

Rational Developer EGL

le nouveau langage structuré pour développer des applications WEB !



Développements Modernes

Impose la maîtrise de nombreuses technologies

STRUTS

JAVA ou C#

ASP

EJB

SQL

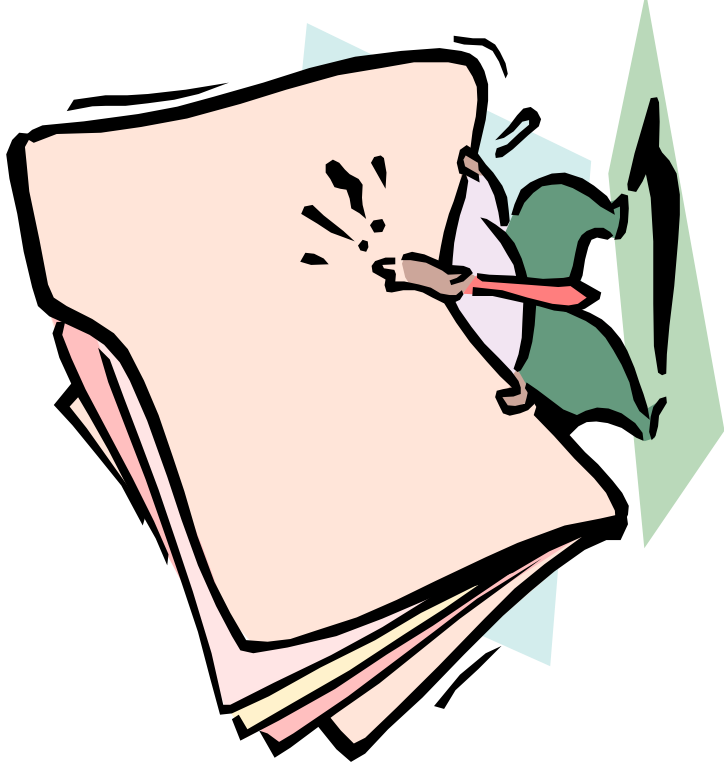
HTML

JSF

XML

SERVLETS

SOAP



I had a Dream

Un langage de 4eme génération ...

Permettant de développer de manière traditionnelle

Permettant des développements WEB

Disposant d'un langage procédural ...

... Mais également orienté objet

Indépendant de la base de données

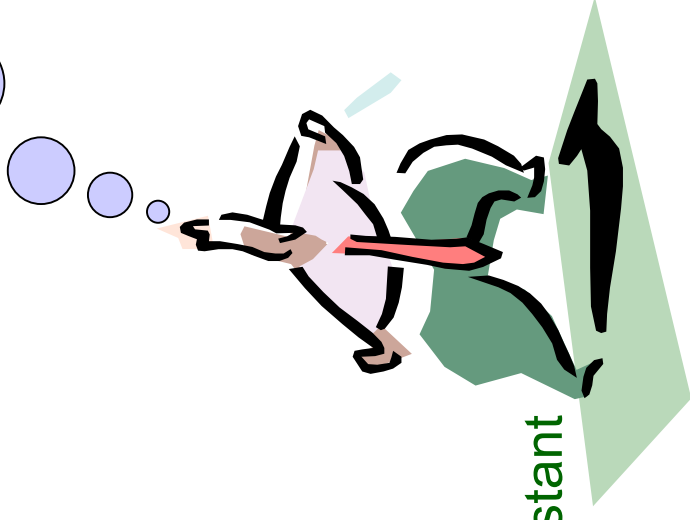
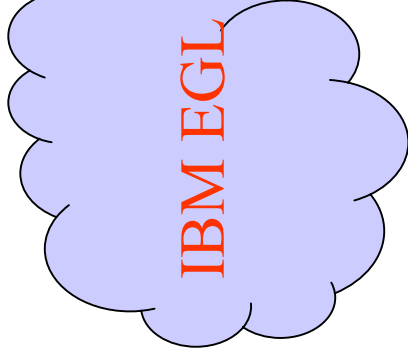
Non soumis à Run-Time

Indépendant du système d'exploitation

Disposant d'un outil moderne

Me permettant de récupérer une partie de l'existant

M'offrant une grande pérennité



Qu'est EGL ?

- Un langage de haut niveau (aussi confortable que COBOL, RPG, 4GL)
 - Permet d'utiliser très simplement toute la panoplie des technologies modernes (à mesure qu'elles émergent)
 - Permet d'utiliser les mêmes outils que ceux utilisés par les développeurs Java.
- Permet de masquer la complexité de certaines technologies
 - J2EE, CICS, i-series, MQ, SQL, Web Services, Portlets, XML, ...
- Destiné à des programmeurs ne connaissant pas J2EE

En particulier les développeurs familiers de:

- Informix 4GL
 - Cobol
 - RPG
- Visual Basic
Oracle Forms
Visual Age Generator



Pourquoi un nouveau langage ?

- Besoin d'un langage de 4^{ème} génération basé sur Java.
 - I-4GL est au C ce que EGL est à Java
 - Les composants Java ainsi générés vont tourner naturellement dans une JVM ou un serveur J2EE
 - Besoin d'un langage simple nécessitant peu d'apprentissage
- IBM a la problématique de faire migrer 2 catégories de développements:
 - Visual Age Generator 4GL
 - Informix 4GL



Comment se présente-t-il ?

- **Outil de développement intégré (IDE)**

EGL est inspiré de “Visual Age Generator 4GL”

EGL est intégré dans tous les produits “Rational Developer”

- Rational Web Developer (RWD)
- Rational Application Developer (RAD)

- **Run-time Environment**

EGL génère du code Java;

Pour être exécuté, ce code Java aura besoin de (au choix):

- Java Run-time Environment (JRE) pour les applications caractères
- Serveur d’application Java (WebSphere, Apache Tomcat)



Capitaliser sur les investissements

Humains

- Outil RAD très performant
 - Fournit des templates, exemples qui font gagner beaucoup de temps
 - Couche d'abstraction, permet de travailler de façon similaire sur:
 - Tables Relationnelles
 - Fichiers standards
 - MQ Series data queues
- Codage de l'application dans un seul langage
 - Peu d'apprentissage
- Possibilité de valoriser les compétences multiples
 - 4GL, Java au sein d'un même projet
- Programmation identique quelle que soit l'interface utilisateur
 - Mode Caractère et/ou mode Web
 - Développement simple d'applications Web
 - Facilite le passage du mode caractère vers le Web



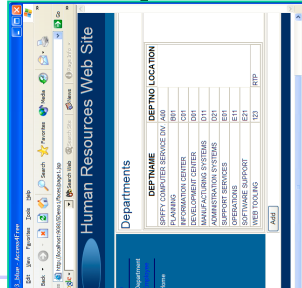
Principales caractéristiques

- EGL business logic est utilisable par des applications J2EE
- EGL génère du Java
- EGL supporte le mode caractère aussi bien que le mode Web
- EGL supporte les environnements Windows / Linux / Unix / AS400 ...
- EGL est extensible (appel de code Java, C)
- EGL dispose de connectivité JDBC
- EGL dispose d'un outil de développement moderne et complètement intégré (debugging, designer html, travail en team ...)
- EGL peut être déployé vers les environnements zSeries et iSeries
- EGL permet de bénéficier des fonctionnalités de “Message Queuing” entre applications hétérogènes (MQ series)
- Permet de développer des “Web Services”



Définition de DDL

- Dictionnaire de données utilisable dans l'ensemble des programmes
- Definit les Validations, Formats, Messages, etc... une seule fois
 - ces règles seront appliquées partout où les données apparaissent (TUI, Page, business logic, ...)
- la notion de “DataItem” permet de créer des types de données ou d’enregistrements très riches



Human Resources Web Site

DEPTNAME	DEPTNO	LOCATION
SUPPORT SERVICES	100	REP
TRAINING	101	
DEVELOPMENT CENTER	102	
DEVELOPMENT CENTER	103	
MANAGEMENT SYSTEMS	104	
SUPPORT SERVICES	105	
SOFTWARE SUPPORT	106	
WEB TOOLS	107	

```
// Customer SQL Record
Record Customer SQLRecord
{ tableNames=("PIZZABYIBM.CUSTOMER"),
  keyItems=("customerId") }
customerId CustomerId;
firstName FirstName;
lastName LastName;
...
state State;
end
```

```
DataItem CustomerId int
{ column=CUSTOMER_ID,
  range=(1, 1000),
  displayName="Customer Number",
  format= "#####" }
end
```

```
DataItem State char(2)
{ displayName="State Abbr.",
  format= "AA",
  validatorTable=StateTable }
end
```

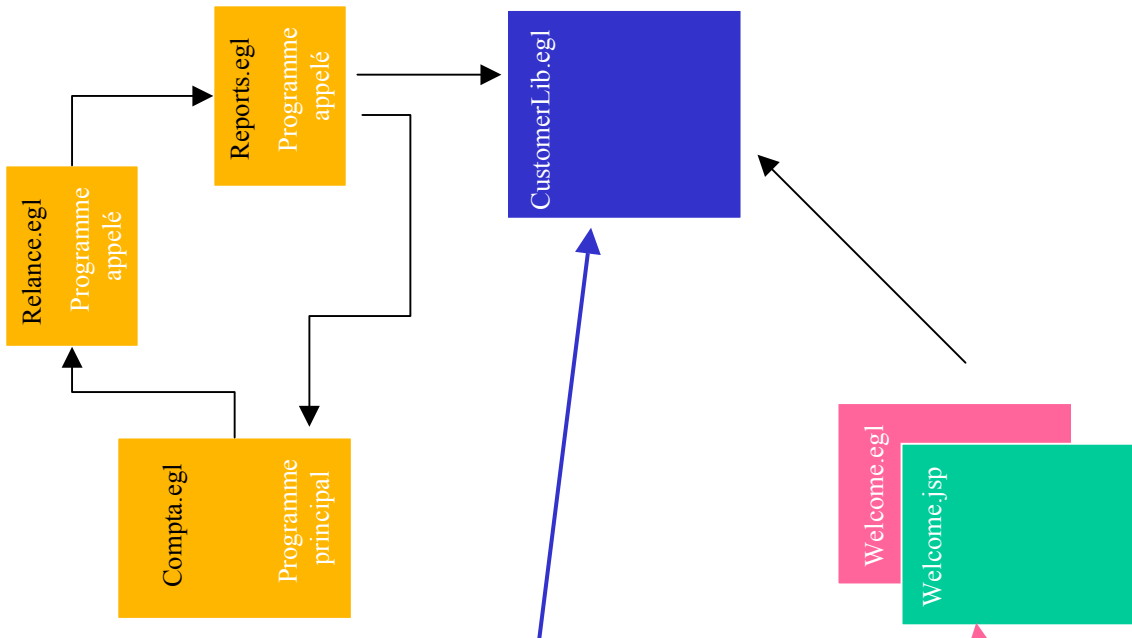
```
// Customer Stores SQLRecord
Record Stores SQLRecord
...
state State;
end
```

```
DataTable StateTable type matchValidTable
StateAbbreviation char(3);
{ contents = [{"NC"}, {"MN"}, {"TX"}, {"VA"} ...] }
end
```



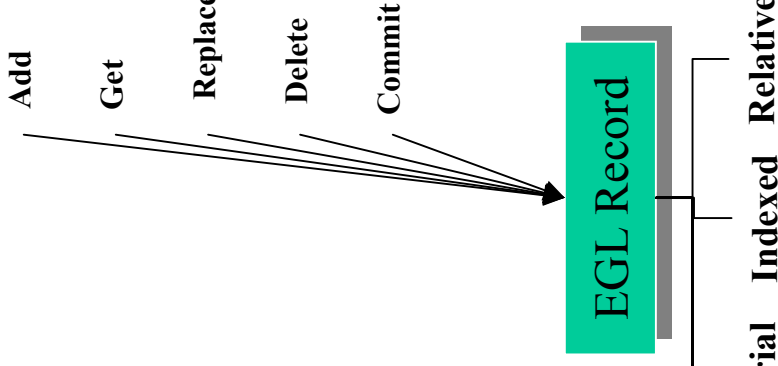
Architecture des Application EGL

- Programmes classiques
 - Point d'entrée unique
 - Mode texte, Batch
- Librairies
 - Points d'entrées multiples
 - Contient la "Business Logic"
 - Contient les fonctions et déclarations de données
- Gestionnaires de pages WEB
 - Contient les données liées aux fichiers .jsp
 - Fonctions "On Page Load"
 - Déclaration de fonctions associées à des boutons
 - Principalement la logique de "Controlle"



Accès aux données

- Offre une même couche d'abstraction vis à vis de:
 - Bases de Données Relationnelles
 - Messages MQ
 - Fichiers séquentiels Indexés, Fichiers plats
 - DL/I (v6.0)
- Ensemble commun de verbes pour l'accès aux données
 - Connect, Add, Get, Replace, Delete, Commit ...
- Erreurs d'I/O gérées de façon identique quelle que soit l'origine et la disparité des types de données
- Conversion de données effectuée automatiquement



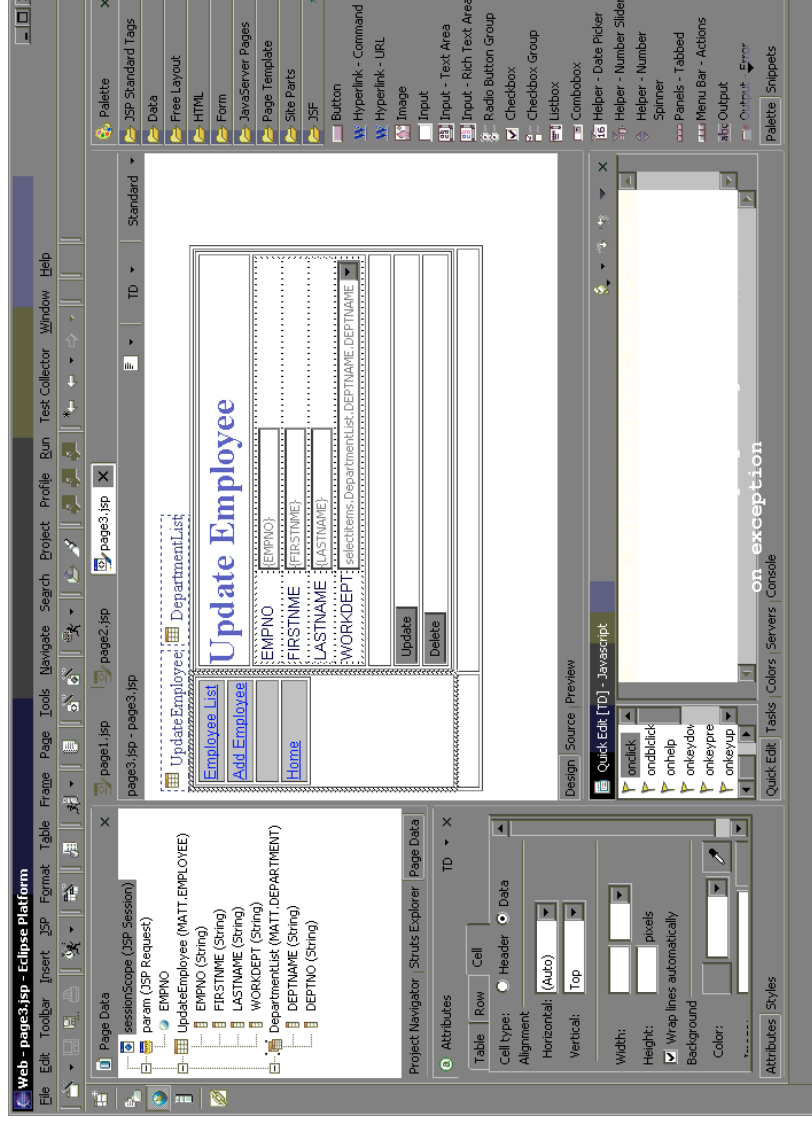
EGL dispose d'un Designer/Java Server Faces

Création des pages JSP

- ▶ Par simple drag'n drop
- ▶ A chaque page JSP correspond une page EGL (pagehandler)
- ▶ Fournit une fonction "onPageLoad"

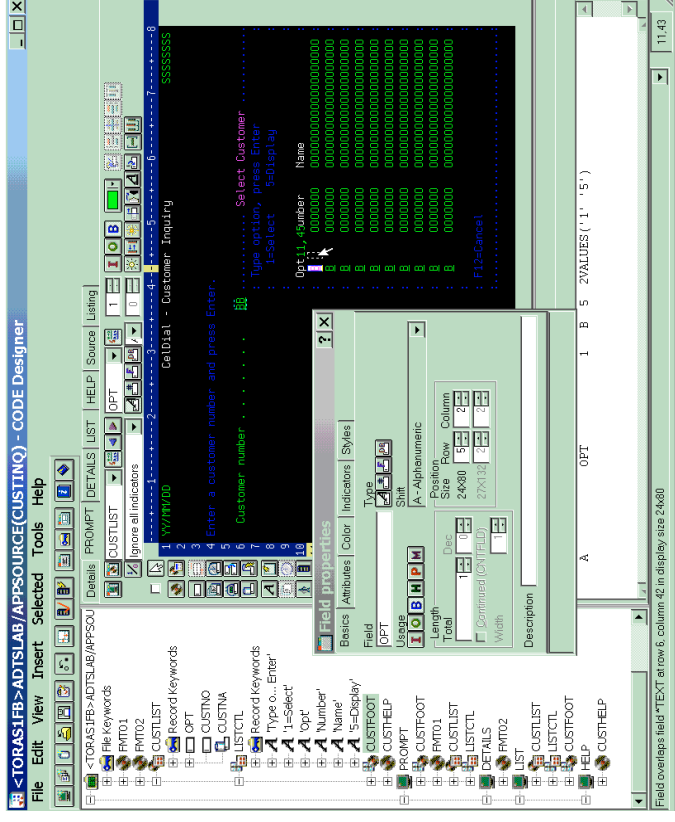
EGL met en oeuvre les principes du développement MVC

Séparation de la présentation / logique métier



EGL et les applications traditionnelles ?

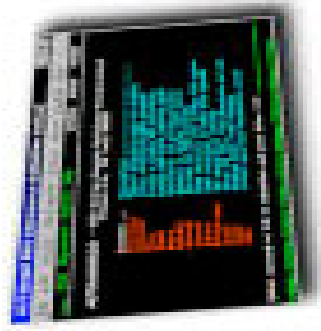
- Appels natifs aux applications COBOL, RPG, C, ou Java depuis une application EGL via l'instruction "Call"
- Support du mode caractère pour:
- Support graphique "Client Lourd" pour 2006



La famille Rational Developer

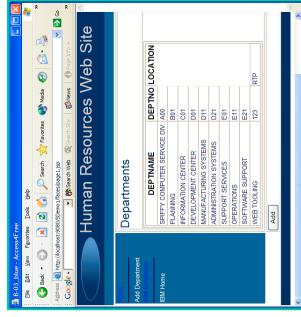


Déploiement de l'application



Runtime Java

- JRE pour Windows, Unix, Macintosh ...
- Gratuit
- Fourni par Sun, IBM, ...



Serveur d'Applications JAVA

- WebSphere Express
- Apache Tomcat



EGL vs 4GL



EGL et 4GL : 2 langages très proches

- EGL et 4GL sont différents à l'origine
 - Cependant tous les 2 sont des langages procéduraux
 - Ils adressent tous les deux les interfaces caractères
 - Les LabsIBM (WebSphere) ont travaillé pour les rendre très proches
- Les mots clé 4GL ont tous un équivalent en EGL
 - Soit par similitude de nom
 - Soit par similitude de fonctionnalité
- Les types 4GL ont tous un équivalent en EGL
 - Notamment pour les types SGBD propriétaires
- Tout est désormais réunis pour que la migration soit possible
- Un utilitaire de conversion a été développé



Le point sur La Migration 4GL -> EGL

À 14h00

