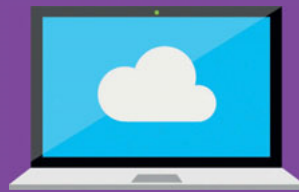




Visual Studio®



Productivité
Collaboration
Mobilité
DevOps



Microsoft

Productivité, collaboration, multiplateforme et multi-devices

Depuis 20 ans, Microsoft ne cesse de faire évoluer Visual Studio : nouvelles fonctionnalités, optimisations, nouveaux supports systèmes et matériels. Visual Studio n'est pas qu'un IDE, mais une véritable plateforme de développement pour les mobiles, le web, le Cloud et le desktop. La nouvelle version 2015 est dans la continuité des versions précédentes, tout en apportant des améliorations notables pour le développeur.

Amélioration de la productivité du développeur ! Pour Microsoft, c'est un objectif. Comment, au quotidien, aider et faciliter le développement et *in fine*, la vie du développeur ? Cela passe par des fonctions comme IntelliSense, CodeMap et une intégration plus poussée avec les SDK et les API (mobiles, cloud, web, desktop, IoT, les tests...).

DevOps. Visual Studio est au cœur du DevOps tel que proposé par Microsoft. Le but est d'améliorer et de rendre toujours plus fluides la communication, les échanges et le travail entre les développeurs et la production. Comment rendre plus souples les processus de développement, de builds, de déploiement et de production des applications ? Visual Studio 2015, comme vous le verrez dans ce livre blanc, s'intègre plus profondément avec Visual Studio Online et tous les services liés (Application Insights, build...)

Mobilité. Plus que jamais, Visual Studio 2015 est l'outil pour développer des applications mobiles (smartphones, tablettes). Il intègre Cordova, supporte parfaitement Xamarin. Et propose des fonctions avancées telles que le C++ pour mobile ! Et demain, ce sera la possibilité de migrer des projets Android et iOS pour les compiler pour Windows Phone(*).

Multi-devices. Pour autant, Visual Studio 2015 n'en oublie pas ce qui a fait sa force : le développement Windows et .NET. L'environnement est naturellement l'outil idéal pour créer des applications Windows 10, Windows 10 Mobile. Et avec la notion de plateforme Windows 10 (allant de l'IoT au Desktop) et des Universal Apps, vous pouvez générer, avec un seul code, des apps fonctionnant sur un smartphone, une tablette, un PC...

Qu'attendez-vous pour tester Visual Studio dès aujourd'hui ?



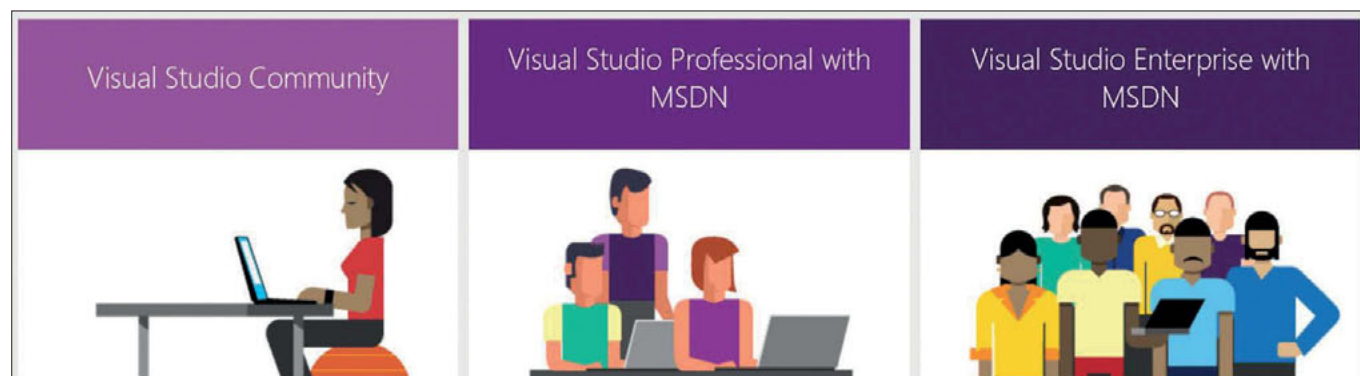
François Tonic
Éditeur & rédacteur en chef de Programmez !

(*) Ces outils sont en pré-versions à l'heure où nous imprimons ce document.

Édito	2
Aperçu de la plateforme	3
Productivité avec Visual Studio 2015	4
Visual Studio Code / Visual Studio Community	8
L'ALM dans les nuages (TFS / VSO)	10
Installation & migration	14
Services Azure	16
Développer pour Windows 10	21
Dev Mobile	27
Containers, Docker...	31
IoT	32
MSDN Platforms	34
Ressources	35

Aperçu de la plateforme Visual Studio

Visual Studio est devenu bien plus qu'une plateforme de développement : IDE, ALM, tests, déploiement, Cloud Computing, multiplateforme. A chaque développeur son Visual Studio !



Jusqu'à la version 2015, Visual Studio se déclinait en plusieurs éditions : Express, Ultimate, Premium, Professional et Test Professional ... Chaque édition correspondait à des usages et des profils de développeurs précis et avec des niveaux de fonctionnalités parfois très différents. A cela se rajoutaient les éditions de Team Foundation Server et de Visual Studio Online (VSO).

Les éditions de Visual Studio 2015

	Visual Studio Community	Visual Studio Professional avec MSDN	Visual Studio Enterprise avec MSDN	Visual Studio Test Professional
Fonctionnalités IDE	**	***	****	-
Architecture et modélisation	*	*	****	-
Outils de tests	*	*	****	**
Debug	***	***	****	-
Accès services VSO	**	**	****	***
Team Foundation Server	-	***	****	****
Outils de collaboration	****	****	****	***
Souscription et bénéfices MSDN	-	***	****	** / ***
Tarif	gratuit (1)	1 199 \$ (2)	5 999 \$ (3)	

Les éditions Premium et Ultimate ont fusionné en Enterprise. Les tarifs indiqués sont ceux du MS Store pour 1 an.

(1) Visual Studio Community est gratuit pour les développeurs indépendants, les projets open source, la recherche universitaire, les formations, l'enseignement et les petites équipes professionnelles. Pour plus d'informations sur les critères d'éligibilité, consultez la page produit Visual Studio Community.

(2) Tarification en ligne Microsoft Store : 1 199 \$ (nouveau) et 799 \$ (renouvellement). Des remises supplémentaires pourront également être obtenues auprès des revendeurs de licences en volume Microsoft.

(3) Visual Studio Enterprise avec MSDN acheté via Microsoft Store en ligne coûte 5 999 \$ (nouveau) et 2 569 \$ (renouvellement), par rapport à Visual Studio Ultimate avec MSDN, qui revenait à 13 299 \$ (nouveau) et 4 249 \$ (renouvellement). Des remises supplémentaires pourront également être obtenues auprès des revendeurs de licences en volume Microsoft.

Les autres produits de la gamme

Visual Studio Code

Environnement de développement gratuit et très léger. Il supporte les langages web les plus populaires. Disponible sur Linux, OS X et Windows.

Visual Studio Express 2015

Éditions gratuites de Visual Studio. Elles se déclinent en Express pour Desktop, Express pour Web, Express pour Windows, Team Foundation Server 2015 Express. Chaque édition se limite à un domaine précis de développement.

MSDN Platforms

MSDN Platforms permet au personnel informatique et aux développeurs Java d'accéder à une large gamme de ressources qui aident votre entreprise à réussir sans le coût supplémentaire de Visual Studio. L'accès à des logiciels passés, présents et futurs permet au personnel informatique de disposer de tout ce dont il a besoin pour installer des environnements de développement/test et permet à des développeurs hétérogènes de générer des applications de qualité élevée pour le Web, le cloud et Windows. De plus, les abonnés peuvent se connecter, collaborer et livrer à temps avec l'aide de la solution ALM de Microsoft. ▲



Productivité avec Visual Studio 2015

Comme pour chaque version majeure de Visual Studio, Microsoft apporte son lot de nouvelles fonctionnalités. D'autres, quant à elles, deviennent plus matures et accessibles sur des versions moins complètes. Tout cela permet d'améliorer grandement la productivité des développeurs.



Pierre-Henri Gache,
Consultant Senior à Cellenza

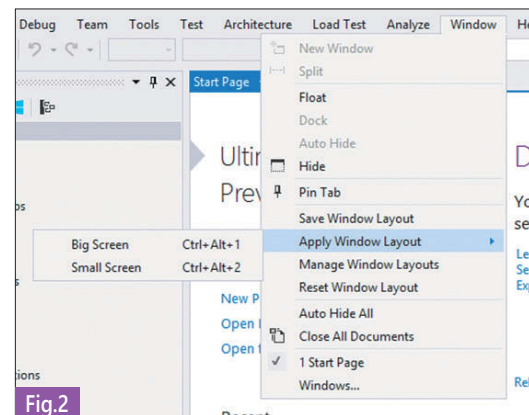
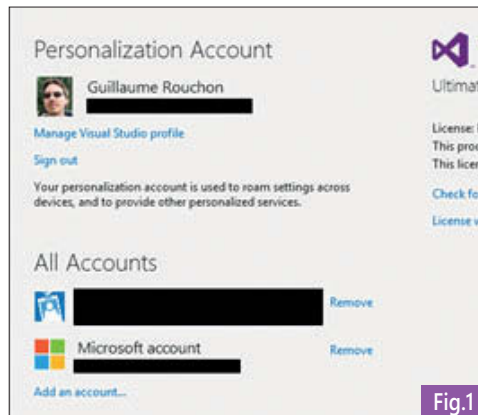


Michel
Perfetti
Consultant
Senior à
Cellenza et
MVP VS ALM



Guillaume Rouchon
Consultant Senior
à Cellenza
et MVP VS ALM

cellenza
DOES IT BETTER



Avec la version 2015 de Visual Studio, Microsoft revisite les différentes distributions. On retrouve maintenant les versions Community, Express, Test Professionnel, Professionnel et Entreprise. L'apport majeur de productivité vient de la fusion des anciennes versions Premium et Ultimate en Entreprise. Les développeurs vont donc pouvoir profiter de l'ensemble des outils présents dans Ultimate. CodeLens est un bon exemple et nous le verrons plus en détail dans la suite de cet article. Auparavant uniquement disponible en version Ultimate, peu de personnes pouvaient en disposer. CodeLens devient donc plus accessible en étant même accessible dès la version Professionnel. Tous les développeurs qui passent à Visual Studio 2015 auront donc CodeLens s'ils travaillent sur TFS 2013-2015, Visual Studio Online ou un repository Git local. Il y a bien d'autres nouveautés que nous vous proposons de découvrir au travers de moments clés dans le développement (l'écriture du code, puis le débogage et l'analyse de performance). Voyons en détail ce que nous propose cette version pour vous aider à améliorer votre flux de travail.

IDE

Setup

Le premier ajout important à Visual Studio 2015 se retrouve dès l'installation. En effet l'installateur propose, en plus des composants classiques, des outils tiers qui s'intègrent directement à Visual Studio tel que :

- ▶ Apache Cordova et Xamarin Starter Edition ;
- ▶ Le SDK Android et un émulateur ;
- ▶ SQL Server Data Tools ;
- ▶ Office Developer Tools ;
- ▶ Applications Universelles Windows ;

- ▶ Les outils de développement pour Windows 8.1 et Windows Phone ;
- ▶ Les outils de développement pour Python ou PowerShell ;
- ▶ Les outils pour se connecter à GitHub.

Ces composants, accessibles directement dans l'installateur de Visual Studio, vont nous éviter de chercher à droite et à gauche ces extensions et de les maintenir à jour, étant donné qu'ils s'intègrent aussi au système de notification interne pour les mises à jour. Ce nouveau système de notification gère aussi les packages de composants de Visual Studio et est capable de mettre à jour lui-même certains packages.

Gestion des comptes

Dans la précédente version, il était possible d'associer un seul compte à Visual Studio qui permettait de sauvegarder son paramétrage. Par contre, dès que l'on accédait à Azure avec plusieurs comptes, il fallait jongler entre les différents identifiants de ces comptes. Avec Visual Studio 2015, afin de ne pas avoir à ressaisir perpétuellement ses identifiants, Microsoft a ajouté un gestionnaire de comptes (**Fig. 1**).

Lorsque des identifiants sont nécessaires, Visual Studio propose de les sélectionner dans une liste existante. Dans le cas de compte d'entreprise, il est aussi possible de filtrer les éléments qui seront affichés.

Gestion de l'affichage

Les écrans évoluent et il n'est plus rare de trouver des développeurs avec des écrans tactiles. Pour améliorer la productivité de ces développeurs, Visual Studio supporte maintenant des gestes :

- ▶ Pinch pour zoomer ou dézoomer ;
- ▶ Double tap pour sélectionner un mot ;

- ▶ Tap dans la marge pour sélectionner une ligne ;
- ▶ Appui long pour faire apparaître le menu contextuel.

D'autre part, nous travaillons aussi de plus en plus avec des ordinateurs portables, parfois connectés à un écran secondaire. Or ces différents écrans n'ont généralement pas la même résolution et il fallait dans ce cas réarranger les fenêtres dans Visual Studio afin de s'adapter à la résolution de l'écran en cours. Avec la version 2015, il est possible de sauvegarder l'organisation des fenêtres et de rappeler une configuration préalablement sauvegardée (**Fig. 2**). En bonus, ces sauvegardes font partie des informations associées à votre profil et seront donc accessibles de n'importe quel ordinateur dès que vous vous connectez à Visual Studio avec votre compte Microsoft.

Éditeur de code

L'éditeur de code reçoit lui aussi des améliorations afin de permettre de gagner du temps via l'affichage d'informations et d'avertissements en temps réel bénéficiant également d'une meilleure lisibilité. Avec l'intégration du compilateur Roslyn, Visual Studio peut, en temps réel, analyser le code saisi et ainsi afficher directement le résultat de ces analyses. On retrouvera par exemple les résultats des analyseurs de code (anciennement FxCop et Code Analysis). Code Analysis a d'ailleurs lui aussi été repensé afin de prendre en compte Roslyn. Les règles sont maintenant distribuées sous forme d'assemblées à ajouter dans ses projets via des packages NuGet. L'analyseur pourra aussi proposer le code à mettre en place pour être en adéquation avec la règle (**Fig. 3**). Le refactoring de code a lui aussi évolué. L'ensemble des fonctionnalités

présentes en C# est maintenant disponible en natif pour les langages C++ et VB.Net. On retrouvera par exemple pour le C++ les options suivantes :

- ▶ Renommer un élément (variable, méthode, classe...);
- ▶ Extraire une fonction d'un code existant ;
- ▶ Implémenter toutes les méthodes virtuelles ;
- ▶ Créer la définition d'une fonction depuis sa déclaration ou inversement ;
- ▶ Déplacer la définition d'une fonction entre le fichier d'en-tête et le fichier source ;
- ▶ Convertir une chaîne contenant des caractères d'échappement en chaîne littérale (C++ 11).

L'accès au refactoring se fait via le nouveau raccourci CTRL + <point> et la fonctionnalité est extensible permettant à la communauté de proposer ses propres fonctions.

Le renommage (l'une des fonctionnalités les plus utilisées) a aussi reçu une mise à jour spécifique. En sélectionnant un élément et en lançant le renommage, l'ensemble des références sont sélectionnées et mises à jour en direct au fur et à mesure de la frappe (Fig. 4).

La gestion des points d'arrêt a été intégrée à l'IDE. Plus de fenêtre modale, mais à la façon des « peek windows ». Dans la même idée, la gestion des exceptions en « Debug » a été améliorée : plus besoin non plus d'ouvrir une fenêtre modale, cette fois-ci les exceptions déménagent dans une fenêtre. Le changement de configuration n'aura jamais été aussi facile.

XAML & Blend (MPE)

Le XAML est partout sur la plateforme Microsoft, que ce soit dans les applications type Desktop en WPF ou les différentes

déclinaisons de Windows Phone ou Windows 8 à 10. D'un simple éditeur XML au départ, l'éditeur XAML intègre maintenant le même genre de fonctionnalités qu'un éditeur de code classique dans Visual Studio. Côté édition, IntelliSense a été amélioré : plus de frustration quand rien n'apparaît alors que la donnée attendue est basique. L'intégration de Blend a été aussi renforcée, ainsi que la gestion des différentes résolutions, fonctionnalité critique lorsque l'on écrit des applications mobiles. Pour le « Debug », deux fonctionnalités à elles seules valent le coup de passer à Visual Studio 2015 : le « live visual tree » et le « live property explorer ». Pour résumer simplement : nous pouvons naviguer dans l'arbre des composants et pour chaque composant, nous avons désormais la possibilité d'en modifier les propriétés.

Nuget 3.0 Nouvelle interface

La première chose que l'on remarque de suite est le changement total de l'interface graphique. Fini le pop-up qui s'ouvrait dans une fenêtre, le gestionnaire de package prend désormais place dans un onglet de Visual Studio. Ainsi vous pouvez gérer vos packages tout en conservant la possibilité de consulter les éléments de votre projet. Les différentes commandes ont été réagencées, mais on retrouve immédiatement ses petits : un champ de recherche, une liste des packages et le descriptif de chacun. Vous me direz qu'à ce stade il n'y a pas grand-chose de nouveau, passons à la suite pour vous faire changer d'avis. (Fig. 5)

Choix de la version & Mise à jour

La première nouveauté notable est la possibilité de choisir la version du package

que l'on souhaite installer. Vous avez donc le choix d'installer par défaut la dernière version ou bien une version antérieure disponible dans la liste de sélection. Ensuite si vous souhaitez mettre à jour vos packages, vous pouvez au préalable visualiser la liste des projets contenant le package en question et quelle est la version utilisée. Vous avez alors le loisir de mettre à jour tout ou partie de vos packages. Mais la fonctionnalité la plus intéressante et qui vous fera gagner le plus de temps est l'action « Consolidate ». En effet lorsque vous sélectionnez cette option, vous allez uniformiser la version du package choisi. Admettons que vous ayez les versions 6.0.4 et 6.0.8 du package « Newtonsoft.Json », chose assez courante quand vous avez une solution avec des projets web. En fonction du choix effectué, seuls les packages concernés seront mis à jour avec une montée ou une descente de version.

IntelliTest

Imaginons qu'il soit possible d'avoir un outil capable de générer ses propres tests unitaires, nous pourrions gagner beaucoup de temps dans nos développements. Bien évidemment, un tel outil n'existe pas et il vous faudra toujours faire l'effort de tester votre code. Mais IntelliTest peut vous donner un sérieux coup de main et vous faire gagner du temps dans cette tâche. Ceci notamment dans l'écriture des tests des cas limites, d'autant plus que ce sont toujours ces tests que l'on néglige le plus. Mais avant cela, regardons comment fonctionne l'outil. Basé sur « Pex », votre code va être analysé pour déterminer tous les chemins d'exécution possibles. Chaque chemin donnera lieu à au minimum un test unitaire. Ensuite cette même analyse donnera lieu à la génération des paramètres des méthodes utilisées dans le test. Prenons

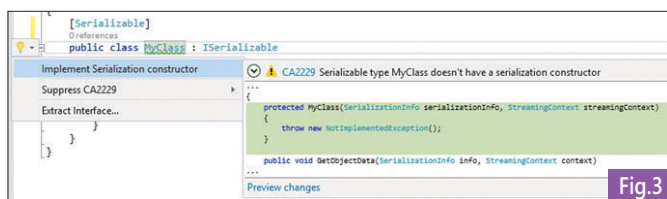


Fig.3

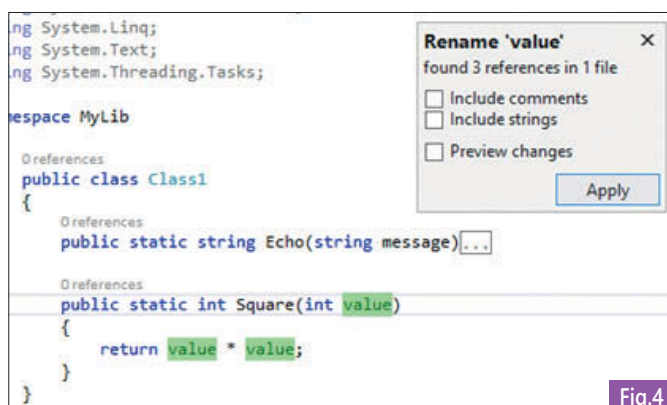


Fig.4

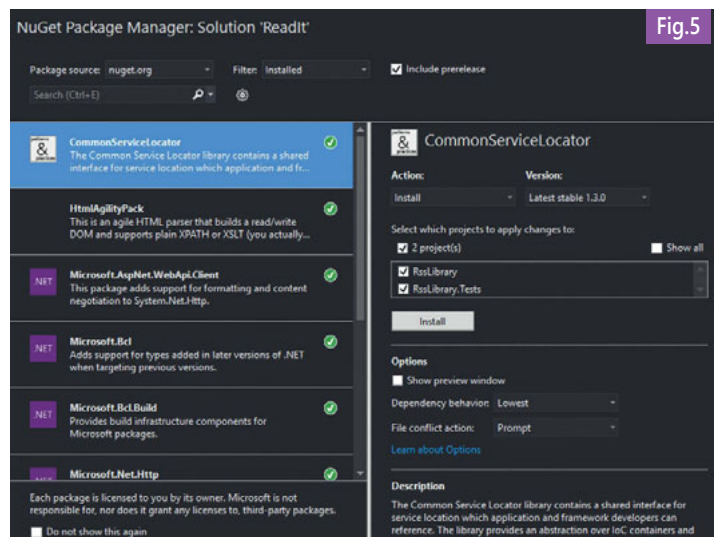


Fig.5

comme exemple une méthode effectuant une division, IntelliTest générera un appel à la méthode avec « 0 » comme paramètre afin de tester si la division par zéro est prise en charge. De même des valeurs maximum et minimum pour les paramètres seront utilisées pour vérifier qu'il n'y a aucun dépassement de capacité dans le code. Il est également possible de donner des instructions à IntelliTest. Si une méthode comprend un entier en paramètre dont la valeur est bornée, il est possible de l'indiquer. Les tests générés auront alors des paramètres dont la valeur sera toujours comprise entre les bornes indiquées. Pour conclure sur ce sujet, il faut prendre IntelliTest pour ce qu'il est : une aide à la génération des tests. Il ne pourra pas vous remplacer dans cette tâche, mais il devrait vous permettre de vous concentrer sur l'essentiel : écrire les tests à valeur ajoutée.

Git + GitHub

Arrivée officiellement dans Visual Studio 2013, l'intégration de Git pouvait être qualifiée de légère, car beaucoup trop de manipulations basiques nécessitaient encore la ligne de commande ou des outils annexes comme « SourceTree » ou « GitHub for desktop ».

Avec Visual Studio 2015, l'intégration a été beaucoup plus poussée :

- La fenêtre d'historique rend compte des branches et de leurs interactions ;
- Les branches sont plus faciles à gérer via un arbre dans le Team Explorer ;
- Github est intégré via une extension qui fournit une expérience similaire à Visual Studio Online ou Team Foundation Server.

Diagrammes d'architecture & Code Maps

Le menu « Architecture » a été de nouveau revu pour être encore plus simplifié et pour produire des « code maps » encore plus simples à utiliser :

- Des codes couleurs permettent d'identifier les différents types de projets (en particulier les tests) ;
- Les graphes complexes s'affichent plus vite, comme pour le rafraîchissement (il n'y a plus de rebuild à chaque changement dans l'affichage !) ;
- Ils sont disponibles pour C#, C++ et VB.Net.

Les diagrammes de couche sont aussi disponibles pour valider le découplage de vos applications.

Très peu de personnes auparavant utilisaient ces fonctionnalités, car elles étaient limitées à la version Ultimate, vous

aurez donc l'occasion de les mettre à profit lors de vos développements.

CodeLens

CodeLens est beaucoup plus intégré à Visual Studio qu'avant. Si vous aviez déjà l'habitude de travailler avec, vous remarquerez que la nouvelle gestion de l'historique affiche une timeline des modifications, un peu comme le fait GitHub sur ses repositories.

Dans Visual Studio 2013, CodeLens n'était disponible que pour C# ou Visual Basic. Maintenant pour tous les autres fichiers les informations génériques sur l'historique, les auteurs, c'est-à-dire tout ce qui n'est pas spécifique à un langage s'affichent en bas du fichier.

Une remarque qui peut vous éviter bien des frustrations : il arrive que même sur TFS 2013 ou TFS 2015, CodeLens ne soit pas complètement disponible. Cela vient du fait que cette fonctionnalité est désactivable côté serveur.

Et généralement, c'est le cas pour éviter l'usage de ressources inutiles côté serveur. Dans ce cas, il faut demander à votre administrateur TFS de le réactiver sur vos collections et d'attendre l'indexation de vos codes sources.

Le débogueur & Diagnostic de performances

Depuis le début de Visual Studio, nous avons la possibilité d'utiliser des outils de diagnostics. Les versions avançant, de nouveaux outils sont apparus, spécifiques à des plateformes ou non, d'autres se sont perfectionnés. Il n'était jamais vraiment facile de savoir quel outil il fallait pour une technologie donnée ou même si un outil existait. Avec Visual Studio 2015, tous les outils de diagnostics sont rassemblés dans un Hub. Ce dernier est dynamique et s'adapte en fonction des applications que

vous voulez analyser : plus besoin d'aller sur MSDN pour savoir quel outil fait quel boulot ! De plus, l'optimisation des performances est devenue un enjeu majeur du développement. Proposer aujourd'hui une application optimisée qui consomme peu de ressources, mais dont l'interface n'est pas fluide, n'est pas envisageable.

De même si votre usage du réseau n'est pas parfait dans le cadre d'applications mobiles, ayant souvent un débit limité, l'expérience utilisateur ne sera pas à son niveau optimum et le seul feed-back que vous aurez de vos utilisateurs sera une mauvaise note avec comme commentaire « ça rame ». Nous allons voir le genre de scénarios pris en compte dans Visual Studio 2015.

Analyse des performances

Généralement lorsqu'un développement arrive à terme, le besoin d'analyser et d'améliorer les performances se fait sentir. Il existe plusieurs méthodes pour effectuer ces opérations, comme par exemple poser des sondes dans son code, juger « au feeling » la réactivité de son travail ou encore utiliser un profiler. C'est ce dernier type d'outil qui offrira bien évidemment les meilleurs résultats : c'est en quelque sorte une façon fiable d'ajouter des sondes pour analyser son application.

Cependant, avant l'arrivée de Visual Studio 2015, il fallait recourir à des outils tiers pour disposer des meilleures solutions. Mais les outils se sont grandement améliorés et il est désormais possible d'avoir les mêmes bénéfices que la concurrence et mieux encore.

La première chose à savoir est que ces outils sont accessibles depuis le menu « Debug » sous la dénomination « Diagnostics Tools » et sont de deux types. Le premier « Show Diagnostics Tools » est un peu différent de ce que vous auriez pu voir auparavant, car il fournit des informations « en live » durant la

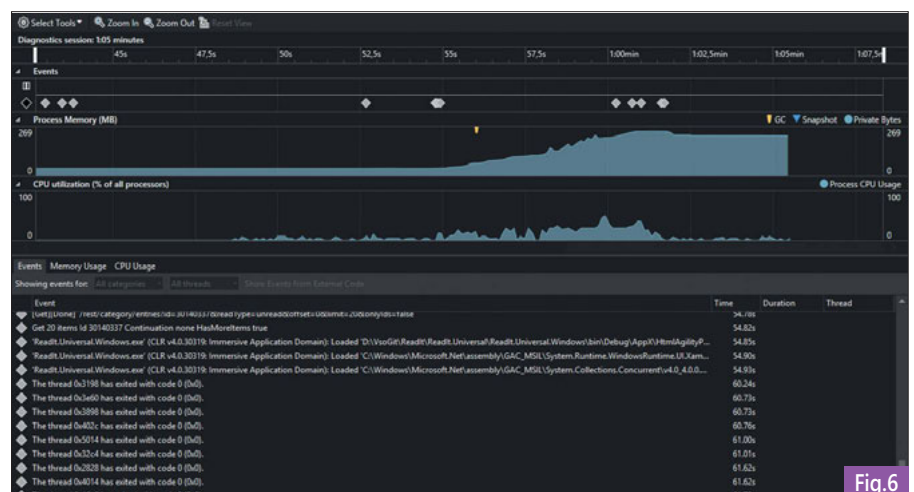


Fig.6

session de débogage. Le second est plus classique, car il se rapproche d'un profiler standard avec une capacité d'analyse plus étendue. Voyons maintenant en détail ces deux outils.

Diagnostics en live

Ce nouvel écran s'active automatiquement lorsque la session de débogage est lancée et se décompose en deux parties. La première comporte des graphiques avec les différents événements de l'application, la consommation processeur et l'occupation mémoire. La seconde partie de l'écran contient le détail des événements et permet de prendre des instantanés de la mémoire.

(Fig. 6)

La force de cet outil est de corréler tous ces éléments ensemble avec un affichage en live. Imaginons que vous ayez un algorithme dans votre code très consommateur en processeur ou en mémoire, vous pouvez insérer dans votre code un message (via le fameux « Debug.WriteLine » par exemple) indiquant les points clés de votre code. Ensuite ces éléments apparaîtront dans les événements et vous pourrez alors constater s'il y a surconsommation de ressources à ce

moment précis. Ensuite il est fort probable que votre application ait généré quelques exceptions, surtout dans les phases d'optimisation, car l'application n'est pas forcément finalisée. Ces exceptions apparaîtront sous forme de losanges rouges sur le graphique et dans la liste des événements. Dans cette dernière, si vous sélectionnez une de vos exceptions, vous pourrez accéder à « l'Historical debugging ». Cette fonctionnalité permet de diagnostiquer une exception comme si vous étiez arrêté dessus. Vous verrez le code où elle s'est produite ainsi que sa description via la fenêtre « Locals ». Dans la fenêtre « Historical Stack Trace », vous pourrez également avoir accès à la pile d'appels.

Le dernier point intéressant est sa capacité à prendre des instantanés de la mémoire. En effet, bien que l'on puisse surveiller son évolution, on a souvent besoin d'avoir un état des lieux précis pour diagnostiquer un problème. Le principe des instantanés est relativement simple : il s'agit de prendre plusieurs photos du nombre et de la taille des instances d'objets puis de les comparer. On peut ainsi en déduire quels sont précisément les types à l'origine d'une

surconsommation par exemple. Pour se faire, rien de plus simple, un bouton « Take snapshot » est disponible dans l'interface. En accumulant le nombre d'instantanés, vous aurez la possibilité de mesurer l'évolution de la mémoire par comparaison. (Fig. 7)

Profiler

Le second outil, accessible sous la dénomination « Start Diagnostic Tools Without Debugging » requiert tout d'abord de compiler le projet en mode « Release ». Ensuite avant de lancer l'analyse, il faut choisir les métriques que l'on souhaite obtenir. Voici une liste des principaux profils en dehors des classiques « processeur » et « mémoire » :

- Application Timeline : permet de mesurer la réactivité de l'application ;
- Network : permet de monitorer l'ensemble des appels réseaux ;
- HTML UI Responsiveness : similaire au premier point pour une application web.

(Fig. 8)

Le premier point positif de cet outil, que l'on peut remarquer dès le début de la session, est que les performances de l'application mesurée ne sont pas impactées par l'analyse. Ce n'est pas anecdotique, car les profilers ont trop souvent l'habitude d'allonger les temps d'exécutions et donc de rendre pénible l'analyse des performances. Ensuite le fait de proposer des outils d'analyse de la réactivité de l'application est un vrai point positif, car les développements actuels sont massivement multithreads, il n'est donc pas rare de bloquer le thread d'interface. Enfin, la possibilité de générer des rapports très complets devrait finir de vous convaincre d'utiliser toutes ces possibilités.

Conclusion

Comme à chaque version de Visual Studio, nous nous posons tous la même question : est-ce que je peux migrer sans risque ? Depuis Visual Studio 2012, Microsoft a travaillé énormément pour la compatibilité ascendante des projets. Les projets web sont plus liés à une version de .NET qu'à une version de Visual Studio, de même que d'autres composants externes comme la partie BI ou OLTP avec SQL Server Data Tools ou Typescript qui sont communs à toutes les versions. Visual Studio 2015 comme son prédécesseur sait cibler les versions inférieures du framework .NET donc pas besoin de passer à 4.6. Si vous faites du web ou client lourd (WPF/Winforms) ou C+++, allez-y.

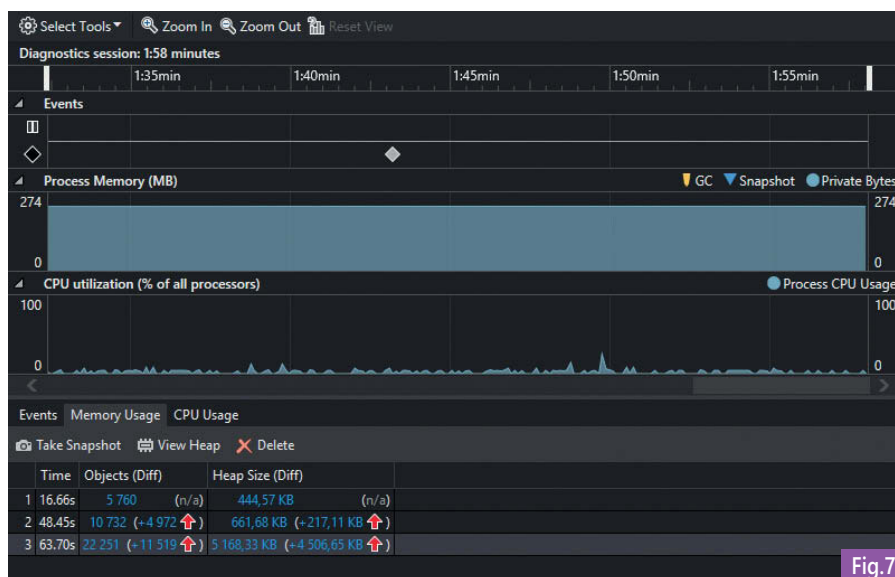


Fig.7

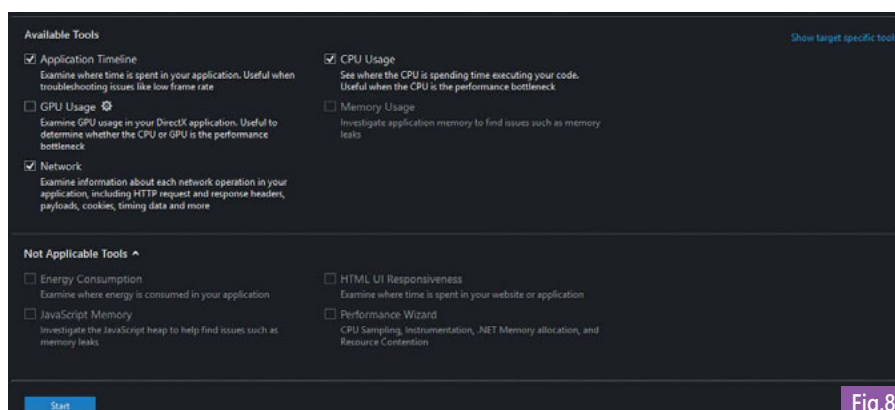


Fig.8

Visual Studio Code/Community

Parmi les solutions gratuites proposées par Microsoft, on retrouve 2 éditeurs de code offrant des fonctionnalités intéressantes et différentes : Visual Studio Code et Visual Studio Community.



Georges DAMIEN
Consultant .NET chez Cellenza
Blog : <http://georgesdamien.com>



Visual Studio Code est un éditeur de code allégé à l'image de « Sublime Text » qui est plutôt orienté Web et présente l'avantage d'être disponible sur Windows, OS X et Linux. Visual Studio Community est une version plus complète qui permet le développement d'applications tout comme sur les versions payantes de Visual Studio (Professionnel/ Entreprise) avec comme avantage une utilisation commerciale notamment pour les développeurs indépendants. Cette version présente moins de fonctionnalités que les versions payantes, mais propose tout de même la majorité des fonctions intéressantes de l'IDE telles que le debugging, le testing, etc.

VISUAL STUDIO CODE



Visual Studio Code (VS Code), une version light et gratuite de « Visual Studio » orientée Web.

C'est quoi ?

Visual Studio Code est tout comme les autres versions de VS un éditeur de code. Cette nouvelle monture est très légère et s'installe assez rapidement (contrairement aux versions payantes qui sont également plus exigeantes en ressources et occupent plusieurs Gigaoctets sur votre disque). Ça se passe ici : <https://code.visualstudio.com/>

Partout et pour tous

Visual Studio Code est cross-plateformes : vous pouvez l'installer sur votre Mac, ou sur Linux et, bien sûr, Windows. Visual Studio Code est gratuit.

Un IDE riche pour le Web

Visual Studio Code propose une coloration syntaxique par défaut pour la majorité des langages orientés Web, mais pas seulement. Les langages déjà pris en charge sont : C++, jade, PHP, Python, XML, Batch, F#, DockerFile, Coffee Script, Java, HandleBars, R, Objective-C, PowerShell, Luna, Visual Basic, Markdown. L'intelliSense est disponible pour les langages suivants : JavaScript, JSON, HTML, CSS, LESS, SASS.

Quelques fonctionnalités intéressantes

Chargement des fichiers, projets et dossiers

Avec VS Code, on peut charger des projets Visual Studio standards, ouvrir directement des fichiers, mais charger également un dossier. Par exemple, si vous avez un petit site Web contenant quelques pages HTML, PHP, css... vous pourrez ouvrir directement l'arborescence du dossier contenant votre site. (Fig. 1 et 2).

J'apprécie personnellement cette fonctionnalité, car elle permet d'avoir accès à toute l'arborescence de notre site sans pour autant avoir la notion de « projet ou solution Visual Studio » qui n'est pas forcément nécessaire pour un petit site Web.

L'édition « Side by Side »

Une autre fonctionnalité intéressante est la possibilité d'ouvrir côte à côte plusieurs pages : on pourrait imaginer, par exemple, ouvrir une page HTML, la page CSS ainsi que la page JS associée directement : (Fig. 3). Le plus de cette fonctionnalité réside dans

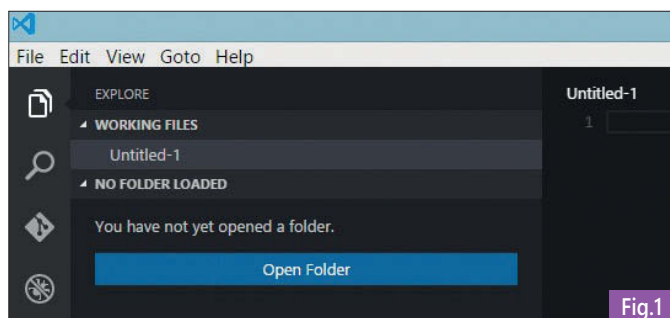


Fig.1

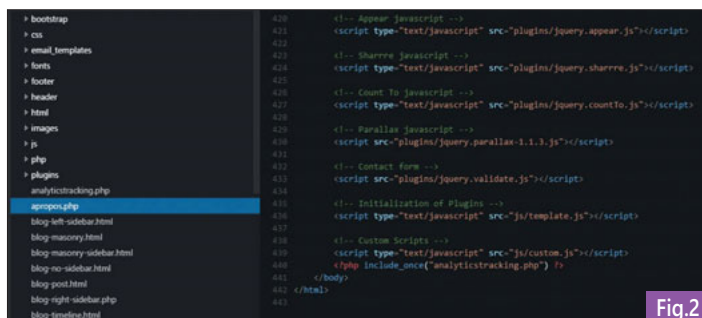


Fig.2

VS Code : Prise en charge des dossiers

VS Code : Aperçu de l'arborescence d'un dossier + coloration syntaxique

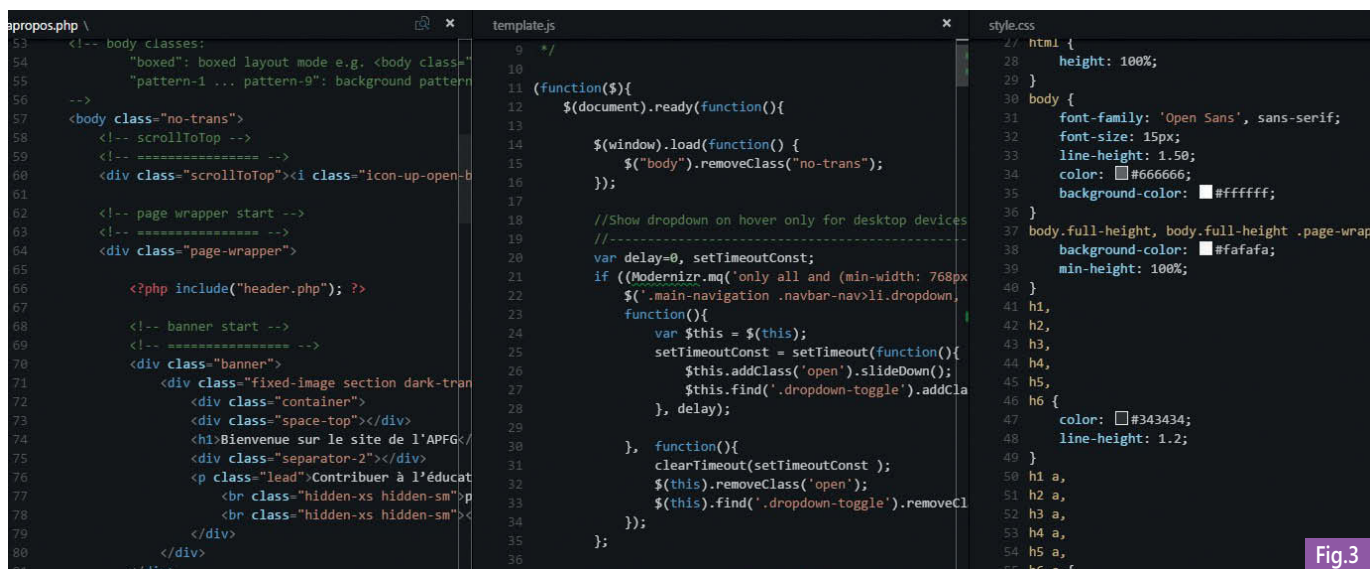


Fig.3

VS Code : Side by Side Editing

l'ergonomie de son utilisation : un simple « drag and drop » affiche les fichiers voulus côte à côte dans la limite de 3 fichiers.

La documentation

Vous trouverez sur le lien <https://code.visualstudio.com/Docs> une documentation assez complète de l'outil présentant notamment l'ensemble des fonctionnalités de façon concise vous permettant une prise en main plus aisée.

User Voice

Toujours pas convaincu(e) ? Vous n'êtes pas satisfait(e) des fonctionnalités proposées ? Ça tombe bien, Microsoft a pensé à vous : Vous avez des suggestions pour améliorer ou rajouter des fonctionnalités ? Microsoft offre la possibilité à tout le monde via « User Voice » de proposer et de voter pour des fonctionnalités souhaitées qui manqueraient à ses produits afin de les améliorer.

Lien d'accès : <https://visualstudio.uservoice.com/forums/121579-visual-studio>

Conclusion

Visual Studio Code est simple, rapide et pourrait remplacer dignement un « Notepad++ » voire même un « WebStorm ». Après une première utilisation, la chose qui me vient l'esprit est le fait de l'utiliser comme éditeur de texte par défaut (surtout si vous êtes habitué à travailler avec Visual Studio au quotidien). De nombreuses évolutions sont à venir. Parmi les fonctionnalités non citées, on retrouve le contrôle de version (Git) intégré, la prise en charge d'ASP.NET 5, de NodeJS, le debugging (pour certains langages), la gestion de Tasks (pour faire du

Make, Ant, Gulp, Jake, Rake or MSBuild), etc. En bref, l'outil semble être suffisamment complet et à la fois léger pour faire du Web dans la partie « FRONT END ». Un de ses points forts est le côté multiplateformes, et les développeurs .NET ne seront pas dépayés en voyant l'IDE. Reste à voir les limites de l'outil si on souhaite aller plus loin, mais ce n'est probablement pas l'objectif de cet IDE : c'est justement là où Visual Studio Community vient combler les manques de cet IDE qu'on pourrait plutôt qualifier de simple éditeur de code.

VISUAL STUDIO COMMUNITY

Visual Studio Community est une version gratuite de Visual Studio. C'est un véritable IDE avec un grand nombre de fonctionnalités à l'image des versions payantes de Visual Studio. Cette nouvelle mouture est installable uniquement sur Windows contrairement à Visual Studio Code.

Un IDE riche et gratuit

Visual Studio Community est gratuit et vise essentiellement les petites structures (moins de 250 PC ou avec un chiffre d'affaires relativement faible) ainsi que les organisations non commerciales ou les indépendants par exemple. La dernière version est téléchargeable depuis le lien suivant : <https://www.visualstudio.com/fr-fr/downloads/download-visual-studio-vs.aspx>

Q : Qui peut utiliser Visual Studio Community ?

A : Pour les développeurs individuels : n'importe quel développeur individuel peut utiliser Visual Studio Community pour créer ses propres applications gratuites ou payantes. Pour les organisations :

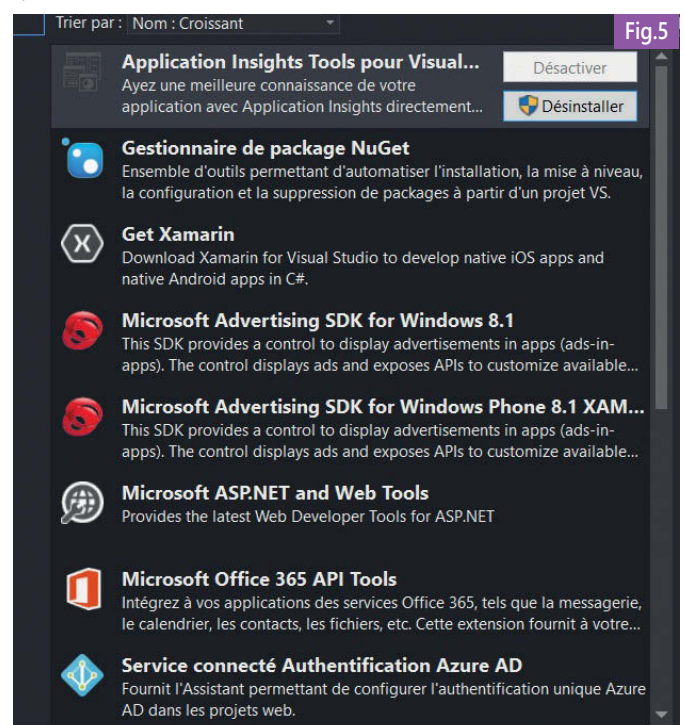
- ▶ Dans un environnement d'apprentissage en classe, pour la recherche universitaire ou pour la contribution à des projets open source, un nombre illimité d'utilisateurs au sein d'une organisation peut utiliser Visual Studio Community
- ▶ Dans les entreprises de moins de 250 PC ou qui génèrent moins de 1 million de dollars en recettes annuelles, jusqu'à 5 utilisateurs peuvent utiliser Visual Studio Community.
- ▶ Dans les entreprises de plus de 250 PC ou qui génèrent plus de 1 million de dollars en recettes annuelles, aucune utilisation n'est autorisée au-delà de l'open source, la recherche universitaire, et les scénarios d'apprentissage en classe décrits ci-dessus.

Les fonctionnalités

Visual Studio Community est un véritable IDE. Vous pouvez développer des projets d'envergure à l'identique des versions payantes : vous pourrez profiter des fonctionnalités de debugging et diagnostics, d'outils de tests, d'intégration, d'outils

Expand all	MSDN Subscriptions					Fig.4
	Visual Studio Community	Visual Studio Professional	Visual Studio Enterprise	Visual Studio Test Professional	MSDN Platforms	
Debugging and Diagnostics	■■■	■■■	■■■	■■■	■■■	
Testing Tools	■■■	■■■	■■■	■■■	■■■	
Integrated Development Environment	■■■	■■■	■■■	■■■	■■■	
Development Platform Support	■■■	■■■	■■■	■■■	■■■	
Architecture and Modeling	■■■	■■■	■■■	■■■	■■■	
Lab Management	■■■	■■■	■■■	■■■	■■■	
Team Foundation Server	■■■	■■■	■■■	■■■	■■■	
Visual Studio Online services	■■■	■■■	■■■	■■■	■■■	
Collaboration Tools	■■■	■■■	■■■	■■■	■■■	
Supported Usage Scenarios	■■■	■■■	■■■	■■■	■■■	
MSDN Subscription Software and Services for Production Use	■■■	■■■	■■■	■■■	■■■	
MSDN Subscription: Software for Development and Testing	■■■	■■■	■■■	■■■	■■■	
Other MSDN Subscription Benefits	■■■	■■■	■■■	■■■	■■■	

Extrait du comparatif des différentes versions de Visual Studio



Aperçu de l'interface de téléchargement d'extensions

collaboratifs, etc. Un comparatif plus détaillé se trouve à l'adresse suivante : <https://www.visualstudio.com/en-us/products/compare-visual-studio-2015-products-vs.aspx> (Fig. 4)

Possibilité d'extension

Visual Studio Community donne la possibilité d'installer une multitude d'extensions (plus de 2000) telles que : Web Essential, Xamarin (pour le développement multiplateforme). Vous y accédez depuis l'onglet « outils », « extensions et mise à jour... » de l'IDE. (Fig. 5).

Vous pouvez également développer et intégrer vos propres extensions notamment grâce au SDK Visual Studio. Pour en savoir plus :

<https://msdn.microsoft.com/en-us/library/bb166441.aspx>

Vous pouvez également utiliser le compilateur Roslyn pour créer votre propre analyseur et générateur de code.

Développement multiplateforme

Grâce à un ensemble d'extensions telles que « Xamarin for Visual Studio », « Apache Cordova », « VS Tools for Unity 2D-3D » « C++ for cross-platform », vous pourrez développer des applications pour Android, iOS et Windows.

Fonctionnalités manquantes

Parmi les fonctionnalités manquantes on peut citer :

- ▶ outils de testing : absence du code coverage, du code UI Testing, l'IntelliTest,... (à noter que ces fonctionnalités sont également absentes de la version Professionnel).
- ▶ environnement de développement intégré : absence de « CodeLens » et de « Code Clone ».
- ▶ architecture-modélisation : absence des diagrammes de couches d'architecture & CodeMap.

A noter également l'absence de la prise en charge de Team Foundation Server. Et bien sûr pas d'abonnement MSDN attaché...

Conclusion

Visual Studio Community, grâce à sa gratuité, donne la possibilité d'accéder à un IDE riche et relativement complet pour envisager des projets conséquents de développement. Les extensions sont multiples et donnent accès à un large panel de fonctionnalités pour envisager par exemple le développement d'applications multiplateformes. ▲

L'ALM dans les nuages

Que ce soit dans le Cloud ou dans sa version On Premise, la nouvelle version de Team Foundation Server recèle de très nombreuses nouveautés, et jamais aucun éditeur n'était allé aussi loin dans l'intégration des différents services nécessaires au développement, au déploiement, et à l'exploitation des applications.

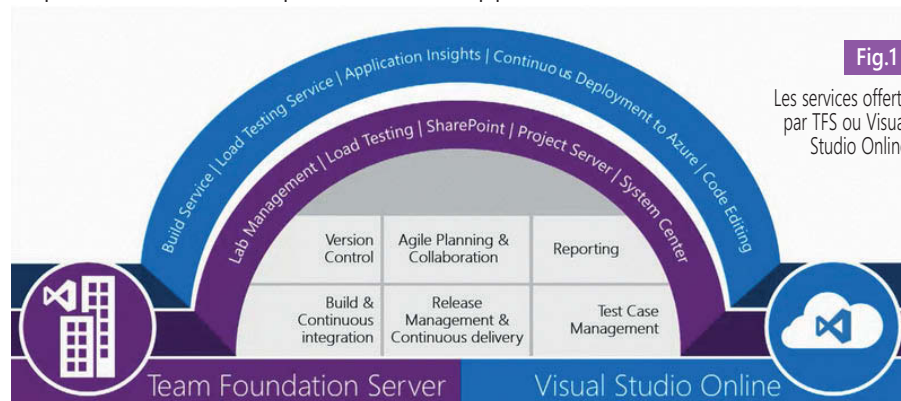


Fig.1

Les services offerts par TFS ou Visual Studio Online



Sylvain Gross
MVP sur Visual Studio ALM,
Sylvain est ALM /EPM Practice
Manager au sein de Neos-SDI.
Il anime de nombreuses
conférences sur l'ALM. Neos-SDI
est Centre de Compétences Gold

sur le domaine ALM, dans le
partenariat Microsoft Partner
Network.



Ces applications sont multi-plateformes, multi-devices, hébergées dans le Cloud ou en interne, mettant la démarche ALM et DevOps à la portée de toutes les entreprises : startups, PME, grands comptes ou institutions. Ce chapitre sera consacré aux services d'ALM (Application Lifecycle Management) offerts par Visual Studio Online, en fournissant une vue d'ensemble des outils à la disposition des services informatiques pour accomplir leurs différentes missions liées à la gestion des applications.

Team Foundation Server, la solution ALM de Microsoft

Team Foundation Server est la solution proposée par Microsoft pour supporter les processus de l'ALM. Il s'agit principalement de gérer :

- ▶ Les exigences de l'application, en permettant leur hiérarchisation sur plusieurs niveaux (fonctionnalités, user story...), et en fournissant des mécanismes de priorisation,
- ▶ Les tâches des membres de l'équipe, en tenant compte de leur capacité de travail, et de l'effort nécessaire pour réaliser ces tâches,
- ▶ Les codes sources de l'application, en intégrant les mécanismes de gestion des branches, de manière centralisée (avec TFVC) ou distribuée (avec GIT)

- ▶ Les builds de l'application, pour intégrer les sources d'une version spécifique, et leur faire suivre un processus de gestion (qualité, disponibilité pour le test...),
- ▶ Les différents types de tests tant techniques (unitaires, performance) que fonctionnels (tests manuels ou automatiques),
- ▶ Les déploiements des différents composants de l'application sur les serveurs de chaque environnement,
- ▶ La collecte d'informations de surveillance des applications en cours d'exécution.

Ces différents services sont disponibles à la fois sous forme de logiciels serveurs à installer sur l'infrastructure de l'entreprise (On Premise), et sous la forme de services hébergés dans le Cloud. (Fig. 1).

De l'ALM dans le Cloud ?

Visual Studio Online est le nom du service Cloud offert par Microsoft, pour supporter les processus ALM. Désormais, tous les services proposés par TFS sont également disponibles dans le Cloud via VSO. Bien entendu, ces services peuvent être disponibles sous une forme différente. Il est intéressant de noter que, très souvent, les fonctionnalités sont plus avancées sur Visual Studio Online que sur la version On Premise. L'architecture Cloud permet en effet à Microsoft de diffuser ses mises à jour plus rapidement dans le Cloud, que dans les versions On Premise.

Le choix de chaque entreprise entre la version Cloud ou On Premise se fera en fonction de critères stratégiques ou opérationnels. Un guide de choix est disponible sur le site de Visual Studio (1).

VISUAL STUDIO ONLINE, VOTRE SOLUTION ALM

Un portail pour votre projet d'équipe

Visual Studio Online permet de travailler sur des Projets d'Equipe. Ce sont des espaces qui contiennent les éléments nécessaires pour travailler sur un projet. Les projets d'équipe sont créés selon différents modèles, dépendant de la méthodologie utilisée : Scrum, Agile, CMMi, etc.

Le projet doit être découpé en itérations, afin de gérer la planification des livraisons des versions de projets, et en zones, permettant un découpage fonctionnel du projet. Les membres de l'équipe sont déclarés sur le portail Web dédié au projet, et les droits d'accès sont automatiquement attribués. La capacité de l'équipe est calculée en fonction de la durée de l'itération et du temps disponible pour chaque membre d'équipe sur ce projet. Le portail est configuré de manière à donner facilement accès aux éléments intéressants du projet : les tâches, les bugs, les builds, les releases, ainsi qu'à tous les artefacts du projet. (Fig. 2)

Les exigences et le travail

Dans Visual Studio Online, l'élément de base est le work item, ou élément de travail. Le nom des éléments de travail varie d'une méthodologie à l'autre. Tous les éléments de travail sont affectés à une personne, et ont un statut (ouvert, en cours, fermé...) qui peut évoluer au fil du temps, selon un workflow défini dans la méthodologie. Les principaux types d'éléments de travail sont :

- ▶ Les exigences, qui vont désigner des besoins, fonctionnels ou techniques. On y trouvera par exemple les Récits Utilisateurs. Ces exigences pourront être rattachées aux fonctionnalités, qui sont des exigences plus globales,
- ▶ Les tâches, qui désigneront les actions à réaliser. Ces tâches sont rattachées à une exigence, pour indiquer pourquoi cette tâche doit être réalisée,
- ▶ Les cas de test, qui décrivent la manière dont une exigence doit être testée : les étapes de test, les valeurs attendues.

Les tâches et les exigences peuvent être créées et éditées depuis le portail Web, soit à l'aide de listes, soit à l'aide de tableaux Kanban. Ce Kanban est d'une efficacité remarquable pour éditer rapidement les tâches, changer les affectations, définir le reste à faire. Sa simplicité d'utilisation permet aux équipes de mettre à jour le statut des tâches directement dans le portail, lors de la mêlée (Scrum) quotidienne. (Fig. 3)

Ce Kanban peut montrer les tâches

regroupées par personne, ou par récit utilisateur. De nombreuses possibilités de personnalisation permettent d'adapter le contenu, tant sur la forme (mise en forme conditionnelle, colonnes supplémentaires), que sur le fond (nouveaux champs).

Les builds

La gestion de build est l'une des activités fondamentales de l'ALM. Cette build permet d'assembler tout le code source d'un projet, pour en faire un package complet. Cette compilation est réalisée sur un serveur, et non sur un poste de développement, c'est pourquoi seule la build permet de garantir la maîtrise du code complet de l'application, tant au niveau du code source que des assemblées référencées. Désormais, un nouveau type de build est disponible dans Team Foundation Server 2015 et Visual Studio Online : Team Foundation Build 2015 (2). Ce nouveau système de build permet de créer des définitions de build pour Windows, iOS, Android, Java. Son utilisation est beaucoup plus simple que le système précédent basé sur XAML : une interface

Web permet de créer une nouvelle définition de build, intégrant :

- ▶ Des étapes personnalisées,
- ▶ Des tests,
- ▶ Des étapes de packaging,
- ▶ Des étapes de déploiement.

Aucune connaissance en programmation n'est nécessaire : il s'agit simplement de configurer les différentes étapes, en fournissant les éléments à chaque composant. L'exécution de la build est déclenchée par l'interface Web, et le cours de son exécution est affiché sur une console en temps réel. Le déclenchement d'une build peut être automatisé sur certains événements :

- ▶ À chaque archivage, pour mettre en place une intégration continue,
- ▶ Chaque nuit, pour disposer quotidiennement de la dernière version de l'application
- ▶ La build peut également conditionner l'acceptation d'un archivage : l'archivage contrôlé est un excellent moyen de s'assurer que le code archivé est correct, car en cas d'erreur de build, ou de test en échec, l'archivage est rejeté, permettant de ne pas pénaliser toute l'équipe. (Fig. 4)

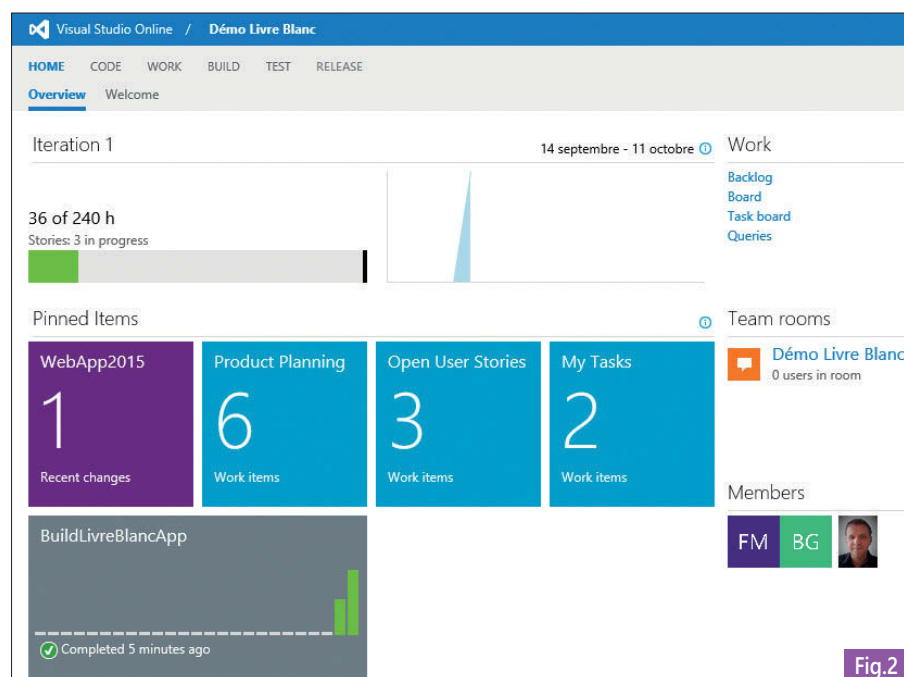


Fig.2

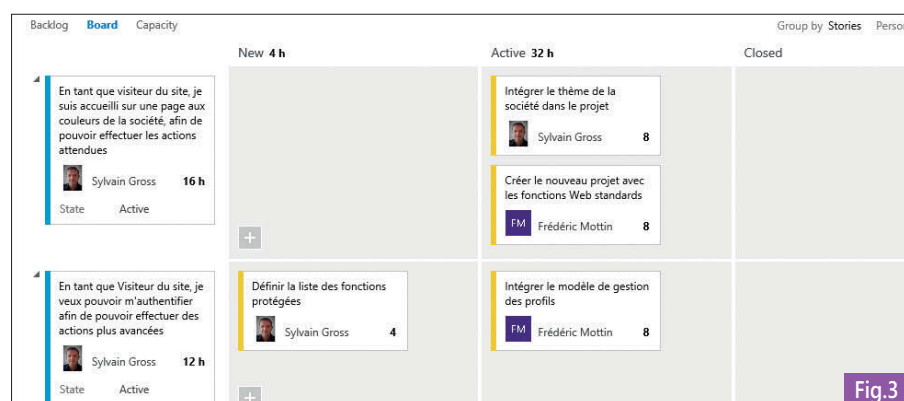


Fig.3

Le Kanban des tâches

Les tests

Au-delà des tests permis par Visual Studio, la plateforme ALM de Microsoft permet la mise en œuvre de tests :

- Fonctionnels : des cas de tests sont définis et paramétrés, puis exécutés depuis le portail Web,
- Techniques : des tests de charges sont définis dans Visual Studio, puis exécutés depuis le portail Web.

Les tests fonctionnels

Les tests fonctionnels permettent de s'assurer que l'application répond aux exigences fonctionnelles. Des cas de tests décrivent les différentes étapes d'un test, et précisent le résultat attendu à chaque étape. Des paramètres peuvent être fournis au cas de test, afin d'appliquer un jeu de données représentatif à l'application, en entrée, ou en sortie. Les tests peuvent être manuels ou automatiques. Les tests manuels peuvent être conçus et exécutés depuis le portail Web, dans le Hub Test. Les tests permettent de détecter des anomalies, qui peuvent être matérialisées sous forme de Bugs directement depuis l'outil. Le contexte de cette anomalie est automatiquement fourni, facilitant ainsi la reproduction du bug par les équipes de développement : la résolution sera facilitée. Les tests fonctionnels peuvent être automatisés : Visual Studio intègre des outils permettant d'enregistrer une séquence d'utilisation (tests unitaires, tests codés d'interface utilisateur), et de lier ce code de test à des cas de tests. De cette façon, les équipes QA peuvent exécuter les tests depuis leur outil Test Manager de manière automatique.

Les tests de charge

Les tests de charge permettent de garantir qu'une application continuera à fonctionner lorsqu'un nombre important d'utilisateurs est connecté. Pour mettre en œuvre ce type de test, il s'agit de définir des scénarios

d'utilisation sous forme de tests de performance Web, ou de tests unitaires, puis de programmer leur parallélisation. La montée en charge peut être progressive ou par palier, en intégrant des temps de chauffe. Les différents scénarios vont être exécutés successivement par chacun des utilisateurs virtuels, dans un ordre paramétrable, de façon à se rapprocher le plus possible de la réalité. Des indicateurs de performance sont positionnés, de manière à déterminer la consommation de chaque composant au niveau mémoire, processeur, espace disque, sur chacune des machines. Ce qui rend traditionnellement les tests de charge difficiles à mener, ce sont les ressources système qu'il faut monopoliser pendant les campagnes de tests. Ces tests requièrent une quantité importante de ressources, et sont très impactant au niveau de l'infrastructure en place. Le Cloud permet de soulager l'infrastructure, en mettant à disposition des équipes un environnement toujours disponible, aux capacités machines quasiment illimitées. L'exécution de campagne de tests sur l'environnement Visual Studio Online devient par conséquent très accessible. De plus, un mode simplifié permet de réaliser des tests de charge basiques, sans préciser de scénario.

Release Management as a Service

Le déploiement des applications sur les environnements de production est la dernière étape du processus de développement. Un package est généré par la Build : il s'agit de copier ces packages sur les serveurs cibles.

Release Management pour Visual Studio(3) est le composant de la suite Visual Studio ALM qui permet de répondre à cette problématique. Release Management permet :

- De modéliser les différents environnements, en y incluant les serveurs,

- De définir pour chacun de ces environnements les actions à réaliser pour déployer les composants de chaque machine de l'environnement (package IIS, script SQL...)
- De définir des processus de retour arrière en cas d'erreur : en cas d'échec des scripts de mise à jour de la base SQL, la base de données est restaurée dans son état précédent, puis l'application d'origine est remise en place,
- De définir un chemin permettant de passer d'environnement en environnement : de développement vers intégration, d'intégration à production par exemple avec possibilité d'ajouter des approbations.

Release Management pour Visual Studio permet par conséquent d'automatiser le déploiement des mises à jour de manière sécurisée, même dans des environnements complexes : plusieurs serveurs Web, des serveurs WCF, des serveurs SQL. Actuellement, Release Management se présente sous la forme d'une application WPF, liée au projet d'équipe à l'aide d'un serveur de release. Cette application permet de définir et d'exécuter le déploiement des applications. Une nouvelle version de Release Management est actuellement en phase de Preview : Release Management as a Service. Comme son nom l'indique, il s'agit d'offrir les mêmes services, mais à l'aide du portail projet de Visual Studio Online. Un nouveau Hub est disponible : Release. À l'instar de Build, il permet de définir une nouvelle séquence de déploiement, à l'aide de blocs disponibles depuis le Designer Web. (Fig. 5). Le déclencheur d'une séquence de déploiement est une Build. Une liste de tâches est disponible pour permettre de déployer les artefacts de la build vers les serveurs de destination. De nombreux composants sont disponibles, permettant d'exécuter des actions en parallèle, ou d'exécuter des scripts SQL ou Power Shell, de lancer des packages MSI, ou de lancer un test automatique de validation du déploiement.

Surveiller les applications avec Application Insights

Lorsque l'application est déployée, la responsabilité est transférée aux équipes d'exploitation, chargées de garantir le niveau de service requis : temps de réponse, absence d'erreurs... Pour faciliter ce travail de surveillance, et travailler de manière collaborative entre les équipes de développement et les équipes d'exploitation, Application Insights(4) propose un service de surveillance des applications :

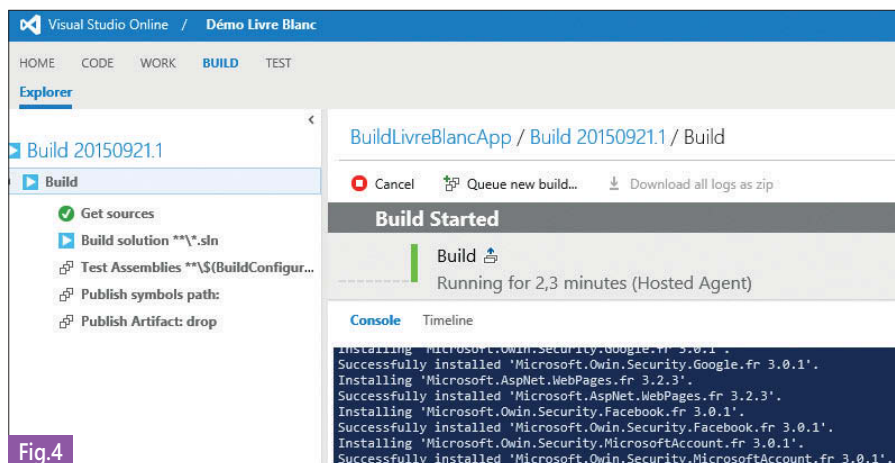


Fig.4

Exécution d'une build

- Détection de problèmes de performance,
- Détection d'erreurs et d'exceptions,
- Rapport d'exécution pour connaître les fonctions réellement utilisées.

La mise en œuvre d'Application Insights est très simple. Il suffit de créer une instance d'Application Insights dans Azure, et d'intégrer le package Application Insights dans le projet. L'exécution de l'application enverra automatiquement des données d'utilisation sur l'instance d'Application Insights. Ces données seront présentées sous forme de tableau de bord dans Azure, vous fournissant ainsi une donnée d'utilisation très précieuse. (Fig. 6)

Power BI

TFS fournit un grand nombre de données sur les différents projets en cours :

- Le nombre de bugs, de tâches, d'exigences,
- Le nombre de build, de tests, de déploiement,
- Le nombre d'archivages, etc...

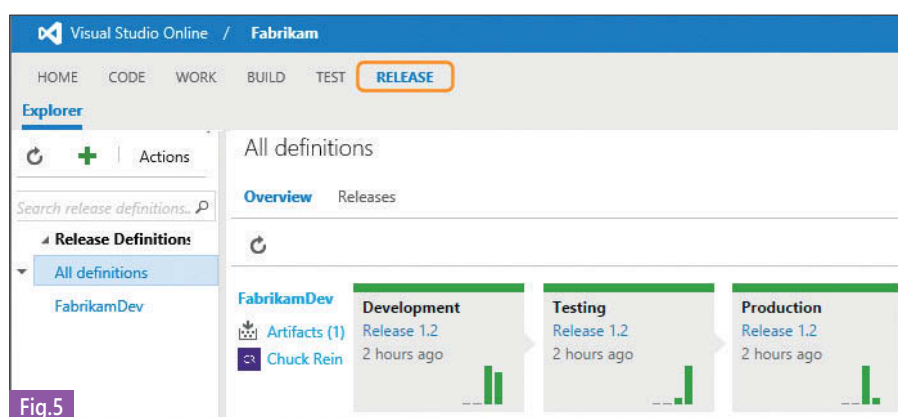
L'exploitation de ces données permet au management d'identifier des pistes d'amélioration de la performance des équipes, aussi bien sur la qualité que sur la productivité. C'est un moyen assez facile de mettre en œuvre une démarche d'amélioration continue. L'outil privilégié pour réaliser des rapports dans le Cloud est Power BI. Cet outil permet d'intégrer de nombreuses sources de données du Cloud, ou de l'infrastructure On Premise en utilisant des passerelles. Un connecteur Power BI pour Visual Studio Online est disponible, et permet de mettre à disposition des concepteurs de rapports un modèle de

données très riche, mis à jour en quasi-temps réel. Des tableaux de bord peuvent être réalisés, et mis à disposition du management, ou des équipes. Les outils de diffusion des rapports Power BI sont des applications Windows, des applications mobiles (iOS, Android, Windows Phone) ou le portail Office 365. (Fig. 7).

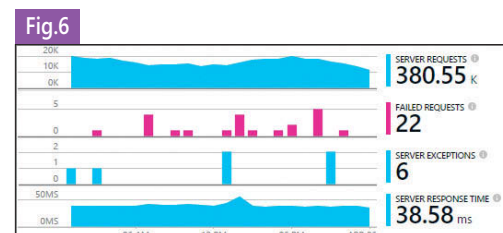
VISUAL STUDIO ONLINE, VOTRE ALM DÈS AUJOURD'HUI

Ce survol des différents modules de Visual Studio Online a montré sa capacité à couvrir tous les besoins ALM pour les applications d'aujourd'hui, et de demain.

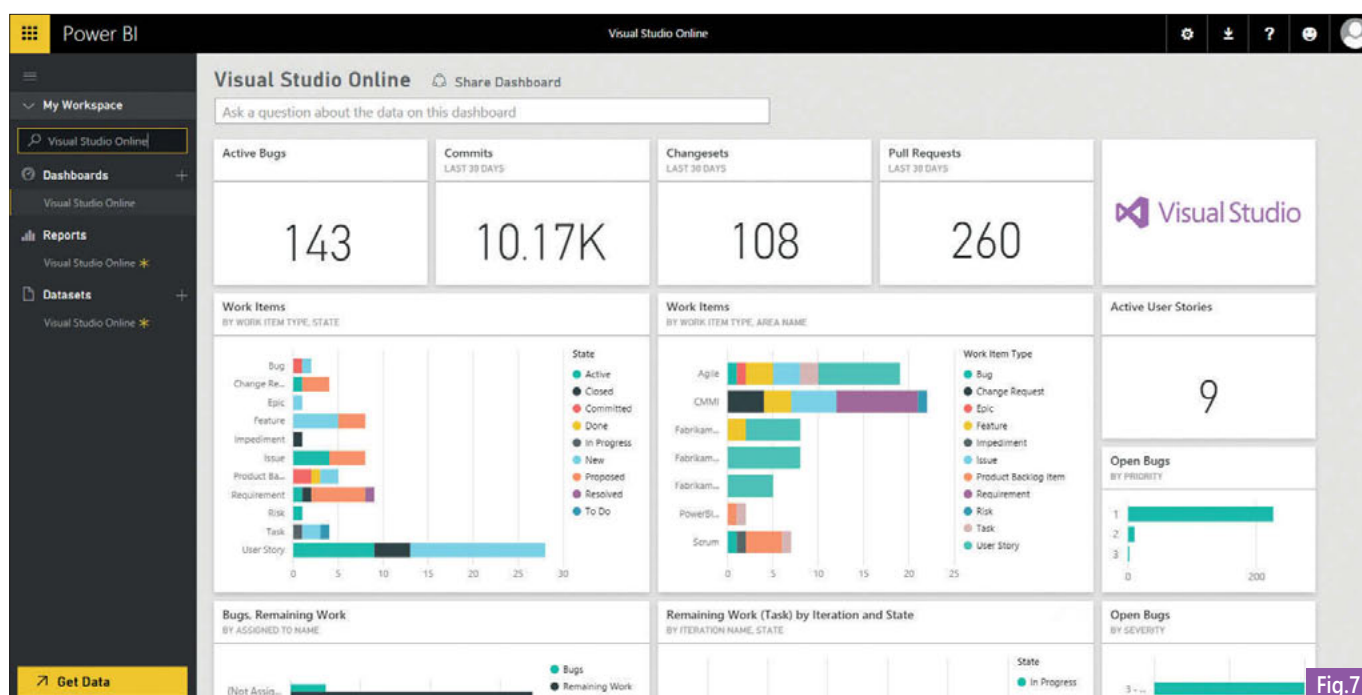
Avec cette version, c'est un nouveau palier qui a été franchi, ouvrant des perspectives enthousiasmantes : des applications multi-devices, connectées au Cloud, pour fournir rapidement des services innovants aux utilisateurs, avec un niveau de qualité irréprochable. L'adoption d'une démarche ALM est désormais accessible à toutes les formes de structures, et peut se faire de manière progressive, en activant les fonctionnalités au fur et à mesure, en fonction des priorités de chaque entreprise.



Le Hub Release Manager



Surveillance de l'application avec Application Insights



Exemple de rapport Power BI pour Visual Studio Online

Installation et migration

La sortie de Visual Studio 2015 a été un événement majeur de la communauté des développeurs .NET. Annoncée avec beaucoup de nouvelles fonctionnalités, je pense que cette nouvelle version était très attendue par nombre d'entre nous.



Gaël Dupire
Consultant Expert Technique
MERITIS



Avant de pouvoir tester ces nouveautés, la première étape est la réussite de l'installation du nouvel opus. Nous allons voir ensemble comment se passe cette procédure. Cela nous permettra aussi de parcourir quelques-unes de ses nouvelles fonctionnalités et d'évoquer les évolutions depuis les précédentes versions. J'ai choisi pour ma part d'installer la version Entreprise. On trouve une version d'essai pour 3 mois sur le site de Microsoft. Il est possible d'essayer d'autres versions ou encore de télécharger la version Community, qui est gratuite. J'ai également installé cette dernière version. La procédure est la même, seul le contenu change. L'installation de la version Entreprise nécessite :

- un processeur de 1.6 GHz (au minimum),
- 1 Gb de RAM (au minimum),
- 10 Gb d'espace disque (au minimum),
- une carte vidéo capable d'utiliser Direct-X 9 (au minimum),
- Windows 8.1 ou Windows Server 2012. (Fig. 1)

L'installation se fait en ligne. On dispose également d'une option pour télécharger un iso. Au démarrage du processus, on arrive sur un premier choix permettant de déterminer le type d'installation souhaitée. L'installation standard dispose des packages nécessaires pour démarrer et nécessite 958 Mo. Les packages non installés pourront être téléchargés par la suite si on souhaite les utiliser. L'installation avancée permettra, quant à elle, de sélectionner les parties à installer. Cela permettra soit de gagner de la place sur l'installation, soit de s'assurer de démarrer avec les bons outils. Microsoft essaie de nous donner, dans ce nouvel opus de Visual Studio, un maximum de polyvalence. Il s'agit donc de cibler au mieux l'environnement sans alourdir le logiciel. Si, à

terme, des packages venaient à manquer, il suffirait de les installer au moment voulu. Si vous avez installé une version bêta du SDK Windows 10, l'installateur vous demandera automatiquement de le supprimer. Il s'agit d'un prérequis. Parmi les packages proposés, on retrouvera les outils habituels, ainsi qu'un certain nombre de nouveautés « embarquables » nativement dans l'EDI. On trouvera notamment : (Fig. 2)

- des outils Python,
- les outils de développement pour application universelle Windows,
- le SDK pour Windows 10,
- des outils de développement cross-plateformes (Xamarin, développement pour Android et iOS),
- Apache Cordova,
- Node.js,
- GIT.

L'installation de l'ensemble des packages requiert jusqu'à 26 Go. J'ai personnellement lancé tous les packages, mais certains n'ont pu être installés. En fin de processus, un bilan nous liste clairement les problèmes survenus afin de pouvoir les corriger.

Parmi les outils installés, on trouvera notamment la nouvelle version de Blend. Il s'agit de l'outil de création d'interfaces graphiques XAML / WPF. (Fig. 3) Une fois l'installation terminée, il y a deux principaux réglages à effectuer. Le premier est classique puisqu'il s'agit de paramétrer son environnement Visual Studio (thème et langage de démarrage). Concernant le thème, de nombreux autres réglages sont disponibles (menu Tools puis Options). On pourra notamment modifier le thème choisi au démarrage si ce dernier ne nous convenait plus. Le second paramétrage est, quant à lui, plus novateur. Il s'agit de s'enregistrer avec un compte Microsoft. Il permet plusieurs fonctionnalités. Lorsque nous démarrons Visual Studio pour la première fois, on remarque que la licence est accordée pour 1 mois (contre 3 mois annoncés sur le site de Microsoft). Pour débloquer les 3 mois d'essai, il faut se connecter à un compte Microsoft dans Visual Studio. (Fig. 4 et 5).

(Fig. 6) L'un des intérêts est aussi de partager les paramètres entre les différents outils Visual Studio (vers Blend par exemple) et à travers les différents PC utilisés. Cet enregistrement lie le compte utilisé à un compte Visual Studio. Cela nous

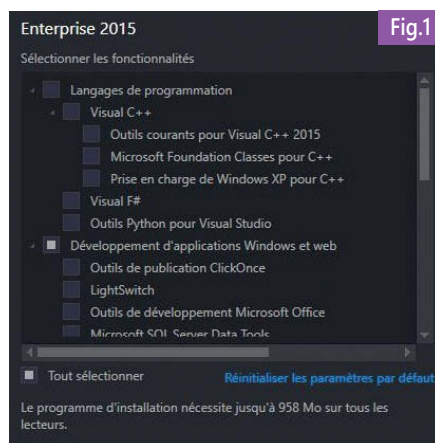


Fig.1



Fig.3

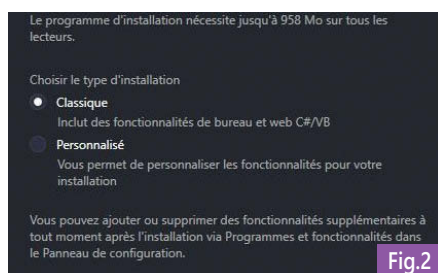
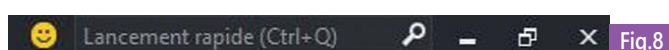
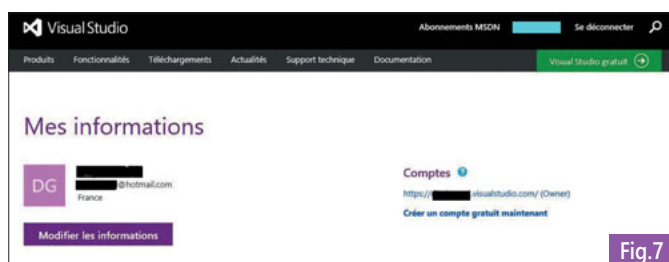
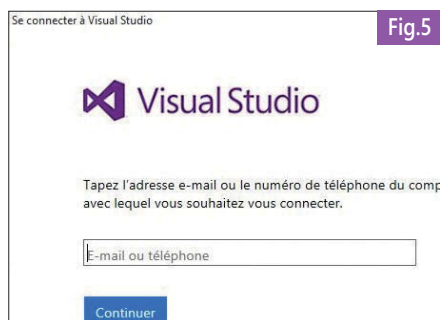


Fig.2



Fig.4

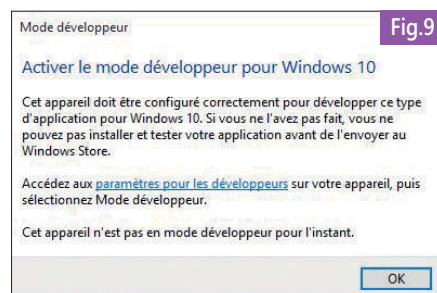
permet d'accéder à ce compte via Team Explorer. De la même façon, les ressources Azure et sites Web Azure seront visibles dans l'explorateur de serveur. Par ailleurs, il est possible d'enregistrer plusieurs comptes afin de gérer les différents aspects de nos projets. On pourra bien entendu ajouter de nouveaux comptes après l'installation. Le changement de compte est possible à n'importe quel moment souhaité. La liste des comptes associés à un utilisateur fait partie de l'itinérance des paramètres. (Fig. 7). Une fois le profil paramétré et sélectionné, on voit apparaître dans le coin en haut à droite de l'EDI un badge avec nos initiales, et à côté, nos nom et prénom. Juste au-dessus, un petit smiley jaune nous permet de communiquer avec les équipes de Microsoft et de leur faire des retours sur les différents outils. (Fig. 8). En cliquant sur cette icône puis en choisissant « Settings », on pourra s'enregistrer pour le programme d'amélioration du produit. Avec « More Options » ou « Help », on ouvrira une page Internet dans Visual Studio qui nous redirige vers une autre page permettant de faire des



feed-back ou d'obtenir des informations sur comment faire ces feed-back. Concernant la compatibilité, mon ordinateur accueille Visual Studio 2010, 2012, 2015 Community et 2015 Entreprise. J'avais un certain nombre de projets en cours ou terminés correspondant à chacune de ces versions. Sans aucun effort particulier, j'ai pu ouvrir différents projets de bibliothèque en C#, C++ et F# ayant été créés sous Visual Studio 2010 (C# et C++) et 2012 (C#, C++ et F#). Nous verrons le détail de ces portages par la suite. Concernant les projets de bibliothèque C# et F#, l'ouverture, la compilation et l'exécution se sont déroulées sans aucun problème et sans adaptation.

La migration

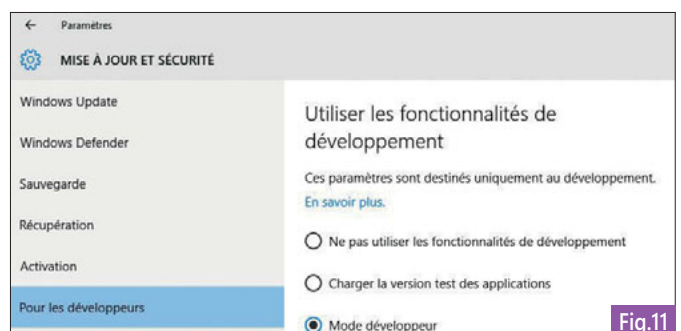
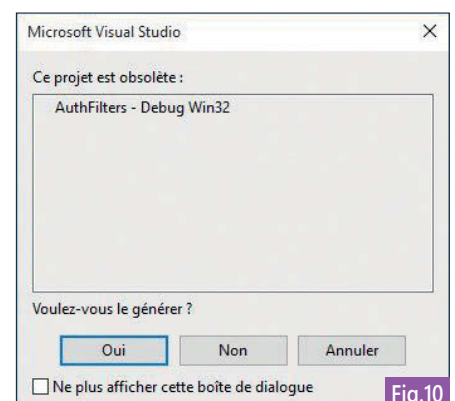
Pour tester le portage de projets de développement ciblant Windows 8.1, j'ai utilisé deux projets disponibles en téléchargement sur MSDN. Le premier projet est la démonstration des capacités d'authentification Web : « Web authentication broker sample » disponible sur MSDN. Ce projet est particulièrement intéressant, car il contient des objets graphiques (XAML) du code C# et C++. Le second projet est une démonstration d'utilisation de la bibliothèque Prism, il s'agit de « Developing a Windows store business app using C# and Prism with MVVM ». Ce projet est également très intéressant, car il utilise NuGet. Le choix d'utiliser des projets dont la source est MSDN vient du fait que les projets ciblant Windows 8.1 ne sont ni créables ni ouvrables avec une version inférieure à Visual Studio 2013. Mes projets issus de cette plateforme étant sur une machine virtuelle, j'ai opté pour cette solution. Cette dernière est bien adaptée à des projets de démonstration particulièrement complets. (Fig. 9 et 10).



J'ai commencé par le projet de Web authentification : l'ouverture s'est bien passée, mais on m'a demandé d'activer le mode développeur disponible dans Windows 10 à la compilation (il s'agit de la version de Windows que j'utilise pour ces tests). Cette requête est faite via une petite fenêtre qui nous demande de modifier les paramètres. Un lien nous permet d'effectuer la modification demandée. Concrètement, ce lien ouvre les paramètres de Windows 10, il ne nous reste donc qu'à sélectionner la bonne option. (Fig. 11).

Un autre événement s'est produit durant cette compilation : un élément du projet était obsolète. De la même façon, une fenêtre nous avertit du problème. Comme il est possible de le générer immédiatement, j'ai validé cette option pour voir où cela conduisait : le projet a été converti et la compilation s'est finie sans autre difficulté. Pour le second projet, aucun événement notable n'est à signaler. Le chargement a été particulièrement long, cela me laisse penser que même si aucune fenêtre de dialogue ne s'est manifestée, il y a certainement des actions de conversion qui se sont déroulées silencieusement. (Fig. 12).

Ceci m'amène au portage des projets C++. Si nous venons de projets issus de Visual Studio 2010 ou 2012, après le chargement de la solution, il nous est proposé une étape de conversion. Cette étape n'est pas obligatoire, mais requiert très peu d'action (cliquer sur OK uniquement!!). Après le chargement du projet, une boîte de dialogue nous demande si l'on souhaite effectuer la mise à niveau. À la fin de cette action, un résumé des actions est



disponible. Finalement, même si quelques incompatibilités ont été relevées dans les projets utilisés, elles ont toutes été résolues sans problème majeur avec des outils embarqués. En tant qu'utilisateur je n'ai eu que très peu d'interaction avec Visual Studio pour permettre l'adaptation de ces codes. Il semble donc que la migration soit en grande partie automatisée et demande peu de temps. Toujours concernant le portage des projets, MSDN a bien documenté ce sujet (rechercher le thème « migration de projet C# avec Visual Studio 2015 » avec Bing). L'article détaille de nombreux cas possibles. C'est dans cet article que j'ai appris que l'étape d'adaptation des projets C++ est optionnelle pour les projets issus de Visual Studio 2010 et 2012 (.vcxproj), mais nécessaire dans les autres cas (.vcproj). On trouvera également des conseils dans le cas où les outils embarqués ne fonctionnent pas. Même si cette situation ne s'est pas présentée dans mes tests, il est utile de savoir où trouver ce type de conseil. Des explications sont données dans les cas où les projets :

- ▶ ne s'exécutent pas, car des fichiers sont introuvables ;
 - ▶ ne peuvent être générés, car ils contiennent des références incompatibles.
- Dans les deux cas, il s'agira d'ouvrir le fichier du projet dans un éditeur de texte et de trouver les références à des versions de Visual Studio ou au framework (réciproquement) puis de corriger à la main. Au travers de mes différentes expériences professionnelles, j'ai eu l'occasion d'utiliser les versions 2005 et 2008 : le chemin parcouru depuis ces versions est immense. Bien entendu, un utilisateur venant directement d'une de ces deux dernières versions (ou même de la version 2010) pourra se sentir un peu perdu par le design ou l'organisation. Cependant, on est clairement dans la continuité des versions précédentes. Finalement, on peut dire que cette version vise l'universalité et la polyvalence dans son contenu tout en facilitant la mobilité des développeurs et l'accessibilité de leurs projets. L'aspect communautaire est aussi très fort. En empruntant cette voie, je pense que Visual Studio continuera de s'améliorer. ▲

Services Paas, IaaS sur DevTest, ouverture d'un compte VSO, crédit Azure dans MSDN

Nous allons voir comment Visual Studio 2015 porte les principes DevOps dans le Cloud Microsoft Azure, le monde de l'Open Source ou les solutions d'éditeurs tiers. Le but est d'accélérer le « **Dev-Lead Time** », un KPI (Key Performance Indicator) très prisé des directeurs de projets, puisqu'il mesure le temps requis pour traverser un cycle complet de développement, de l'idée jusqu'à la mise en production. Les questionnements relatifs aux licences et économies à considérer seront évoqués.



Vincent Thavonekham
Microsoft Practice Manager VISEO

Notions de base et définitions

Avant d'aborder pleinement le vaste et passionnant sujet du DevOps, rappelons quelques définitions simplifiées :

- ▶ L'« **ALM** » (Application Lifecycle Management) correspond à un ensemble de processus et d'outils permettant de gérer le cycle de vie complet de création d'un logiciel. Par cycle de vie complet, nous entendons les étapes successives de la collecte des besoins, les spécifications, la réalisation, les tests unitaires, la recette jusqu'à la livraison en production et la gestion des bugs. L'ALM permet d'extraire de nombreux indicateurs et KPI afin d'optimiser les processus et la qualité globale.
- ▶ Une « **Usine logicielle** » représente la partie outillage et serveur mise en œuvre dans l'ALM pour automatiser les tâches répétitives et chronophages du processus (ex : compilations, exécution de tests, édition de rapports, déploiements, etc.).
- ▶ Le « **DevTest** » est une approche agile permettant aux développeurs d'accélérer leurs développements en testant et livrant automatiquement leur code sur un environnement d'intégration plusieurs fois par jour,
- ▶ Le « **DevOps** » est une approche Agile qui repose sur le « DevTest » et pousse le principe plus loin, en incluant dans la boucle tous les acteurs jusqu'à la mise en service et support du produit. Cette livraison en production auprès du service d'Exploitation (« Operations » en anglais, d'où l'« Ops » de DevOps) suit un workflow très rigoureux de validations et inclut une forte traçabilité ; le DevOps est parfois assimilé, à tort, à l'ALM.

Le « DevOps » représente le « Graal » de toute équipe souhaitant accélérer le « Dev-Lead Time » tout en garantissant un niveau de qualité de leur produit.

VSO et Microsoft Azure comme accélérateurs de développement

Pour réduire de façon drastique le « **Dev-Lead Time** », nous allons d'abord nous concentrer sur Visual Studio Online (VSO), c.-à-d. un Team Foundation Server toujours à jour et disponible « en tant que service » sur Microsoft Azure. Nous verrons également comment l'interaction de VSO avec les autres « services » d'Azure permet d'optimiser ce précieux KPI. Par le passé, les habitués des projets de grande ampleur prévoyaient une semaine à un mois pour configurer et roder une usine logicielle « from scratch ». À présent, à périmètre constant, il est fréquent de réaliser tout cela en moins d'une semaine. Mieux, avec de l'entraînement et sur un projet de taille modeste, il est possible de réaliser, toujours « from scratch », une usine logicielle de bout-en-bout en **une heure top-chrono**, documentation comprise :

- ▶ fabriquer un environnement de Dev (virtualisé),
- ▶ créer un projet vierge, avec une coquille de Tests Unitaires associée (TU),
- ▶ monter un serveur de Build, avec les outils pour lancer les TU, et calculer le taux de couverture de code par les tests,
- ▶ monter VSO-GIT (un référentiel GIT combiné à la puissance de VSO pour gérer les projets),
- ▶ créer un hébergement pour une application Web avec sa base de données,
- ▶ brancher les outils de monitoring applicatifs et techniques.

Oui, tout cela en une heure pour des cas

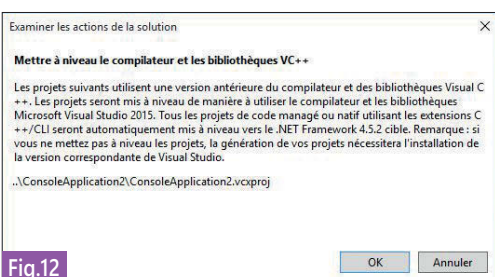


Fig.12

simples, ou 3 heures pour des projets de type « BI ». Le secret consiste simplement à puiser ces briques dans Azure. Ensuite, il peut être intéressant, par exemple d'ajouter du reporting sur l'état d'avancement des développements au travers de la brique Power-BI (V2). Ce dernier pointera sur le cube OLAP de VSO.

Le schéma ci-contre survole les services d'Azure, où chaque icône encapsule plusieurs fonctionnalités ; ce qui constitue la richesse du Cloud Microsoft : (Fig. 1).

Ce foisonnement de fonctionnalités se traduit par un bon positionnement dans les classements GARTNER. En effet, Azure est positionné comme **unique Cloud** à figurer comme Leader dans les différentes analyses (respectivement IaaS, stockage, PaaS, virtualisation, base de données en tant que service) : (Fig. 2).

Entrer dans le détail de chaque service ci-dessus dépasserait le cadre de ce document.

Dans un contexte DevOps, une offre très couramment utilisée est Azure App Service, puisqu'elle permet de créer rapidement des applications Web et mobiles sur différentes plateformes en mode « DevTest », mais dispose également de mécanismes robustes pour déployer **en production** et assurer simplement un SLA élevé. C'est l'objet du paragraphe suivant.

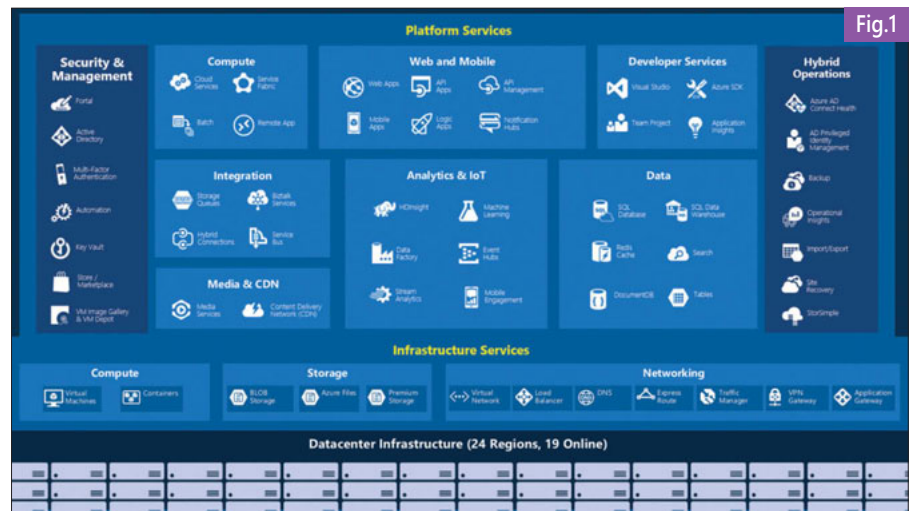


Fig.1

Des services Azure simples pour assurer le DevTest et DevOps : Azure App Service

Historiquement Azure App Service était composé de plusieurs briques séparées ; Maintenant ces différents éléments sont rassemblés afin de pouvoir réaliser un ensemble cohérent de services fédéré autour d'un « **Hosting Plan** ». Pour le DevTest, on peut utiliser les offres gratuites ou peu chères d'Azure App Service. Si l'on souhaite aboutir à un SLA global élevé avec des backups/restores rapides, une simple question d'ajustement de curseur, impliquant un coût supplémentaire, permet d'aboutir à

une version robuste taillée pour la production. Créer de tels ensembles peut se faire sous forme de « templates ». Cette simplification permet d'accélérer davantage la « vélocité » d'une équipe puisqu'elle peut se concentrer sur la « valeur métier » à produire, et non sur la tuyauterie « PaaS » qui est prise en charge par Microsoft. D'ailleurs, la communauté Open Source met à disposition de nombreux « Azure Resource Group Templates ».

(Fig. 3). Un simple clic dans GitHub ouvre le portail Azure et instancie ce template : (Fig. 4). Ensuite, passer du mode « Dev » au mode « Production » est facilité et sans surprise, puisque le template a été utilisé à de nombreuses reprises en Dev et dans l'usine

- Les Applications Web ou « Azure Web Apps »,
- Les Mobile Apps,
- Les API Apps,
- Les Logic Apps.

Le **premier** permet de déployer du code sans se soucier de la plateforme, fournie et gérée par Microsoft, d'où le terme de « PaaS » (Platform as a Service) pour vos

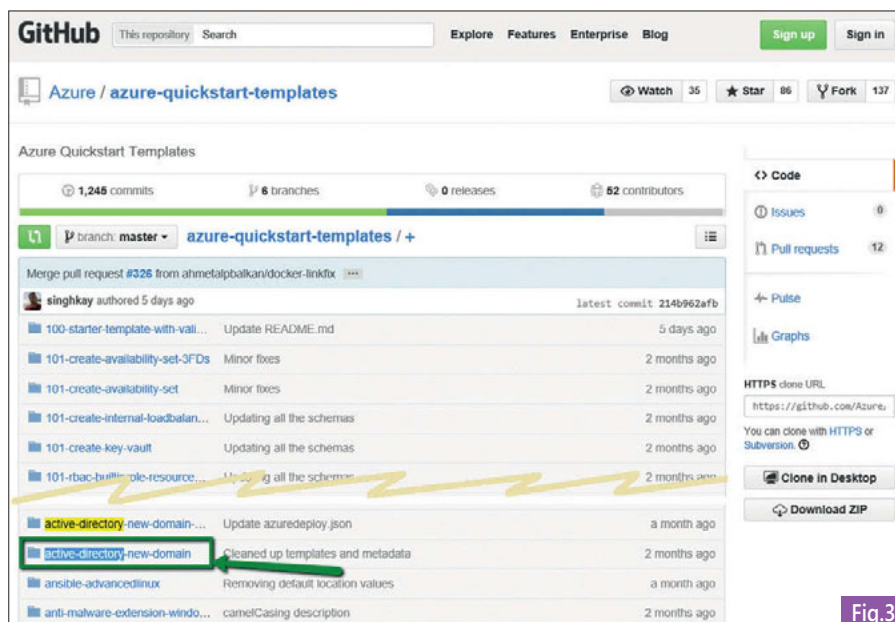


Fig.3

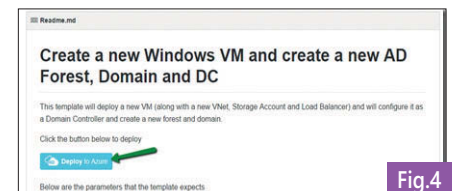


Fig.4

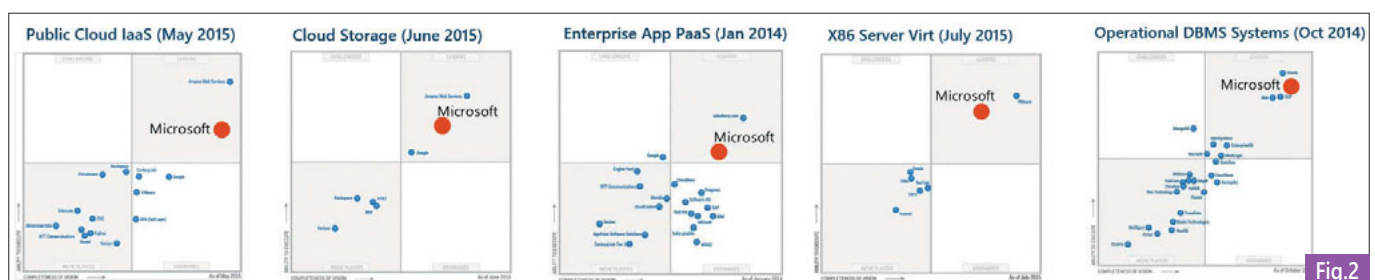


Fig.2

applications Web. Aussi, est-il possible de réellement coder en PHP, en Java, en .Net, avec AngularJS,... et de déployer le code selon le moyen privilégié (Visual Studio Online-GIT, Github, Dropbox, FTP, upload de fichier...). Il ne reste plus qu'à choisir ses options en quelques clics : haute disponibilité « Always On », « Backup WebApp + Base de données », version de PHP, Web Socket ou non, ... (Fig. 5).

Le second service concerne les **Applications Mobiles.** On pourrait se demander la légitimité de cette brique, mais à y regarder de plus près, des problématiques récurrentes de coder des CRUD dans les Web API, de « Push Notifications » sur de l'iOS, mais aussi sur de l'Android, ou encore en natif sur Windows Phone sont prises en charge, et la complexité est encapsulée.

Le troisième service répond à de nombreuses problématiques liés à la mise en production de **REST API**, comme les aspects sécurité, la mise à l'échelle des REST API, le monitoring, la documentation, où le bon « vieux » SOAP et son WSDL étaient bien pratiques. Ici, les API seront protégées par un proxy et authentifiées au travers d'un mécanisme de Token-based-SSO, reposant par exemple sur Azure Active Directory, comme Identity Provider.

Le quatrième et dernier service propose des

briques de logiques « visuelles » toutes faites, qui permettraient, par exemple, de se connecter à Salesforce et de réaliser des traitements selon des workflows spécifiques. (Fig. 6).

Salesforce n'est qu'un exemple d'ouverture de Microsoft. Les problématiques B2B (Business to Business) et EAI (Enterprise Application Integrator) sont simplifiées, qu'elles soient « Cloud » comme Salesforce, ou « On Premise » afin de laisser vos données dans votre SI. Pour ce faire, des connecteurs fournis en standards sont disponibles, dont voici un extrait : (Fig. 7).

Bien débuter avec les crédits Azure

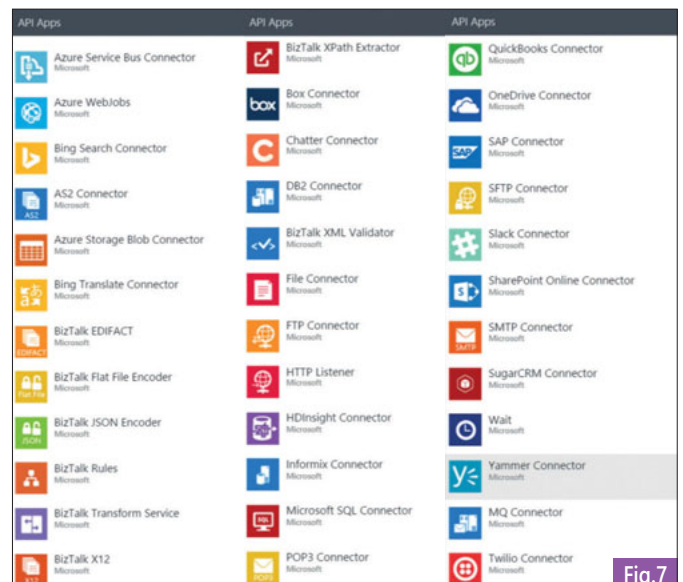
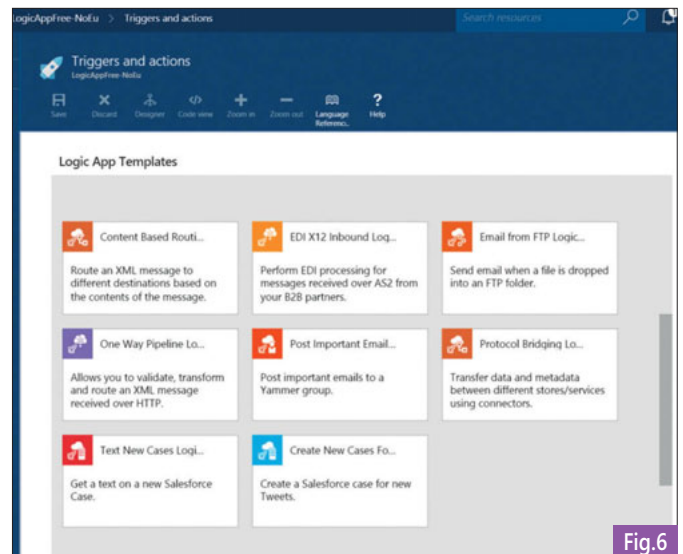
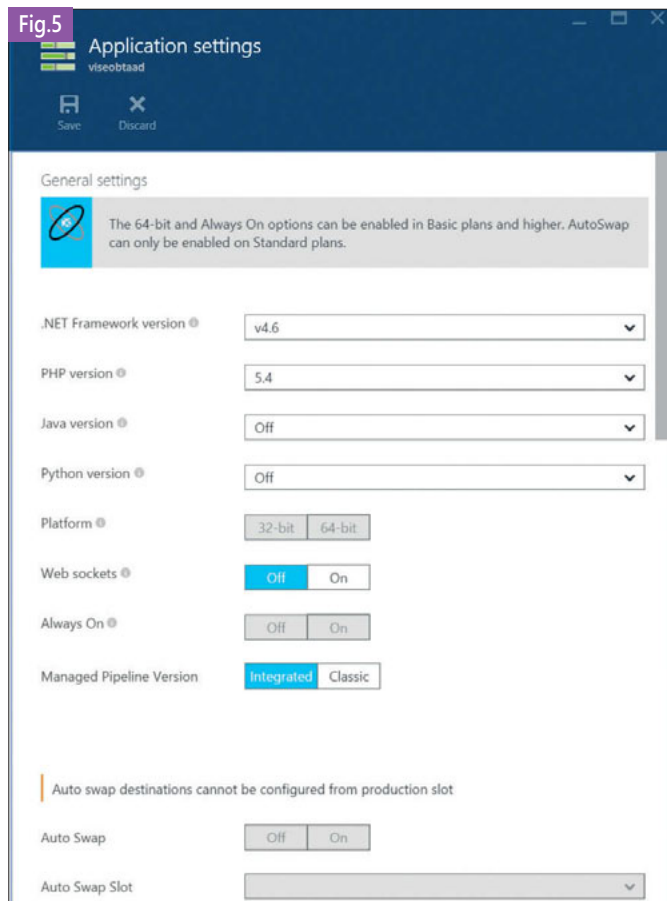
L'utilisation de la plateforme Azure s'apparente à la consommation d'un téléphone portable : le fournisseur propose un crédit à utiliser et le décompte se fait à l'usage. Du coup, les économies sont immédiates : une ressource n'est facturée que si elle est utilisée, contrairement à une ferme de serveurs physiques ; qu'ils soient ou non utilisés, ces derniers sont ou seront à payer. Il existe plusieurs types de crédits Azure :

► Version d'essai gratuit 30 jours :

Microsoft vous met à disposition un crédit de **170 €** dans ce laps de temps. En choisissant des services gratuits, l'enveloppe des 170 € sera encore moins impactée. Aussi, allez — vous pouvoir tester des environnements complets avec des scénarios de bout en bout,

► Souscription MSDN :

Contient **11 To** de logiciels Microsoft auxquels s'ajoute notamment un accès à la plateforme Azure. Vous disposez de 45 €, 85 € ou 130 € de crédits selon les abonnements, respectivement VS Professionnel avec MSDN, MSDN Platforms et VS Entreprise avec MSDN. Dédié aux développeurs, MSDN dispose également d'un autre avantage : le coût des ressources Azure est proposé à un tarif préférentiel. Par exemple, avec 130 € il sera possible de consommer plus d'Azure qu'avec un abonnement non MSDN, permettant ainsi de réaliser de nombreux tests et démos clients (il est interdit d'utiliser les licences en production),



- **Paiement « Pay-as-you-go »** : en tant que particulier ou entreprise, ce moyen est pratique à court terme pour dépanner et utiliser rapidement Azure. Il suffit de saisir son N° de carte bleue, et pour éviter les désagréments de date d'expiration de cartes CB, il est aussi possible de choisir des prélèvements automatiques,
- **Pour les Start-ups** : de 4 175 € à 49 000 € de ressources offertes pour aider les start-ups,
- **Contrats d'entreprise « en volume »** : Pour bénéficier de tarifs préférentiels par rapport au tarif public, il est requis de réaliser une projection de ce que l'on va payer en 3 ans, avec des « resynchronisations des montres » tous les ans. Cela permet de réaliser des économies, de bénéficier d'avantages Microsoft, d'un portail Web dédié pour administrer Azure à l'échelle d'un grand groupe, d'un support de proximité et de tarifs préférentiels et contractuellement garantis, et enfin d'accéder à différents documents techniques ou juridiques concernant la plateforme Azure.

En comprenant le modèle de licences et les avantages du Cloud, des économies considérables peuvent être réalisées grâce au Cloud Azure.

Prenons le cas réel d'une société qui paye un demi-million d'euros en licences hors coût des plateformes virtualisées. Afin de fiabiliser les développements, une plateforme de DevTest (et à terme DevOps) est envisagée.

Or, cela implique la création à la volée de nombreux environnements éphémères de tests, d'intégration et de recette. Par conséquent, certaines licences, comme celles des bases de données ORACLE entraîneraient des coûts astronomiques, du fait des contraintes actuelles de licensing. Cependant, puisqu'Azure repose sur un mode de « paiement à l'usage », et puisqu'ORACLE a réalisé des accords avec Microsoft, la solution la plus rapide et la plus économique consiste à monter tout cela sur Azure, si le projet le permet : le coût des licences est alors directement intégré (si on le souhaite) à la location des VM éphémères. Ainsi, dans la limite d'un quota fixé, et depuis Visual Studio, les développeurs sont à même de déployer et tester rapidement leur code, sans passer par des lourdeurs de demandes de provisionning de VM.

Focus sur VSO : comment bien débuter ?

Visual Studio Online (VSO) fait partie des nombreux services fournis par la plateforme Azure. (Fig. 8).

Il est possible de s'authentifier de façon sécurisée, soit via un compte « Microsoft Account » (ex. « Live ID »), soit via un « Corporate Account » c.-à-d. un compte Active Directory d'entreprise (Azure AD), pour peu que vous utilisiez le fournisseur d'identité Azure Active Directory ou celui d'Office 365 (qui repose en fait sur Azure AD). Considérons la société Contoso. Son

compte VSO serait accessible depuis l'URL contoso.visualstudio.com ; un utilisateur s'y connecterait via SSO avec votre Login/mot de passe d'entreprise habituel, comme Vincent.Thavonekham@contoso.com

(Intérêt d'Azure AD vs Microsoft Account : par exemple, si un salarié quitte l'entreprise ou si une politique d'expiration de mot de passe est active sur votre AD, la propagation sera également effective sur VSO, via Azure AD).

Bien débuter avec le DevTest : simplicité et encapsulation

Le portail d'administration Azure comporte à ce jour 29 rôles (Owner, Contributor, Reader,...) ; cela permet, par exemple, au service marketing de n'accéder qu'en lecture seule à une sous-partie du projet Azure, sans avoir les autorisations de détruire des VMs, ni de visualiser les rapports d'audit techniques (Application Insights). En outre, cela permet de séparer les coûts et de ventiler les budgets. (Fig. 9).

Les développeurs vont pouvoir étendre leur environnement de développement dans le Cloud en quelques clics, et parfois même entièrement depuis Visual Studio. En effet, Microsoft a rendu simple l'usage d'Azure (avec ou sans Visual Studio). Prenons le cas classique de création d'une application Web ; rappelons que Visual Studio peut ajouter les bons exécutables et configurer le projet afin d'en réaliser un monitoring précis via « **Application Insights** ». Pour cela, il suffit de préciser le login adéquat vers Azure. (Fig. 10).

Cette simplicité s'applique également aux ressources IaaS, qui sont pourtant intrinsèquement plus complexes à utiliser que des services PaaS. En effet, Microsoft a encapsulé ces notions afin que le développeur puisse se concentrer sur son travail et non sur les subtilités du Cloud. D'autant plus que, parfois, ces notions sont peu maîtrisées. L'image (Fig. 11) illustre que le développeur peut déployer son code sur Azure, que ce soit sur une Web App (PaaS)

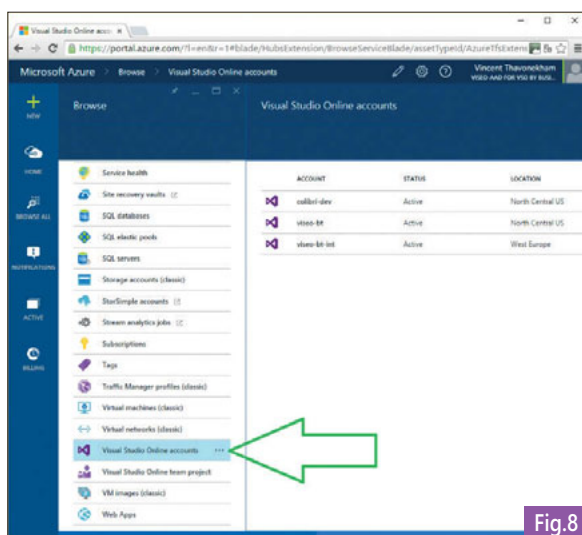


Fig.8

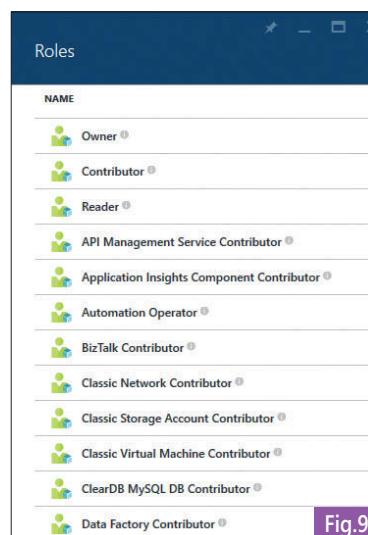


Fig.9

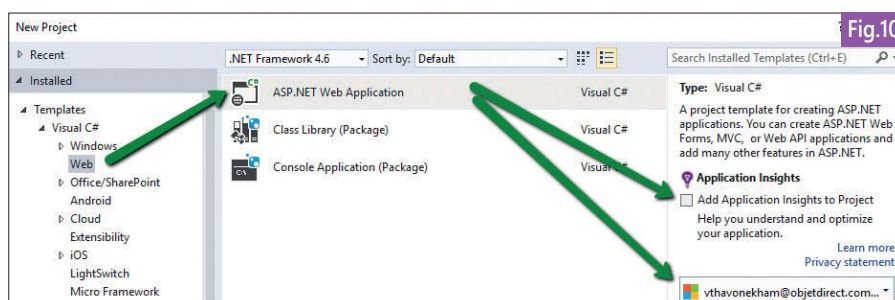


Fig.10

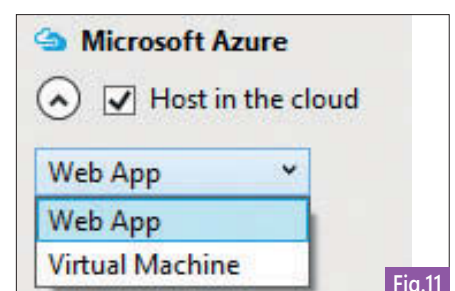


Fig.11

ou une Virtual Machine (IaaS).

Cette encapsulation de complexité est également applicable à VSO, ce qui économise des heures, voire des jours d'installations et de configurations de machines de Builds, de réinitialiser des IIS, gérer des notifications ... (Fig. 12). Pour cela, pas besoin de quitter Visual Studio, l'administration basique d'Azure est directement intégrée dans la « View » nommée « Server Explorer ».

Pour implémenter le DevTest ou DevOps, quel composant cloud choisir ?

Les limites entre IaaS et PaaS sont parfois floues ; en conséquence, Amazon, Google et Microsoft n'ont pas exactement la même implémentation.

De plus, même au sein de l'offre Microsoft, il sera possible de trouver plusieurs moyens d'héberger son application en PaaS : Azure Web App et les Web Roles/Worker roles. Tout va dépendre des performances, du SLA, des coûts souhaités.

Alors comment s'y retrouver, surtout avec le nombre considérable de services sur la plateforme Azure ? Sans entrer dans le détail de tous les composants (plus de 500 nouveautés par an), voici les recommandations générales :

D'une part, il est nécessaire de maintenir son niveau de connaissance en Cloud, qu'il soit Azure ou non, en pratiquant les concepts reposant sur les principes d'**architectures distribuées**. Aussi, lors de la réalisation d'applications « business critical », différents types d'architectures comme les « lambda architectures », le CQRS ... ou les concepts clefs comme l'idempotence ou le « retrial policy » seront évalués. D'autre part, il est recommandé de connaître le classement suivant : IaaS – PaaS – SaaS.

► Plus le curseur tend vers la droite en direction du SaaS, plus les services sont simples à utiliser, avec de nombreuses options gratuites ou payantes incluses.

Cependant, vous perdrez en « liberté », car ces services ont déjà été développés et testés pour vous,

► Plus le curseur est orienté vers la gauche en direction du IaaS, plus le degré de liberté à configurer sera important, mais plus le temps de maintenance sera long, ce qui représentera un coût plus important en production. Par ailleurs, il est connu que le coût réel d'une application n'est pas localisé dans la phase de développement, mais dans la phase de MCO « Maintien en Condition Opérationnelle », puisque ce dernier dure en principe des années.

Pour concevoir et réaliser des applications robustes, suivons les recommandations suivantes : « **Privilégier au maximum les services PaaS pour économiser et simplifier le Dev et la Prod, et choisir le IaaS si et seulement si son équivalent PaaS ne propose pas assez de personnalisations** ». Ainsi nombre d'applications sont actuellement en production avec une architecture idéale en « Always On » induisant un haut SLA, sans toutefois avoir passé plusieurs mois à « réinventer la roue ». Oui, il restera des scripts d'automatisation à réaliser, complétés par les fonctionnalités de « auto-healing » du Cloud : à la manière d'un satellite en orbite, lorsqu'il est endommagé de façon imprévisible par des débris, cependant le satellite bien conçu pourra se recolmater sans intervention humaine afin de reprendre du service.

Ouverture des services Microsoft Azure

Afin de promouvoir cette ouverture vers de nombreux services et surtout permettre aux utilisateurs de modéliser leurs besoins, Microsoft réalise de nombreux partenariats. Certains sont parfois même surprenants dans les cas de concurrents historiques. Citons 3 exemples d'ouverture :

► **Ouverture du Cloud Azure** : Sur des projets d'ampleur, il est courant de mêler sur Azure : du Docker + Apache Cassandra + ASP.NET Web API + Apache Redis Cache + VSO + Jenkins + ORACLE database + Bus TIBCO...

► **Ouverture de Visual Studio** : Prise en compte du fait que tous les développeurs ne travaillent pas en environnement Microsoft, comme les développeurs HTML5, NodeJS, AngularJS, ... aussi, la structure des projets a été entièrement revisitée en ASP.Net 5, permettant ainsi à tous les développeurs d'ouvrir des projets avec leurs outils non-Microsoft auxquels ils sont habitués,

► **Ouverture du code source** : De nombreux frameworks ont été rendus ouverts. Par exemple, Entity Framework, ou encore l'ASP.NET 5, où ce dernier subit en ce moment même des améliorations considérables. Cette vitesse de développement est déroutante dans le monde Microsoft ; aussi, comme tout produit Open Source, des « Breaking Changes » nombreux et nécessaires sont présents en début de vie du produit.

Pour que tous ces différents services puissent interagir entre eux et s'emboîter à la manière de Legos, Microsoft a misé sur des standards reconnus et bien établis (OpenID Connect, OAuth 2.0, SAML 2.0 ...). C'est ainsi que le liant à tout cela repose sur de nombreux Web API protégés par le Token-based SSO (Single Sign On). Celui de Microsoft est nommé **Azure Active Directory** (Azure AD) ; cela dit tout fournisseur d'identités compatibles avec ces standards fera l'affaire.

L'avancée du sujet Token-based SSO est considérable, poussée par de nouveaux usages :

► Le BYOD (Bring Your Own Device),
 ► La mobilité avec des « modes déconnectés »,
 ► Les éditeurs tiers proposent de plus en plus d'applications en mode SaaS, Par conséquent, les DSI investissent de plus en plus sur la mise en place d'un nouveau système de SSO pour permettre cette ouverture vers le monde ; l'actuel étant soit trop souvent limité à l'intranet, soit peu sécurisé sur Internet, soit non synchronisé avec leur annuaire LDAP.

Pour aller plus loin en DevOps

Le sujet du DevOps est vaste ; cependant, l'infime partie évoquée précédemment permet déjà d'être opérationnel en DevTest en quelques heures. Il reste cependant nécessaire :

► de comprendre les subtilités des architectures distribuées,
 ► d'en maîtriser les bonnes pratiques (de type public cloud, private cloud ou hébergement classique),
 ► quelques particularités de la plateforme Azure.

Donc, ce monde où la complexité du Cloud est encapsulée replace le produit et le besoin du client au centre d'une DSI agile et réactive. Bon DevOps avec VS 2015 et Microsoft Azure ! Et n'oubliez pas, l'outillage ne fait pas tout. La réussite d'une transformation DevOps passe également par la gestion du changement, et les méthodologies et processus robustes. ▲

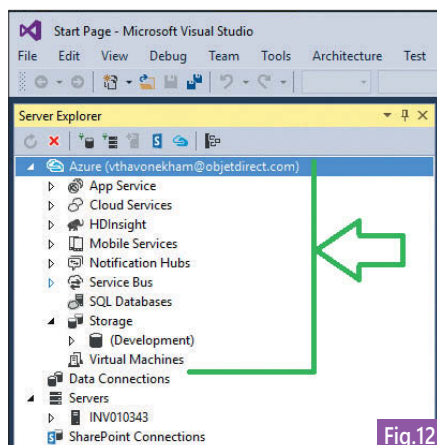


Fig.12

Développer pour Windows 10

Avec la sortie de Windows 10, les développeurs peuvent désormais s'amuser avec le nouveau Windows Software Development Kit (SDK) pour le nouveau système d'exploitation. Un nouveau SDK implique de nouvelles fonctionnalités, et le SDK Windows ne déroge pas à cette règle. Ce SDK permet de réaliser des Universal Windows Platform (UWP) apps, comprendre ici applications universelles pour les familles de devices Windows 10 : Windows Desktop, Windows Mobile, Windows IoT, Xbox...



Le concept d'application universelle est une notion très présente de nos jours. Nous avons l'exemple avec les applications universelles Windows 8.1 et l'envie de permettre au développeur de partager son code métier entre les différentes plateformes, tout en développant des vues spécifiques pour celles-ci. Chaque famille apporte son API spécifique en plus de l'API commune, pour pouvoir offrir à l'utilisateur une expérience plus adaptée en fonction du device utilisé. (Fig. 1). C'est dans cet esprit universel que le développeur d'applications Windows 10, avec cette notion de mise en commun de la logique métier, peut utiliser les nouveaux concepts, outils et contrôles fournis par ce nouveau SDK.

Windows 10 et le XAML

Le développement d'applications Windows 10 cible de nombreux devices, c'est pourquoi il faut, plus que jamais, se préparer à développer des interfaces utilisateurs pour de nombreux formats. C'est pourquoi nous disposons maintenant de nouveaux contrôles et éléments qui vont permettre aux développeurs de fournir une expérience user-friendly, quel que soit le périphérique sur lequel l'utilisateur est, que ce soit sur téléphone, tablette, ou des écrans plus larges. D'une manière générale, on peut constater que le XAML s'oriente vers un format plus clair, en essayant de gommer les fioritures du langage, afin de pouvoir écrire un XAML moins verbeux, et plus simple à comprendre, sans s'y perdre.

RelativePanel, un peu de fraîcheur dans notre XAML

Parmi les nouveaux contrôles, notamment de type conteneur, nous avons maintenant le RelativePanel. Depuis de nombreuses années, Grid, Canvas, StackPanel, ScrollView et compagnie ont pour tâche d'agencer les différents éléments des vues de nos applications. Dorénavant, une nouvelle façon d'organiser son contenu est à notre disposition. Le nom du contrôle en dit beaucoup sur son principe : la relativité. (Fig. 2).

En effet, ce nouveau contrôle dispose les éléments en déclarant des relations entre ses enfants. C'est-à-dire que l'on va pouvoir positionner un élément A à droite d'un élément B, ce qui signifie que le déplacement de l'élément B entraînera le déplacement de l'élément A, celui-ci étant toujours positionné à la droite de B. Cette nouvelle façon de penser semble intrigante à première vue, mais cela offre réellement de nombreuses possibilités une fois qu'on

Fig.2

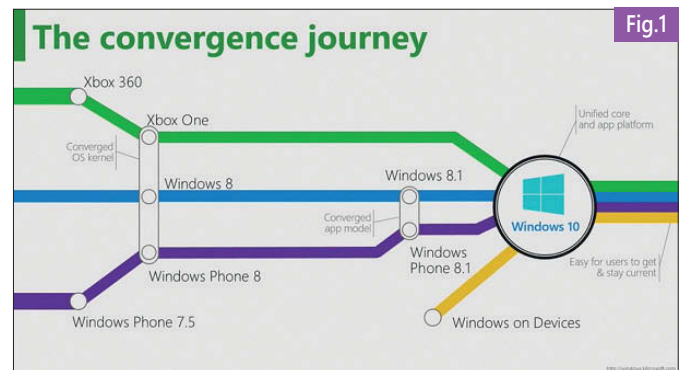
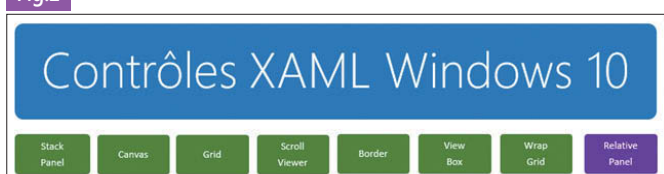


Fig.1

s'est habitué à son fonctionnement. C'est avec plus de 16 Attached Properties que l'on peut positionner les enfants du RelativePanel entre eux, mais aussi entre le RelativePanel lui-même et un enfant particulier, voyons un peu comment cela se passe au niveau du code.

On s'aligne vis-à-vis du RelativePanel

Tout d'abord, on déclare notre RelativePanel et on y ajoute un élément sans rien lui spécifier.

Code :

```
<RelativePanel>
  <Rectangle x:Name="Rect1" Width="50" Height="50" Fill="Crimson"/>
</RelativePanel>
```

Résultat :

On observe un comportement plutôt normal, l'élément vient se placer tout en haut à gauche du RelativePanel. (Fig. 3)

Vous pouvez jouer avec 6 différentes propriétés pour agencer vos enfants en fonction du Panel :

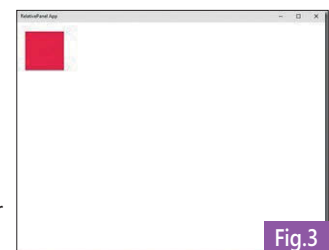


Fig.3

```
<Rectangle RelativePanel.AlignVerticalCenterWithPanel="True"
  RelativePanel.AlignHorizontalCenterWithPanel="True"
  RelativePanel.AlignRightWithPanel="True"
  RelativePanel.AlignLeftWithPanel="True"
  RelativePanel.AlignTopWithPanel="True"
  RelativePanel.AlignBottomWithPanel="True"/>
```

Agencement entre éléments

C'est l'essence même du RelativePanel, on est toujours sur des Attached Properties comme pour le RelativePanel, cependant, à la place du booléen, il est cette fois-ci nécessaire de spécifier l'élément relatif à l'action. On a donc par exemple :

Code :

```
<RelativePanel>
  <Rectangle x:Name="Rect1" Width="50" Height="100" Fill="Crimson"
    RelativePanel.AlignHorizontalCenterWithPanel="True"
    RelativePanel.AlignLeftWithPanel="Rect2"/>
```

```

RelativePanel.AlignVerticalCenterWithPanel="True"/>

<Rectangle x:Name="Rect2" Width="50" Height="50" Fill="CadetBlue"
RelativePanel.AlignVerticalCenterWith="Rect1"/>
</RelativePanel>

```

Résultat : (Fig. 4)

Ici, on a notre rectangle bleu (Rect2) qui va s'aligner verticalement avec le rectangle rouge (Rect1). Supposons que dans notre développement, nous changeons d'avis, et nous voulons que le rectangle bleu soit à la même hauteur que le rectangle rouge, nous avons juste à remplacer le `AlignVerticalCenterWith` par `AlignTopWith`. On peut donc jouer avec les propriétés suivantes :

```

<Rectangle x:Name="Rect2" Width="50" Height="50" Fill="CadetBlue"
RelativePanel.AlignHorizontalCenterWith="Rect1"
RelativePanel.AlignVerticalCenterWith="Rect1"
RelativePanel.AlignLeftWith="Rect1"
RelativePanel.AlignRightWith="Rect1"
RelativePanel.AlignTopWith="Rect1"
RelativePanel.AlignBottomWith="Rect1"/>

```

Cependant, la puissance du `RelativePanel` ne s'arrête pas là. Outre l'alignement des enfants, on peut aussi définir une position.

LeftOf, RightOf, Above, Below

Les propriétés étant assez explicites, voyons immédiatement ce qui se passe en spécifiant `LeftOf` et `Above` sur notre rectangle bleu (Rect2) :

Code :

```

<Rectangle x:Name="Rect2" Width="50" Height="50" Fill="CadetBlue"
RelativePanel.LeftOf="Rect1"
RelativePanel.Above="Rect1"/>

```

Résultat :

En reprenant les exemples précédents, le rectangle bleu (Rect2) va venir se coller au-dessus du rectangle rouge (Rect1). Cela va nous permettre ainsi de gérer des positions en même temps que des alignements dans notre vue. (Fig. 5)

Le `RelativePanel` est un contrôle qui apporte un peu de fraîcheur à notre architecture XAML, mais là où il va exceller, c'est dans cette optique d'unifier la vue, où l'on va pouvoir définir nos différents positionnements et alignements pour telle ou telle vue.

AdaptiveTrigger, adaptons notre vue

En tant que développeur d'applications Windows Phone et Windows Store, vous avez probablement eu affaire au `VisualStateManager` pour manipuler votre vue en fonction de la taille de la fenêtre, pour gérer le snap de Windows 8.1. Cependant il fallait s'abonner à l'événement `SizeChanged` de la Frame de l'application, puis changer l'état soi-même grâce à la méthode `GoToState` du `VisualStateManager`.

Tout d'abord, il n'est plus obligatoire de définir une animation pour chaque état de son `VisualStateManager`, car celui-ci supporte

maintenant les `Setters`, cela nous permet d'avoir, encore une fois, un XAML moins dense.

Prenons un exemple classique : on désire avoir un label à gauche d'une `TextBox` lorsqu'on dispose d'assez d'espace en longueur, et avoir le label au-dessus de la `TextBox` lorsque l'espace est restreint. En somme, on aimerait avoir un formulaire responsive, voici un exemple avec le `ResponsivePanel` :

```

<Grid x:Name="LayoutRoot" Background="White">
    <VisualStateManager.VisualStateGroups>
        <VisualStateGroup>
            <VisualState x:Name="NormalState">
                <VisualState.StateTriggers>
                    <AdaptiveTrigger MinWindowWidth="800" />
                </VisualState.StateTriggers>

                <VisualState.Setters>
                    <Setter Target="TextBox.(RelativePanel.RightOf)"
                        Value="Label"/>
                </VisualState.Setters>
            </VisualState>

            <VisualState x:Name="SpecialState">
                <VisualState.StateTriggers>
                    <AdaptiveTrigger MinWindowWidth="0" />
                </VisualState.StateTriggers>

                <VisualState.Setters>
                    <Setter Target="TextBox.(RelativePanel.Below)"
                        Value="Label"/>
                </VisualState.Setters>
            </VisualState>
        </VisualStateGroup>
    </VisualStateManager.VisualStateGroups>

    <RelativePanel Padding="24">
        <TextBlock x:Name="Label" Text="Label" />
        <TextBox x:Name="TextBox" Background="Blue" />
    </RelativePanel>
</Grid>

```

On définit ici que le `VisualState` « `NormalState` » est utilisé lorsque la `Width` est supérieure à 800 avec :

```

<VisualState.StateTriggers>
    <AdaptiveTrigger MinWindowWidth="800" />
</VisualState.StateTriggers>

```

Dans cet état-là, nous avons assez d'espace, donc nous utilisons la propriété `RelativePanel.RightOf` pour indiquer que la `TextBox` doit se positionner à la droite du `Label`.

```

<Setter Target="TextBox.(RelativePanel.RightOf)" Value="Label"/>

```

Dans le cas contraire, si nous disposons d'une `Width` inférieure à 800, l'autre `VisualState` « `SpecialState` » est déclenché, et c'est donc la propriété `Below` qui est utilisée. Cet exemple est relativement simple, mais montre bien que l'`AdaptiveTrigger` va devenir essentiel au développement d'applications responsives, afin d'avoir un affichage adapté à chaque format d'écran.

SplitView, un panneau latéral

Parmi ces nouveaux contrôles XAML, nous avons dorénavant la possibilité d'utiliser une `SplitView` dans nos applications. L'objectif de ce contrôle est de permettre au développeur d'intégrer la gestion

d'un panneau latéral dans son application. Attention tout de même, ce n'est pas un Hamburger menu intégré et déjà prêt à l'utilisation. Si vous désirez un Hamburger menu authentique, il va falloir combiner la `SplitView` avec un bouton qui va manipuler la `SplitView`, rien de bien compliqué en soi !

Nous avons donc le `SplitView.Pane` qui regroupe les différents menus de l'application, la gestion du profil, les paramètres, par exemple. Puis le `SplitView.Content` qui, comme son nom l'indique, va accueillir la page de contenu.

```
<SplitView>
<SplitView.Pane>
    // Le menu latéral
</SplitView.Pane>
<SplitView.Content>
    // Le contenu
</SplitView.Content>
</SplitView>
```

Pour manipuler notre `SplitView`, pas de secrets, certaines propriétés sont indispensables.

La propriété **`IsPaneOpen`** permet d'ouvrir/fermer votre menu.

`PanePlacement`, qui peut être égal à `Left` ou `Right`, détermine le côté par lequel le menu apparaît, et **`OpenPanelLength`**, la taille qu'il prend. La propriété **`DisplayMode`** vous permet de choisir entre plusieurs types d'affichages différents, on va pouvoir déplacer le contenu de la page pour faire apparaître le menu, ou tout simplement placer le menu au-dessus du contenu.

Ce nouveau contrôle permet au développeur de mettre en place des menus plus aisément, et ainsi offrir la meilleure expérience utilisateur possible, d'un point de vue ergonomique et dynamique.

Une vue différente pour chaque type d'écrans

Nous avons vu que l'`AdaptiveTrigger` est très utile lorsqu'on souhaite faire du responsive, cependant, en réalité, il est souvent nécessaire de développer une UI complètement différente pour chaque famille (Desktop, Mobile, Xbox, IoT, etc.), et donc par conséquent obtenir un fichier XAML très imposant.

C'est pourquoi il est dorénavant possible de créer des vues spécifiques pour une famille. Prenons l'exemple du célèbre `MainPage.xaml` : nous avons notre `MainPage.xaml` à la racine de notre projet, celui-ci va être le XAML standard. Supposons ensuite que nous voulons un design particulier sur Windows Mobile. Il nous suffit de créer le dossier suivant : `DeviceFamily-Mobile` et d'y ajouter une vue XAML avec le même nom : `MainPage.xaml`.

Ainsi, lorsqu'on affiche cette vue sur Windows Mobile, c'est le fichier présent dans `DeviceFamily-Mobile` qui est utilisé, et non celui présent à la racine du projet. Les développeurs ont désormais toutes les cartes en main pour pouvoir implémenter des vues adaptées à toutes les familles, tout en gardant une logique commune, unique.

Et si l'on parlait IoT

Depuis plusieurs années, on assiste à l'essor des cartes programmables permettant aux bidouilleurs de fabriquer des objets connectés simples ou de prototyper des objets complexes. Pour Microsoft, cela signifie que l'universalité de leurs applications doit devenir concrète et s'étendre jusqu'aux objets connectés. Microsoft adopte le mouvement Maker et ne le fait pas à moitié.

(Fig. 6)

Windows 10 IoT Core

Fort du succès de cette démarche d'applications universelles ne nécessitant qu'un seul développement (un seul et même code) pour cibler toutes les plateformes, Microsoft pousse jusqu'à inclure les mini-ordinateurs et cartes programmables dans la partie. Une version Windows 10 IoT Core est née pour satisfaire ces besoins spécifiques. Similaires aux cartes Arduino, les cartes Raspberry Pi 2, Galileo ou Minnowboard Max ont désormais leurs propres versions de Windows 10 leur permettant également de faire fonctionner les mêmes applications que leurs grandes sœurs.

Universal Apps

Comme nous le disions, Microsoft a pris les choses au sérieux et fournit l'intégration complète du développement sur ces plateformes. Créer une app pour IoT Core revient à créer un nouveau projet Windows 10 ou Universal Windows Platform comme un autre. Pour cela, Windows 10, Visual Studio 2015 et le Windows 10 SDK sont nécessaires, mais tous disponibles gratuitement.

Quelques nouveautés seront par contre bien utiles selon le contexte de projet IoT comme les modes « Headed » et « Headless » permettant à Windows 10 de fonctionner sur des machines avec ou sans écran. En mode Headed une seule application avec UI pourra fonctionner à la fois alors qu'en mode Headless, les backgrounds apps auront la part belle. Vos applications Windows 10 fonctionneront sans problème sur ces petits appareils mises à part certaines limitations spécifiques, mais la majeure partie des applications n'auront aucun mal à tourner et ne nécessiteront aucun développement supplémentaire. En revanche, vous pourrez ajouter des fonctionnalités à vos applications s'exécutant sur ces



plateformes, en utilisant le SDK Windows 10 IoT Core. Celui-ci est disponible sous forme de package NuGet gratuit et offre à nos applications Windows 10 les fonctionnalités supplémentaires d'accès aux entrées/sorties des broches d'un Raspberry Pi ou toute autre carte compatible. (Fig. 7)

Développez dès maintenant

Pour commencer à développer pour ces plateformes vous devrez installer Windows 10 IoT Core sur votre machine cible et développer depuis un PC sous Windows 10 avec Visual Studio 2015. Vous trouverez toutes les étapes nécessaires à vos premiers pas sur le site (<http://www.windowsondevices.com>).

Arduino

Windows 10 devient le premier système "Certifié Arduino". Le partenariat annoncé avec Arduino se traduit par une intégration très profonde de la stack IoT de Microsoft avec Arduino.

Windows Remote Arduino

L'idée est de s'appuyer sur les capacités de la plateforme Arduino à gérer le hardware en lui proposant une communication bidirectionnelle, depuis ou vers les applications Universal Windows Platform. La couche d'accès aux BDD ou au réseau pourrait ainsi être déléguée aux app Win10 et les interactions hardware aux cartes Arduino dans des scénarii IoT beaucoup plus complets qu'aujourd'hui.

La communauté Arduino étant énorme, ce partenariat peut être un véritable essor pour l'intégration des app Windows 10.

Windows Virtual Shield for Arduino

L'aspect le plus marquant de ce partenariat est la possibilité de créer des Shields virtuels pour Arduino. Ici ce sont les devices tournant sous Windows 10 qui sont au service des cartes Arduino en permettant à celles-ci d'accéder aux capacités matérielles et aux capteurs du device Win10. Par exemple, l'accéléromètre ou le GPS de votre téléphone peuvent être utilisés sur votre Arduino comme si vous aviez monté des Shields spécifiques. Cette intégration aux couches IoT est tout simplement bluffante tant au niveau technique qu'au niveau philosophie d'ouverture de la plateforme Windows.

Allons plus loin

Pour trouver des idées de projets ou tout simplement s'émerveiller devant la créativité technologique de certains, vous pourrez parcourir les exemples de code mis à disposition sur la plateforme hackster.io et ainsi découvrir l'étendue des fonctionnalités disponibles : <https://www.hackster.io/windowsiot>

Les nouveautés côté code

Avec l'arrivée des applications Universal Windows Platform, des nouveautés côté code font également leur apparition. Nous allons tout de suite découvrir les principales.

Nouveauté d'accès à Windows via Windows.System.Launcher

Avec Windows 8.1, il était possible d'ouvrir des fichiers ainsi que des URL à partir de l'application en utilisant les méthodes proposées telles que `LaunchFileAsync`, `LaunchUriAsync`. Windows 10, de par son SDK, nous propose de nouveaux accès aux fichiers, dossiers.

Ouvrir des dossiers (LaunchFolderAsync)

La méthode `LaunchFolderAsync` (<https://msdn.microsoft.com/library/windows/apps/windows.system.launcher.lau>

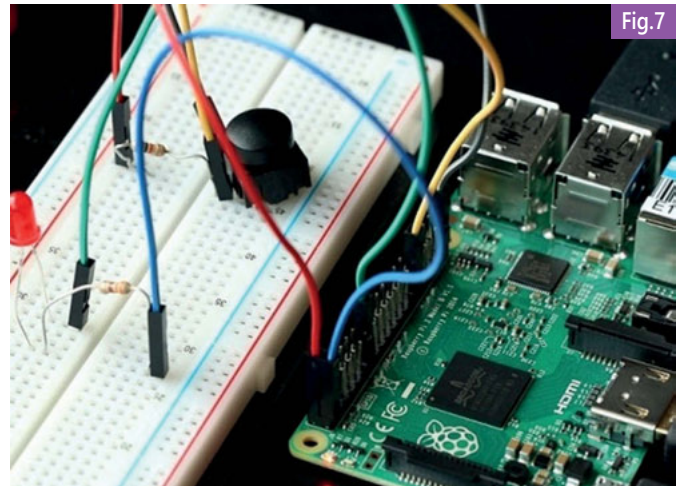


Fig.7

`nchfoldersync.aspx`) fait son apparition, elle permet de demander l'ouverture d'un explorateur de fichier vers le dossier spécifié. Cependant, comme pour l'ouverture de fichier, des restrictions sont mises en place. Il est donc parfois nécessaire de spécifier ce dossier dans le Manifest de l'application (via les Capabilities), pour obtenir le droit d'interagir avec.

Ouvrir des paramètres natifs (LaunchUriAsync)

La méthode `LaunchUriAsync` (<https://msdn.microsoft.com/library/windows/apps/windows.system.launcher.lau> `nchuriasync.aspx`), utilisée pour ouvrir une adresse internet est étendue. Cette méthode qui permet d'ouvrir des URL peut être couplée au nouveau protocole (`ms-settings`) pour afficher les paramètres natifs de Windows 10 depuis votre application.

C'est idéal pour des scénarii où votre application nécessite un environnement précis, pour ainsi guider l'utilisateur dans la configuration de celui-ci. Voici un exemple pour ouvrir l'écran de gestion du WiFi sous Windows 10 :

```
var success = await Launcher.LaunchUriAsync(new Uri("ms-settings:network-wifi"));
```

Le protocole « `ms-settings` » expose de nombreuses pages de paramètres, une liste est disponible sur MSDN (<https://msdn.microsoft.com/fr-fr/library/windows/apps/xaml/Dn741261.aspx>).

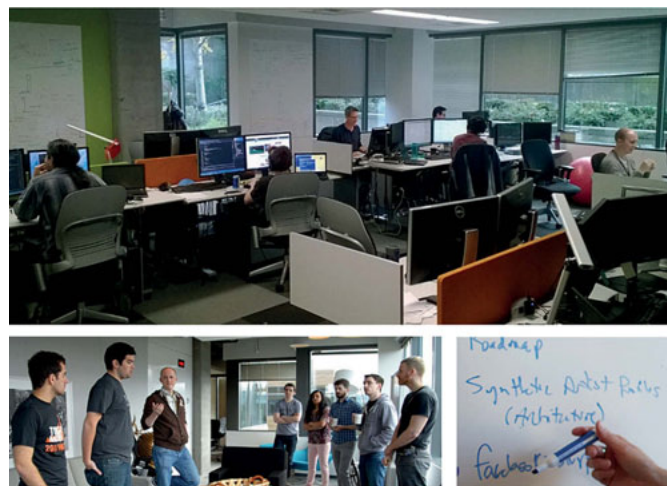
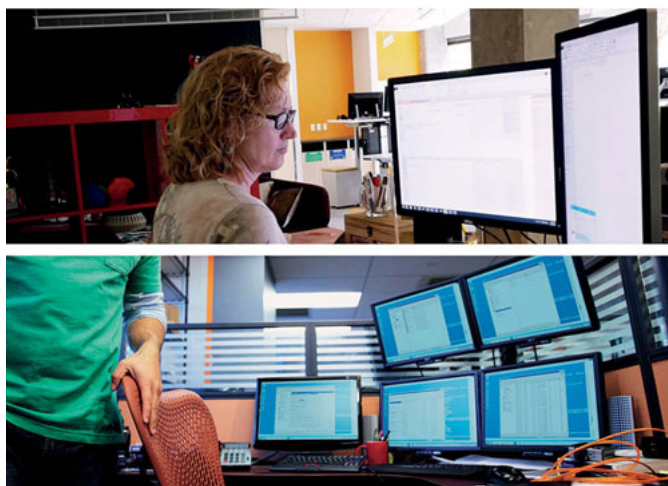
Communication entre applications

Sous Windows 8.1, notamment dans des scénarii d'entreprise, nous étions amenés à faire communiquer nos applications Windows 8.1 avec d'autres applications de l'environnement. Mais les solutions étaient minces et très limitées. Une des solutions utilisée par défaut est de passer par le « partage de fichier », en créant et en remplissant un fichier avec les données à échanger et en demandant à Windows de l'ouvrir. Cependant, ce procédé a ses limites :

- ▶ parfois Windows n'autorise pas l'ouverture de l'application,
- ▶ l'application recevant le fichier ou le protocole doit être configurée pour pouvoir le lire (parfois un développement supplémentaire est nécessaire),
- ▶ et surtout, le dialogue ne se fait que dans un sens ! (nous ne pouvons pas avoir de retour de l'application appelée).

Échanger des données entre applications (LaunchUriForResultsAsync)

Avec « App-to-App », Microsoft nous met à disposition une vraie manière de dialoguer avec une autre application de l'environnement utilisateur. « App-to-App » nous permet de résoudre les limites que nous avons avec la méthode précédemment évoquée. Ceci passe notamment par l'utilisation d'un protocole personnalisé (dans



l'exemple nous utiliserons le protocole infinitesquare) pour dialoguer entre les applications. Prenons l'exemple suivant :

- une application « A », qui joue le rôle de « client »,
- une application « B » qui joue le rôle de « serveur »,
- l'application « A » va communiquer avec l'application « B » (les deux doivent être configurées).

Configuration du « serveur » B

Dans l'application « serveur », il faut spécifier dans son Manifest (dans l'onglet déclarations) qu'elle peut répondre à un protocole défini, en renseignant le nom et le type de retour (sans, avec ou parfois). Côté code, il faut surcharger la méthode `OnActivated` dans le fichier `App.xaml.cs` pour obtenir les arguments passés lors de l'appel et faire le traitement adéquat. Dans mon exemple, je fais ce traitement dans ma `MainPage`, c'est pourquoi je lui transmets les arguments.

```
protected override void OnActivated(IActivatedEventArgs args)
{
    Frame rootFrame = Window.Current.Content as Frame;

    // Créer la frame si elle n'est pas encore définie
    if (rootFrame == null)
    {
        rootFrame = new Frame();
        Window.Current.Content = rootFrame;
    }

    // Obtention des arguments passés lors de l'appel
    var protocolForResultsArgs = (ProtocolForResultsActivatedEventArgs)args;

    // Transfert des arguments à la MainPage
    rootFrame.Navigate(typeof(MainPage), protocolForResultsArgs);

    Window.Current.Activate();
}
```

Et dans la méthode `OnNavigatedTo` de la page `MainPage`, j'effectue le traitement :

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    // Obtention des arguments passés depuis le App.xaml.cs
    var protocolForResultsArgs = e.Parameter as ProtocolForResultsActivatedEventArgs;

    if (protocolForResultsArgs == null)
```

```
{
    return;
}

// On définit l'objet stockant les valeurs à retourner
var result = new ValueSet();

// Si les arguments transmis contiennent celui que l'on attend
if (protocolForResultsArgs.Data.ContainsKey("Request"))
{
    string dataRequest = protocolForResultsArgs.Data["Request"] as string;

    if (dataRequest == "Est-ce que tu m'entends ?")
    {
        // On définit une réponse
        result["Response"] = "Hey ho !";
    }
}

// On termine l'opération et on retourne le résultat
protocolForResultsArgs.ProtocolForResultsOperation.ReportCompleted(result);
}
```

Appel à l'application B dans l'application A et exploitation du résultat

Dans l'application « client », il faut ici configurer la communication à l'application « Serveur ». Il suffit de connaître quelques informations sur ce qu'attend l'application « Serveur » :

- le `PackageFamilyName` (présent dans le manifest),
- le nom du protocole,
- les clés utilisées pour l'échange (à l'entrée, mais aussi à la sortie).

Dans notre cas, nous avons ces informations de configuration facilement puisque c'est nous qui les avons définies, mais dans le cas d'échange avec une application tierce, nous devons les obtenir ou les fournir à l'autre application.

L'appel utilise la méthode `LaunchUriForResultsAsync`

(<https://msdn.microsoft.com/en-us/library/windows/apps/dn956688.aspx>).

Voici comment utiliser cette méthode et échanger des données entre A et B :

```
var options = new LauncherOptions()
{
    // On définit le PackageFamilyName
    TargetApplicationPackageFamilyName = "5f926382-faf0-4039-9847-e085b01b248d_jsdmg9dqt3zzj"
};
```

```
// On renseigne nos données à échanger
var inputData = new ValueSet();
inputData["Request"] = "Est-ce que tu m'entends ?";

// On lance l'appel avec les données
var result = await Launcher.LaunchUriForResultsAsync(new Uri("infinite-square://"), options, inputData);

// Si le résultat est positif et qu'il contient ce que l'on attend
if (result.Status == LaunchUriStatus.Success &&
    result.Result != null &&
    result.Result.ContainsKey("Response"))
{
    // On affiche à l'utilisateur le retour
    var messageDialog = new MessageDialog(result.Result["Response"],
    ToString());
    await messageDialog.ShowAsync();
}
```

Il est bon de savoir que Microsoft bride volontairement l'échange des données entre les applications via ce système à 100ko (la valeur de l'objet dans ValueSet).

Savoir si une application supporte l'appel (QueryUriSupportAsync)

La méthode QueryUriSupportAsync (<https://msdn.microsoft.com/en-us/library/windows/apps/dn893515.aspx>) peut se combiner à la méthode précédente, car elle permet de savoir si une application peut répondre à une demande précise. Cette demande est définie en deux parties :

- son Uri (adresse d'un fichier ou protocole),
- son type (Uri ou Uri avec résultat).

Cette méthode permet de mieux anticiper l'environnement de l'utilisateur et pouvoir l'avertir s'il est nécessaire d'installer un logiciel tiers pour profiter au maximum des fonctionnalités que lui met à disposition l'application Windows 10.

Gestion des modes d'affichage (plein écran, fenêtre, tactile ou non)

Les applications Windows 10 s'affichent nativement en mode fenêtre et sont donc « responsives ». Pour offrir une meilleure expérience utilisateur, il est parfois nécessaire de savoir comment l'application est utilisée pour adapter cet usage en fonction. Avec la propriété UserInteractionMode de UIViewSettings, nous sommes capables de savoir si Windows est utilisé en mode « touch » ou à la souris, grâce à l'énumération obtenue via :

UIViewSettings.GetForCurrentView().UserInteractionMode.

Nous pouvons également connaître le mode d'affichage de l'application (fenêtré, plein écran) et agir sur celui-ci :

- entrer en mode plein écran,
- sortir du mode plein écran,
- définir la taille de la fenêtre de l'application.

Pour interagir avec la fenêtre de l'application, nous devons obtenir la fenêtre courante en utilisant la méthode GetForCurrentView (<https://msdn.microsoft.com/en-us/library/windows.ui.viewmanagement.applicationview.getforcurrentview.aspx>) qui permet d'obtenir la vue active.

Toast Notifications

Avec les UWP, les notifications sont grandement enrichies et peuvent désormais avoir des « actions » utilisateurs.

Les notifications sont constituées de 3 parties :



- La partie « Visual » : ici il n'y a qu'un seul template de base que nous pouvons personnaliser en ajoutant/modifiant/supprimant l'icône, des images, du texte ... Ce visuel s'adapte automatiquement aux différentes plateformes (smartphone, PC, tablette, Xbox).
 - La partie « Actions » : elle est située en dessous de la partie « Visual » et peut accueillir au maximum 3 boutons de type « Input » et/ou une case de texte pour permettre à l'utilisateur d'interagir avec la notification et donc avec votre application.
 - La partie « Audio » : comme sur Windows Phone 8.1, la partie audio reste inchangée, elle peut jouer un son personnalisé depuis l'application (ms-appx, ms-appdata).
- À ceci s'ajoute, le choix du type de notification qui est émis. Ce type permet à Windows d'interpréter de manière différente la notification (plein écran, son en boucle ...). Ce type est appelé « scénario » dans le code XAML de l'élément racine toast et peut avoir plusieurs valeurs possibles (default, alarm, reminder, incomingCall). En fonction du besoin, ce type peut être grandement utile pour que l'utilisateur ne rate pas la notification lorsqu'elle est émise. Avec toute cette personnalisation, l'utilisateur peut effectuer des choix et agir via la notification avec l'application. Voici quelques exemples d'interactions :
- Ouvrir l'application au premier plan avec une action spécifique,
 - Ouvrir l'application en fond,
 - Ouvrir une autre application,
 - Effectuer une action système qui se limite aux alarmes et aux rappels (pour le moment).

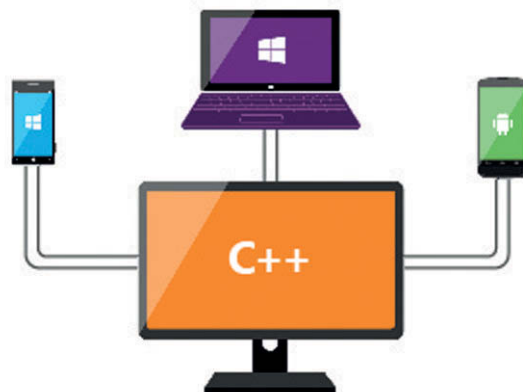
Cortana

Cortana, la célèbre assistante présente dans Windows 10, peut également être utilisée dans vos applications ! En l'intégrant dans notre programme, nous donnons la possibilité aux utilisateurs d'interagir avec celui-ci sans avoir besoin de le lancer. Par le biais de Cortana, l'utilisateur peut effectuer des actions simples avec notre application. Prenons le cas d'un logiciel de réservation d'hôtel, l'utilisateur demande à Cortana un hôtel proche. Celle-ci pourra lui faire des propositions à partir des réponses générées par notre programme. Il pourra ensuite aller jusqu'à la réservation sans même avoir ouvert le logiciel ni avoir touché à son clavier ! De plus, si notre application est paramétrée pour interagir avec Cortana, celle-ci peut naviguer, effectuer des actions, écrire dans l'application. Les scénarios peuvent être assez vastes. Dans la plupart des cas, Cortana va permettre de gagner du temps, et également réduire les efforts de l'utilisateur (rechercher ce dont il a besoin, prendre en main l'application ...) Cette nouveauté demande un développement particulier, l'intégration étant assez longue, nous n'allons pas nous y attarder ici. Cependant, les personnes intéressées et qui souhaitent mettre à profit Cortana dans leurs applications trouveront la marche à suivre sur MSDN

(<https://msdn.microsoft.com/library/windows/apps/xaml/dn974230.aspx>) ▲

Mobilité

Le développement mobile cross-plateformes est un enjeu qui devient de plus en plus important. Comment fournir une application de qualité et adapter aux utilisateurs des différentes plateformes tout en gardant un coût de développement et de maintenance non exponentiel ? Plusieurs outils / Frameworks proposent des approches différentes pour répondre à ces défis. Microsoft a décidé d'adresser toutes ces solutions en intégrant un ensemble d'outils dans Visual Studio 2015, de manière à nous laisser le choix de la meilleure solution en fonction de notre contexte.



Kévin ALBRECHT
Teddy DESMAS
Thomas LEBRUN
Sébastien OLLIVIER
Thomas OUVRE
Consultants **Infinite Square**



Dans cet article, nous étudierons notamment Apache Cordova, les possibilités de développement en C++ dans la version 2015 de Visual Studio, ainsi que les possibilités offertes par Xamarin. Nous verrons également la solution des Windows Bridges permettant de déployer une application iOS ou Android sur le store Windows.

Apache Cordova

Apache Cordova est un Framework open source développé par la fondation Apache, permettant de créer des applications mobiles cross-plateformes en utilisant les langages HTML / CSS / JavaScript. L'idée est de pouvoir cibler toutes les plateformes mobiles majeures en développant une seule application et en s'émancipant en partie des langages natifs comme Java, C#, Objective-C, ou autre. Pour pouvoir interagir avec le device, Apache Cordova met à disposition un ensemble d'API JavaScript permettant d'accéder aux fonctionnalités natives, de manière unique quelle que soit la plateforme.

Le principe

Le principe est de développer une application Web, en HTML / CSS / JavaScript, et de la packager sous la forme d'une application native. Pour que le rendu de l'application Web s'adapte à tous les devices, il est nécessaire de développer une application Web responsive, par exemple à l'aide du Framework CSS

Bootstrap. Pour proposer un rendu plus fini, il est également recommandé de créer une application Web de type Single Page Application (SPA), par exemple à l'aide des Frameworks AngularJS ou KnockoutJS. L'idée est donc ici d'avoir un seul code métier et un design unique sur toutes les plateformes. Techniquement, le principe du Framework Cordova est de créer une application native composée uniquement d'un composant WebView, propre à la plateforme ciblée. Ce

composant WebView naviguera simplement sur l'application, dont les fichiers (HTML, CSS, JavaScript, font, images, etc.) auront été embarqués en tant que ressources locales de l'application native.

Accès aux fonctionnalités du device

Pour pouvoir accéder aux fonctionnalités natives du téléphone, l'application développée va pouvoir s'appuyer sur les plugins Cordova. Un plugin Cordova permet d'exposer une fonctionnalité native du device via une API JavaScript, qui sera utilisable depuis l'application Web. Le rôle du plugin est alors de faire le lien entre l'appel JavaScript et l'exécution du code natif, en fournissant une implémentation par plateforme. L'interface JavaScript étant la même pour chaque plateforme, l'utilisation d'un plugin se fait de manière identique, quelle que soit la cible.

Par exemple, l'accès aux contacts du téléphone se fait en utilisant le plugin « Contacts », qui expose les fonctions create, find et pickContact, permettant respectivement de créer un contact, de rechercher un contact et d'afficher un composant natif de sélection d'un contact.

Apache Cordova met à disposition un ensemble de plugins permettant d'accéder aux fonctionnalités basiques du device, comme les notifications, la caméra, les contacts, l'accéléromètre, etc. D'autres acteurs ont également créé des plugins, y compris Microsoft via sa filiale « Microsoft Open Technology », avec notamment un connecteur Cordova pour Azure Mobile Services, disponibles pour les plateformes Windows 8, Windows Phone 8, mais aussi iOS et Android.

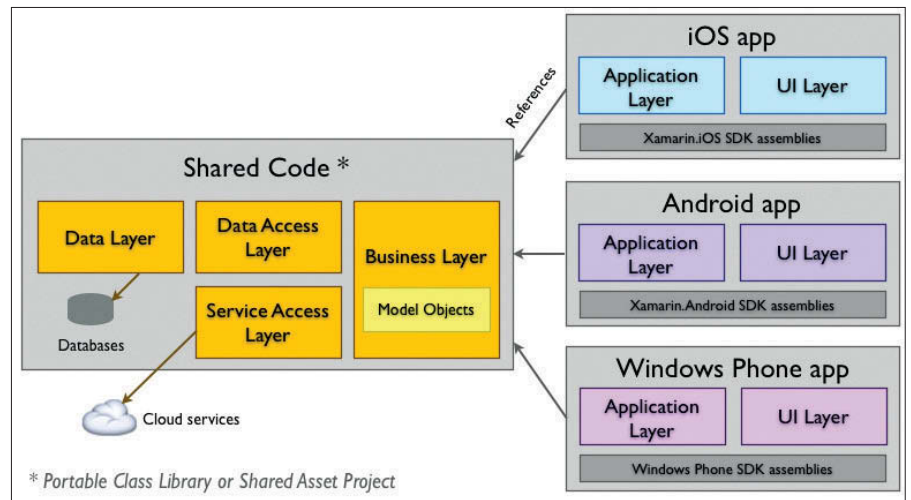


Apache Cordova s'appuie également sur une communauté active pour développer de nombreux plugins, majoritairement sur GitHub, comme un scanner de code-barre ou de QR Code, des connecteurs Facebook ou encore des notifications « Push ». L'engouement grandissant autour de Cordova fait que ce framework devient de plus en plus complet et permet de développer des applications de plus en plus évoluées, tout en gardant l'avantage de n'avoir qu'un seul développement à faire pour cibler plusieurs plateformes.

Intégration avec Visual Studio

Visual Studio 2015 inclut un ensemble d'outils permettant de faciliter le développement d'applications avec Cordova. Alors qu'il est nécessaire d'utiliser des lignes de commandes (cli) pour manipuler une application Cordova (ajouter une plateforme, ajouter un plugin, compiler une application, etc.), Visual Studio encapsule toutes ces commandes dans des tâches de Build, pour retrouver les réflexes habituels. Pour cela, à l'installation, Visual Studio vous proposera un ensemble d'outils tiers, comme iTunes, node.js, la JVM, le SDK Android, etc.

Ces outils sont également disponibles sur Visual Studio 2013, via l'extension *Multi Device Hybrid Apps*. Le débog d'applications Cordova est également un point difficile lorsque l'on se passe de Visual Studio. En effet, pour avoir accès aux DOM, à la console, etc. il est nécessaire de passer par des outils externes, comme VorlonJS ou Weinre. Ces outils ne permettent pas, par contre, de débogger directement le JavaScript de l'application. Visual Studio quant à lui propose une expérience de débog riche, de manière assez identique à ce qu'on a l'habitude de voir sur cet IDE, et ce sur toutes les plateformes (Windows Phone, Android et iOS). Enfin, pour toute la partie industrialisation des développements, Microsoft a fait le choix d'intégrer à son éditeur des outils open source très populaires dans la communauté Web, plutôt que de développer ses propres outils. NPM est utilisé pour gérer toutes les dépendances serveur, bower pour les dépendances clients. Grunt ou Gulp sont là quant à eux pour mettre en place tous les workflows de build (compilation du TypeScript, bundlification des fichiers CSS, compression automatique des images, etc.). Petit bonus, il est possible de lier notre application sur le store et de créer l'app package directement depuis l'application Cordova ce qui facilite grandement la publication.



Pourquoi Cordova ?

Cordova offre le gros avantage de factoriser la totalité du code applicatif. Toutes les vues et tout le code métier seront réutilisés sur toutes les plateformes.

L'un des avantages de Cordova vient également de sa base technique : HTML / CSS / JavaScript. Si vos équipes possèdent déjà des compétences dans le développement Web, le pas à franchir pour créer des applications natives sera très simple, comparé à l'effort nécessaire pour monter en compétence sur un nouveau langage ou sur un nouvel environnement. Il y a quelques années, Cordova était critiqué pour son manque de performance et son manque d'intégration avec le device. Ces points ont été corrigés, d'une part par les navigateurs mobiles (utilisés par le WebView Cordova) qui ont des moteurs d'exécution JavaScript beaucoup plus optimisés, et d'autre part par le nombre très important de plugins actuellement disponibles. Il est maintenant possible de créer des applications Cordova ressemblant à s'y tromper à des applications natives.

Xamarin

Xamarin peut être assimilé à un framework permettant de développer des applications mobiles pour iOS, Android et Windows Phone en s'appuyant sur un langage de développement unique (C# ou F#). Xamarin permet de mutualiser toute la logique applicative, grâce à une bibliothèque de classes portable ou sur un projet partagé. L'un des points forts de Xamarin réside dans ce que le produit offre aux développeurs : *tout ce qui est réalisable techniquement avec Objective-C ou Java peut être fait en C# avec Xamarin !*

Concrètement, cela signifie que tout ce qui concerne la logique métier, les accès aux données, la communication réseau, etc. est partagé sur l'ensemble des plateformes

tandis que l'interface graphique est développée en utilisant les APIs natives, rendues accessibles en C#.

Xamarin est composé de 4 produits :

- **Xamarin.Android** : qui représente la surcouche permettant l'exécution d'applications C# sur les plateformes Android
- **Xamarin.iOS (MonoTouch)** : qui permet aux applications Xamarin de s'exécuter sur iOS
- **Xamarin.Mac** : qui offre aux développeurs la possibilité d'exécuter leurs applications sur OS X
- **Xamarin Studio** : qui n'est rien d'autre qu'un environnement de développement, basé sur [MonoDevelop](#), permettant de développer des applications C# pour Android (sans disposer de Visual Studio) ou des applications iOS sous OS X.

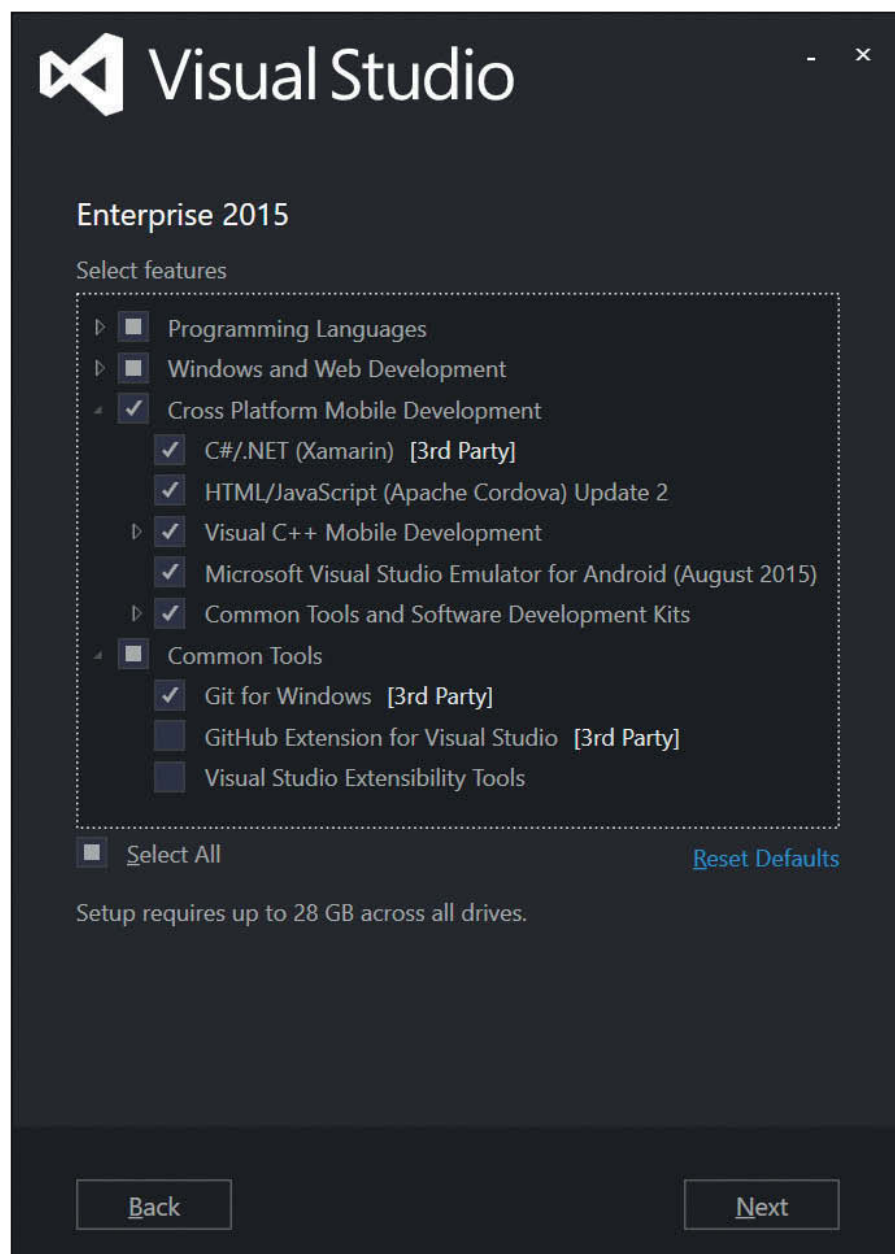
Pourquoi Xamarin ?

Si l'on en croit les possibilités offertes par le produit, Xamarin semble donc être prometteur et se présente comme la solution idéale pour les développements d'applications mobiles multiplateformes. Mais, au-delà des fonctionnalités en elles-mêmes, l'intérêt de Xamarin réside aussi et surtout dans le principe de « réutilisation ». En effet, comme nous l'avons vu juste avant, Xamarin offre aux développeurs la possibilité de développer leurs applications iOS/Android en utilisant Visual Studio et C# et donc, implicitement, de réutiliser le savoir-faire qu'ils ont acquis lors de leurs développements précédents. La courbe d'apprentissage est donc extrêmement simple pour les développeurs ayant l'habitude de l'environnement Microsoft : seules les nouvelles APIs liées à l'environnement cible sont à connaître, même s'il reste nécessaire de connaître les paradigmes et les concepts des différentes plateformes pour lesquelles une application

est développée, ceci afin de respecter au mieux les codes de l'expérience utilisateur. Un autre avantage de Xamarin concerne les possibilités techniques offertes aux développeurs. En effet, lorsque l'on regarde de plus près des applications iOS ou Android, on constate que celles-ci sont très souvent peaufinées dans les moindres détails (animations, gestes, etc.). Avec Xamarin, il sera possible de reproduire l'ensemble de ces petits détails, car **100%** des APIs sont couvertes ! Une collaboration forte avec **Microsoft** : en plus des points cités précédemment, il faut savoir que Microsoft et Xamarin travaillent conjointement pour faciliter le développement d'applications iOS / Android, au travers de Visual Studio. Ainsi, si un partenariat important a été annoncé entre les 2 sociétés il y a un peu moins de 2 ans, le produit est maintenant plus fortement intégré à Visual Studio, en proposant, à l'installation, la possibilité d'installer Xamarin.

Visual C++ Cross-Platform Mobile

Visual C++ Cross-Platform Mobile de Visual Studio 2015 permet maintenant aux développeurs de créer des applications visant les trois grandes plateformes mobiles du moment avec un unique langage, le C++. À la différence d'une technologie comme Cordova, le but n'est pas de s'abstraire des patterns et composants de chaque plateforme, mais plutôt de proposer un langage permettant de les manipuler directement, sans abstraction. Ainsi, il devient possible de créer des applications natives tout en mutualisant du code entre les différentes plateformes. L'émergence d'un tel produit vient en partie d'un constat très évocateur : 80% des applications les plus populaires sous Android ont été développées en C++. Le choix de ce langage par les grands éditeurs derrière ces applications n'est pas un hasard : il s'agit du dénominateur commun entre ces trois plateformes. Il offre à la fois toute la souplesse requise pour manipuler les APIs natives de chaque plateforme tout en permettant une mutualisation maximum. De plus, des bibliothèques maintenant très populaires parmi les développeurs C++ (comme openCV pour le traitement d'image ou Qt pour les interfaces utilisateurs) sont disponibles sous chacune de ces plateformes, ce qui montre d'une part l'universalité des langages C et C++, et offre d'autre part des possibilités intéressantes pour les développeurs d'applications souhaitant bénéficier de ces



bibliothèques. Visual C++ Cross-Platform Mobile offre tout l'outillage nécessaire au sein d'un unique outil : Visual Studio. En effet, il peut être assez contraignant et ardu de maîtriser les différents compilateurs C/C++ de chacune des plateformes : CLang, GCC ou encore VCC. Comme à son habitude, Microsoft fournit un outil maniable de par la configuration, mais prêt à l'emploi ce qui permet de se concentrer rapidement sur le code. Le développeur habitué de Visual Studio disposera de la plupart des outils qui font sa force. On y trouve par exemple un débogueur avec la possibilité d'inspecter du code à l'exécution, de l'exécuter en pas-à-pas, etc. Il est aussi possible d'y développer une application Android sans device à l'aide de l'émulateur intégré. En revanche, il sera nécessaire de disposer d'une machine exécutant OS X et Xcode pour compiler une application iOS. Toutefois, le debugging se fera au travers de Visual Studio.

Concrètement, à qui s'adresse Visual C++ Cross Platform Mobile ?

Tout d'abord et naturellement, aux développeurs familiers du langage C++ souhaitant continuer à utiliser ce langage sans devoir absolument apprendre celui d'une des plateformes cibles. Mais aussi aux développeurs de jeux : il s'agit en effet d'une solution permettant de travailler « au plus près de la machine » et d'attaquer des bibliothèques graphiques comme OpenGL (pour Android et iOS) et DirectX (pour Windows) et d'obtenir les meilleures performances possible. Ainsi Visual Studio dispose même de template de projet prêt à l'emploi générant un projet Android et un projet Xcode avec une base de code commune écrite en C++, dont le rendu graphique est effectué via OpenGL. Enfin les développeurs d'applications au sens large souhaitant mutualiser une partie de leur

code, pourront le faire en C++. En effet, on peut imaginer qu'une partie de l'application soit développée dans le langage de la plateforme, notamment pour tout ce qui concerne les interfaces utilisateurs (en effet, même si ces APIs sont attaquables en C++, cela peut représenter un effort conséquent) et que d'autres briques soient développées en C++, pour des raisons de performances et/ou pour la possibilité de mutualiser du code logique. Chacune des plateformes dispose ensuite de moyens d'accéder aux branches de codes écrites en C++ depuis le code Java, Objective-C ou C#.

Windows Bridge

Dans le cas où l'application existe déjà sous Android et iOS, la solution du développement cross-plateformes n'est pas forcément la plus intéressante.

Pour cette situation, Microsoft propose les Windows Bridges. Windows Bridge est une passerelle proposée par Microsoft pour que les développeurs iOS, Android puissent porter leurs applications vers Windows 10. Cet outil open source, encore en preview aujourd'hui (octobre 2015), permet, avec une légère adaptation, de garder le code natif (Objective-C ou Java) de l'application pour créer une version fonctionnelle de celle-ci sur ce nouveau système d'exploitation. Avec Visual Studio, le développeur iOS ou Android peut donc réaliser une version de son application pour ce nouveau marché qui couvre plusieurs périphériques (ordinateur, tablette, smartphone, Xbox ...), et ce sans connaître l'environnement Windows ! La version Android n'est disponible pour le moment que pour quelques chanceux, mais la version iOS est disponible sur GitHub (<https://github.com/Microsoft/WinObjC>).

Debug Cross-Plateformes

Visual Studio 2015, offre un ensemble complet d'outils de Build et de Debug pour les plateformes Windows, Windows Phone, iOS et Android.

Ripple

Ripple est une extension du navigateur Chrome permettant d'exécuter l'application Cordova dans le navigateur, en émulant un environnement se rapprochant de l'environnement mobile.

Ripple propose actuellement d'émuler plusieurs devices, notamment des iPhone, iPad, BlackBerry ou Android. L'avantage de cette extension est de pouvoir déboguer rapidement son application depuis le navigateur, sans avoir à la déployer sur un device, ou à utiliser un émulateur. Il est

également possible d'utiliser les outils développeurs de Chrome pour déboguer l'application, ce qui arrêtera le débogueur de Visual Studio.

Il est possible de modifier certaines propriétés du device, comme son orientation, sa position géographique, la batterie, de modifier l'accéléromètre, de déclencher certains événements (Passage en mode avion, etc.) et bien d'autres. Ripple s'exécutant sur le navigateur, il ne peut pas exécuter les plugins Cordova, mais propose un mécanisme permettant de saisir manuellement les données qui auraient dû être retournées par l'appel au plugin. Ripple reste donc une solution très restreinte, mais fonctionnelle aux premières phases de création d'une application, mais ne permet pas de se substituer totalement aux émulateurs et aux devices.

Windows Phone

Visual Studio 2015 permet de déboguer et déployer une application native, Xamarin ou Cordova sur Windows / Windows Phone que ça soit un émulateur ou un device physique. De grandes améliorations ont été apportées pour le débogage, notamment au niveau du DOM explorer afin d'inspecter une application pendant son utilisation en temps réel.

Android

Visual Studio 2015 intègre un nouvel émulateur Android basé sur la version Kitkat fonctionnant sous Hyper-V comme les émulateurs Windows Phone, à ne pas confondre avec « Google Android Emulator » qui correspond à l'émulateur fourni avec le SDK Android. Il est utilisable pour déboguer des projets Apache Cordova, C++ mobile ou Xamarin. Contrairement à l'émulateur fourni avec le SDK Android, l'émulateur de Visual Studio est très performant et permet de profiter d'un environnement de test fidèle aux devices réels avec une expérience de debug quasi similaire à celle sur Windows Phone.

iOS

Le développement d'une application iOS nécessite l'utilisation d'un Mac et de l'IDE Xcode. Visual Studio 2015 ne permet pas d'échapper à cette contrainte, mais intègre un mécanisme permettant de déboguer directement depuis un device iOS branché à un Mac. Ce mécanisme nécessite l'installation d'un agent sur le Mac, qui servira de lien entre Visual Studio et Xcode et des Xcode Command-line tools. Cet agent est disponible sous la forme d'un package npm : « remotebuild ».

Une fois la communication établie entre Visual Studio et l'agent, il est possible de déboguer les applications sur un iPhone, branché au Mac, ou sur un émulateur iPhone. Techniquement, Visual Studio enverra en HTTP les informations du projet qui seront interprétées par Xcode. Ensuite, le Mac transmettra les données du device ou de l'émulateur à Visual Studio qui pourra retranscrire les informations comme s'il s'agissait d'un debug en local.

Conclusion

Depuis les premières versions bêta de Visual Studio 2015, Microsoft a mis un accent particulier à l'intégration d'outils de développement cross-plateformes. Dans sa version finale, Visual Studio se positionne ainsi en tant qu'outil incontournable pour développer des applications cross-plateformes, aussi bien en utilisant Xamarin, C++ ou Cordova. Microsoft a également fait le choix de s'ouvrir encore plus au monde de l'open source en intégrant à son IDE l'utilisation de frameworks très populaires comme npm, bower ou gulp. Ajoutez à cela tous les outils déjà à disposition depuis quelques années, comme le mécanisme de plugin (Resharper, GhostDoc, etc.) ou l'intégration d'outils collaboratifs (TFS, VSO, Git, etc.), Visual Studio est un IDE plus complet que jamais.

Reste maintenant pour vous à choisir la meilleure solution pour vos développements cross-plateformes. Il n'y a pas de choix évident, tout dépend de votre contexte, de vos connaissances, de vos habitudes. Mais pour établir des règles générales, on pourrait dire que Apache Cordova est plus adapté si vous avez l'habitude du développement Web ou si vous souhaitez également créer un site responsive. Si, au contraire, vous êtes plus un développeur .NET et que votre application possède beaucoup de règles métiers, Xamarin est sans doute l'outil dont vous avez besoin.

Si vous êtes à l'aise avec C++, C++ Cross Platform Mobile vous permettra de créer des assemblies ou framework cross-plateformes afin de gagner en productivité et maintenabilité. Enfin, Windows Bridge est à utiliser lorsque vous souhaitez porter une application Android ou iOS sur Windows à moindre coût.

Dans tous les cas, l'outil de développement universel capable de répondre à vos besoins sera Visual Studio ! ▲

Containers, Docker, Azure, Windows Server et Visual Studio...

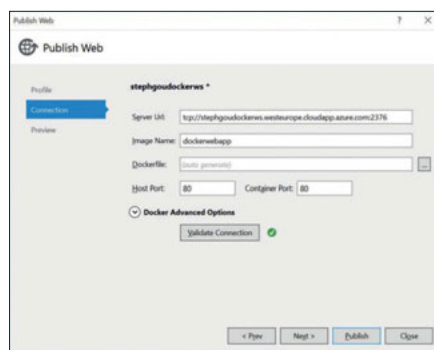
Régulièrement, l'industrie des systèmes d'information voit son évolution rythmée par des innovations ou des transformations majeures. Les points d'inflexion de cette courbe technologique vont alors généralement de pair avec l'ampleur de l'intérêt qu'ils suscitent. Et aujourd'hui, nombreux sont ceux qui s'intéressent aux containers et à Docker...



Stéphane Goudeau
Microsoft DX Architect

En particulier Microsoft, avec le support de Docker sur la plateforme Microsoft Azure, sa Marketplace et Windows Server 2016. Rien d'étonnant à ce que cet engagement se manifeste maintenant avec Visual Studio : les caractéristiques d'un container (isolation, optimisation des ressources, vitesse de déploiement et de démarrage) en font un candidat idéal pour les tests et développement.

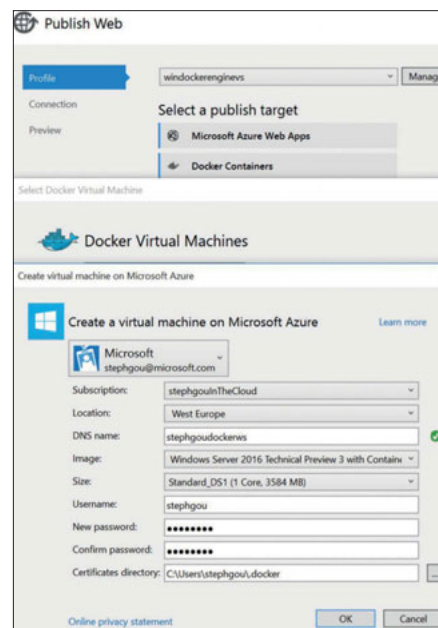
En effet, le principe d'un container est de donner l'illusion à un processus qu'il est le seul à s'exécuter sur la machine, en isolant, dans un espace de noms virtualisé, les ressources auxquelles on souhaite lui donner accès. Ainsi, un conteneur ne peut accéder aux fichiers ni interagir avec des applications qui se situeraient en dehors de son espace. Voilà qui favorise le partage et la réutilisation du container ou de son image par d'autres acteurs du cycle de vie logiciel. La consommation des ressources est également optimisée, car fichiers, répertoires et services en cours d'exécution sur l'OS sont partagés entre les conteneurs. En conséquence, le déploiement et le lancement des conteneurs multiples sur le même hôte sont extrêmement rapides. Cette efficacité du conteneur est donc de nature à favoriser les multiples itérations du développement, avec une vision très « Lean » de l'optimisation des ressources... À eux seuls, ces éléments suffisaient à justifier une évolution de Visual Studio ciblant l'utilisation de la nouvelle génération de containers incluse dans Windows Server 2016. Mais avec le support de Docker, l'impact va bien au-delà. En effet, Docker propose un ensemble de commandes communes pour la gestion des containers, un modèle de packaging permettant de décrire leur contenu (leur image) et un mécanisme de déploiement permettant de les distribuer sur n'importe quel hôte, qu'il soit Linux ou Windows, voire de les



télécharger vers des référentiels privés ou publics (Docker Hub). Il s'inscrit donc parfaitement dans une démarche DevOps par sa capacité à accélérer le cycle de vie de l'application et facilite l'implémentation d'architectures Micro-Services par la décomposition en containers.

DevOps et architecture Micro-Services

Visual Studio 2015 se devait donc d'accompagner cette évolution. Il est donc désormais possible de déployer un container directement depuis Visual Studio grâce aux Visual Studio Tools pour Docker. L'extension Docker pour Visual Studio permet de configurer l'environnement Docker, fondé sur une architecture client-serveur. Le client Docker communique avec le serveur hôte Docker grâce à un service, le « Docker engine », qui assure la génération et la publication des images ainsi que leur exécution au sein de containers. Elle installe, sur le poste de travail Windows du développeur, l'interface en ligne de commande Docker 1.8.1, pour la gestion et le diagnostic des conteneurs. Elle automatise la configuration du serveur hôte Docker, une machine virtuelle Windows ou Linux, et la déploie sur la plateforme Microsoft Azure ou en local (custom host). Cette machine virtuelle est décrite par un template Azure Resource Manager (ARM) et son déploiement est assuré par un script Windows PowerShell. Enfin, elle génère automatiquement les certificats permettant de communiquer en toute sécurité (via Transport Layer Security - TLS) depuis



l'environnement de développement vers le serveur hôte Docker.

Elle permet de directement publier une nouvelle image de container ou une application dans un container sur le serveur hôte Docker. Elle génère également les fichiers MSBuild, Bash (pour Linux ou Mac), ou PowerShell, afin de pouvoir lancer cette publication, que l'on peut également déclencher directement en ligne de commande. Enfin, elle génère un Dockerfile dont le contenu variera en fonction du système Windows ou Linux et que l'on peut utiliser comme point de départ pour ajouter ou modifier le conteneur. En effet, ce fichier contient toutes les commandes qu'il faudrait exécuter manuellement pour parvenir à la construction d'une image de Docker. Dans l'environnement proposé aujourd'hui, il s'agit d'une application console ou d'une application Web ASP.NET 5. Comme l'indique le Dockerfile, lorsque la cible est un environnement Linux, c'est le runtime Mono qui sera utilisé alors que sous Windows, c'est le .NET Core runtime. Nul doute que le type de projets Visual Studio, que l'on pourra déployer dans des containers par le biais de cette extension, devrait progressivement s'enrichir...

Un fonctionnement de Docker au sein de containers Windows rigoureusement symétrique à celui que l'on connaissait déjà pour Linux, outillé avec un environnement de développement aussi complet que Visual Studio, complété par les facilités offertes par Microsoft Azure, voilà de quoi assurer un futur prometteur à Docker et, qui sait, peut-être transformer le mythe du « Write once, run anywhere » en réalité ?

Visual Studio 2015 & IoT

L'Internet des Objets deviendra tôt ou tard une commodité. De plus en plus de projets seront impactés par ce phénomène. Au même titre que l'approche mobile en son temps, nos solutions auront de grandes chances de devoir traiter un volet objets connectés. La capacité à délivrer rapidement des solutions intégrées pourra faire la différence entre un acteur et un autre.



Alex Danvy
Evangeliste chez Microsoft

Les possibilités sont nombreuses, nous allons citer quelques exemples de projets liés à l'objet lui-même. Pour cela, nous allons nous appuyer sur le projet Open Source « Connect The Dot » disponible sur GitHub

<https://github.com/Azure/connectthedots>.

Ce projet illustre la connexion d'objets au Cloud en utilisant Microsoft Azure et des objets très variés.

Raspberry Pi

Le très versatile Raspberry Pi 2 est un véritable ordinateur qui permet l'exécution d'un système d'exploitation à la différence de cartes moins évoluées. Historiquement, diverses distributions de Linux sont disponibles, mais aussi depuis peu, Windows 10 IoT Core, une version allégée et dédiée aux objets connectés de Windows 10 offre une option supplémentaire pour les concepteurs. Plusieurs langages peuvent servir à animer les objets. Que vous souhaitiez travailler sur des scripts Python (Linux uniquement) ou Javascript/Node.js

(utiliser le plug-in Node.js Tools IoT extensions <https://github.com/ms-iot/ntvsiot>), ou réutiliser vos connaissances C# (Mono sur Linux avec la librairie RaspberryPi.net <https://github.com/cypherkey/raspberrypi.net> et Universal Windows Platform sur Windows IoT Core) vous profiterez de toute la puissance de l'environnement intégré de Visual Studio 2015.

La création d'une application multi-appareil (PC, Tablette, smartphone, IoT), qu'elle soit visuelle ou non, est rendue possible par Windows IoT Core et son modèle de développement Universal Windows Platform (Fig. 1). De la conception visuelle avec Blend au débogage pas-à-pas en passant par le partage de code ou l'optimisation assistée des performances, c'est toute la puissance du Visual Studio qui est mise à contribution pour la mise au point de vos objets. Il existe un kit de démarrage officiel complet et très abordable qui permet d'évaluer la plateforme, disponible chez Adafruit <http://www.adafruit.com/windows10iotpi2>

Arduino

La fantastique plateforme Arduino se voit accompagner d'un outil de développement parfaitement fonctionnel, mais assez rudimentaire. Grâce à ses capacités d'extensions par plug-ins notamment, Visual

Studio 2015 permet d'être enrichi de nouvelles fonctionnalités. Visual Micro (<http://www.visualmicro.com>) en est un excellent exemple. Compatible avec toutes les cartes Arduino, il découplera votre productivité. Tout y est : Coloration syntaxique détaillée, IntelliSense, débogueur, etc. (Fig. 2)

.net Micro Framework

Le .net Micro Framework est la version embarquée du .net Framework, fonctionnant sur des micro contrôleurs (sans système d'exploitation). Très léger, peu énergivore, il ouvre toute une palette de scénarii. Son caractère Open Source (<http://netmf.github.io>) a favorisé le développement de nombreuses cartes, notamment des cartes compatibles broche à broche avec Arduino, dénommées Netduino (<http://www.netduino.com>), mais aussi Gadgeteer (<http://www.netmf.com/gadgeteer>), une initiative originale issue des laboratoires de Microsoft Research Cambridge. Gadgeteer relève le défi de permettre à des développeurs de logiciels de réaliser des montages sans peine, car assisté visuellement et sans soudure. (Fig. 3)

Cloud

Si les objets connectés sont les yeux, les oreilles, voire les bras du système, le cerveau reste les serveurs. Pour disposer d'un

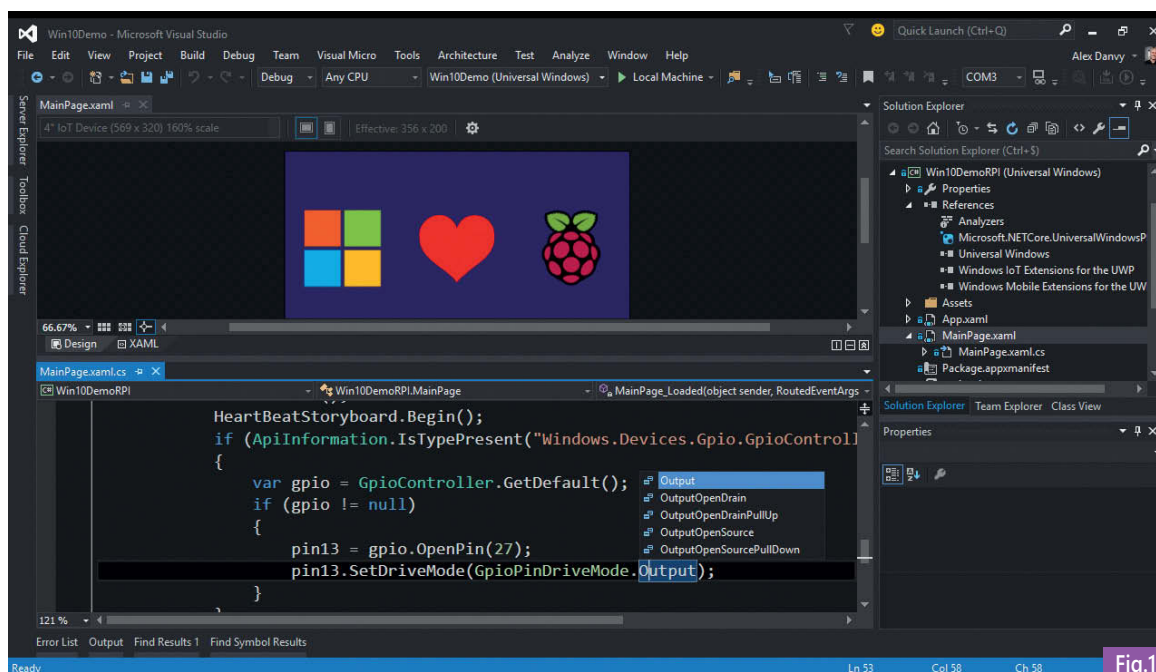


Fig.1

Conception visuelle et non visuelle pour Windows IoT Core pour Raspberry Pi 2

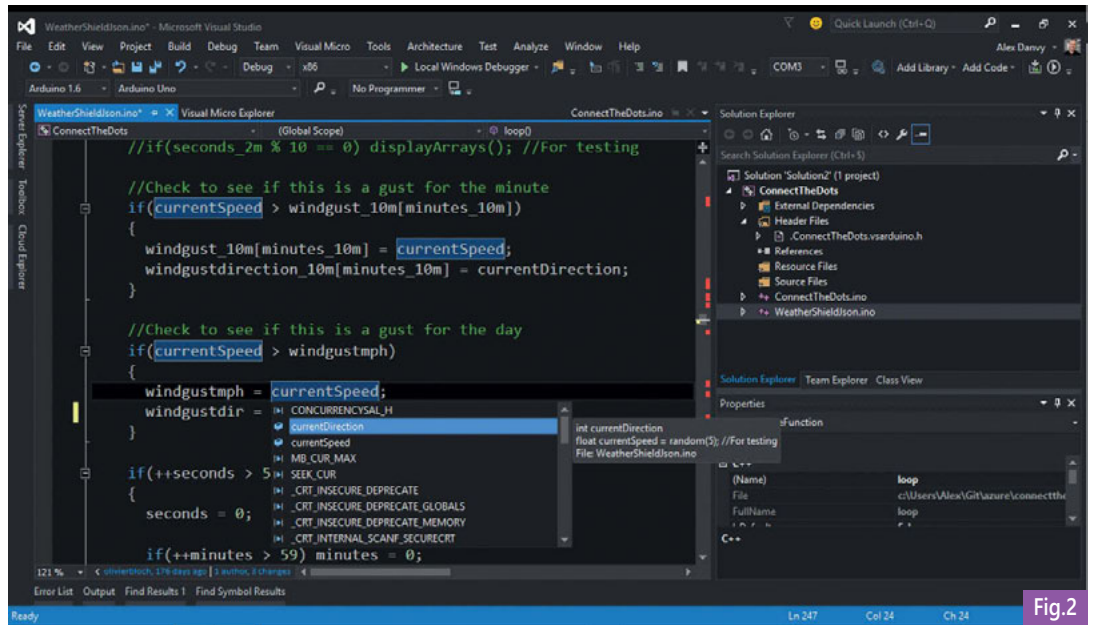
maximum de puissance à un coût très modeste, il faut s'orienter vers le Cloud. Des services spécialisés tels que Azure IoT Services permettent de surmonter sans effort les problématiques auxquelles l'IoT doit faire face :

- ▶ Sécurité
- ▶ Continuité de service
- ▶ Mise à l'échelle
- ▶ Enregistrement et suivi des appareils
- ▶ Ingestion massive de données
- ▶ Calculs en temps réel
- ▶ Stockage
- ▶ Scénarios intelligents, machine learning
- ▶ Business Intelligence

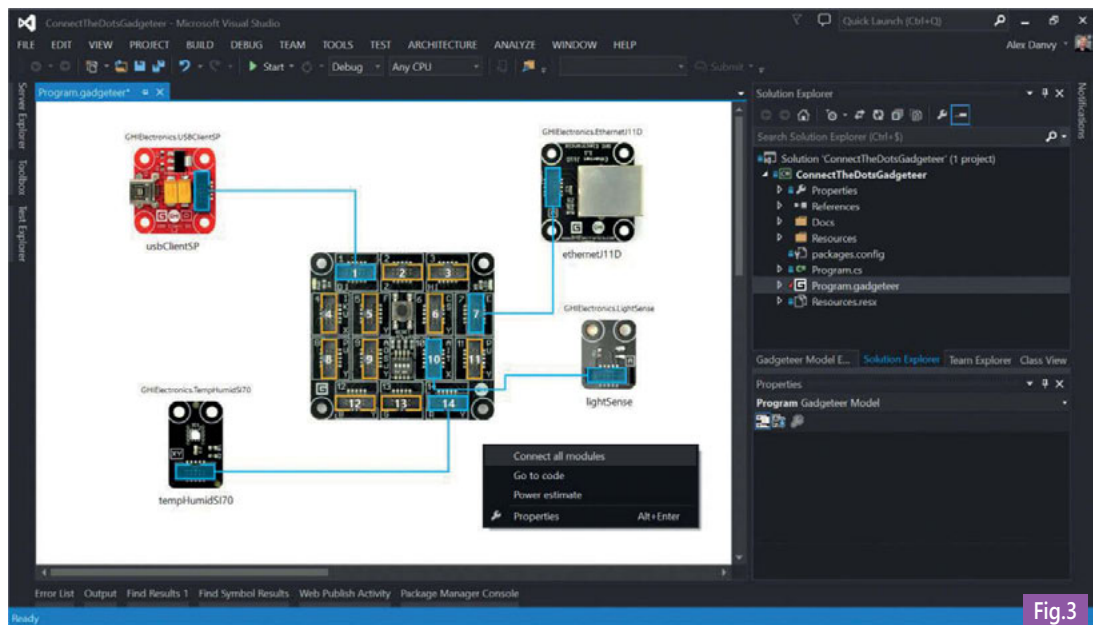
Visual Studio 2015 permet de développer et de déployer une solution Cloud IoT très complète simplement, sans ligne de commande complexe, directement depuis l'éditeur. (Fig. 4)

Solutions connectées

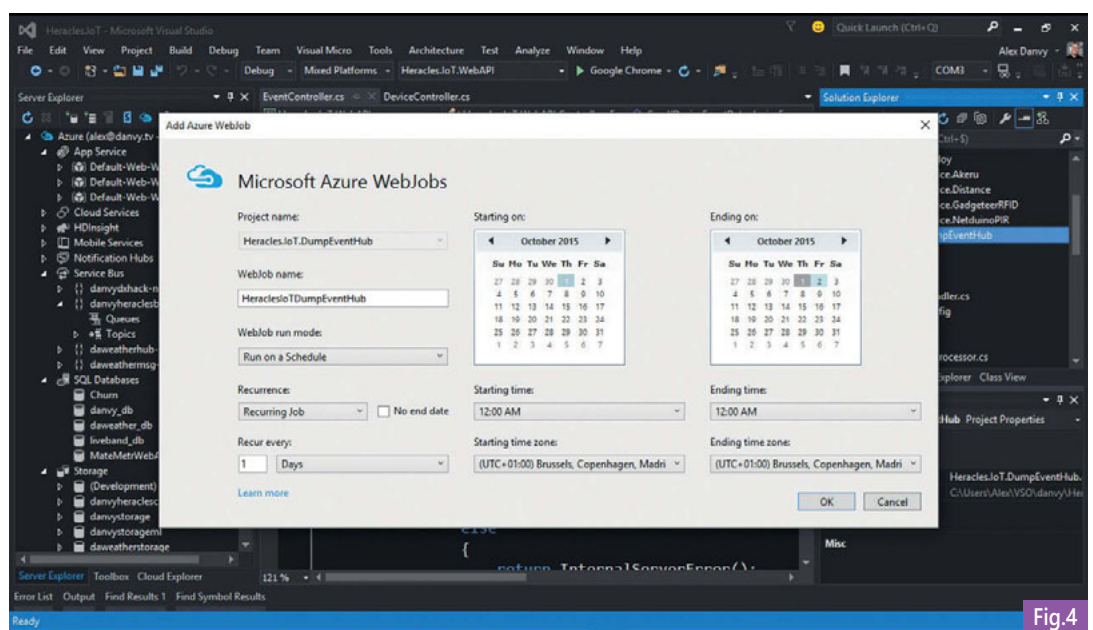
Une solution connectée implique la mise en œuvre de projets très différents, qu'il s'agisse d'un backend dans le Cloud ou « On Premise », d'un frontend web pour le pilotage, ou de code à déployer sur les objets. Visual Studio 2015 permet d'intégrer l'ensemble de ces projets au sein d'une même solution. Vous bénéficierez ainsi d'une productivité maximale : l'environnement que vous connaissez, une vision globale de la solution logicielle, une méthodologie unique, un seul dépôt de source et une seule chaîne de production. Vous pouvez même déboguer votre code pas à pas et suivre le cheminement de la donnée de l'objet vers le Cloud !



Edition d'un projet Arduino dans Visual Studio grâce à VisualMicro



Conception visuelle de circuits Gadgeteer



Déploiement d'une unité de traitement programmée (WebJob)

MSDN Platforms

L'abonnement MSDN Platforms fait partie des offres MSDN, il propose un accès à tous vos environnements hors production. Il n'inclut pas l'outil de développement Visual Studio. Pourtant, il est celui dont vous avez peut-être le plus besoin pour le développement de vos applications, encore plus si vous adoptez une démarche de type DevOps.



Cédric Leblond
Consultant **Meritis**
Accompagne les équipes sur ses
sujets préférés
DevOps et Agilité.
Est aussi
MVP ALM.



Il s'adresse potentiellement à toutes les parties prenantes dans le cycle de vie de vos applications : Ops, administrateur IT, DBA, chef de projet, manager, ScrumMaster, Product Owner...
Il s'adresse également aux développeurs qui codent avec d'autres outils de développement que Visual Studio. Ils bénéficient alors eux aussi de l'accès à tous les environnements hors production, outils et services décrits dans cet article.

Des nombreuses ressources

Cet abonnement offre un accès complet à la plateforme de collaboration ALM de Microsoft, qu'elle soit mise en place par vos soins Team Foundation Server (TFS) ou hébergée Visual Studio Online (VSO). Le téléchargement, et surtout l'utilisation pour le développement et test sont inclus pour tous les environnements et serveurs Microsoft avec en plus des classiques Windows et SQL Server : SharePoint, Exchange, Power BI, Dynamics ... Votre abonnement MSDN Platforms vous permettra également de bénéficier automatiquement des derniers correctifs et versions des logiciels dès leur disponibilité.

L'abonnement propose aussi de l'aide en cas de problème technique. Vous êtes prioritaires sur les forums MSDN et également sur les forums Technet. Si le problème est important, vous pouvez discuter en ligne sur le MSDN chat, ou encore demander de l'aide à Microsoft via les 2 tickets de support technique inclus dans chaque abonnement MSDN. L'abonnement vous aide aussi à maintenir vos connaissances techniques en incluant 2 cours par an du catalogue

Microsoft e-learning et 30 cours Pluralsight à votre disposition pendant 1 an.
J'ai récemment accompagné une équipe dans sa démarche DevOps. Cette équipe agile développe une solution web avec un Backend basé sur .Net, SharePoint et SQL Server, et le Front sur HTML5/Javascript avec AngularJS. Elle est composée de 3 développeurs .Net, 1 expert SQL, 1 Ops (administrateur SharePoint/Windows), 2 développeurs AngularJS, 2 spécialistes du test, 1 ScrumMaster et 1 Product Owner. Les développeurs .Net utilisent principalement Visual Studio 2015, ils s'orientent vers un abonnement MSDN Visual Studio Enterprise. L'expert SQL et les développeurs AngularJS utilisent d'autres outils pour coder, ils accèdent eux aussi à SQL Server et SharePoint pour leurs développements : un abonnement MSDN Platforms, par personne, le permet. Les spécialistes du test travaillent avec Microsoft Test Manager, ils accèdent aussi à SharePoint : MSDN Test Professional répond à ce besoin. Pour la gestion du backlog, du travail, du code source et déploiement continu, l'équipe exploite les fonctionnalités de TFS. Le ScrumMaster et le ProductOwner doivent à minima acquérir une CAL. Ils aident l'équipe et surtout demandent des feed-back aux parties prenantes, un abonnement MSDN Platforms est alors approprié.

Avec 85 €/mois de crédit Azure offert, chacun bénéficie en outre d'environnements performants de développement et test.

Bien sûr, un abonnement MSDN n'est pas systématiquement nécessaire. Si un manager suit l'avancement de l'équipe via des comptes-rendus créés à partir des données TFS, cela est gratuit. Les utilisateurs finaux peuvent utiliser les environnements et applications développées dans le cadre de tests

MSDN PLATFORMS

Outils

Team Foundation Server + CAL
(accès à toutes les fonctionnalités)

Visual Studio Online Advanced Plan
(accès à toutes les fonctionnalités)

Crédit Azure gratuit 85€/mois

Produits en téléchargement pour le développement et le test

Windows, Windows Server,
Windows embedded

SQL Server

SharePoint, Exchange, Dynamics

Power BI Pro

Et les autres
(liste complète <http://bit.ly/ProdMSDNAbos>)

Formation

30 cours Pluralsight sur 12 mois
(liste <http://bit.ly/Plural30MSDN>)

2 cours par an dans les collections
Microsoft E-Learning

MSDN Magazine

Support

2 tickets sur incidents techniques

Priorité sur les forums MSDN et Technet

Discussion en ligne MSDN

Offres exclusives

-30% sur l'abonnement annuel Pluralsight

-50% sur le service UserVoice la
première année

Les offres changent régulièrement
(liste <http://bit.ly/OffresMSDN>)

Juillet 2015

d'acceptation (UATs) sans licence particulière. Ils peuvent aussi créer et suivre des bogues ou évolutions grâce au niveau d'accès Stakeholder gratuit pour TFS 2015 ou VSO.

En résumé

Les possibilités offertes par MSDN Platforms sont multiples et riches. Pour en tirer le maximum, je vous invite à consulter régulièrement la page de votre compte, tous les bénéfices y sont regroupés et mis à jour régulièrement. ▲

<http://www.visualstudio.fr> : le site de l'équipe France. Retrouvez toutes les informations produits, livres blancs en Français et l'actualité de Visual Studio et

MSDN. Si vous avez un projet ou une question, contactez l'équipe marketing et vente au 01 45 36 77 17 ou msdnfr@microsoft.com

La liste de nos partenaires revendeurs et services est aussi disponible sur ce site.



Microsoft Virtual Academy :

Ceux qui réussissent ne cessent jamais d'apprendre et les

technologies majeures ne cessent jamais d'évoluer. Microsoft Virtual Academy (MVA) propose des formations Microsoft en ligne entièrement gratuites, menées par des experts dans 12 langues différentes. Nous vous recommandons notamment le cours suivant : <http://aka.ms/mvavs>



L'assistant Visual Studio :

en ajoutant un assistant à Visual Studio vous pourrez suivre des tutoriaux interactifs, vous pourrez lancer tous les pas-à-pas créés par la communauté. L'Assistant Visual Studio est un plugin additionnel à Visual Studio qui permet de lire des cours directement dans l'IDE tout en permettant d'interagir avec celui-ci pour effectuer des tâches comme Ouvrir un fichier, Ouvrir une solution, Remplacer du texte, Ajouter un item dans un projet, Build une solution... et bien d'autres !

<http://aka.ms/assistantVS>

Developer code samples :

Besoin d'aide pour utiliser une nouveauté du framework ou encore débiter un nouveau projet ? Quoi de mieux qu'un bon exemple

pour découvrir comment utiliser la fonctionnalité attendue ? C'est ce que propose ce site : un ensemble d'exemples sur des points précis qui vous permettront de progresser et d'appréhender facilement les différents aspects du langage.

<http://code.msdn.microsoft.com/>



Channel 9 :

Ce site regroupe les vidéos des personnes qui conçoivent les produits chez Microsoft. Il est également un lieu d'échange autour des technologies. Les nombreux événements Microsoft qui

punctuent l'année sont pour la plupart disponibles sur ce site, notamment l'événement Connect() et l'évènement de lancement mondial de Visual Studio 2015.

<http://channel9.msdn.com>

Le site **MSDN** (Microsoft Developer Network) réunit l'ensemble des ressources techniques et des services dont le développeur sur technologies Microsoft a besoin. Vous pouvez également participer aux discussions dans les Forums MSDN, rejoindre les communautés de développeurs et rencontrer les experts Microsoft France. Inscrivez-vous sur ce site à la newsletter MSDN. Elle propose le meilleur des informations techniques en provenance des équipes de développement de Microsoft France et de Microsoft Corp, mais aussi des communautés francophones.

<http://msdn.microsoft.com/>

Visual Studio Community :

un IDE complet gratuit, des outils de développement mobiles cross-plateformes pour Windows, iOS et Android, et l'accès à des milliers d'extensions. Cette édition de Visual Studio est disponible gratuitement pour les étudiants, les contributeurs open source, les start-ups, les développeurs indépendants et sous certaines conditions les PME. Pour télécharger cette édition :

<http://aka.ms/VScommunityFR>



Essayez Microsoft Azure gratuitement pendant un mois :

inscrivez-vous et obtenez 170 € à utiliser sur tous les services Azure : machines virtuelles, bases de données SQL, sites Web, Push notifications sur mobile, diffusion en continu de médias, Active Directory...

Tous les détails sur <http://azure.com>



Pépinière Microsoft Azure et Windows

Bénéficiez d'un programme d'accompagnement personnalisé et gratuit, de la mise à disposition de ressources techniques à un coaching technique et du conseil business, pour vous aider dans le développement de vos applications Windows ou Cloud. Plus d'informations sur ce lien : <http://www.pepinieremicrosoft.com>



BizSpark :

est un programme réservé aux start-ups

de logiciels : sociétés de moins de 3 ans, moins de 1 million de \$ de CA et dont l'activité principale est la création d'une solution logicielle on-premise ou dans le Cloud. Les sociétés membres de BizSpark bénéficient de licences de développement via un accès à MSDN Enterprise, du support, 130 € par mois de crédits Azure pour 5 associés pendant 3 ans. Plus d'informations : <http://www.microsoft.com/bizspark/>

L'Accélérateur Microsoft Ventures à Paris :

Un des 7 accélérateurs start up de Microsoft dans le monde, notre programme de 3 mois propose un vrai tremplin pour votre entreprise. Nous travaillons notamment avec des startups à fort potentiel international sur le développement de leur produit, business et aussi le financement de leur entreprise.

Pour plus d'informations, rendez-vous sur microsoftventures.com/paris



Événements

Chaque mois plusieurs événements gratuits vous sont proposés pour découvrir nos outils <http://aka.ms/vseventsfr>

