



Introduction aux Applications Blocks

18/08/2004

Par Elise Dupont elise.dupont@europe.com

Droit de diffusion:

L'ensemble ou partie de ce document ainsi que le code mis à disposition, ne peut être diffusé sur d'autres sites Web sans l'autorisation au préalable de son créateur.

Avant Propos :

Quand on commence à développer plus de 2 applications .Net, on se rend vite compte qu'il existe des parties génériques à toute application qui peuvent être centralisées. Certains peuvent partir dans l'optique de développer leur propre framework ou encore socle technique, mais il existe aussi d'autres solutions.

Microsoft a développé et testé des composants que tout le monde peut utiliser dans ses applications, et qui répondent à des besoins fréquents. Ces briques appelées Application Block évitent d'avoir à développer soi même et à tester (ce qui est long et coûteux) une solution qui répond de façon générique à un problème donné. Il suffit de télécharger l'application block et d'ajouter une référence dans votre projet .Net afin de pouvoir profiter des fonctionnalités offertes.

Ces applications blocks sont des concrétisations de toutes les recommandations données par Microsoft en terme d'architecture et de solution technique. On peut voir une relation directe ici <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/daab-rm.asp> (ainsi que la liste de toutes les recommandations) On peut en effet constater que pour la majorité des recommandations d'architectures faites par Microsoft, correspond un application block.

Je vais expliquer chacun de ces applications blocks, et donner les liens correspondants, chaque application block est correctement documenté et garni d'exemples concrets.

C'est parti !

Sommaire:

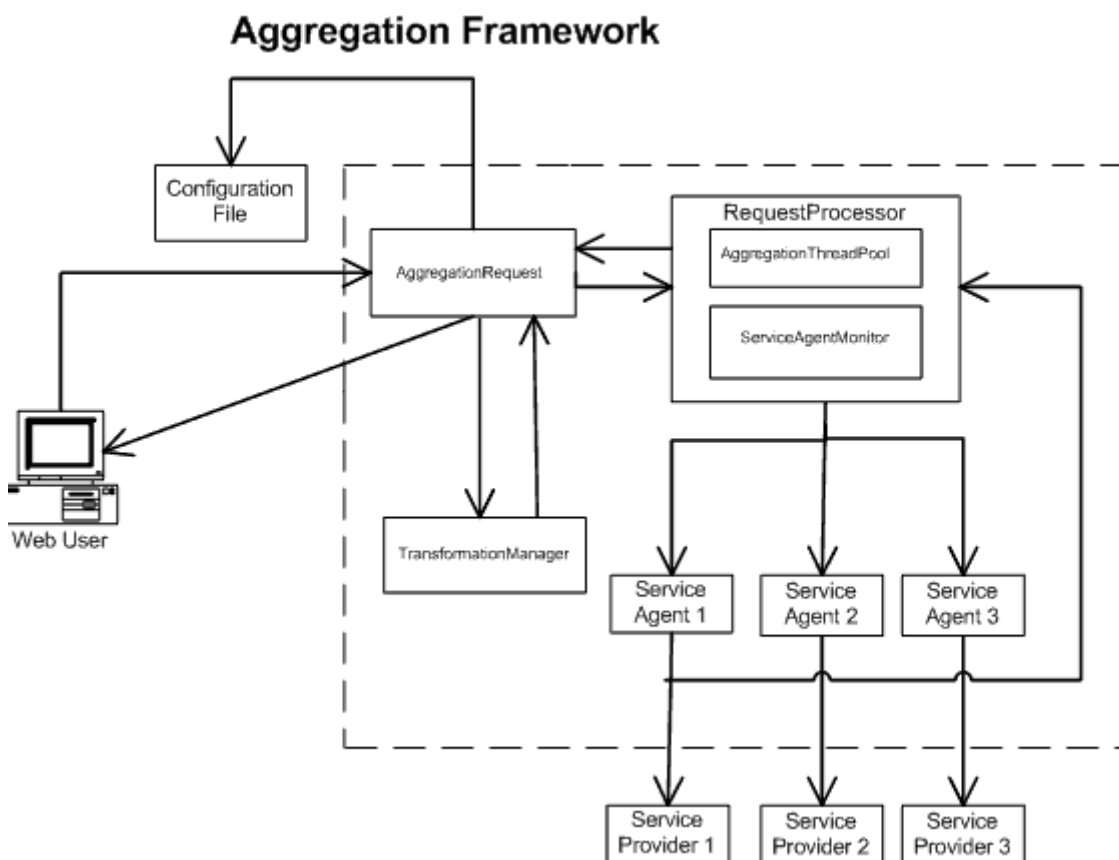
1. Aggregation Application Block
2. Asynchronous Application Block
3. Caching Application Block
4. Configuration Management Application Block
5. Data Access Application Block
6. Exception Management Application Block
7. Logging Application Block
8. Updater Application Block
9. User Interface Process Application Block
10. Conclusion

1. Aggregation Application Block

Cette brique vous permet d'agréger différentes sources de données (par exemple une base SQL serveur et un web service) afin de pouvoir présenter de façon commune ces informations.

L' Aggregation Application Block se charge d'envoyer les différentes requêtes à chaque source de données et de transformer le résultat dans un seul et unique fichier XML qui pourra par la suite être exploité par votre application.

Microsoft fournit un schéma qui résume ce composant :



Lien utile :

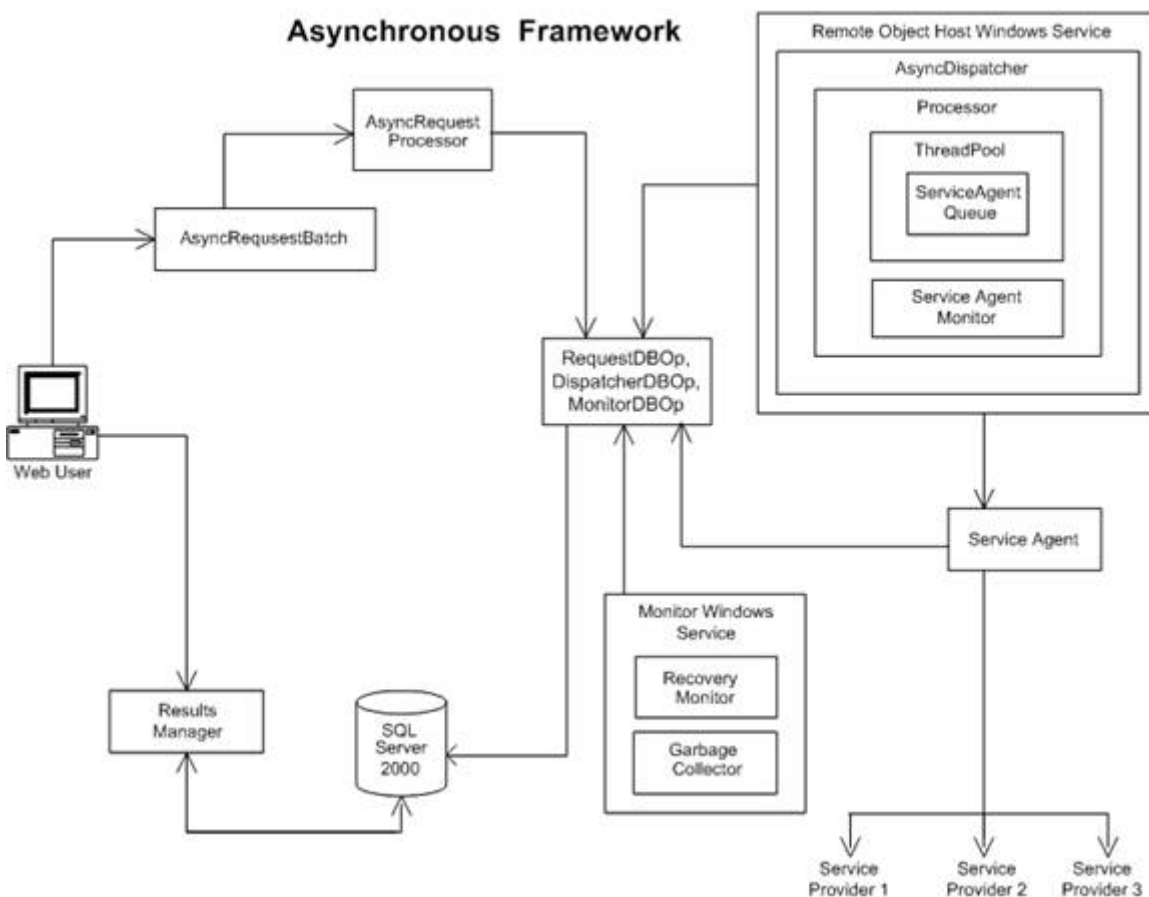
- Vous pourrez trouver plus d'informations ici :

2. Asynchronous Application Block

Cette brique vous permet d'améliorer les performances de vos applications asp.net si celles-ci font appel à divers services ou font appel à des services dont le temps de réponse dépend de la qualité de la connexion. En effet en général le mode par défaut est le mode dit synchrone, qui par défaut attend la fin d'exécution d'une ressource avant de passer à la suite des opérations.

Le mode asynchrone lui est bien connu pour pouvoir d'effectuer une opération, de ne pas attendre la fin, de pouvoir continuer ses traitements et d'être prévenu quand l'opération demandée a été exécutée. En mode asynchrone on peut donc permettre d'optimiser le temps d'attente pour le chargement d'une page web par exemple.

Microsoft fournit un schéma qui résume ce composant :



Lien utile :

- Pour plus d'information sur cet application block c'est par ici : <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag/html/PAIBlock.asp>

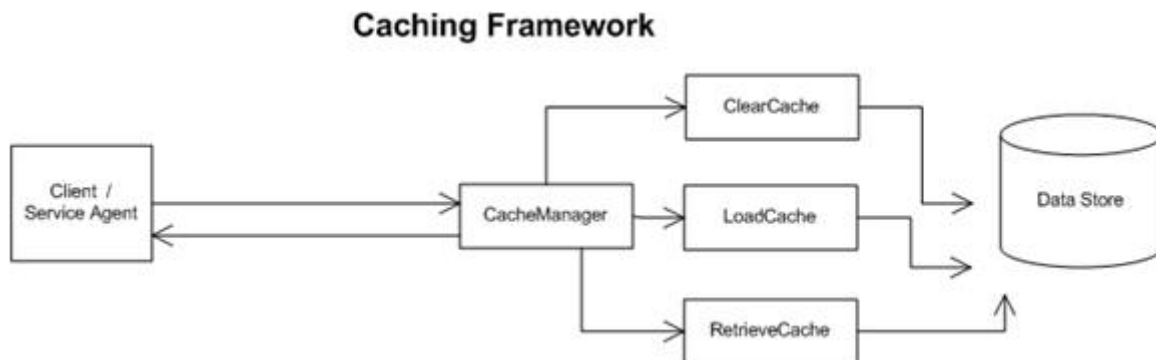
3. Caching Application Block

Cet application block vous facilite la vie en matière de gestion de caches. Le cache peut concerner SQL Serveur, un objet, un fichier chargé en mémoire etc...

Cet application block gère sans problème tout ce qui concerne l'expiration, l'accès, et la protection des données mises en cache.

Il existe cependant encore une petite faille dans cette brique (dans les scénarios d'applications multi thread notamment), mais les équipes de Microsoft travaillent sur le sujet pour perfectionner ce block.

Microsoft fournit un schéma qui résume ce composant :



Lien utile :

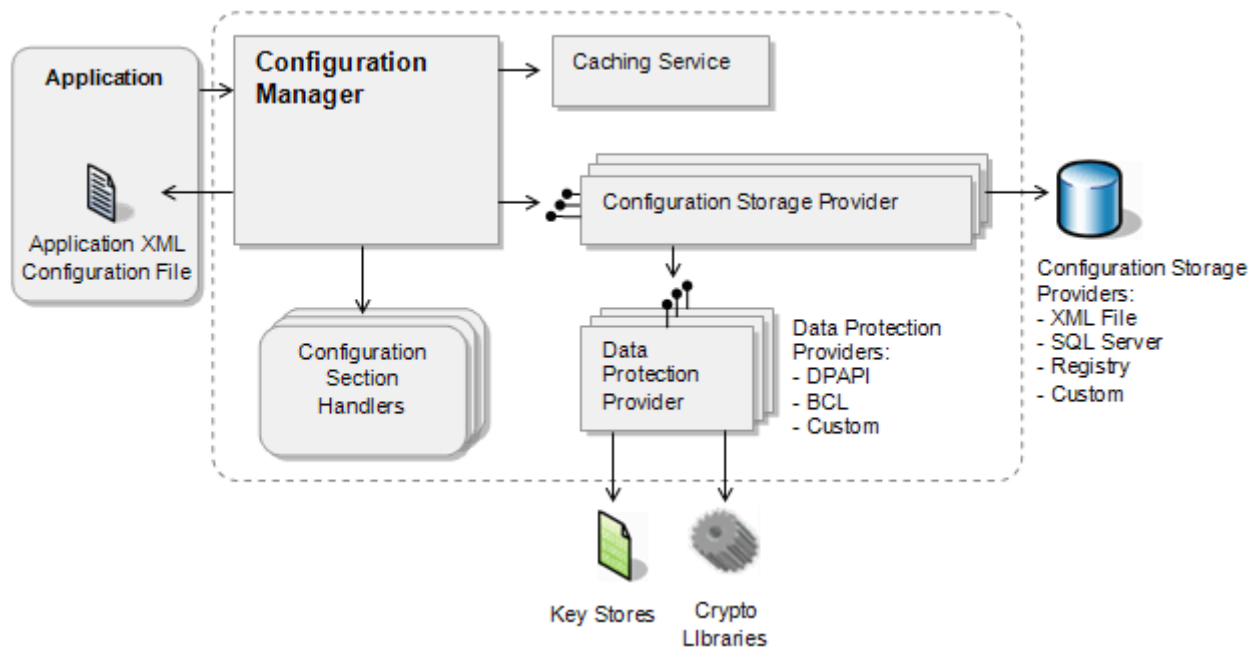
- Les informations détaillées sont disponibles ici : <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag/html/CachingBlock.asp>

4. Configuration Management Application Block

Si vous trouvez que stocker les configurations dans un fichier XML est un moyen bien trop limitatif pour votre application, ce block applicatif est fait pour vous.

Il permet en effet de gérer de façon transparente et indépendante des informations de configurations provenant de diverses sources : XML, Base de données, base de registre...

Microsoft fournit un schéma qui résume ce composant :



Lien utile :

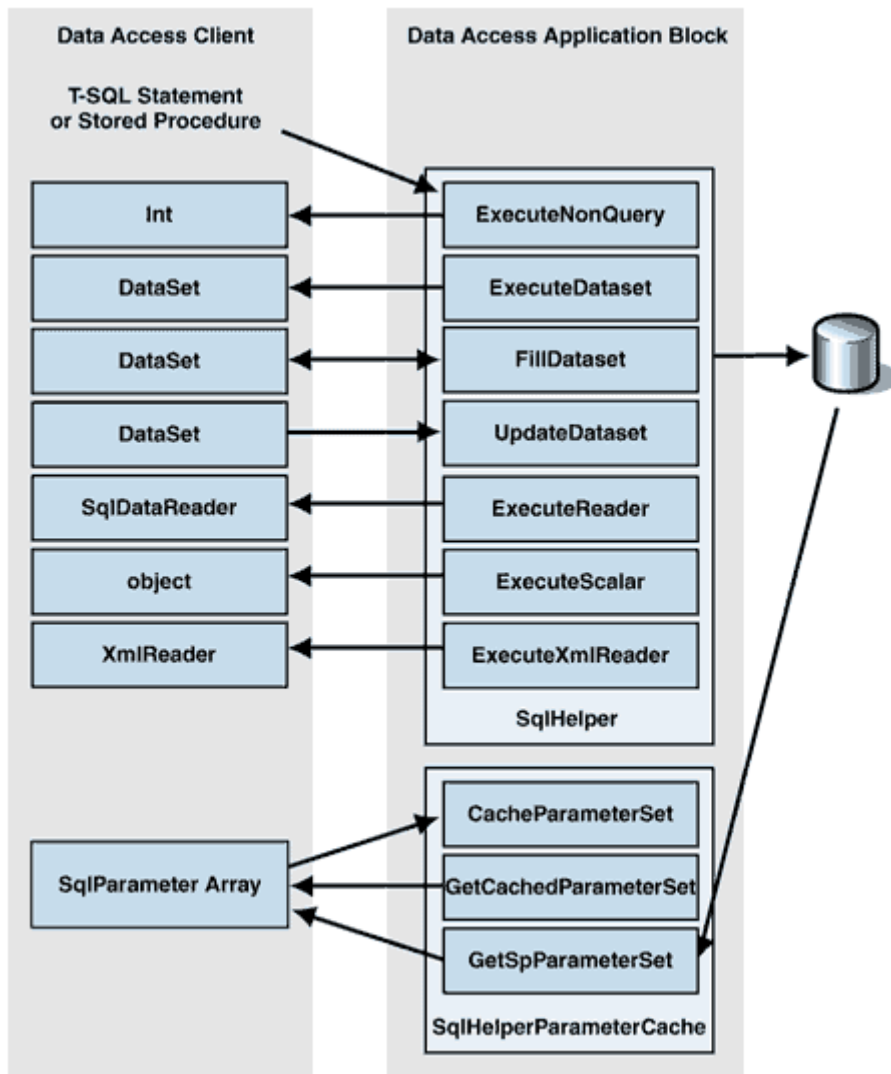
- Plus d'infos ici : <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/cmab.asp>

5. Data Access Application Block

Pour ceux qui sont fatigués de toujours écrire le même code pour accéder aux données, ce block est fait pour vous : il permet de centraliser de façon générique les principales méthodes d'accès aux données telles qu'exécuter une procédure stockée avec des paramètres, récupérer un DataSet ou un DataReader à partir d'une requête SQL...

De façon très simple, on peut par exemple créer une table de paramètres SQL et fournir ce tableau avec le nom de la procédure stockée concernée.

Microsoft fournit un schéma qui résume ce composant :



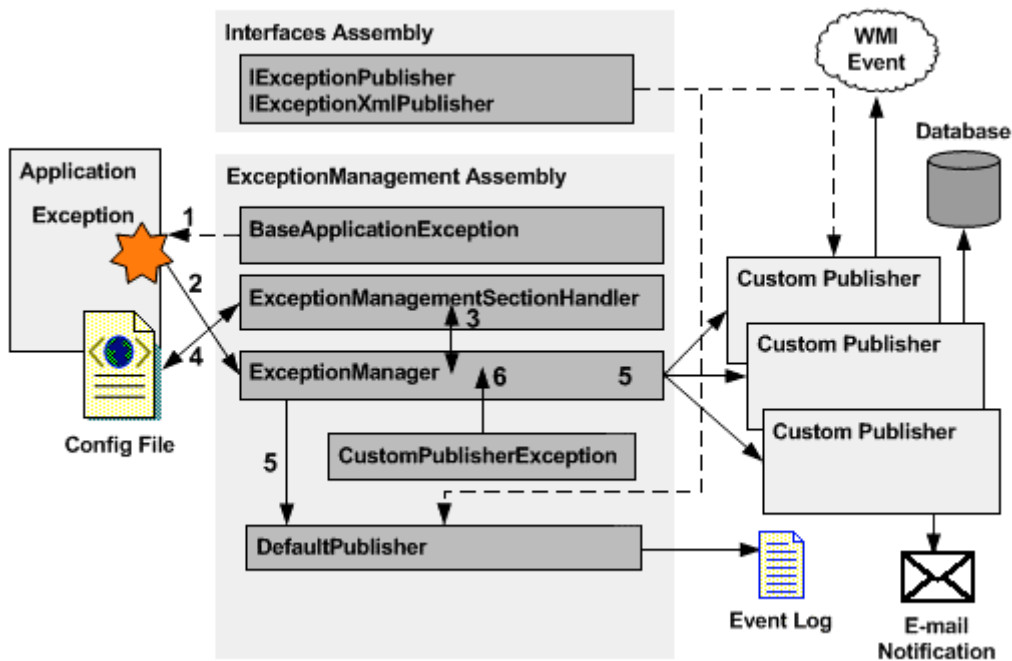
Lien utile :

- Ca se passe ici : <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/daab-rm.asp>

6. Exception Management Application Block

Cet application block fournit des méthodes pour gérer de façon très simple les exceptions dans vos applications avec le moins de code possible, et de la façon la plus conseillée et la plus propre qu'il soit. Je vous conseille d'aller y jeter un coup d'œil, vous y apprendrez à coup sur plein de choses en matières d'exception, un domaine ou on a tendance à ne pas vraiment pousser loin la compréhension.

Microsoft fournit un schéma qui résume ce composant :



Lien utile :

- Pour plus d'informations : <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/emab-rm.asp>

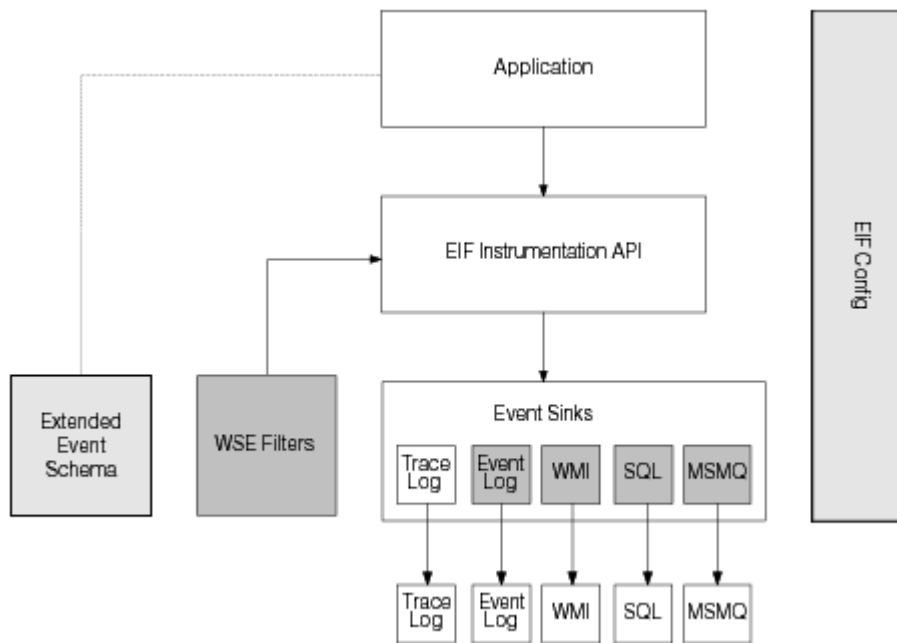
7. Logging Application Block

Cet application block peut de façon tout à fait logique bien se compléter avec l' Exception Management Application Block puisqu'une fois les erreurs remontées, encore faut-il les tracer correctement. Cette brique applicative vous permet donc de stocker les informations les plus précieuses en cas d'erreur (comme l'heure, la source de l'évènement, l'ID, de nom du compte sous lequel a été exécuté l'application etc...).

Le stockage se fait de manière très simple sur différents formats sans douleur.

A noter que cet application block a besoin du Enterprise Instrumentation Framework (EIF) afin de fonctionner.

Microsoft fournit un schéma qui résume ce composant :



Liens utiles :

- Logging application block : <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag/html/logging.asp>
- EIF : <http://go.microsoft.com/fwlink/?LinkId=20558&clcid=0x409>

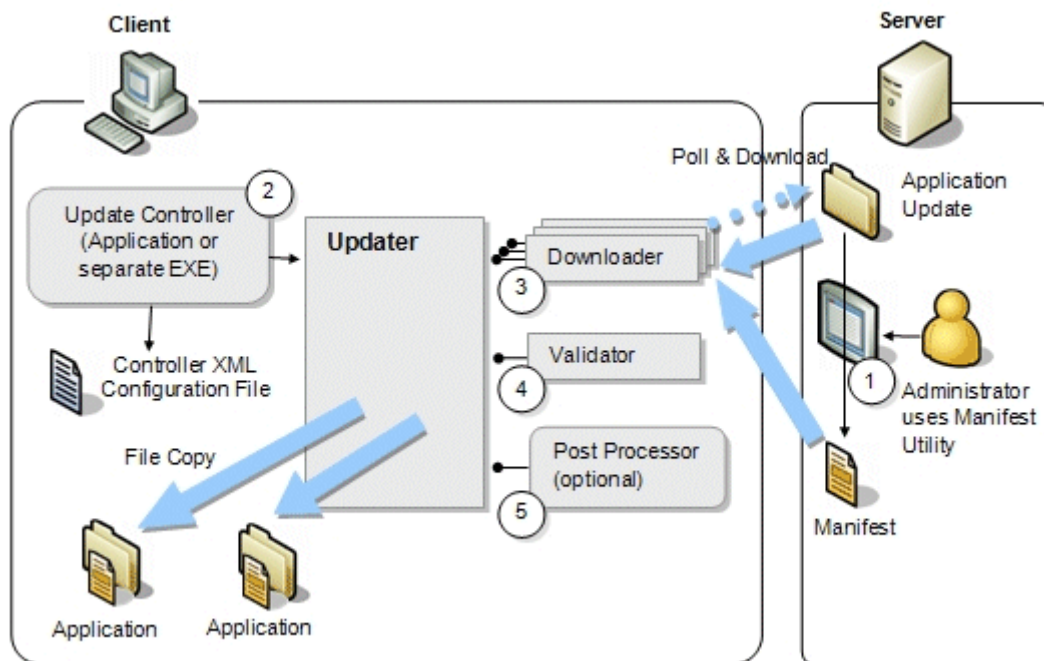
8. Updater Application Block

Pour tous ceux qui développent des applications Windows et qui connaissent donc la problématique de la mise à jour des versions clients avec un moindre coût de déploiement, le principe de l'updater est parfois la solution idéale (surtout pour ceux qui n'ont pas les moyens de mettre un place un système basé sur SMS).

L'update application block permet d'obtenir une solution simple qui « poussera » les dernières versions vers les postes clients.

Cette brique vous permet donc de détecter, valider et télécharger simplement les dernières versions de vos applications, librairies et j'en passe.

Microsoft fournit un schéma qui résume ce composant :



Lien utile :

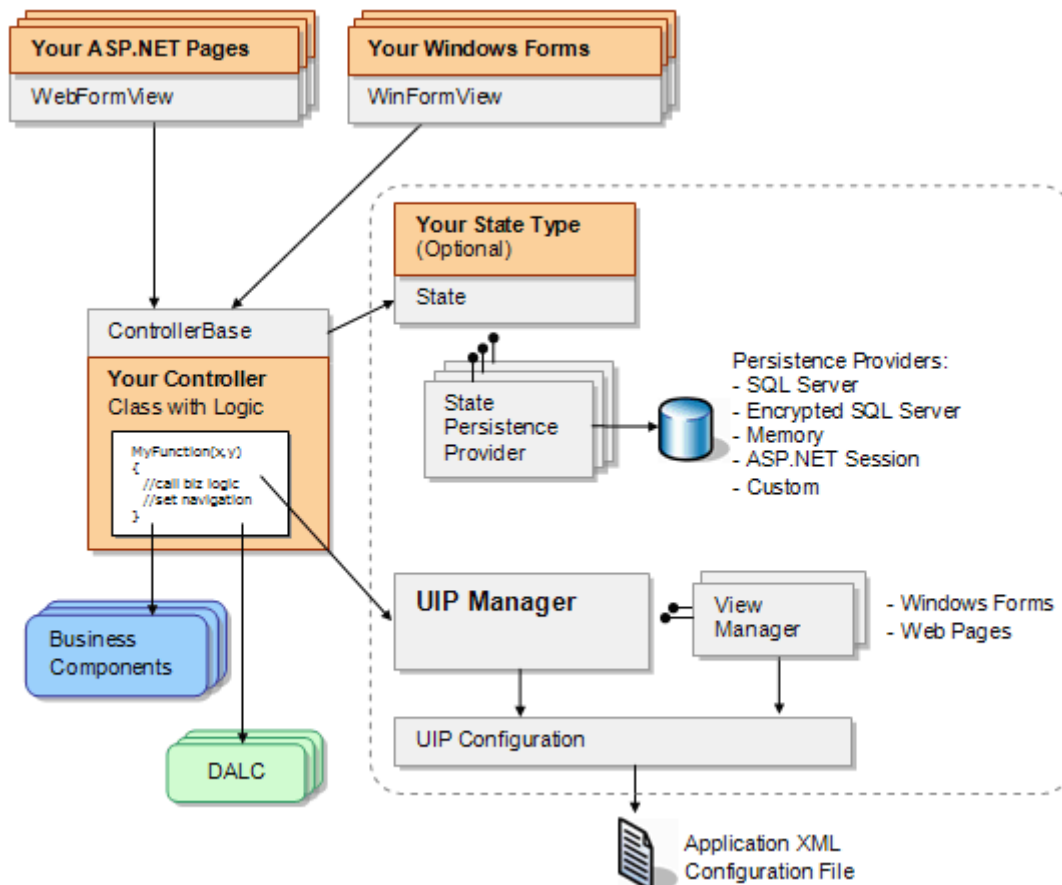
- Pour les curieux, les informations utiles se trouvent ici : <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/updater.asp>

9. User Interface Process Application Block

Ce block réponds à un besoin qui peut d'avérer très complexe à développer soi-même : ce qu'on appelle les User Interface Process. Ce terme un peu barbare désigne le processus que l'on retrouve parfois dans certaines applications qui permettent d'interrompre un processus en cours (avec un bouton du genre « retourner plus tard ») pour ensuite pouvoir reprendre l'opération plus tard avec un bouton du genre « Reprendre l'action ». En gros toute application qui a besoin de gérer des « états » au niveau de l'interface connaît en général ce genre de problématique.

Ce block pour donne des fonctions génériques qui vous permettent de gérer les états et le flux de vos « transactions » et sessions au niveau des interfaces.

Microsoft fournit un schéma qui résume ce composant :



Lien utile :

- Ca a l'air un peu barbare mais cela s'avère vraiment très pratique : <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/uiip.asp>

10. Conclusion

Pour conclure, je ne donnerai qu'un seul conseil : regarder de plus près tout ceci grâce aux nombreux liens que j'ai donné, cela vous fera gagner beaucoup de temps dans vos futures applications, surtout si vos développements ont une envergure de "développement industriel". Pour ceux qui trouvent très léger les solutions apportées par les Application Blocks, de part votre logique métier particulière ou des développements très spécifiques, dans ce cas je ne peux que vous conseiller de vous lancer dans le développement de votre propre socle technique si vous ne l'avez déjà fait.



L'ensemble ou partie de ce document ainsi que le code mis à disposition, ne peut être diffusé ailleurs sans autorisation préalable