Généralités

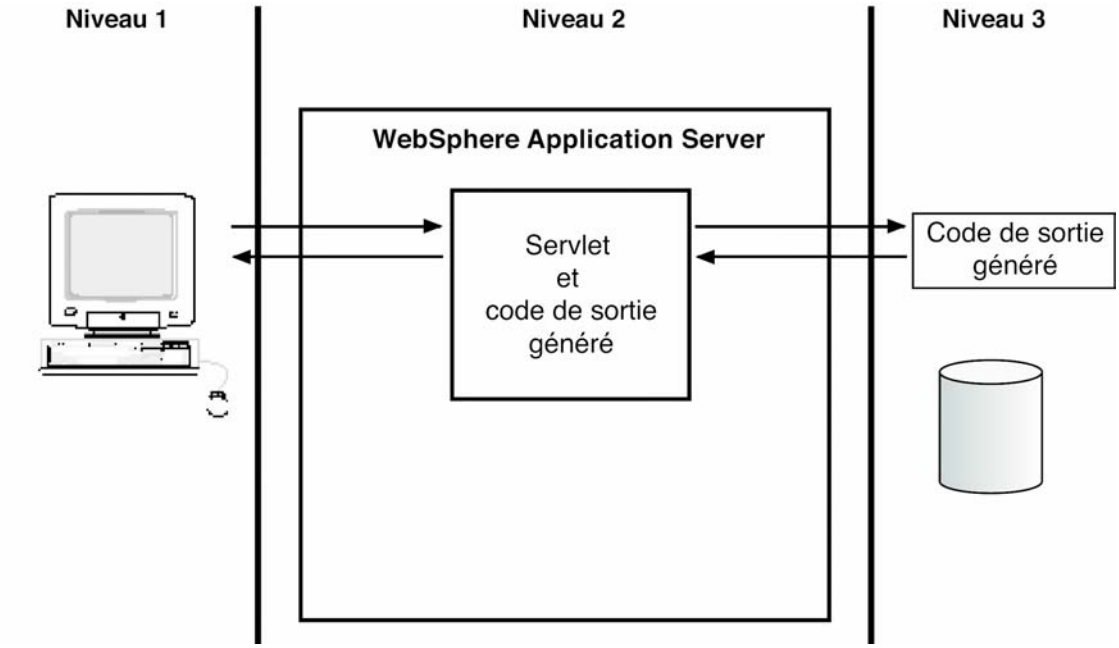
Introduction à EGL

EGL (*Enterprise Generation Language*) est une technologie de développement qui vous permet de générer rapidement des applications complètes fonctionnant dans les environnements suivants :

- Java™ (J2EE ou non J2EE),
- z/OS, que ce soit dans le cadre d'un processus de traitement par lots (sous le contrôle de JCL) ou sur la plate-forme CICS® (en mode interactif ou en mode batch).

EGL vous permet de délivrer des données d'entreprise aux navigateurs, même si vous ne possédez qu'une expérience minimale des technologies Web. Vous pouvez notamment utiliser cette technologie pour :

- développer un service Web accessible à d'autres applications Internet ;
- développer un programme EGL afin de l'utiliser dans une architecture trois tiers (ou trois niveaux), comme le montre la figure suivante.



Le niveau 1 se compose d'un navigateur Web dans lequel l'utilisateur entre une adresse URL ou clique sur un lien pour appeler un programme. Le niveau 2 inclut un serveur d'applications Web, tel que WebSphere® Application Server, qui sert d'intermédiaire entre le navigateur et le programme appelé. Le niveau 3 contient le programme généré, qui prend en charge les données du niveau 2, interagit avec les sources de données, manipule les données et les renvoie au niveau 2.

Ces trois niveaux correspondent généralement à des machines distinctes, et la couche logique du niveau 2 ou 3 peut elle-même être fractionnée sur plusieurs machines.

La version courante d'EGL permet de générer :

- des objets Java exécutés sur les plates-formes de niveau 2 ou 3 suivantes :
  - AIX,
  - iSeries,
  - Linux (à base Intel),
  - Windows 2000/NT/XP,
  - Z/OS UNIX System Services ;
- des programmes COBOL fonctionnant sur CICS pour z/OS, qui est une plate-forme de niveau 3.

À votre demande, EGL prépare les composants générés pour la phase d'exécution :

- Il envoie chacun de ces composants vers une plate-forme cible ;
- Il supervise l'étape de préparation qui consiste à compiler les programmes Java, à transcoder, compiler et éditer les liens des programmes COBOL et à associer les modules de chargement à une base de données DB2 ou Informix.
- Renvoie un message d'accomplissement.

Bien qu'il puisse être utilisé de façon indépendante, EGL s'intègre dans les environnements Struts Tools et JavaServer Faces, qui vous permettent de structurer vos applications Web.

Reposant sur la technologie VisualAge® Generator, EGL offre les avantages suivants :

- Il vous permet d'implémenter rapidement votre logique applicative en utilisant le langage procédural EGL, un débogueur interprétatif vous est aussi fourni;
- Il vous permet de vous concentrer sur la fonction principale de votre code, au lieu d'avoir à résoudre les difficultés techniques liées à la plate-forme ou à ses ressources. Vous pouvez, par exemple, utiliser des instructions d'E/S similaires pour accéder à différents types de source de données externe (fichiers, bases de données relationnelles ou files d'attentes de messages MQSeries®).
- Il vous permet de personnaliser votre code en fonction des spécifications de la plate-forme courante, sans avoir à vous préoccuper des problèmes de migration future. Un programme EGL développé pour une plate-forme cible donnée peut être converti pour pouvoir être utilisé sur une autre plate-forme.
- Il vous permet de générer les différents composants d'un système applicatif à partir du même code source. Ainsi, après avoir développé un programme EGL, vous pouvez générer un encapsuleur Java, un bean de session Enterprise JavaBean (EJB) et un programme de niveau 3. Ce gain de productivité est particulièrement évident lorsque vous développez un logiciel pour permettre aux utilisateurs d'accéder à un servlet de niveau 2, qui transmet les données à un encapsuleur Java généré, lequel accède à son tour à un programme généré de niveau 3 ou à un serveur EJB.

Lorsque vous utilisez CICS comme plate-forme d'exécution cible, vous bénéficiez de deux avantages supplémentaires :

- Premièrement, vous vous affranchissez des difficultés techniques traditionnellement liées à l'utilisation d'un langage de troisième génération pour développer un programme CICS pseudo-conversationnel. Il vous suffit de programmer votre logique applicative comme si l'utilisateur était en train de « dialoguer » avec un programme résident, sans vous préoccuper du fait que pendant la phase d'exécution, le code fait l'objet de transferts incessants vers et hors de la mémoire au cours des multiples interactions entre le programme et l'utilisateur.
- Deuxièmement, vous pouvez éviter d'avoir à configurer un connecteur CICS lorsque vous déployez un programme généré qui est accessible depuis le niveau 2. Dans ce cas, un encapsuleur Java généré, situé au niveau 2, reformate les données qui sont appelées à transiter entre les niveaux 2 et 3.

Processus de développement

Le processus de développement sous EGL se compose des étapes suivantes :

Configuration
Vous configurez votre environnement de travail, en définissant des préférences et des projets, par exemple.

Création et ouverture des fichiers EGL
Vous commencez à générer le code source.

Déclaration
Vous définissez et spécifiez les caractéristiques de votre code.

Validation
EGL vérifie périodiquement vos déclarations (chaque fois que vous sauvegardez un fichier, par exemple), en vous indiquant si elles sont correctes sur le plan syntaxique et, dans une certaine mesure, cohérentes en termes de contenu.

Débogage
Lorsque vous générez un programme Java, vous interagissez avec un débogueur intégré pour vous assurer que votre code répond bien à vos spécifications.

Compilation
Le code source est converti en code de sortie au moyen des procédures suivantes :

Génération
EGL valide vos déclarations et génère un code de sortie qui intègre le code source.

Préparation
EGL prépare le code source de façon à générer des objets exécutables. Cette procédure consiste, dans certains cas, à transférer le code source de la plate-forme de développement vers une plate-forme de déploiement, à préparer le code source sur cette dernière et à renvoyer un fichier de résultats à la plate-forme de développement.

Déploiement
EGL génère un code de sortie qui facilite le déploiement des objets exécutables.

Configurations d'exécution

EGL permet de générer, entre autres, les types de programme suivants :

- un programme Java unique pour toutes les plates-formes supportées, telles qu'elles sont répertoriées dans l'Introduction. Vous pouvez déployer ce programme en dehors de l'environnement J2EE ou dans les conteneurs J2EE suivants :
  - application client J2EE,
  - application Web J2EE,
  - conteneur EJB ; dans ce cas, vous générez également un bean de session EJB.

Le programme Java généré via EGL ne peut être utilisé dans un service Web que s'il est non interactif et qu'il obtient la main au moyen d'un appel. Dans ce cas, il peut être déployé aussi bien dans l'environnement J2EE qu'en dehors de celui-ci.

- un programme COBOL destiné à être exécuté, en mode interactif ou mode batch, dans l'environnement CICS pour z/OS. Un programme généré via EGL qui est exécuté en mode de traitement par lots sur une plate-forme CICS peut être utilisé au sein d'un service Web EGL.
- un programme COBOL non interactif, destiné à être exécuté sous z/OS, mais sur une plate-forme autre que CICS ; ce type de programme entre dans la catégorie des programmes batch z/OS.

EGL permet en outre de développer des applications Web capables de :

- délivrer des pages graphiques aux navigateurs Web ;
- stocker et extraire des données pour un nombre potentiellement important d'utilisateurs ;
- s'intégrer dans l'environnement JavaServer Faces.

Pour plus d'informations sur le support spécifique des applications Web, reportez-vous à la section PageHandler part.

Enfin, EGL peut être utilisé pour générer un encapsuleur Web, comme cela est expliqué dans la section suivante.

Utilisation d'un encapsuleur Java

L'encapsuleur Java généré via EGL se compose d'un ensemble de classes qui vous permettent d'appeler un programme EGL à partir d'un code Java non EGL (notamment à partir d'un programme Java non J2EE ou d'une classe d'action propre à une application Web J2EE reposant sur la technologie Struts ou JSF). La procédure d'intégration Java-EGL est la suivante :

1. Générez des classes d'encapsulation Java spécifiques au programme EGL.
2. Incorporez-les dans le code Java non EGL.
3. À partir du code Java non EGL, appelez les méthodes associées à ces classes d'encapsulation pour générer l'appel réel et convertir les données entre les deux formats suivants :
   - le format type de données utilisées par Java,
   - le format de type élémentaire requis pour transférer les données vers et depuis le programme EGL.