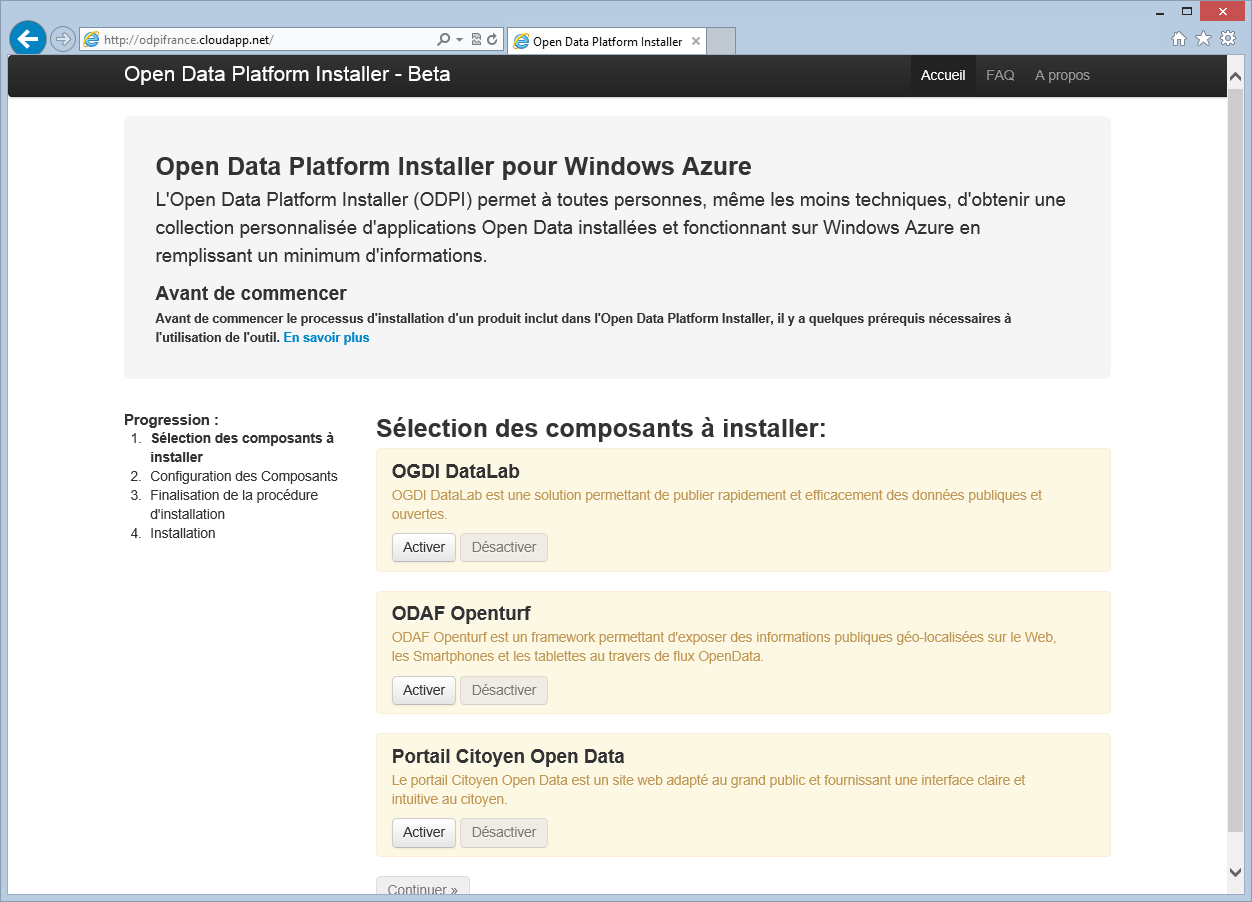
Celui-ci constitue également un excellent moyen de se tenir informé des évolutions proposées par la plateforme.

# Présentation rapide d’ODPI

ODPI est une solution dont l’objectif est de permettre à toutes personnes, même les moins techniques, d’installer des solutions Open Data simplement et rapidement. ODPI gère donc pour vous la configuration et le déploiement des solutions Open Data proposées dans Windows Azure moyennant une saisie d’informations simplifiées. S’adressant en premier lieu au grand public ODPI n’en demeure pas moins intéressant pour les développeurs désireux d’optimiser les déploiements ou de gagner du temps dans leurs développements.

ODPI propose actuellement plusieurs solutions permettant d’entrer de plein pied dans le mouvement Open Data telles que le kit de démarrage interactif OGDI (*Open Government Data Initiative*), ODAF (*Open Data Application Framework*) ou encore le Portail Citoyen Open Data permettant d’offrir une interface simple et intuitive aux citoyens.

Pour plus de renseignements sur ODPI nous vous invitons à visiter le blog [OGDI France](http://blogs.msdn.com/b/ogdifrance/) et plus particulièrement le billet [ODPI, la facilité de mise en œuvre des solutions Open Data](http://blogs.msdn.com/b/ogdifrance/archive/2012/07/13/odpi-la-facilit-233-de-mise-en-uvre-des-solutions-open-data.aspx).



# Organisation du code source d’ODPI

L’intégralité du code source du site ODPI est située dans le sous-répertoire *Source* du répertoire d’extraction de l’installeur automatique ODPI.

Compte tenu de ce qui précède, ce répertoire contient 3 sous-dossiers principaux :

1. Le dossier *blob files* contient l’ensemble des solutions Open Data actuellement préconfigurées dont 3 accessibles par défaut sur le site web ODPI que vous déploierez. Ces 3 solutions Open Data sont OGDI DataLab (*Open Governement Data Initiative*), ODAF Openturf (*Open Data Application Framework*) et le Portail Citoyen Open Data.
2. Le dossier *ODI* correspond au site Web qui sera déployé. Il regroupe différents dossiers qui constituent, au final, l’application Web ASP.NET MVC3 destinée à l’environnement Windows Azure et permettant de visualiser les données provenant d’un service Open Data;
3. Le dossier *ODI.Azure* est un projet permettant de déployer le projet ODI décrit précédemment dans Windows Azure;

La suite de cette section décrit les dossiers/éléments de configuration de ces différents sous-dossiers de façon à mieux cerner la solution proposée par ODPI et ainsi vous l’accaparer plus facilement en vue par exemple de sa personnalisation (légère ou avancée).

Tous les sites, services et composants afférents sont écrits en C#. Ils sont disponibles pour la version v4.0.30319 du Microsoft Framework .NET. Les fichiers de solution associés et présents dans ces dossiers sont destinés à l’environnement de développement Visual Studio 2010.

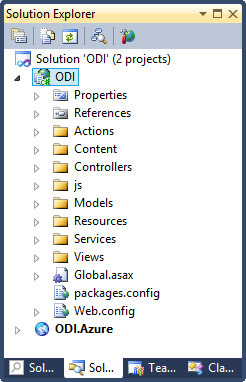
## ODI (Open Data Installer)

Cette première composante de la solution correspond à la plateforme d’installation automatique sous forme d’une application Web ASP.NET MVC3 destinée à l’environnement Windows Azure.

important_ddRemarque importante :

La publication requiert un compte et une souscription active avec Windows Azure. Vous pouvez accéder à Windows Azure avec un paiement à l’usage, sans engagement, ou bien des forfaits comme décrit à l’adresse <http://www.microsoft.com/france/windows-azure/Offres.aspx>.

ODI est composé des 8 dossiers suivants :



| **Dossier** | **Description** |
| --- | --- |
| ***Actions*** | Ce dossier contient les actions qui seront effectuées lors de la configuration et du déploiement des solutions Open Data. Il contient notamment les classes de validation des connexions aux sources de données et les actions à effectuer une fois le déploiement réussi. |
| ***Content*** | Dossier classique d’un site web ASP.NET MVC 3, il contient les images et feuilles de styles. |
| ***Controllers*** | Dossier classique d’un site web ASP.NET MVC 3. |
| ***Js*** | Ce dossier est un des points clé d’ODPI. C’est à travers ses fichiers JavaScript que l’interaction avec l’utilisateur est possible. |
| ***Models*** | Au-delà de son utilité dans l’architecture ASP.NET MVC 3, il contient la définition et la configuration à adopter des différentes solutions Open Data qui seront accessibles sur le site web. |
| ***Resources*** | L’ensemble des fichiers de localisation sont contenus dans ce répertoire. Le site est actuellement entièrement traduit en français et en anglais. |
| ***Services*** | Cette couche service permet entre autres la génération de certificats nécessaires à l’identification sur le portail Windows Azure et permet également de configurer les packages en fonction d’un template précis puis de déployer dans Windows Azure. |
| ***Views*** | Dossier classique d’un site web ASP.NET MVC 3. Il contient l’ensemble des pages accessibles en fonction de la route demandée. |

## ODI.Azure

ODI.Azure est le projet permettant de déployer l’application ODPI dans Windows Azure. Il référence donc le projet *ODI* décrit précédemment et permet de le configurer.

L’ensemble des paramètres de configuration de la solution ODPI est regroupé au niveau des fichiers de configuration de services *ServiceConfiguration.Cloud.cscfg* et *ServiceConfiguration.Local.cscfg* du projet Windows Azure *ODI.Azure* relatif au rôle *ODI*;

# Mise en place de la configuration du compte Windows Azure

Cette section décrit les éléments de configuration d’un compte Windows Azure à mettre en place préalablement à la publication de la solution ODPI en tant que telle dans Windows Azure.

important_ddRemarque importante :

Les étapes suivantes requièrent un compte et une souscription active avec Windows Azure. Vous pouvez accéder à Windows Azure avec un paiement à l’usage, sans engagement, ou bien des forfaits comme décrit à l’adresse <http://www.microsoft.com/france/windows-azure/offres.aspx>. A la date de publication de ce document, des offres d’essai gratuites sont proposées également comme décrit à l’adresse <http://www.microsoft.com/france/windows-azure/tester/autres-offres_windowsazure.aspx>.

## Etape 1 – Création du compte de stockage



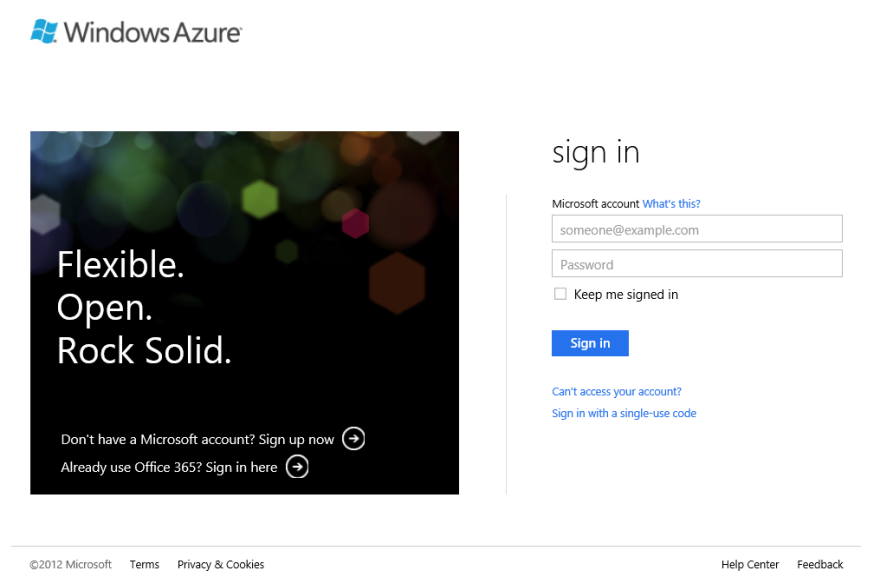
ODPI (*Open Data Platform Installer*) nécessite 1 compte de stockage Windows Azure pour son fonctionnement.

Ce compte de stockage à travers le [service de blobs Windows Azure](http://msdn.microsoft.com/en-us/library/dd135733)[[5]](#footnote-5) permet de stocker les packages Azure contenant les solutions Open Data ainsi que leur templates associés.

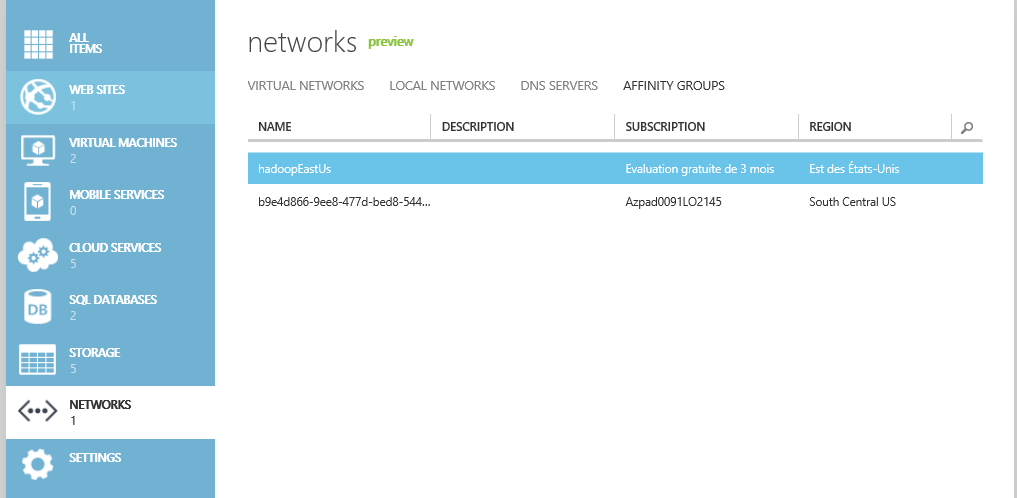
Lors de cette étape, vous allez créer le compte de stockage d’ODPI depuis le portail de gestion Windows Azure à l’adresse <http://manage.windowsazure.com/>.

Description : procedure_ddProcédez comme suit :

1. Ouvrez votre navigateur, naviguez vers <http://manage.windowsazure.com/> et connectez-vous en utilisant votre compte microsoft associé à votre compte Windows Azure.



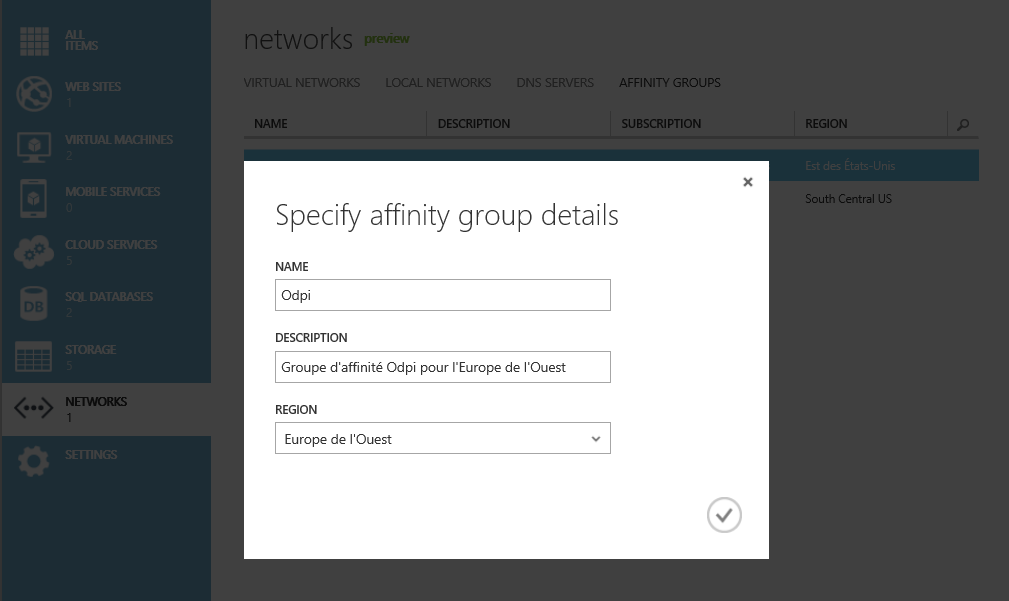
1. La première chose à faire est de créer un groupe d’affinité. Pour cela sélectionnez l’onglet **NETWORKS** puis le menu **AFFINITY GROUPS** et choisissez ensuite**CREATE*.***



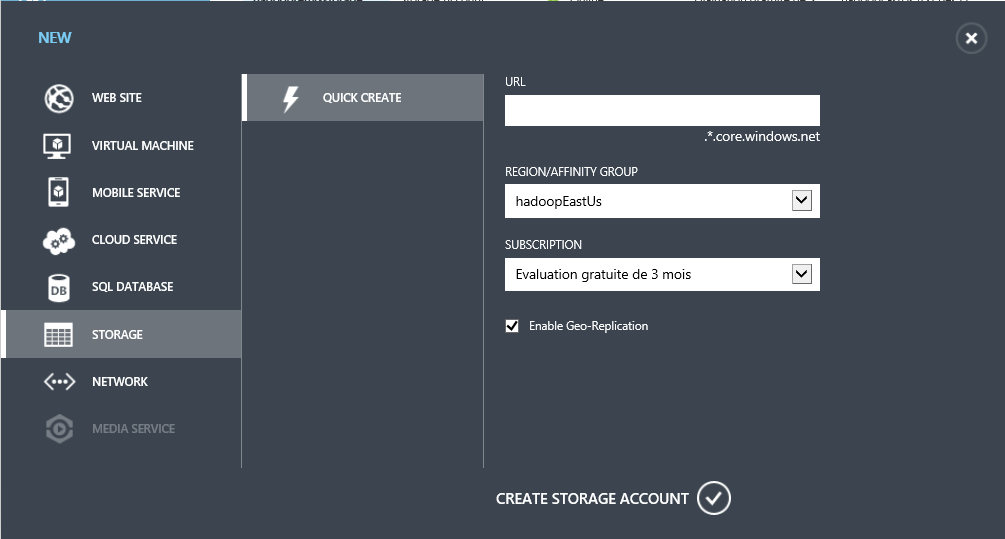
note_ddNote :

Vous créez un groupe d’affinités ici de façon à pouvoir ensuite déployer la solution ODPI dans la même localisation géographique, ceci afin d’optimiser la bande passante et de réduire la latence entre les applications et les données dont elles dépendent.

1. Dans la boite de dialogue **Specify affinity group details**, choisissez un nom pour votre groupe, par exemple *Odpi* ainsi qu’une localisation géographique, **Europe de l’Ouest** étant conseillé pour un accès depuis la France métropolitaine.

****

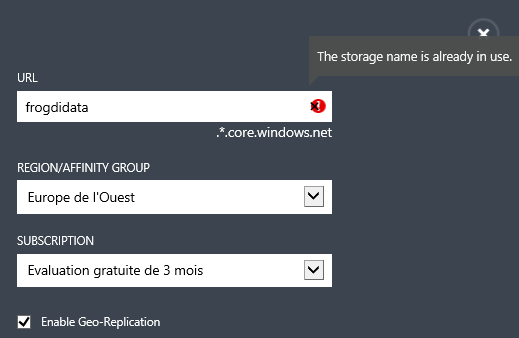
1. Cliquez sur le bouton pour valider la création du groupe d’affinité. Après un temps de chargement pouvant durer quelques minutes, la création du groupe devient effective.
2. Maintenant, pour créer le compte de stockage, cliquez sur le bouton **New** en bas à gauche puis sur **Storage**.



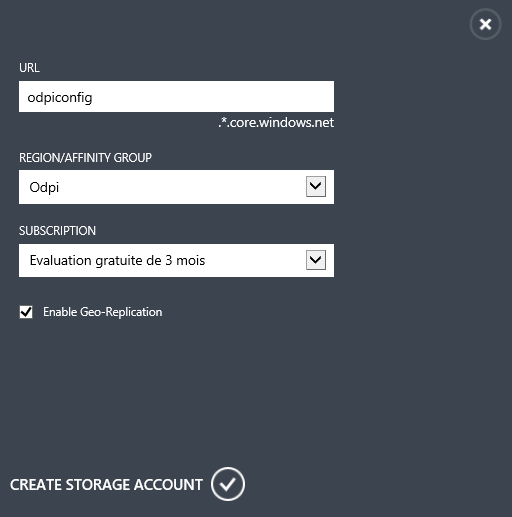
1. Dans la fenêtre apparue, sélectionnez **Quick Create** puis dans la zone de texte **URL** entrez le nom de votre compte de stockage, par exemple *odpiconfig* dans ce cas. Windows Azure va utiliser cette valeur pour générer les URLs servant de points d’entrée aux services (tables, blobs et files d’attente) du compte de données. Attention, ce nom doit être unique, contenir entre 3 et 24 caractères et n’utiliser que des lettres en minuscule et des chiffres !

note_ddRemarque :

Le portail s'assure que le nom est valide en vérifiant i) que le nom est conforme aux règles de noms et ii) qu’il est actuellement disponible. Une erreur de validation sera affichée si vous entrez le nom qui ne satisfait pas les règles.



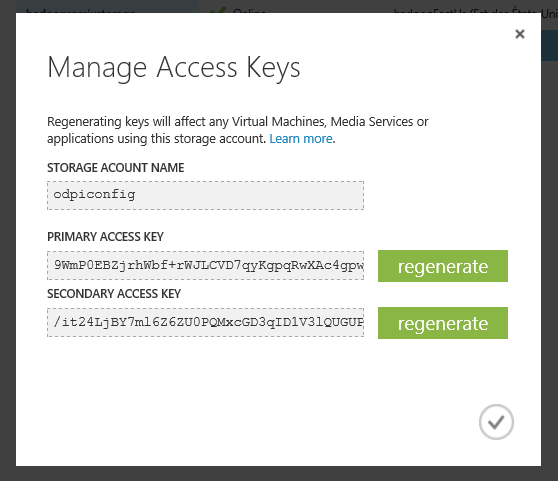
1. Sélectionnez la région créée précédemment dans la liste **REGION/AFINITY GROUP*.***



1. Cliquez sur **CREATE STORAGE ACCOUNT** pour valider la création du compte de stockage. Après un temps de chargement pouvant durer quelques minutes, la création du compte devient effective.

C:\Users\t-rolivi\Desktop\odpiconfigcreated.png

1. Notez que le bouton **MANAGE KEYS** donne accès à la clé primaire du compte nécessaire pour pouvoir accéder à ces services via les [interfaces programmatiques (*API*) de gestion Windows Azure](http://msdn.microsoft.com/fr-fr/library/ee460799.aspx)[[6]](#footnote-6).
2. Cliquez sur le bouton **MANAGE KEYS** de la zone **Primary access key** pour voir les clés associées à ce compte.



Votre compte de stockage est à présent créé. Vous devez maintenant le préparer pour fonctionner avec ODPI.

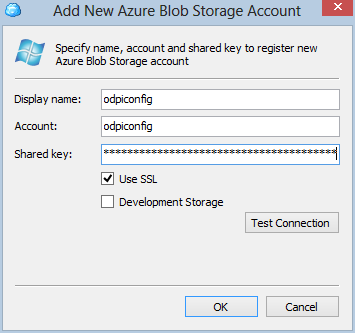
## Etape 2 – Configuration du compte de stockage



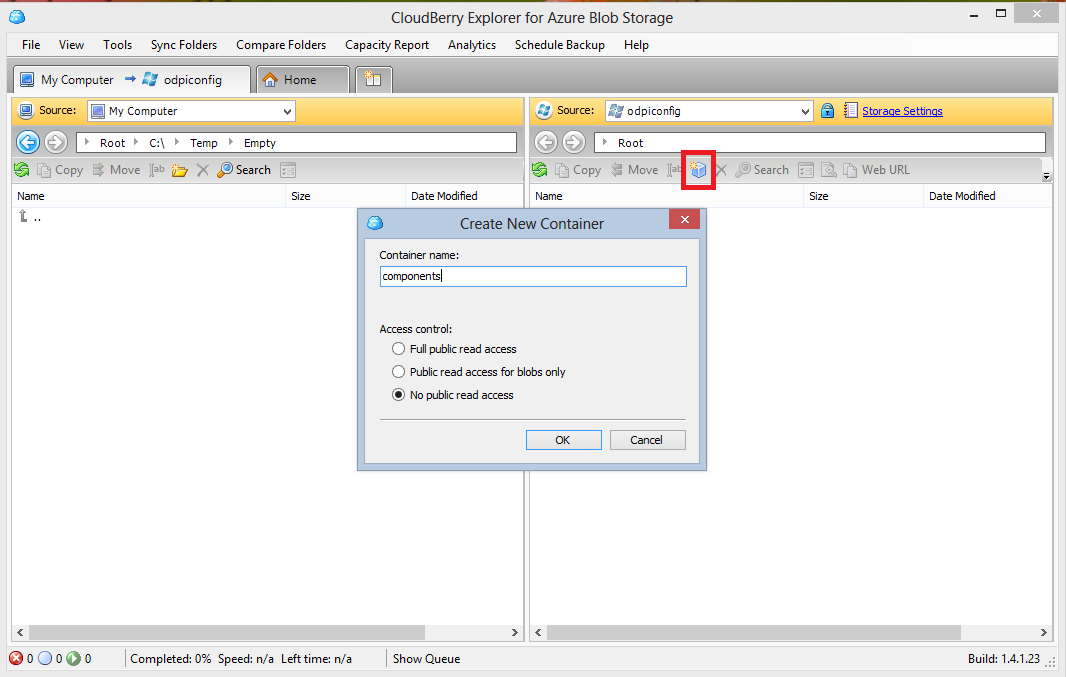
L’installeur automatique ODPI (*Open Data Platform Installer*) nécessite 1 seul compte de service en plus d’un compte de stockage pour s’exécuter. Le compte de stockage sera utilisé pour ses fonctionnalités blobs destinées à héberger les paquets de déploiement des solutions Open Data ainsi que leur templates.

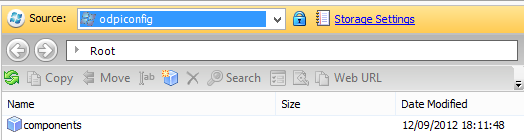
Dans la suite de cette étape vous aurez besoin d’un outil permettant d’explorer et de gérer les stockages blobs dans Windows Azure. Nous utiliserons ici l’application [CloudBerry Explorer For Windows Azure](http://www.cloudberrylab.com/) qui a l’avantage d’être gratuite en plus d’être parfaitement adaptée à notre besoin.

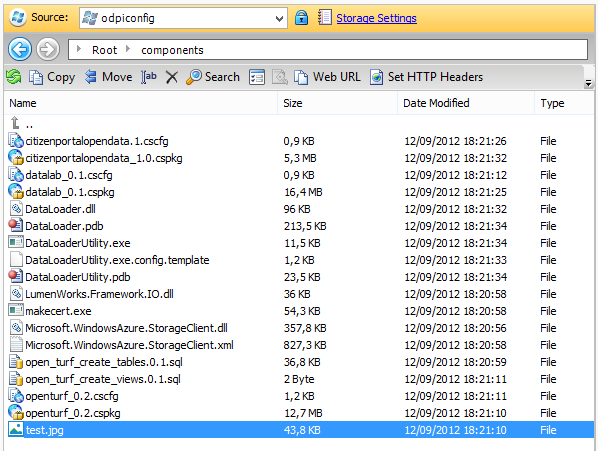
1. Dans CloudBerry Explorer, dans la liste déroulante **Source** cliquez sur **<New Azure Blob Storage Account>** pour ajouter votre compte de stockage précédemment créé.
2. Dans la fenêtre **Add New Azure Blob Storage Account** remplissez les informations nécessaires. **Shared key** correspond à la clé d’accès primaire énoncée dans l’étape 1.



1. Cliquez sur **OK** pour vous connecter et explorer votre compte de stockage qui pour l’instant est vide.
2. Vous allez à présent créer un conteneur pour héberger les packages de services Azure contenant les solutions Open Data. Pour cela créez un conteneur nommé **components** en cliquant sur le bouton en forme de cube.





1. A présent faîtes glisser l’ensemble des fichiers du dossier **blob files,** fournis avec la solution ODPI, dans le conteneur **components**. 
2. Votre compte de stockage est à présent configuré pour permettre à ODPI de déployer les solutions Open Data.

# Publication d’ODPI dans Windows Azure

Vous allez à présent aborder les différentes étapes visant à déployer totalement votre solution personnalisée de l’installeur automatisé ODPI dans votre compte Windows Azure.

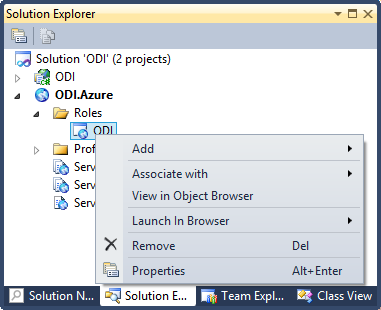
## Etape 1 – Configuration des paramètres



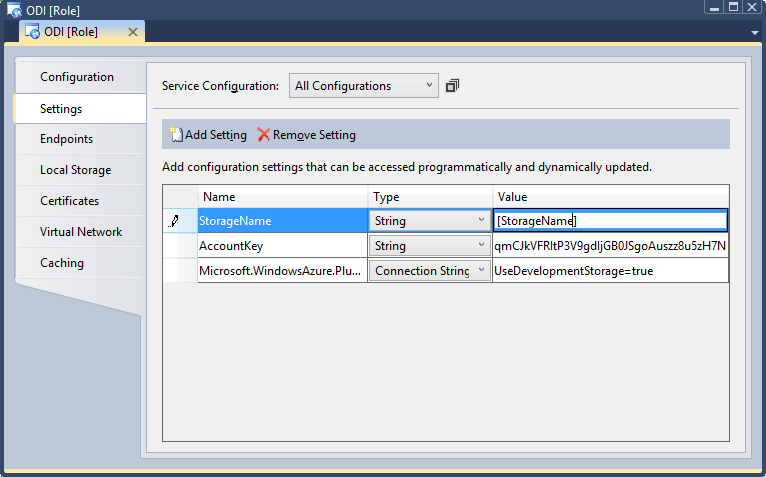
La configuration centralisée d’ODPI regroupe les éléments nécessaires au fonctionnement du site Web et notamment au bon déploiement des solutions.

Description : Description : Description : Description : Description : Description : Description : Description : Description : procedure_ddPour correctement configurer l’installeur, procédez comme suit :

1. Dans Visual Studio 2010, dans **Solution Explorer**, avec la solution *ODI.sln* ouverte, développez le dossier **Roles** du projet *ODI.Azure*. Ce projet de type Windows Azure vous sert à déployer le projet *ODI*.
2. Faites un clic droit sur *ODI* sous le dossier **Roles**, puis sélectionnez **Properties.**

****

1. Un assistant de configuration s’ouvre. Sélectionnez l’onglet **Settings**.



1. Saisissez les informations du compte de stockage créé dans l’étape 1 et contenant vos packages de services Open Data.

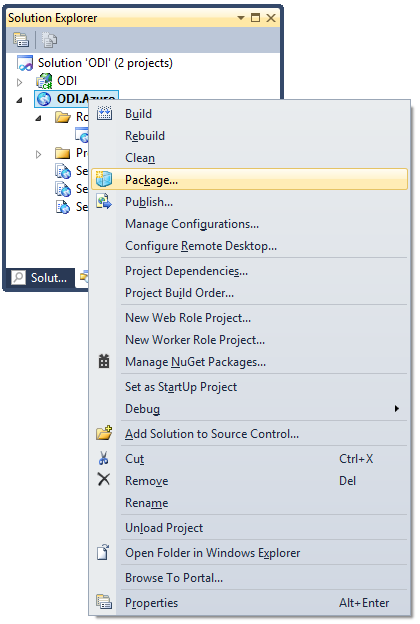
## Etape 2 – Déploiement d’ODPI



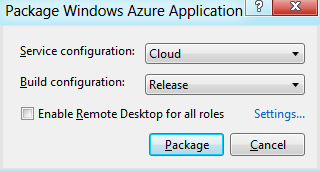
Vous venez de configurer l’installeur automatique ODPI. Dans cette étape, vous allez procéder au déploiement d’ODPI dans votre compte de service Windows Azure associé. Pour cela il est nécessaire de créer dans un premier temps un paquet qui sera ensuite déployé via le portail de gestion Windows Azure.

Description : Description : Description : Description : Description : Description : Description : Description : Description : procedure_ddProcédez comme suit :

1. Toujours dans Visual Studio 2010, faites un clic-droit sur le projet*ODI.Azure* précédent et choisissez **Package**dans le menu contextuel.

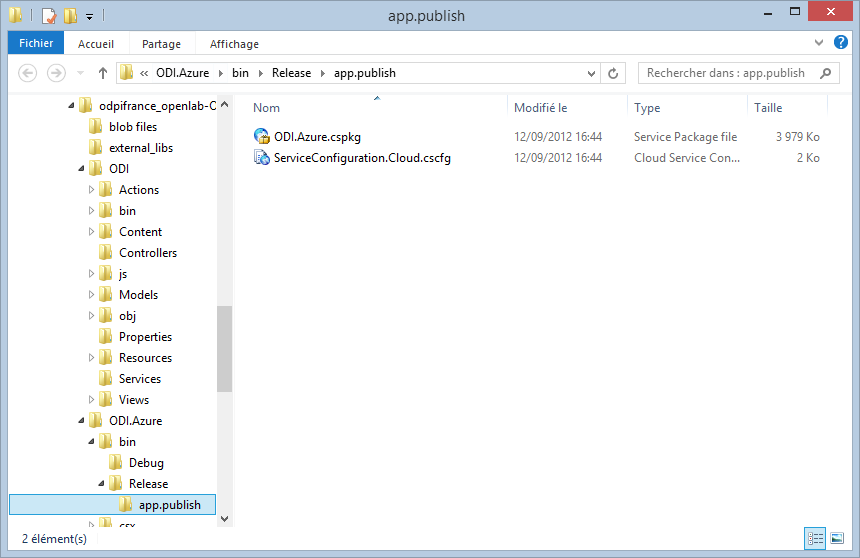


1. Dans la boite de dialogue **Package Windows Azure Application**, choisissez les options **Service Configuration : Cloud**, **Local Configuration : Release** et cliquez sur **Package** pour confirmer.



1. Une fois le package créé, une fenêtre Windows s’ouvre dans le dossier app.*publish* correspondant. Ce dossier contient le package *ODI.Azure.cspkg* et le fichier de configuration *ServiceConfiguration.Cloud.cscfg*associé.

*Vous allez utiliser les fichiers correspondants dans les étapes suivantes. Mémorisez donc le chemin vers ce dossier.*



1. Il ne vous reste plus qu’à déployer ce package via le portail de gestion Windows Azure accessible depuis l’adresse <http://manage.windowsazure.com/>.

# Mise à jour des solutions Open Data dans ODPI

Vous allez à présent aborder les différentes étapes visant à mettre à jour votre solution personnalisée de l’installeur automatisé ODPI dans votre compte Windows Azure. Ceci afin de prendre en compte les modifications effectuées dans une ou plusieurs solutions Open Data déjà présentent dans ODPI.

## Etape 1 – Génération du package de services Azure d’une la solution Open Data

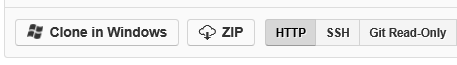
Lors de cette étape nous vous expliquerons comment obtenir un package dans le cadre d’une mise à jour de version d’une solution Open Data déjà présente dans ODPI.

Dans un premier temps, vous devez récupérer cette solution.

Par défaut ODPI propose 3 solutions Open Data. A savoir :

* OGDI DataLab disponible à l’adresse : <https://github.com/openlab/DataLab>
* ODAF Openturf disponible à l’adresse : <https://github.com/openlab/Openturf>
* Portail Citoyen Open Data

Pour télécharger les sources depuis GitHub le moyen le plus simple et rapide est de cliquer directement sur le bouton ZIP. S’il constitue le moyen le plus simple, les autres options se révèlent plus adaptées pour mettre en place une gestion de versions. Pour plus d’informations sur l’utilisation détaillée de solutions Open Data avec GitHub telles qu’OGDI DataLab ou ODAF Openturf nous vous invitons à consulter les billets [OGDI DataLab sur GitHub ! Impliquez-vous dans la communauté](http://blogs.msdn.com/b/ogdifrance/archive/2012/05/14/ogdi-datalab-sur-github-impliquez-vous-dans-la-communaut-233.aspx) et [GitHub for Windows](http://blogs.msdn.com/b/ogdifrance/archive/2012/06/14/github-for-windows.aspx) tous deux présents sur le blog [OGDI France](http://blogs.msdn.com/b/ogdifrance/).



Les sources téléchargées il vous faut à présent générer un package de services Windows Azure qui servira de base lors d’une installation via ODPI.

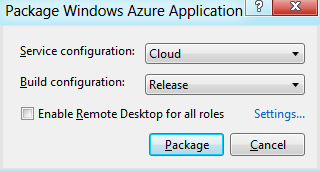
Suivant la solution choisie le processus de génération du package Windows Azure est différent.

### Pour OGDI DataLab…

… rendez-vous dans le dossier **Documents** situé à la racine du dossier récupéré de GitHub. Suivez la documentation *Démarrer avec le Kit de démarrage OGDI DataLab.*docx qui vous guidera dans la configuration, la compilation et la génération du package de services. Pour aller plus vite vous pouvez directement attaquer la partie *Résolution des dépendances avec NuGet* du document précédemment ouvert.

Votre solution maintenant configurée vous devez à présent créer le package de services Windows Azure.

* Pour cela ouvrez Visual Studio en tant qu’administrateur
* Faîtes un clic droit sur *ODAF.Azure* puis **Package**.
* Dans la boite de dialogue **Package Windows Azure Application**, choisissez les options **Service Configuration : Cloud**, **Local Configuration : Release** et cliquez sur **Package** pour confirmer.



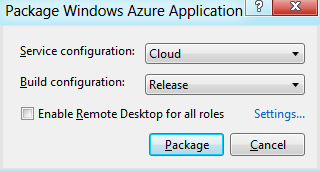
* Une fois le package créé, une fenêtre Windows s’ouvre dans le dossier app.*publish* correspondant. Ce dossier contient donc le package *ODAF.Azure.cspkg* qui nous intéresse. Mémorisez donc le chemin vers ce dossier.

### Pour ODAF Openturf…

… rendez-vous dans le dossier **Documents** situé à la racine du dossier récupéré de GitHub. Suivez la documentation ***Déployer le Kit de démarrage ODAF Openturf dans Windows Azure.docx*** qui vous guidera dans la configuration, la compilation et la génération du package de services. Pour aller plus vite vous pouvez directement vous rendre à **l’étape 6** de la partie *Exécution de la solution ODAF Openturf en environnement local* du document précédemment ouvert.

Votre solution maintenant configurée vous devez à présent créer le package de services Windows Azure.

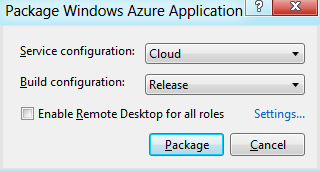
* Pour cela ouvrez Visual Studio en tant qu’administrateur
* Faîtes un clic droit sur *DataBrowser.Cloud* puis **Package**.
* Dans la boite de dialogue **Package Windows Azure Application**, choisissez les options **Service Configuration : Cloud**, **Local Configuration : Release** et cliquez sur **Package** pour confirmer.



* Une fois le package créé, une fenêtre Windows s’ouvre dans le dossier app.*publish* correspondant. Ce dossier contient donc le package *DataBrowser.Cloud.cspkg* qui nous intéresse. Mémorisez donc le chemin vers ce dossier.

### Pour le Portail Citoyen Open Data…

* Ouvrez Visual Studio en tant qu’administrateur
* Faîtes un clic droit sur *DataBrowser.Cloud* puis **Package**.
* Dans la boite de dialogue **Package Windows Azure Application**, choisissez les options **Service Configuration : Cloud**, **Local Configuration : Release** et cliquez sur **Package** pour confirmer.



* Une fois le package créé, une fenêtre Windows s’ouvre dans le dossier app.*publish* correspondant. Ce dossier contient donc le package *DataBrowser.Cloud.cspkg* qui nous intéresse. Mémorisez donc le chemin vers ce dossier.

Le package généré est prêt à l’utilisation. Retenez son emplacement, vous en aurez besoin pour l’étape 3 dans laquelle nous téléchargerons ce fichier vers le compte de stockage blob Windows Azure.

## Etape 2 – Mise à jour des fichiers dans ODPI

L’étape 2 n’est pas requise si les paramètres de configuration présents dans le fichier *ServiceConfiguration.Cloud.cscfg* du projet *ODI.Azure* n’ont pas été modifiés. Si vous êtes dans ce cas passez directement à l’ETAPE 3 – AJOUT DANS LE COMPTE DE STOCKAGE.

A contrario, si des paramètres ont été ajoutés/supprimés vous devez alors suivre les étapes suivantes afin de répercuter les changements sur le site web ODPI.

1. La première étape est de se rendre dans le dossier **Models/Config** et de rajouter ou supprimer les propriétés contenues dans le fichier éponyme à la solution à mettre à jour. De même veillez à modifier la variable *template* afin de répercuter les modifications dans les paramètres.

Illustrons ce premier point avec la solution OGDI DataLab. Le paramètre de configuration *SolutionName* a été implémenté dans le kit de démarrage interactif OGDI DataLab afin de rendre le nom de la solution dynamique. Il faut à présent mettre à jour ODPI pour qu’il propose le nouveau paramètre lors de la configuration d’OGDI DataLab sur le site web.

Pour cela nous ouvrons le fichier **DataLabConfig.cs** et nous rajoutons la propriété suivante :

public string SolutionName { get; set; }

De même la méthode **BuildSettingsString** est à mettre à jour.

public string BuildSettingsString()

        {

            string template = @"<Setting name=""RecaptchaPrivateKey"" value=""{0}""/>

                                  <Setting name=""RecaptchaPublicKey"" value=""{1}""/>

                                  <Setting name=""DataConnectionString"" value=""DefaultEndpointsProtocol=https;AccountName={2};AccountKey={3}""/>

                                  <Setting name=""DiagnosticsConnectionString"" value=""DefaultEndpointsProtocol=https;AccountName={2};AccountKey={3}""/>

                                  <Setting name=""serviceUri"" value=""http://{4}.cloudapp.net:8080/v1/""/>

                                  <Setting name=""IsAnalytics"" value=""0""/>

**<Setting name=""SolutionName"" value=""{6}"" />**

                                </ConfigurationSettings>

                              </Role>

                              <Role name=""DataBrowser.WorkerRole"">

                                <Instances count=""1""/>

                                <ConfigurationSettings>

                                  <Setting name=""serviceUri"" value=""http://{4}.cloudapp.net:8080/v1/"" />

                                  <Setting name=""DiagnosticsConnectionString"" value=""DefaultEndpointsProtocol=https;AccountName={2};AccountKey={3}"" />

                                  <Setting name=""DataConnectionString"" value=""DefaultEndpointsProtocol=https;AccountName={2};AccountKey={3}"" />

                                ";

            return string.Format(template, RecapPrivateKey, RecapPublicKey, BlobAccountName, BlobAccountKey, Dns**, SolutionName** );

        }

Enfin la méthode **BuildFromData** doit, elle aussi, être mise à jour afin de répercuter les données saisies depuis l’interface dans la propriété *SolutionName*.

public void BuildFromData(dynamic data)

{

    Dns = data.dns;

    RecapPrivateKey = data.recappriv;

    RecapPublicKey = data.recappub;

    BlobAccountName = data.storagename;

    BlobAccountKey = data.storagekey;

SolutionName = data.solutionname ;

}

1. La seconde étape va permettre à l’utilisateur de remplir ce paramètre lors de la configuration dans l’interface web.

Cette étape consiste à modifier la vue associée à la solution mise à jour dans le but de rajouter un champ de saisie. L’ensemble des vues relatives à la solution se trouve dans le dossier **Views/ConfTpl/.**

Ce qui donne dans le cadre de la solution OGDI DataLab la vue associée **Views/ConfTpl/DataLab.cshtml**.

Le code à rajouter pour un nouveau champ est le suivant :

<div class="control-group">

<label class="control-label">

    @ODI.Resources.Views.ConfTpl.DataLabResource.[NomDuParamètreDeTraductions] </label>

<div class="controls">

    <input class="input-xlarge" type="text" id="solutionname" name="solutionname" />

</div>

1. ODPI est à présent configuré pour prendre en compte les modifications apportées. Vous devez maintenant déployer la solution dans Windows Azure. Pour déployer ODPI dans Windows Azure reportez-vous à la PUBLICATION D’ODPI DANS WINDOWS AZURE

## Etape 3 – Ajout dans le compte de stockage

Le fonctionnement interne d’ODPI est basé sur un compte de stockage contenant l’ensemble des packages de services Azure des solutions Open Data. Une mise à jour de la solution implique donc une mise à jour du package dans le compte de stockage qui sera directement répercutée dans ODPI.

Pour cette étape vous aurez besoin du package généré lors de l’étape 1. Ceci constitue la dernière modification pour que les changements soient effectifs. Il vous faut donc ajouter ce package dans le conteneur **components** du compte de stockage blob Windows Azure.

Pour plus d’informations sur la manière d’envoyer un fichier vers un compte de stockage blob Windows Azure reportez-vous à l’ETAPE 2 – CONFIGURATION DU COMPTE DE STOCKAGE et notamment à l’outil utilisé : *CloudBerry Explorer for Windows Azure*.

# Ajouter une solution Open Data dans ODPI

Vous allez à présent aborder les différentes étapes visant à déployer totalement votre solution personnalisée de l’installeur automatisé ODPI dans votre compte Windows Azure.

## Etape 1 – Ajout dans le compte de stockage blob

La première étape est de vérifier que vous possédez bien un package de services Windows Azure de votre solution Open Data.

Si tel n’est pas le cas ou que vous ne savez pas comment obtenir, depuis votre solution Open Data personnelle, un package de services pour Windows Azure référez-vous au lien suivant <http://msdn.microsoft.com/fr-fr/library/windowsazure/ff683672.aspx#PublishAzure>.

Ce package est indispensable à l’utilisation de votre solution Open Data dans ODPI car c’est à partir de celui-ci que les sites web seront générés.

Le package créé vous obtenez par la même occasion un fichier de configuration. Celui-ci possède l’extension **.cscfg** et contient les paramètres qui seront configurables depuis ODPI.

Avant d’ajouter ce fichier dans le compte de stockage vous devez le modifier afin d’en faire un template spécifique à ODPI.

important_ddRemarque importante :

ODPI utilisera le template avec la méthode *string.Format*. Il convient donc de l’adapter pour que celui-ci soit compatible.

### Création du template de configuration

Le template de configuration ressemble en beaucoup de points au fichier **.cscfg** généré à l’exception que celui-ci est prévu pour que les paramètres de configuration soient dynamiquement ajoutés par ODPI.

L’objectif, ici, est donc de remplacer l’ensemble des balises **Setting,** qui devront être modifiées dynamiquement, par la chaîne de formatage **{0}**.

Le tableau ci-dessous illustre les modifications à effectuer dans le fichier de configuration ODAF Openturf.

|  |  |
| --- | --- |
| <ServiceConfiguration serviceName="ODAF.Azure" xmlns="<http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceConfiguration>">  <Role name="ODAF.Website.Mvc">  <Instances count="1" />  <ConfigurationSettings>  **<Setting name="DiagnosticsConnectionString" value="DefaultEndpointsProtocol=https;AccountName=XXXX;AccountKey=XXXXXX" />**  **<Setting name="ODAF" value="Server=XXXX.database.windows.net;database=odaf;User Id=XXXXX;Password=XXXXX;MultipleActiveResultSets=True;" />**  **<Setting name="AppName" value="ODAF Openturf" />**  **<Setting name="BitlyLogin" value="XXXX" />**  **<Setting name="BitlyAPIKey" value="XXXXXXX" />**  **<Setting name="tokenExpiryMinutes" value="10080" />**  **<Setting name="UseTokenExpiry" value="false" />**  **<Setting name="BlobStorageEndpoint" value="DefaultEndpointsProtocol=https;AccountName=XXXXX;AccountKey=XXXX" />**  **<Setting name="AdminTwitterUser" value="XXXXXX" />**  **<Setting name="TwitterAppId" value="XXXX" />**  <Setting name="DevMode" value="false" /> </ConfigurationSettings>  </Role>  <Role name="ODAF.SearchIndexRole">  <Instances count="1" />  <ConfigurationSettings>  <Setting name="BlobStorageEndpoint" value="DefaultEndpointsProtocol=https;AccountName=XXXXX;AccountKey=XXXXXXXX" />  <Setting name="DiagnosticsConnectionString" value="DefaultEndpointsProtocol=https;AccountName=XXXX;AccountKey=XXXXXX" />  </ConfigurationSettings>  </Role>  </ServiceConfiguration> | <ServiceConfiguration serviceName="ODAF.Azure" xmlns="<http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceConfiguration>">  <Role name="ODAF.Website.Mvc">  <Instances count="1" />  <ConfigurationSettings>  **{0}**  **<Setting name="UseTokenExpiry" value="false" />**  **<Setting name="DevMode" value="false" />**  </ConfigurationSettings>  </Role>  <Role name="ODAF.SearchIndexRole">  <Instances count="1" />  <ConfigurationSettings>  <Setting name="BlobStorageEndpoint" value="DefaultEndpointsProtocol=https;AccountName=XXXXX;AccountKey=XXXXXXXX" />  <Setting name="DiagnosticsConnectionString" value="DefaultEndpointsProtocol=https;AccountName=XXXX;AccountKey=XXXXXX" />  </ConfigurationSettings>  </Role>  </ServiceConfiguration> |

### Envoi des fichiers vers le compte de stockage blobs

Vous possédez à présent un *package de services Windows Azure* pour votre solution et un *template de configuration* ODPI. Il vous faut à présent envoyer ces fichiers sur votre compte de stockage blobs Windows Azure, plus précisément dans le conteneur **components**.

Pour cela nous vous invitons à vous référer à l’ETAPE 2 – CONFIGURATION DU COMPTE DE STOCKAGE qui explique comment se connecter à un compte de stockage blob Windows Azure et comment envoyer des fichiers dans le conteneur **components.**

## Etape 2 – Création des nouveaux fichiers dans ODPI

Chaque solution Open Data présentée par ODPI est basée sur un ensemble de fichiers pour fonctionner.

Une solution Open Data dans ODPI se décompose de la manière suivante :

* Une classe de configuration située dans le dossier **Models/Config**
* Une vue permettant de saisir les informations et située dans le dossier **Views/ConfTpl**
* Les fichiers de traduction situés dans **Views/ConfTpl**
* Une classe de post déploiement **optionnelle** située dans le dossier **Actions/PostDeploy**

Afin d’illustrer la création d’une application Open Data nous allons décrire dans la suite de cette étape l’ajout d’une solution nommée **OpenIntel**.



1. Dans Visual Studio rendez-vous dans le dossier **Models/Config** puis ajoutez une classe nommée **OpenIntelConfig.cs**.

important_ddRemarque importante :

La convention de nommage pour les fichiers de configuration est la suivante *[NomDeLaSolution]Config.cs*.

1. Cette classe contient l’ensemble des propriétés qui seront exposées et donc configurées par l’utilisateur via ODPI. De même elle contient le template, dans la méthode **BuildSettingsString**, permettant de générer le fichier de configuration Windows Azure adéquat, une méthode **BuildFromData** permettant de mettre à jour les propriétés en fonction des données saisies par l’utilisateur et un nom de **Template** utilisé pour récupérer la vue associée.

    public class OpenIntelConfig : **IOdiAppConfig**

    {

**// Propriétés**

        public string DbName { get; set; }

        public string DbHost { get; set; }

        public string DbUserName { get; set; }

        public string DbUserPassword { get; set; }

        public string BingServiceKey { get; set; }

        public string StorageName { get; set; }

        public string StorageKey { get; set; }

**// Méthode contenant le template de configuration Windows Azure**

        public string BuildSettingsString()

        {

            string template = @"<Setting name=""OIConnectionString"" value=""Server=tcp:{0};Database={1};User ID={2};Password={3};Trusted\_Connection=False;Encrypt=True;"" />

                                <Setting name=""BingServiceKey"" value=""{4}"" />

                                <Setting name=""mapdotnet.adminservicesettings.AzureStorageFactoryAccountName"" value=""{5}"" />

                                <Setting name=""mapdotnet.adminservicesettings.AzureStorageFactoryAccessKey"" value=""{6}"" />

                                ";

            return string.Format(template, DbHost, DbName, DbUserName, DbUserPassword, BingServiceKey,StorageName, StorageKey);

        }

**// Méthode de liaison entre la vue et les propriétés**

        public void BuildFromData(dynamic data)

        {

            DbName = data.dbname;

            DbHost = data.dbhost;

            DbUserName = data.dbusername;

            DbUserPassword = data.dbpassword;

            BingServiceKey = data.bingservicekey;

            StorageName = data.storagename;

            StorageKey = data.storagekey;

        }

**// Nom permettant de retrouver la vue associée**

        public string Template { get { return "OpenIntel"; } }

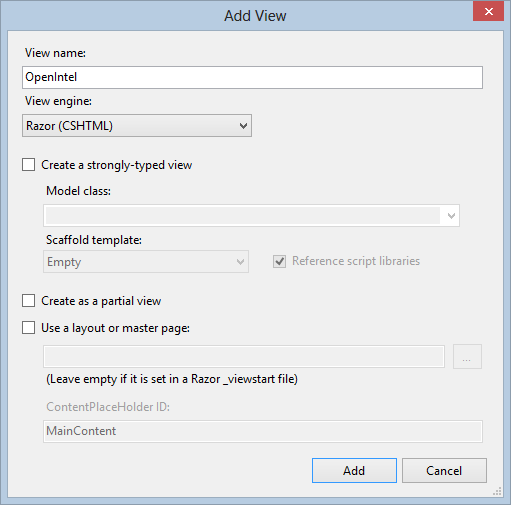
    }

1. La classe de configuration créée il vous faut à présent créer la vue qui sera fournie à l’utilisateur via ODPI. Pour cela faîtes un clic droit sur le dossier **Views/ConfTpl** -> **Add** -> **View**.

Dans la fenêtre **Add View** apparaissant entrez le nom de la vue en fonction du nom présent dans la propriété *Template* de la classe **OpenIntelConfig.cs**.

public string Template { get { return "**OpenIntel**"; } }

Soit :



1. Cliquez sur **Add**. Dans la vue, ajoutez les informations disponibles à la saisie. Typiquement l’ensemble des propriétés ajoutées dans la classe **OpenIntelConfig.cs**.

@{

    Layout = null;

}

<div class="page-header">

    <h4>

        @ODI.Resources.Views.ConfTpl.OpenIntelResource.ApplicationSettings</h4>

        <p>

        @ODI.Resources.Views.ConfTpl.OpenIntelResource.ApplicationSettingsWillAllowDataLab

    </p>

</div>

<div class="control-group">

    <label class="control-label">

        @ODI.Resources.Views.ConfTpl.OpenIntelResource.BingServiceKey</label>

    <div class="controls">

        <input class="input-xlarge" type="text" id="bingservicekey" name="bingservicekey" placeholder="@ODI.Resources.Views.ConfTpl.OpenIntelResource.KeyProvidedByBingAPI" />

    </div>

</div>

<div class="page-header">

    <h4>

        @ODI.Resources.Views.ConfTpl.OpenIntelResource.SQLAzureInfo</h4>

    <p>

        @Html.Raw(ODI.Resources.Views.ConfTpl.OpenIntelResource.IfYouDoNotHaveASqlAzure)</p>

    <p>

        @ODI.Resources.Views.ConfTpl.OpenIntelResource.OnceTheDataBaseIsCreated</p>

</div>

<div class="control-group">

    <label class="control-label">

        <a href="#" data-toggle="modal" data-target="#GetDbHostName" rel="tooltip" data-original-title="@ODI.Resources.Views.ConfTpl.OpenIntelResource.ClickToSeeHow">

            @ODI.Resources.Views.ConfTpl.OpenIntelResource.DatabaseHostname</a></label>

    <div class="controls">

        <input class="input-xlarge" type="text" id="dbhost" name="dbhost" placeholder="@ODI.Resources.Views.ConfTpl.OpenIntelResource.YourServerDatabase" />

    </div>

</div>

...

Pour ajouter un champ de saisie le template type est le suivant :

<div class="control-group">

    <label class="control-label">

        @ODI.Resources.Views.ConfTpl.OpenIntelResource.[NomDeLaVariableDeLocalisation]</label>

    <div class="controls">

        <input class="input-xlarge" type="text" id="[iddeliaison]" name="[iddeliaison]" placeholder="@ODI.Resources.Views.ConfTpl.OpenIntelResource.[NomDeLaVariableDeLocalisation2]" />

    </div>

</div>

important_ddRemarque importante :

*[iddeliaison]* ci-dessus représente la variable saisie dans la methode **BuildFromData** de la classe **OpenIntelConfig.cs** soit pour ajouter un champ lié à la propriété *StorageKey* :

StorageKey = data.**storagekey**;

*[iddeliaison]* sera remplacé par *storagekey*.

Ce qui nous donne :

<div class="control-group">

    <label class="control-label">

        @ODI.Resources.Views.ConfTpl.OpenIntelResource.[NomDeLaVariableDeLocalisation]</label>

    <div class="controls">

        <input class="input-xlarge" type="text" id="**storagekey**" name="**storagekey**" placeholder="@ODI.Resources.Views.ConfTpl.OpenIntelResource.[NomDeLaVariableDeLocalisation2]" />

    </div>

</div>

1. Enfin, l’application étant entièrement localisée il convient de créer les fichiers de ressources pour la traduction qui seront situés dans le dossier **Views/ConfTpl.**

## Etape 3 – Mise à jour des fichiers existants

Les fichiers propres à la nouvelle solution créés, il est temps de configurer ODPI pour afficher cette solution.

La configuration se décompose en 2 parties :

1. La première partie consiste à référencer la vue dans ODPI afin que celle-ci soit automatiquement chargée en mémoire lors du premier lancement. Pour charger la vue associée ouvrez le fichier **main.js** situé dans le dossier **js**.

Naviguez jusqu’à la fin du fichier. Vous trouverez une ligne commençant par **tpl.loadConfigTermplates**.

Cette ligne contient l’ensemble des vues qu’ODPI référence pour les solutions Open Data.

En reprenant l’exemple précédent d’**OpenIntel**, il faut rajouter la valeur *‘OpenIntel’* dans le tableau.

tpl.loadConfigTemplates(['SqlAzure'**, 'OpenIntel'**, 'BlobStorage', 'OpenTurf', 'DataLab', 'CitizenPortalOpenData'], function () { });

1. La seconde et dernière étape de configuration permet d’ajouter la nouvelle application à la liste des applications Open Data disponibles sur la page d’accueil.

* Pour cela ouvrez le fichier **OdiAppRepo** situé dans le dossier **Models**. Dans sa méthode **Initialize()** est défini une liste d’application. Il faut rajouter dans cette méthode votre application à cette liste.
* A la fin de cette méthode ajoutez donc les lignes suivantes pour **OpenIntel**

            Apps.Add(new OdiApp()

            {

                Id = 4,

                DisplayOrder = 4,

                Name = ODI.Resources.Models.OdiAppRepoResource.OpenIntel,

                ConfigHelpText = ODI.Resources.Models.OdiAppRepoResource.OpenIntelConfigHelpText,

                Description = ODI.Resources.Models.OdiAppRepoResource.OpenIntelDescription,

                Configs = new IOdiAppConfig[] { new OpenIntelConfig() },

                Validations = new IValidateAction[] { new SqlAzureValidation(), new TableStorageValidation() },

                PostAction = new OpenIntelPostDeploy(),

                PackageName = "openintel\_0.2.cspkg",

                ConfName = "openintel\_0.1.cscfg",

                RequiredFiles = new string[]

                {

                    "open\_intel\_create\_tables.0.1.sql",

                    "OI\_Sample.mapx",

                    "MapFiles.xml",

                    "ISC.MapDotNetServer.Common.dll",

                    "ISC.MapDotNetServer.Common.Maps.dll",

                    "ISC.MapDotNetServer.Common.Maps.xml",

                    "ISC.MapDotNetServer.Common.xml",

                    "MoveMap.exe",

                    "MoveMap.pdb"

                },

                SiteUrl = "http://{0}.cloudapp.net"

            });

important_ddRemarque importante :

Vous noterez qu’ici aussi le texte est localisé. Pour ajouter ou modifier des informations utilisez le fichier **OdiAppRepoResource.resx** du dossier **Resources/Models.**

* Vous trouverez ci-dessous une description des éléments de configuration de la classe new OdiApp().

| **Méthode** | **Description** |
| --- | --- |
| ***Id*** | Identifiant unique de l’application dans ODPI |
| ***DisplayOrder*** | Définit l’ordre d’affichage des applications |
| ***Name*** | Nom de l’application dans ODPI |
| ***ConfigHelpText*** | Décrit dans les grandes lignes les informations nécessaires lors de la configuration |
| ***Description*** | Description de l’application apparaissant sur la page d’accueil |
| ***Configs*** | Lie une ou plusieurs configurations à l’élément ajouté dans la liste. C’est ici que sera définie l’application correspondant à cette entrée dans la liste. |
| ***Validations*** | Ajoute des éléments de saisie et de validation lors de la configuration. Très utile afin d’inclure automatiquement les éléments de saisie et de valider les informations comme les informations d’une base de données SQL Azure ou encore d’un compte de stockage de type table Windows Azure. |
| ***PostActions*** | Permet d’exécuter des actions post installation. Cette étape permet notamment d’exécuter des scripts SQL, de lancer des programmes automatisés, etc. |
| ***PackageName*** | Nom du package de services Windows Azure présent dans le compte de stockage |
| ***ConfName*** | Nom du template de configuration Windows Azure présent dans le compte de stockage |
| ***RequiredFiles*** | Recense les fichiers qui doivent être obligatoirement dans le compte de stockage hors package de services Windows Azure et template de configuration. |
| ***SiteUrl*** | Url du site final déployé. Généralement http://{0}.cloudapp.net |

Les nouveaux fichiers étant créés il vous faut à présent déployer ODPI dans Windows Azure.

## Etape 4 – Déploiement d’ODPI dans Windows Azure

Cette dernière étape permet de mettre en ligne l’installeur automatique de solution Open Data (ODPI) avec la nouvelle solution que vous avez ajouté.

Pour déployer la solution dans Windows Azure référez-vous à la partie PUBLICATION D’ODPI DANS WINDOWS AZURE

# Pour aller plus loin

Pour de plus amples information sur les solutions proposées par Microsoft à destination du Secteur Public, nous vous invitons à consulter le [site portail Web Microsoft Secteur Public](http://www.microsoft.com/france/entreprises/secteur-public/solutions-pour-secteur-public.aspx)[[7]](#footnote-7).

Pour de plus amples informations sur l'interopérabilité technique des produits et technologies Microsoft avec des logiciels et matériels d'autres fournisseurs, nous vous invitons à consulter le [site portail Web Microsoft Interopérabilité](http://www.microsoft.com/france/interop)[[8]](#footnote-8).

Dans le contexte de la révolution numérique en marche, du foisonnement des technologies, des innovations permanentes, etc. l’interopérabilité, c’est-à-dire la capacité des produits, logiciels et matériels à communiquer entre eux, représente un enjeu fondamental aujourd’hui: à terme, tous les produits, logiciels, matériel doivent pouvoir communiquer entre eux.

Derrière la dimension technique de l’interopérabilité, le véritable enjeu est la liberté de chacun : celle de choisir les outils qui correspondent le mieux à ses besoins, celle de pouvoir accéder à ses données au fil du temps, indépendamment de la technologie utilisée.

Avec l’[annonce majeure](http://www.microsoft.com/presspass/press/2008/feb08/02-21ExpandInteroperabilityPR.mspx)[[9]](#footnote-9) faite au mois de février 2008, Microsoft s’est engagé, pour garantir la pérennité des données et assurer la conformité aux standards de l’industrie informatique, la prise en charge des principales normes internationales, dans une politique d’ouverture sans précédent dans l’histoire informatique, une démarche prospective qui prouve qu’innovation, propriété intellectuelle et interopérabilité ne s’opposent pas.

Cela se traduit notamment par la mise à la disposition de la communauté des développeurs, de plus de 50 000 pages d’informations techniques relatives aux protocoles de communication et aux interfaces de nos produits-phare, et par le développement des relations collaboratives avec les différents acteurs du secteur, y compris les communautés investies dans les logiciels Open source.

Pour de plus amples informations sur les standards d'interopérabilité et normes internationales que Microsoft soutient ou auxquels Microsoft apporte sa collaboration, nous vous invitons à consulter le [site portail Web Microsoft Standards](http://www.microsoft.com/france/standards)[[10]](#footnote-10).

Pour de plus amples informations sur les contributions, le financement et la mise à disposition sous licence libre de logiciels de Microsoft apporte sa collaboration, nous vous invitons à consulter le [site portail Web Microsoft Open Source](http://www.microsoft.com/france/opensource/)[[11]](#footnote-11). Le présent kit en constitue une illustration au même titre par exemple que le [traducteur DAISY XML](http://sourceforge.net/projects/openxml-daisy) à destination des livres audionumériques parlants et vis-à-vis de l’accessibilité des contenus et informations en ligne, notamment dans le cadre du [RGAA (Référentiel Général d'Accessibilité pour les Administrations)](http://references.modernisation.gouv.fr/rgaa-accessibilite)[[12]](#footnote-12).

Pour des informations générales sur les engagements de Microsoft vis-à-vis d’une technologie pour tous (*design for all*) et les solutions d’accessibilité de Microsoft, nous vous invitons à consulter le [site portail Web Microsoft Accessibilité](http://www.microsoft.com/france/accessibilite)[[13]](#footnote-13).

# Restons en contact

 Le groupe LinkedIn [Forum des Architectures Applicatives Microsoft](http://linkd.in/archiappms)[[14]](#footnote-14) regroupe des architectes en informatique qui ont des choix de technologies à faire dans les projets pour lesquels ils travaillent.

L’architecte applicatif, en situation de projet, travaille typiquement aux côtés de la direction de projet pour choisir et assumer des choix techniques en fonction des contraintes du projet (fonctionnalités, délais, ressources). Pour effectuer ces choix à bon escient, il doit connaître ce que le marché offre en termes de technologies. Cela peut prend typiquement deux formes : veille technologique continue, recherches dans le cadre du projet.

L’architecte applicatif a aussi pour rôle de faire le lien entre les équipes de développement et les équipes d’infrastructure et d’exploitation de la future application. Il doit également veiller à ce que ses choix soient bien mis en œuvre pendant le développement.

Ce forum, à l’initiative de Microsoft France, a pour but d’aider les architectes applicatifs dans la dynamique de ce Kit de démarrage, à :

* Faciliter la connaissance de l’offre de Microsoft pour les projets en entreprise (envoi de liens vers des présentations, documents ou livre blanc comme celui-ci, webcasts, conférences, etc.) ;
* Echanger sur des problématiques d’architecture ayant un rapport, même partiel, avec la plateforme.

Cet espace est le vôtre, faites le vivre, nous sommes aussi et surtout là pour vous lire.

1. Microsoft Windows Azure : http://www.microsoft.com/france/windows-azure/ [↑](#footnote-ref-1)
2. Microsoft Windows Azure Training Kit : http://www.microsoft.com/downloads/en/details.aspx?FamilyID=413E88F8-5966-4A83-B309-53B7B77EDF78&displaylang=en [↑](#footnote-ref-2)
3. Microsoft Web Camps Training Kit : http://trainingkit.webcamps.ms/Default.htm [↑](#footnote-ref-3)
4. Windows Azure Team Blog : http://blogs.msdn.com/b/windowsazure/ [↑](#footnote-ref-4)
5. API du service de blobs Windows Azure : http://msdn.microsoft.com/en-us/library/dd135733 [↑](#footnote-ref-5)
6. Windows Azure Service Management API : http://msdn.microsoft.com/fr-fr/library/ee460799.aspx [↑](#footnote-ref-6)
7. Site portail Web Microsoft Secteur Public : http://www.microsoft.com/france/entreprises/secteur-public/solutions-pour-secteur-public.aspx [↑](#footnote-ref-7)
8. Site portail Web Microsoft Interopérabilité : http://www.microsoft.com/france/interop [↑](#footnote-ref-8)
9. Communiqué de presse Microsoft Makes Strategic Changes in Technology and Business Practices to Expand Interoperability : [↑](#footnote-ref-9)
10. Site portail Web Microsoft Standards : http://www.microsoft.com/france/standards [↑](#footnote-ref-10)
11. Site portail Web Microsoft Open Source : http://www.microsoft.com/france/opensource/ [↑](#footnote-ref-11)
12. Référentiel Général d'Accessibilité pour les Administrations : http://references.modernisation.gouv.fr/rgaa-accessibilite [↑](#footnote-ref-12)
13. Site Microsoft France Accessibilité : http://www.microsoft.com/france/accessibilite [↑](#footnote-ref-13)
14. Forum des Architectures Applicatives Microsoft: http://linkd.in/archiappms [↑](#footnote-ref-14)