



Developing Agility

Lettre d'information trimestrielle destinée aux clients EAE et AB Suite de Unisys

Table des matières

- Page 2 **Modernisation des applications EAE grâce à l'outil ASP.NET Generator Client Tool**
Débarrassez-vous des écrans verts en suivant les cinq étapes des meilleures pratiques proposées par George McGowan pour développer des interfaces utilisateur graphiques pour navigateur web
- Page 4 **Le coin des ingénieurs : les multiples facettes (inter-utilisateurs) des applications EAE et AB Suite**
John Papachristos (ACUS) passe en revue des approches avancées pour la mise à niveau des clients ASP.NET générés.
- Page 9 **Nouvelles options d'interface utilisateur graphique : les générateurs personnalisés**
Examen rapide des options supplémentaires susceptibles d'améliorer la présentation à l'utilisateur final grâce à des programmes générateurs personnalisés, disponibles à partir d'une variété de sources.
- Page 11 **Comment obtenir une architecture orientée service (SOA) intelligente**
Quatre livres blancs et une étude de cas offrent de nouvelles perspectives sur les contributions des applications ClearPath à l'élaboration d'une architecture orientée service (SOA).
- Lien vers **ClearPath Connection : avez-vous déjà lu cette lettre d'information ?**
Consultez un numéro récent de cette lettre d'information électronique trimestrielle et restez au fait des derniers développements liés aux produits, services et événements ClearPath. Pour plus de détails, consultez la section des nouvelles de eCommunity.
- Lien vers **Les étudiants lettons mis à l'honneur pour l'acquisition de compétences AB Suite**
Le cru 2008 inclut 23 étudiants ayant étudié les fondements de l'application AB Suite comme l'une des matières principales du tronc commun de leur programme informatique. Reportez-vous à l'article paru dans eCommunity.
- Page 11 **Calendrier**
Consultez notre calendrier pour obtenir les informations relatives aux événements à venir.

Modernisation des applications EAE grâce à l'outil ASP.NET Generator Client Tool

Par George E. McGowan, Jr. CPA, fondateur et président de McGowan Computer Associates

Le coût réel de la modernisation

La modernisation est un sujet d'actualité dans bon nombre de structures informatiques. Même si les applications répondent parfaitement aux exigences organisationnelles et commerciales de l'entreprise, la présence d' interfaces utilisateur de type « écran vert » renvoie un message indiquant que le système est archaïque et requiert d'être remplacé d'urgence.

Nous avons vu un grand nombre d'entreprises relever le défi de remplacer leurs applications EAE (Enterprise Application Environment) développée sur mesure par une solution Package du marché. Toutefois, la probabilité de pouvoir déployer une solution du marché offrant une fonctionnalité équivalente restent toutefois très minces. De plus, seules quelques entreprises seulement parviennent à assimiler parfaitement l'ensemble des règles de gestion contenues dans les applications EAE, car celles-ci ont évolué avec le temps pour répondre aux changements constants tant au niveau des entreprises que du marché.

Le prix à payer pour acquérir de nouveaux logiciels cœur de métier pour les entreprises est élevé : sur le marché américain, les offres commencent généralement à 2 millions de dollars. Il est aussi très courant de sous-estimer le coût de la migration. La mise en correspondance des données (mapping) et leur déplacement d'un système à un autre est chose facile. En revanche, les coûts les plus importants concernent la formation des utilisateurs, la qualification du nouveau logiciel, la reconstruction des interfaces, la conduite d'essais de performances sérieux portant sur les charges de travail des transactions mixtes ainsi que l'élaboration de la fonctionnalité « manquante ». Les entreprises sous-estiment toujours les efforts à déployer pour tester le nouveau système adopté et assurer la formation de TOUS les utilisateurs de l'application.

Le développement d'une nouvelle application personnalisée n'est pas forcément une option possible pour les structures informatiques actuelles : les ressources y sont rares et la liste des maintenances en attente s'allonge tous les jours un peu plus.

En fin de compte, la valeur d'une application EAE développée sur mesure n'a presque pas de prix.

C'est la présentation et non l'application qui fait la différence

Il n'y a aucune raison d'abandonner une application EAE lorsque le problème se situe au niveau de la présentation. Vous pouvez facilement protéger et préserver (tout en continuant à affiner et étendre) les règles de gestion en améliorant l'expérience de l'utilisateur final grâce à Client Tools, et particulièrement au programme générateur ASP.NET Generator. ASP.NET Generator vous aide à créer une interface utilisateur graphique pour un navigateur moderne et dotée de fonctions susceptibles de satisfaire les exigences des travailleurs au savoir les plus avisés en matière d'Internet.

Les formats d'écran ASP.NET fonctionnent remarquablement bien, mieux que les pages ASP que bon nombre d'utilisateurs EAE ont pu utiliser les années passées. ASP.NET permet d'universaliser les pages Web en les rendant accessibles à tous les types de navigateurs et de version ainsi qu'avec l'outil Developer Test. D'un point de vue de support technique pur, il n'existe pas non plus de logiciel client à déployer sur les stations de travail. Une fois que l'entreprise adopte ASP.NET, elle a le choix d'exposer à ses utilisateurs externes via Internet l'ensemble de ses applications, ou une partie seulement de celles-ci.

Meilleures pratiques pour l'élaboration d'interfaces utilisateur graphiques par navigateur

L'utilisation de l'outil ASP.NET Generator est simple, mais comme tout développeur d'applications EAE peut le constater, l'implication des maîtres d'ouvrage est essentielle au processus. Nous avons adopté l'approche suivante en cinq étapes sur bon nombre de projets réussis :

1. Impliquer

Former un comité de réflexion regroupant des représentants des utilisateurs finaux, depuis ceux qui assurent la saisie de données jusqu'aux utilisateurs expérimentés, afin qu'ils participent tous ensemble à l'élaboration des écrans.

2. Expérimenter

Sans une approche mesurée, les formats ASP.NET risquent de se transformer en boîte de Pandore. Travaillez avec vos utilisateurs finaux afin de définir plusieurs types d'écrans génériques comportant menus, tableaux, requêtes et formats de saisie de gros volumes de données. Ceci vous permet de tester les différentes options proposées par ASP.NET jusqu'à ce que vous réussissiez à trouver l'ensemble de fonctionnalités le mieux adapté à votre environnement.

3. Simplifier

Ce n'est pas parce que vous bénéficiez de la possibilité de transformer tous les champs en listes déroulantes et en cases d'options ou d'ajouter une profusion de couleurs et de graphiques que vous devez nécessairement le faire. Il

est fort probable que votre application EAE soit un système vital pour votre entreprise. Par conséquent, évitez les fioritures et les artifices. Une fois la phase d'expérimentation terminée, prenez du recul pour évaluer la valeur ajoutée réelle offerte par les nouvelles options d'interfaces utilisateur graphiques et simplifiez votre approche pour y inclure les options qui répondent véritablement à vos besoins.

4. Normaliser

La majorité des utilisateurs finaux sont tout à fait disposés à adopter une interface utilisateur graphique facile à manier. La courbe d'apprentissage est minimisée lorsqu'une touche de cohérence, de fiabilité et d'intuitivité est appliquée à l'ensemble des formats d'écran ASP.NET. C'est pour cette raison que nous vous recommandons de définir des normes relatives aux interfaces utilisateur graphiques (charte graphique) à l'issue des étapes Expérimenter et Simplifier. Quant aux utilisateurs qui résistent au changement, il faut prendre le temps de travailler avec eux et essayer de répondre à leurs besoins. Le passage d'un émulateur de terminal à une application basée sur un navigateur peut s'avérer être quelque peu traumatisante pour les utilisateurs existants. Par ailleurs, les nouveaux utilisateurs *s'attendent* forcément à une application basée sur un navigateur.

5. Moderniser

Grâce à une approche bien pensée telle que celle qui est esquissée ci-dessus, vous bénéficiez des connaissances, des normes et de l'aval des utilisateurs finaux nécessaires pour réussir la mise à jour de la couche présentation de votre application EAE à l'aide de ASP.NET Generator. Dans tous les cas, essayez de limiter l'impact des premières étapes de la modernisation :

- Envisagez d'adapter les transactions existantes avec une présentation navigateur Web sans introduire d'autres changements, en maintenant la navigation et la fonctionnalité existantes.
- Identifiez un sous-ensemble d'écrans fréquemment utilisés et appliquez de manière sélective vos nouvelles normes à ces seules transactions, en y ajoutant des listes déroulantes, des cases d'options ainsi que d'autres commandes.
- Testez ces changements pendant un certain temps, en y apportant des modifications selon les besoins puis étendez les nouvelles normes à de nouvelles transactions.

Le processus de développement sera modifié si vous choisissez l'outil Developer Painter pour le déploiement de vos interfaces utilisateur graphiques. Sur vos écrans graphiques dessinés, l'accent ne sera plus mis sur les écrans caractères mais sur le générateur graphique. ASP.NET Generator rapproche les formats de ceux utilisées par Developer Painter. Si un écran est accessible uniquement via le navigateur Web, il n'y aura aucune raison d'investir du temps dans le développement de formats permettant d'embellir les écrans caractères. Positionnez les champs sur le générateur de texte dans l'ordre que vous souhaitez pour naviguer entre ces derniers à l'aide des écrans graphiques. N'oubliez pas qu'il vous faut prendre en considération les étiquettes et les titres dans le générateur d'écrans textuels.

Saisie de données au kilomètre

Y aurait-il une catégorie de personnes qui seraient défavorables à l'adoption d'interfaces utilisateur graphiques ? Oui, les opérateurs de saisie intensive de données partout dans le monde ! Il est très important de bien comprendre la démarche suivie par cette population pour saisir les données au sein de vos applications EAE et concevez des interfaces utilisateur graphiques qui répondent à leurs contraintes et se gardent de les augmenter. Ces gens-là n'ont besoin ni de bouton radio ni de liste déroulante !

Pour commencer à utiliser les interfaces utilisateur graphiques

C'est le meilleur moment pour moderniser la couche de présentation de votre application EAE. ASP.NET Generator est une solution éprouvée, comparable à la plupart des outils de développement ASP.NET actuellement disponibles sur le marché. Il s'agit de la meilleure méthode possible pour actualiser rapidement et efficacement vos écrans verts en les introduisant dans le 21^{ème} siècle. Pour le moment, nous n'avons pas encore trouvé de besoin qui ne puisse pas être satisfait par l'utilisation des fonctions standard de ASP.NET Generator. Grâce à cet outil client extrêmement performant, nous avons aidé bon nombre d'entreprises à effectuer leurs premiers pas vers l'adoption d'interfaces utilisateur graphiques basées sur un navigateur et à réaliser de grandes avancées en termes d'utilisation de commandes sophistiquées pour transmettre les données à d'autres applications s'exécutant sur la station de travail.

Modernisez-vous en contrôlant les coûts et restant dans le cadre de vos possibilités, et préservez la valeur ajoutée de votre application EAE existante. La modernisation de vos applications existantes offre à votre entreprise le meilleur rendement possible sur vos applications. Protégez vos investissements en applications en adoptant ASP.NET Generator sans plus tarder.

George E. McGowan, Jr. CPA est président fondateur de McGowan Computer Associates, une société de services qui offre depuis 23 ans des conseils en gestion aux entreprises exploitant LINC, EAE et Agile Business Suite (AB Suite). Pour obtenir de plus amples informations sur les services de conseil, de formation, de support et de migration proposés par McGowan Computer Associates, contactez George McGowan au 801.446.7100 ou à l'adresse électronique GMCGOWANJR@aol.com.

Le coin des ingénieurs : les multiples facettes (user inter) des applications EAE et AB Suite

Amélioration apportées aux clients ASP.NET standard à l'aide de moteurs de rendu et de générateurs personnalisés

Pour poursuivre notre propos, centré sur les interfaces utilisateur, nous consacrons notre rubrique « Le coin des ingénieurs » à la présentation de plusieurs approches innovatrices visant à améliorer les clients ASP.NET générés. Nous vous encourageons à commencer à utiliser Client Tools et notamment ASP.Net Generator sur vos applications EAE afin d'améliorer l'expérience d'utilisateur final dès à présent, et à poursuivre cet effort lors de votre migration vers AB Suite.

Pour vous donner une idée générale des différentes méthodes disponibles pour vous permettre d'étendre vos applications web ASP.NET dans vos applications EAE ou AB Suite, nous nous sommes entretenus avec John Papachristos (ACUS), notre expert attiré en la matière. John est l'ingénieur en logiciels principal et le directeur technique de Client Tools.

Developing Agility: Pour quelle raison un développeur chercherait-il à faire plus que ce qui est disponible via l'outil ASP.NET Generator standard ?

John Papachristos (JP) : Dans notre environnement de développement, nous fournissons un jeu de commandes de base pour la conception des formats d'écrans avec l'outil de dessin d'écran « screen painter ». Du fait qu'il est possible de générer une multitude d'options d'interface client à partir de cette seule définition d'écran, sur une application EAE ou AB Suite, nous devons respecter certaines limites. Cependant, il arrive que les développeurs cherchent à déployer une interface utilisateur plus sophistiquée pour un environnement client spécifique qu'ils souhaitent utiliser, comme dans le cas des pages ASP.NET. C'est dans ce contexte que des possibilités de personnalisation plus sophistiquées entrent en jeu.

Par exemple, avec ASP.NET, un développeur pourrait vouloir ajouter une fonction d'affichage de calendrier, une commande de téléchargement de fichier ou d'autres commandes qui ne sont pas systématiquement fournies par nos générateurs. Nous avons toujours dit que notre environnement client ASP.NET généré constituait véritablement un tremplin pour les clients souhaitant en faire davantage avec leur interface basée sur un navigateur. En utilisant Microsoft(r) Visual Studio(r), il est possible d'améliorer l'application ASP.NET afin d'y inclure des commandes supplémentaires provenant de ASP.NET Framework ainsi qu'un traitement personnalisé permettant de répondre aux exigences commerciales du client.

Qu'est-ce qu'une commande personnalisée ?

Une commande est un objet que vous placez sur un format (telle qu'une page Web, par exemple) en utilisant le générateur afin d'exécuter des fonctions spécifiques. Les commandes courantes comprennent notamment les boutons de commande, les listes déroulantes, les images, etc.

Une commande personnalisée est définie comme ne faisant pas partie du jeu de commandes standard prévu par Microsoft. Web Form Renderer (Moteur de rendu de formats Web) est une commande qui fournit le conteneur pour une application EAE ou AB Suite. Elle gère les communications en direction et en provenance de l'application et présente ou « rend » les formats de transactions (présentation) sur le navigateur Web.

Web Form Renderer est donc un moyen pour ajouter des commandes ou des objets plus sophistiqués ou exécuter des traitements pré et post transaction entre le client et l'application sans avoir à modifier l'application en tant que telle. Grâce à cette commande, vous ne risquez plus de craindre les overwriting car vous utilisez constamment la forme Web telle que celle-ci est générée (sauf, bien entendu, dans le cas d'un événement de type OnAlternateForm). Il s'agit d'un fabuleux moyen d'introduire un traitement personnalisé sur une application frontale ou d'arrière-plan lors du transfert de données entre le navigateur et l'environnement d'exécution. >>

DA : Quels sont les moyens dont disposent les développeurs pour améliorer la forme ASP.NET générée ?

JP : Il existe trois options principales pour ce faire. Les développeurs ont le choix entre les méthodes suivantes :

1. Modifier les fichiers d'infrastructure du client ASP.NET
2. Utiliser le Web Form Renderer (Moteur de rendu de format d'écran Web)
3. Exploiter le Client Generator Customization Kit pour créer un générateur personnalisé

Avant de parler des moteurs de rendu et des générateurs personnalisés, expliquons d'abord comment les développeurs peuvent introduire des traitements personnalisés dans les différentes zones du framework ASP.NET à l'aide des fichiers d'infrastructure client.

DA : Vous parlez de modifier l'infrastructure ? Cela paraît difficile et quelque peu effrayant. Est-ce le cas ?

JP : Ce n'est pas aussi difficile qu'il y paraît. En faisant des modifications dans les fichiers d'infrastructure de l'environnement du client ASP.NET, les développeurs peuvent modifier la présentation et la logique client qui s'exécute dans le navigateur sur le poste de travail de l'utilisateur final, ainsi que la logique traitée sur le serveur Web. Les fichiers d'infrastructure sont intégrés avec le générateur Client Tools ASP.NET et constituent le fondement de notre framework ASP.NET client.

Par exemple, nous fournissons un fichier de script appelé CWFRCommonScript.js. qui regroupe un ensemble de JavaScript s'exécutant dans le navigateur. Selon les besoins d'une entreprise donnée, les développeurs ayant des compétences JavaScript sont à même de modifier les fonctions contenues dans ce fichier de script ou d'y ajouter des fonctions supplémentaires afin d'appeler un traitement personnalisé dans le navigateur. Ce script peut être mis à jour de sorte à capturer les frappes au clavier et exécuter certaines actions basées sur la saisie des utilisateurs finaux ou pour imposer une tabulation automatique vers le champ suivant lorsque l'utilisateur a atteint la fin du champ actuel. J'ai vu des développeurs introduire un script autorisant l'interaction entre le format d'écran du navigateur et les applications externes s'exécutant sur la station de travail locale, comme par exemple l'échange de données entre le format d'écran et Microsoft Word ou toute autre application hébergée sur la station de travail.

DA : Juste pour clarifier votre propos, ces changements sont-ils effectués après la génération du projet ASP.NET ? En raison du fait que vous changez d'« infrastructure », lors de votre prochaine opération de génération, vous n'avez aucune modification à effectuer pour réappliquer ces changements, n'est-ce pas ?

JP : C'est exact. Le fichier CEWFRCommonScript.js fait partie de notre jeu de fichiers d'infrastructure et à ce titre, il n'est pas susceptible d'être écrasé à chaque nouvelle opération de génération. Cependant, les développeurs qui choisissent de modifier les fichiers doivent être conscients du fait que les nouvelles versions sont fournies au format IC (Interim Correction). Par conséquent, si vous appliquez une nouvelle version IC, vous devrez être prudent et sauvegarder vos mises à jour.

DA : Vous avez mentionné d'autres méthodes permettant de modifier les fichiers d'infrastructure.

JP : Il est en effet possible d'introduire des changements en modifiant les fichiers d'infrastructure basés sur un serveur. L'un de ceux-ci, appelé IspecView.CS, définit une classe héritée par toutes les formats Web générées pour une application donnée. Il inclut bon nombre de fonctions côté serveur, exécutées lors du traitement d'un format Web. Ceci offre l'opportunité d'introduire quelques changements personnalisés, susceptibles d'affecter toutes les formats d'écrans.

Un développeur peut apporter des modifications au fichier IspecView.cs pour étendre les attributs de présentation dynamiques afin d'y inclure quelques attributs personnalisés. Par exemple, vous pouvez lancer une vérification de la plage de valeurs des champs numériques ou changer les couleurs des boutons individuels. Toute la logique sous-tendant le traitement des attributs dynamiques est contenue dans ce fichier d'infrastructure, qui peut ensuite être étendu par les développeurs selon leurs exigences particulières.

DA : Quelles sont les documentation susceptibles d'aider nos lecteurs à en savoir plus sur les modalités de modification de ces fichiers d'infrastructure ?

JP : Lorsque l'on connaît bien les questions liées au traitement des formats et au compilateur C#, il est assez facile de comprendre le processus utilisé. En outre, les fichiers d'infrastructure contiennent des commentaires qui donnent des explications supplémentaires sur les différentes fonctions. Certaines informations sont également disponibles dans le kit de personnalisation, mais il s'agit d'un élément à commander séparément, et qui n'est pas nécessaire pour modifier les fichiers d'infrastructure, lesquels font partie du générateur Client Tools ASP.NET et sont installés à ce titre. >>

DA : Existe-t-il d'autres fichiers d'infrastructure ?

JP : Oui, il y en a un certain nombre. Vous pouvez configurer l'environnement ASP.NET pour afficher une page de log-in et les développeurs ont la possibilité de personnaliser la page de log-in que nous fournissons en fonction de leurs besoins. Il existe d'autres formats Web, telles que les pages de temporisation (délai d'attente hors-temps) et les pages d'erreur qui sont également personnalisables.

Une autre possibilité intéressante est le menu déroulant obtenu en cliquant sur le bouton droit de la souris, qui constitue une commande de personnalisation susceptible d'être mise à jour pour exclure les éléments existants ou y introduire un traitement personnalisé. Je peux vous citer le cas d'un client qui a étendu le menu déroulant obtenu via le clic droit de la souris de sorte à y inclure un nouvel élément de menu permettant de copier les données depuis une feuille de calcul Microsoft Excel(r) vers les champs Copy.From de la forme Web ASP.NET. Le client, qui utilisait auparavant une telle fonction sous son émulateur T27, avait constaté qu'il était impossible d'effectuer cette procédure dans un environnement ASP.NET Client standard. En personnalisant le menu déroulant obtenu via le clic droit de la souris et en ajoutant certaines méthodes JavaScript connexes dans CEWFRCommonScript.js, le client a réussi à répliquer cette fonctionnalité sur ses formats Web ASP.NET.

DA : Nous avons à présent couvert la première option d'extension des formats ASP.NET. Qu'en est-il de l'utilisation de la commande Web Form Renderer ?

JP : C'est exact. Une autre approche de la personnalisation de l'environnement ASP.NET consiste à modifier les fichiers générés de sorte à fournir des commandes et un traitement supplémentaires. Bien entendu, c'est là où il convient d'être très vigilant dans la gestion des fichiers personnalisés afin d'éviter l'écrasement de ces derniers lors des opérations de génération ultérieures. À cet effet, l'événement « OnAlternateForm » de la commande Web Form Renderer (Moteur de rendu de formats Web) est fort commode car il vous permet réellement de dérouter cet événement et d'afficher votre forme personnalisée à partir d'un ISPEC spécifique au lieu de la forme « standard » générée, définie dans l'application. Les clients tirent ainsi profit de cet événement dans le cas de certaines formats nécessitant une personnalisation plus étendue au sein de leur application.

DA : Qu'est-ce qu'un « moteur de rendu » et quel est l'usage que peut en faire un développeur? Quelles sont les améliorations que les développeurs peuvent apporter aux interfaces utilisateur final grâce aux moteurs de rendu ?

JP : En fait, le numéro du premier trimestre 2006 de cette nouvelle lettre comporte un grand nombre d'articles instructifs sur les moteurs de rendu. Web Form Renderer (Moteur de rendu de formats Web) est essentiellement une commande personnalisée autorisant l'affichage des commandes de formats Web générées à partir des formats ISPEC, c'est-à-dire les formats conçues à l'aide de Developer Painter, et qui permet au programmeur d'ajouter un traitement spécifique sur la base de certains événements ou déclencheurs.

J'ai déjà mentionné l'événement « OnAlternateForm » qui vous donne la possibilité de substituer des formats que vous avez personnalisés. Il existe d'autres événements, tels que OnPreTransaction, OnPostTransaction et OnStatusLine, que les développeurs utilisent à certains points du traitement de forme Web pour exécuter des actions spécifiques. Par exemple, vous pouvez souhaiter capturer les données en provenance du système d'exécution hôte avant que celles-ci ne parviennent au navigateur et n'occasionnent l'attribution de valeurs par défaut à certains champs.

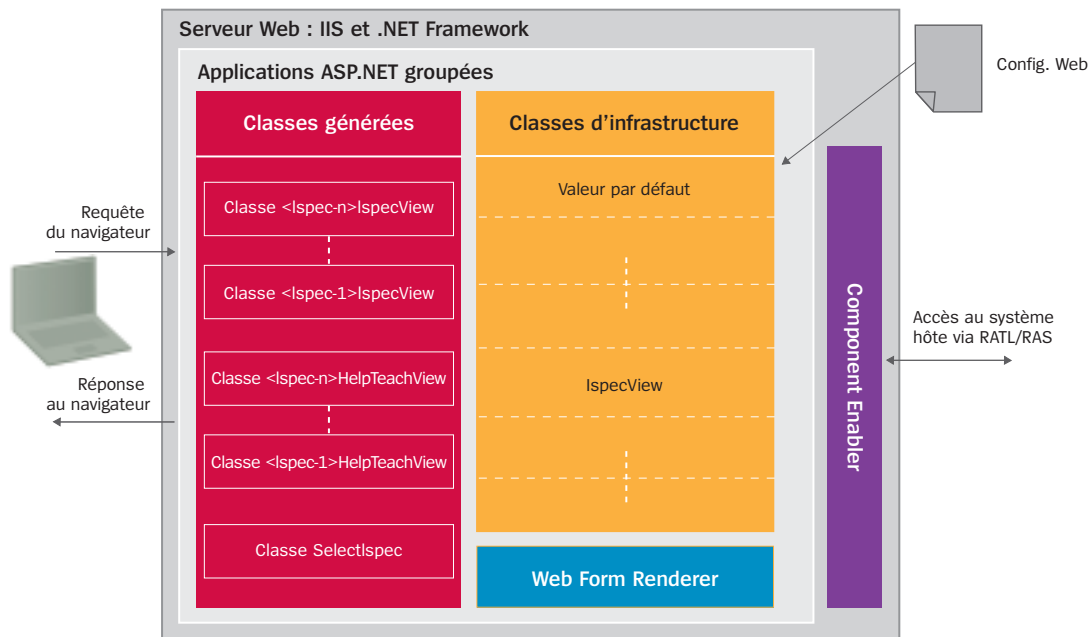
DA : Faut-il prendre des précautions particulières lors de l'utilisation de Web Form Renderer ?

JP : Le seul bémol à signaler réside au niveau du coût de traitement d'exécution, qui risque d'être relativement onéreux en cas de mise en œuvre d'un gros volume de traitement de formats via le moteur de rendu. Car il ne faut oublier que la commande Web Form Renderer s'exécute sur le serveur Web et non sur la station de travail du client.

DA : C'est entendu. Il est temps pour nous à présent d'aborder la troisième option, à savoir le kit Client Generator Customization Kit. Qu'est-ce qu'un générateur personnalisé et quel est l'usage que peut en faire un développeur?

JP : Les développeurs ont la possibilité d'utiliser un générateur personnalisé lorsqu'ils doivent effectuer un grand nombre de changements de formats Web, comme dans le cas des commandes personnalisées et du traitement spécial et qu'ils souhaitent faire de l'application de ces modifications un processus répétable. Si vous êtes préoccupé par les coûts potentiels indirects découlant de l'utilisation de la commande Web Form Renderer pour une personnalisation étendue, la meilleure option pour vous sera de mettre à jour directement les formats Web générées. Une fois que les développeurs sont satisfaits du fonctionnement de leurs formats Web ASP.NET personnalisés, ils peuvent alors recourir à notre Client Generator Customization Kit afin de modifier le générateur ASP.NET source et créer leur propre générateur personnalisé. Le générateur personnalisé génère les formats Web de la manière souhaitée par les développeurs, en créant des commandes supplémentaires et en évitant tout traitement d'exécution imposé par la commande Web Form Renderer. >>

Traitement d'exécution des formats Web sous ASP.NET



Cette illustration montre l'interaction entre les trois niveaux d'une application Web : navigateur Web s'exécutant sur une station de travail, forme Web ASP.NET s'exécutant sur le serveur Web et accès à l'application du système hôte via RATL/RAS.

DA : La création d'un générateur personnalisé semble être une opération complexe.

JP : J'entends souvent cela, pourtant, je peux aider tous ceux qui souhaitent se lancer à le faire en quelques minutes. Nous avons récemment organisé un atelier de travaux pratiques de 40 minutes dans le cadre de la réunion des utilisateurs du Unisys Technology Forum à Queenstown, en Nouvelle-Zélande. Il a suffi de 10 minutes aux participants pour être capables d'ajouter des aides sous forme d'infobulles aux champs, en utilisant le kit de personnalisation. N'ayez pas peur de vous y essayer.

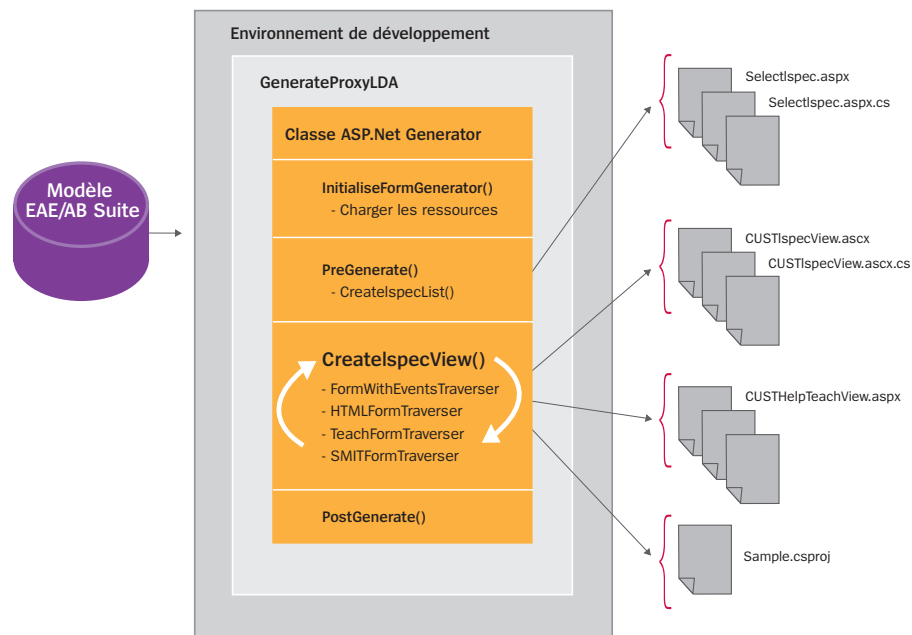
DA : À quel moment un développeur utilise-t-il la commande Web Forms Renderer et quand est-il amené à créer ou à utiliser un générateur personnalisé ?

JP : Vous exécuterez la commande Web Forms Renderer si vous ne voulez pas introduire de changement au format d'écran générée mais souhaitez exercer un impact au niveau du traitement des formats en cours d'exécution. Les développeurs devront utiliser le générateur personnalisé s'ils cherchent à effectuer des modifications aux formats générés ou à réclamer un processus de changement répétable.

DA : Le développeur peut-il effectuer une quelconque tâche sur Client Tools dans AB Suite qu'il ne peut pas effectuer sous une application EAE ?

JP : Non, en fait, vous disposez d'une option qui consiste à utiliser la version AB Suite de Client Tools avec l'application EAE. Client Tools s'utilise de manière interchangeable sur les applications EAE et AB Suite. Il existe, bien évidemment, des fonctions propres au générateur de formats sous AB Suite qui sont indisponibles dans une application EAE. Cependant, les fonctions du client ASP.NET sont les mêmes, que vous utilisiez l'application par défaut, la commande Renderer ou le kit de personnalisation. Il s'agit bien du même client. >>

Les quatre phases du processus de génération



Ceci illustre les quatre phases du processus de génération. Le gros du travail est effectué selon la méthode `CreatelspecView()`. Celle-ci exécute les différentes classes de type « Traverser » qui génèrent les diverses parties d'une forme Web. Une classe « Traverser » traite les données renseignées en mémoire par les appels effectués sur les interfaces `GenerateProxyLDA` depuis l'application EAE ou AB Suite Builder. Un générateur instancie généralement une ou plusieurs classes `Traverser`, en fonction des résultats devant être produits à partir d'une application client spécifique.

DA : Lorsqu'un développeur utilise le générateur ASP.NET, existe-t-il des restrictions ou des limitations qu'il faut respecter, particulièrement pour passer d'une application EAE à une application AB Suite ?

JP : Non, il n'y a aucune restriction ou limitation à signaler dans ce cas. Tout ce que vous effectuez au sein d'une application EAE est susceptible de migrer vers AB Suite en toute transparence, lorsque vous chargez le modèle. Les clients peuvent migrer vers AB Suite, construire leur dossier groupé et compiler leurs clients ASP.NET, et l'interface utilisateur du navigateur sera identique à celle de l'application EAE.

DA : AB Suite Developer utilise Visual Studio. Les clients doivent-ils également détenir une licence Visual Studio afin d'exploiter Client Tools sur les applications EAE ?

JP : Non, ils n'ont pas besoin de licence Visual Studio. Il leur faut par contre le compilateur C#, qui est fourni avec le framework runtime .NET. Cela signifie que vous avez la possibilité de générer les applications AB Suite à l'aide des fichiers batch fournis avec AB Suite, puis de déployer et d'utiliser celles-ci sans Visual Studio. Ceci étant dit, toutes les activités de personnalisation que je viens de passer en revue avec vous exigent un environnement de développement Visual Studio.

Un grand merci à John Papachristos pour tous ces précieux conseils sur la personnalisation des formats ASP.NET et le traitement de formats.

Nouvelles options d'interface utilisateur graphique : les générateurs personnalisés

Récapitulation rapide

Ce numéro de *Developing Agility* est exclusivement consacré aux différentes options dont bénéficient les développeurs pour fournir à l'utilisateur final une présentation graphique sur les applications EAE et AB Suite.

Premièrement, il vous est possible de vous servir des générateurs d'interface utilisateur graphiques sur les applications EAE et AB Suite pour définir la présentation à l'utilisateur final comme faisant partie intégrante du modèle et générer tous les types de clients, avec ou sans navigateurs Web.

Vous souhaitez en faire plus ? Vous pouvez modifier manuellement le client généré, mais nous vous déconseillons de le faire car il vous faut ensuite réappliquer les modifications à chaque nouvelle opération de génération. Cependant, il vous reste une large gamme d'autres options, particulièrement si vous êtes intéressé par le déploiement de clients par navigateur interposé. Comme il a été indiqué plus haut dans la rubrique Le coin des ingénieurs, vous avez la possibilité de modifier les fichiers d'infrastructure client, d'utiliser le Web Forms Renderer pour faire des changements dans le projet ASP.NET généré et/ou d'utiliser Client Tools Customization Kit pour construire votre propre générateur personnalisé. La plupart de ces options requièrent un certain niveau de connaissances relatives au compilateur C# ou à .NET, mais les résultats sont impressionnants et restent en parfaite synchronisation au fur et à mesure de l'évolutivité de vos applications EAE ou AB Suite au fil du temps.

Mais ce n'est pas tout !

Plusieurs entreprises ont développé des générateurs personnalisés spécialisés, reposant sur un paramétrage. Conçus pour les cas de figure où ASP.NET Generator standard n'est pas suffisant, l'entreprise concernée ne souhaitant pas par ailleurs élaborer son propre générateur personnalisé, ces outils facilitent considérablement la tâche des développeurs sans grande expérience du Web ou de .NET qui se lancent dans la personnalisation de l'interface utilisateur graphique. Ils autorisent également une approche répétable de sorte que les modifications sont automatiquement réappliquées à chaque opération de génération du système.

Élaborés par des techniciens ingénieurs qui maîtrisent les applications EAE et AB Suite autant que les interfaces utilisateur graphiques et connaissent parfaitement les besoins des utilisateurs finaux, ces générateurs personnalisés sont une option extrêmement valable, surtout lorsque vous ne souhaitez pas forcément acquérir d'expertise Web ou .NET en interne. Le tableau suivant répertorie plusieurs sources de générateurs personnalisés parmi les plus connues. Au final, la décision vous appartient, mais vous disposez d'un grand nombre d'options. Quel serait le bon choix pour votre entreprise ? Cela dépend de deux facteurs : l'apparence que vous souhaitez donner à vos interfaces utilisateur graphiques et le niveau de connaissances techniques que vous possédez ou voulez acquérir. Considérez les éléments suivants :

Nom du générateur personnalisé Société/Informations de contact	Description
<p>Generator ASP.NET for AB Suite and EAE Disponible sous forme d'offre de service auprès de Unisys Europe en contactant Dominique.Michaut@fr.unisys.com</p>	<p>Ce générateur permet aux développeurs d'améliorer aisément les interfaces utilisateur graphiques en y ajoutant automatiquement des objets graphiques non pris en charge par ASP.NET Generator. Ceci est réalisé par la saisie de paramètres dans l'outil de générateur personnalisé. Cet outil peut servir de solution client/serveur complète, dotée d'une fonctionnalité JavaScript générique susceptible d'être également personnalisée.</p> <p>Les options de commande personnalisée incluent les éléments suivants :</p> <ul style="list-style-type: none"> • Calendrier • Liste de valeurs • Grille de données avec tri et pagination • Utilisation de fichier .css (CSS) • Et beaucoup d'autres éléments <p>Simple et efficace, Generator ASP.NET for AB Suite and EAE est exploité par la majorité des utilisateurs d'applications EAE/AB Suite en France et dans d'autres pays européens. >></p>

Nom du générateur personnalisé Société/Coordonnées	Description
<p>Interface Builder Disponible auprès de <u>Information Exchange Group (IEG), Inc.</u></p>	<p>Interface Builder (IB) est un outil de conception de formats d'écrans particulièrement flexible qui exploite aisément les fonctions de navigation avec un minimum de changements ou de connaissances spécifiques requises. Depuis son lancement en 1999 jusqu'à sa version actuelle, IBDotNET a connu des améliorations en continu et est désormais étroitement intégré à AB Suite. Les formats générées par IB sont utilisées par près de 200 entreprises. En outre, trois sociétés de développement basées en Amérique du Nord ont adopté ce produit.</p> <p>L'atout de IB est son infrastructure souple qui autorise l'ajout de fonctionnalité graphique et la personnalisation client ou le traitement serveur sans la nécessité de recourir à d'importantes compétences de programmation par navigateur. La seule limite est celle de votre imagination.</p> <p>Une copie d'évaluation d'IB peut être téléchargée depuis le site Web d'IEG. Les conditions de validité des périodes d'essai sont consultables à l'adresse http://ieg-ib.iegonline.net/ieg-ib.htm.</p>
<p>Smart Web Client Disponible auprès de <u>McGavin Consulting Group</u></p>	<p>Smart Web Client (SWC) permet aux développeurs d'ajouter aux formats graphiques EAE bon nombre d'éléments graphiques sophistiqués, tels que des boutons de calendrier, des menus déroulants ou des bandes d'onglet. La fonctionnalité AJAX Client de SWC offre aux utilisateurs finaux la possibilité de requérir et de sélectionner des données applicatives sans modifier les Specs. Les utilisateurs chevronnés aiment la fonctionnalité de zoom proposée par le produit, qui étend dynamiquement la taille de l'affichage de la fenêtre. SWC dispose également de fonctionnalités telles que celles des clés de fonction, des touches à accès rapide et des touches numériques programmables réclamées par les mordus de la vitesse de saisie de données lors des opérations de cycle rapide liées aux écrans.</p> <p>SWC est utilisé par les entreprises spécialisées dans le développement d'applications EAE en Nouvelle-Zélande, en Australie et au Royaume-Uni. Le produit est actuellement en cours d'amélioration pour la prise en charge de AB Suite, comprenant la mise à niveau automatique de SWC à partir des applications EAE et AB Suite.</p> <p>Visitez le site www.mcgavin.biz pour obtenir une présentation d'ensemble de SWC ou envoyez vos requêtes à l'adresse électronique questions@mcgavin.biz.</p>
<p>CTC ASP.NET WebForms Generator Disponible auprès de <u>Client Tools Consultancy (CTC)</u></p>	<p>Les grandes lignes du générateur personnalisé ont été présentées dans le <u>numéro du premier trimestre 2008 de Developing Agility</u>. En utilisant CTC Configurator, les développeurs configurent, selon les besoins, l'apparence et le toucher de toutes les commandes sur l'écran, en tirant profit de la gamme complète des fonctions ASP.NET. Ceci inclut la possibilité d'exploiter les commandes de tiers afin de répondre aux exigences des interfaces utilisateur modernes.</p> <p>Pour télécharger une version d'essai du logiciel, rendez-vous sur le <u>site Web de CTC</u>.</p>

- Pour bénéficier d'une interface utilisateur graphique digne de ce nom, qui dépasse de loin les écrans verts d'antan, la plupart des entreprises spécialisées dans le développement sont relativement satisfaites des performances de Developer Painter et de Presentation Client en tant que mécanisme d'affichage.
- Pour se doter d'une interface utilisateur graphique par navigateur haut de gamme, Client Tools ASP.NET Generator est un excellent choix.
- Les entreprises spécialisées dans le développement dotées de solides connaissances relatives au compilateur C# ou à ASP.NET et désireuses d'obtenir des commandes plus sophistiquées de leurs interfaces utilisateur graphiques peuvent envisager d'apporter des améliorations supplémentaires à leurs formats Web via ASP.NET Generator, tel qu'expliqué dans la rubrique Le coin des ingénieurs.
- Les entreprises qui recherchent une interface Web personnalisée mais ne souhaitent pas développer de nouvelles compétences techniques en interne devraient plutôt se pencher sur les options de générateur personnalisé présentées dans le tableau ci-dessus.

Optez pour l'option qui vous semble la mieux adaptée à votre environnement et à vos besoins. Vos utilisateurs finaux vous en seront reconnaissants !

Comment obtenir une architecture orientée services (SOA) intelligente

Dans une étude de cas récente, Galvano Groothandel BV décrit comment la création d'une architecture SOA exploitant ses applications EAE a permis à ce grossiste en accessoires de plomberie haut de gamme d'augmenter le volume de ses commandes de commerce en ligne de 55 %. L'architecture SOA est un sujet d'actualité et il n'a jamais été plus opportun de commencer à explorer comment vos applications EAE et systèmes ClearPath pour en faire partie intégrante.

L'architecture SOA vous aide à accroître la valeur ajoutée de vos applications d'entreprise existantes, y compris celles qui sont basées sur les applications AB Suite ou EAE, et ce, sans nécessairement mettre en péril les flux de revenus générés par ces applications.

Unisis veut vous aider à tirer profit des opportunités SOA tout en misant sur la croissance de la valeur commerciale inhérente à vos applications stratégiques existantes. À cet effet, nous avons récemment publié une série de livres blancs relatifs à l'architecture SOA, qui retracent la progression logique du concept depuis sa présentation théorique jusqu'à l'évocation des intergiciels actuellement disponibles qui permettent à vos applications ClearPath de contribuer à une SOA.

Ces documents sont téléchargeables depuis le site eCommunity :

- [Service-Oriented Architecture: Delivering for Business](#)
- [Service-Oriented Architecture: ClearPath Systems in SOA](#)
- [Middleware Products and Strategies for ClearPath OS 2200](#)
- [Middleware Products and Strategies for ClearPath MCP](#)

L'[étude de cas Galvano](#) peut être consultée dans son intégralité sur le site Web d'Unisis.

Calendrier

Plusieurs rencontres entre les utilisateurs des applications AB Suite et EAE et les employés d'Unisis seront prochainement organisées à travers le monde. N'oubliez pas de consulter le site Web [eCommunity](#) pour les dernières informations à ce sujet.

Quoi ?	Où ?	Quand ?
Enregistrement du WebCast sur les nouveautés AB Suite d'avril 2008	Disponible sur demande	À tout moment
Réunions d'utilisateurs AB Suite et EAE (choix entre deux dates), la deuxième session s'adresse particulièrement aux clients francophones et disposera de traduction simultanée.	St. Paul de Vence, France	22-24 septembre 2008 24-26 septembre 2008
Conférence annuelle sur les technologies : UNITE	Caribe Royale Orlando , Orlando, FL	19-23 octobre 2008

Les spécifications sont susceptibles d'être modifiées sans préavis.

© 2008 Unisis Corporation

Tous droits réservés.

Unisis est une marque déposée d'Unisis Corporation. Microsoft, Excel et Visual Studio sont des marques déposées de Microsoft Corporation. Tous les autres noms de produits et marques mentionnés dans ce document sont considérés comme des marques commerciales ou déposées de leurs détenteurs respectifs.