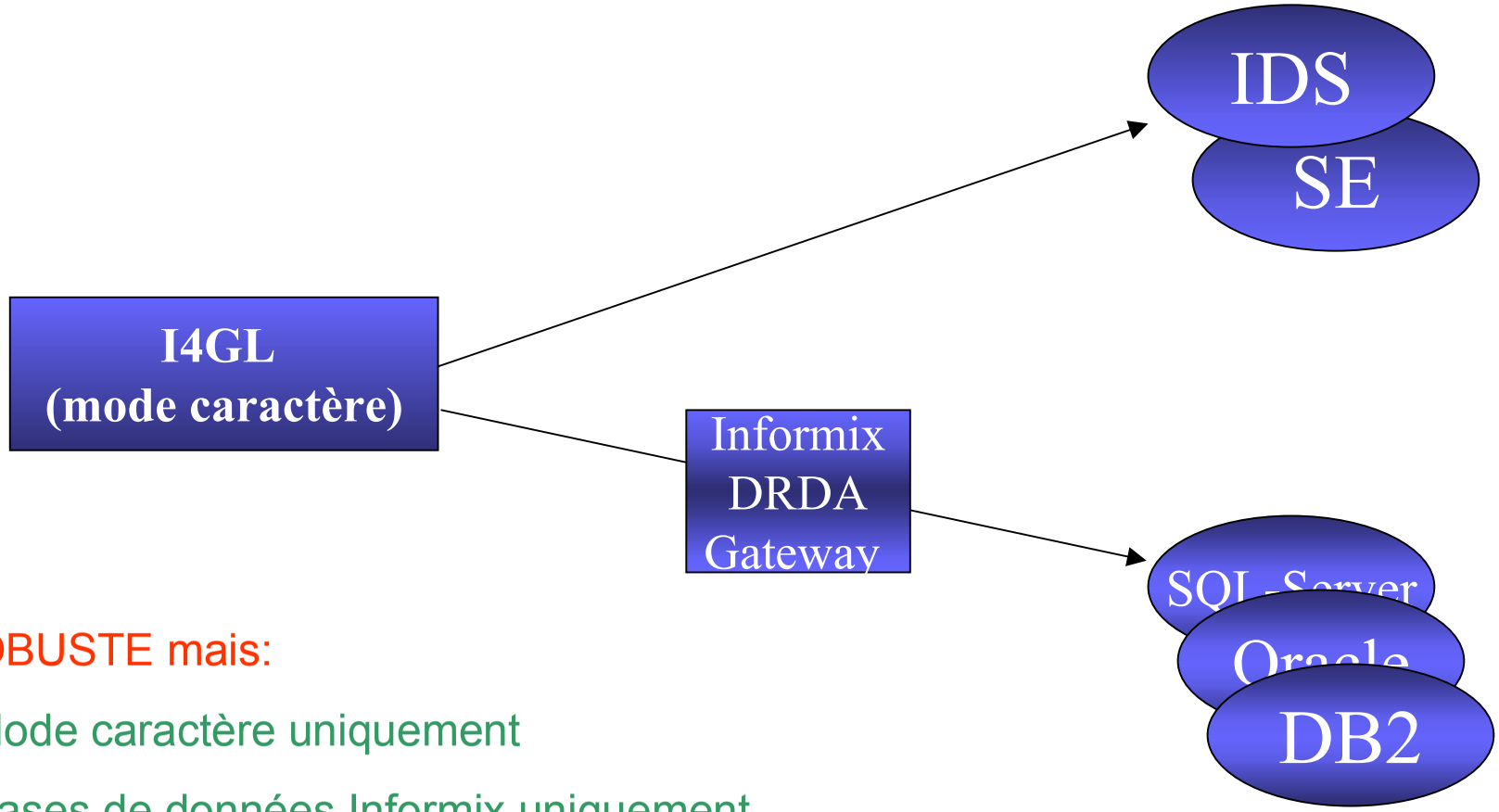
EGL

Outil de migration

Informix 4GL vers le monde Java

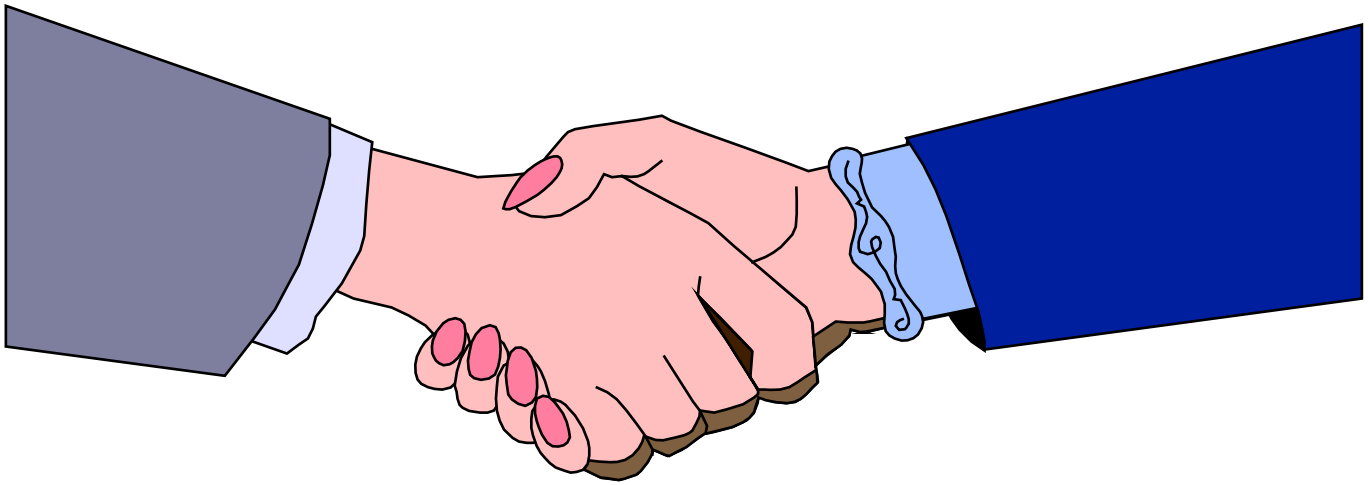
Aujourd'hui I4GL, V7.32.xC2



ROBUSTE mais:

- Mode caractère uniquement
- Bases de données Informix uniquement
- Essentiellement sous Unix
- Langage qui n'évolue plus (essentiellement des correctifs)

Integration Informix 4GL / IBM WebSphere

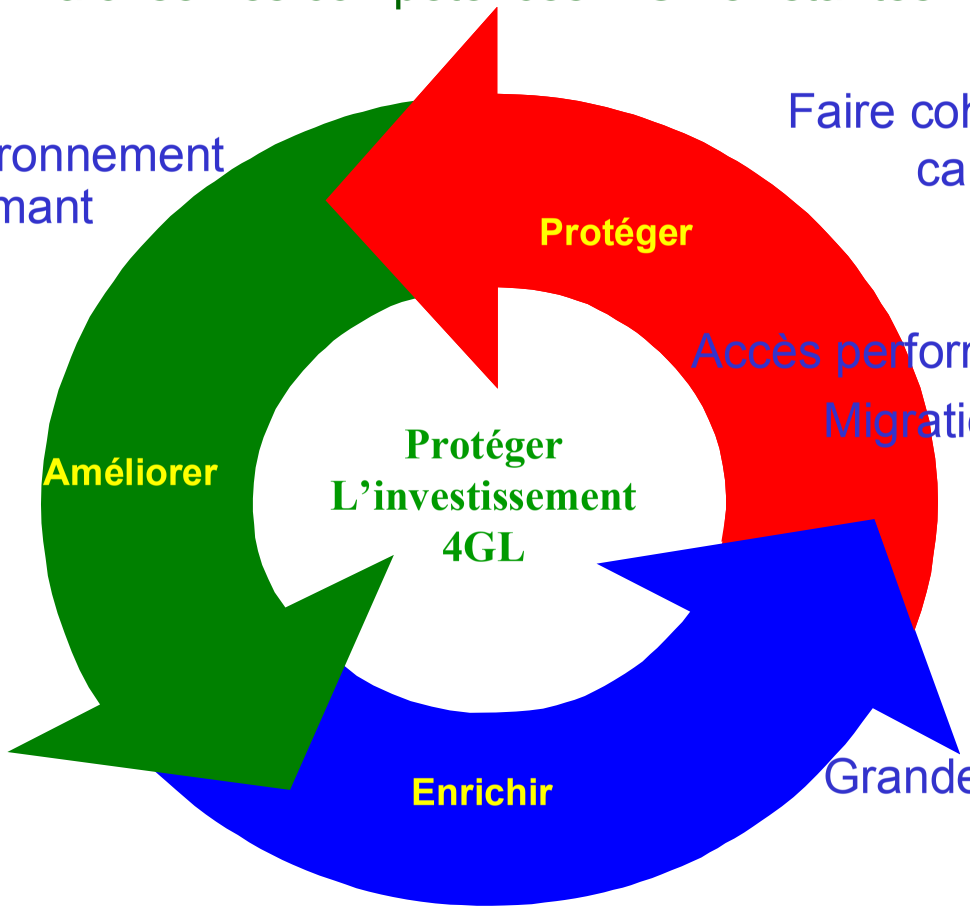


La mission d'IBM

Valoriser les compétences I4GL existantes

Faire cohabiter applications caractères et WEB

Fournir un environnement RAD performant



Fournir les meilleures technologies de reporting

Grande variété de types de déploiements

Outil de développement sophistiqué pour les applications WEB

Protéger l'existant

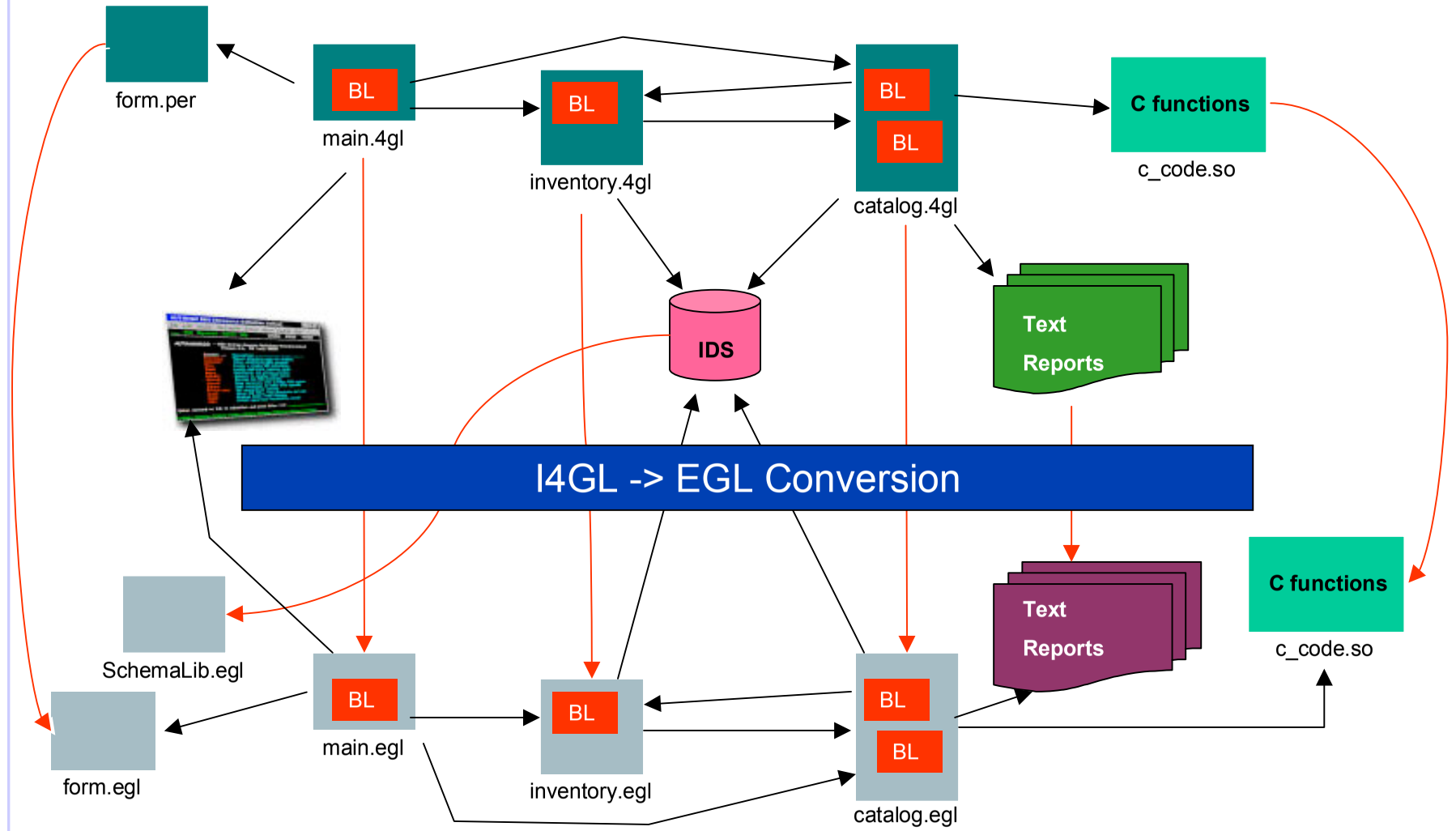
- Valoriser les compétences 4GL existantes
  - EGL est un langage simple, robuste et de haut niveau.
  - Il comprend une majorité de constructions 4GL (toutes à terme)
- Support des interfaces caractères et Web
  - Mode 3270, 5250, Unix et Windows
  - Mais également interface Web (JSP's)
- Amélioration de l'accès aux SGBD IBM
  - Travail important réalisé au niveau des drivers JDBC vis à vis des bases Informix et DB2
  - Support de n'importe quelle database fournissant un driver JDBC
- Support des 4GL Forms
  - Les I4GL forms seront transformées en EGL forms
- Outils de migration automatique 4GL -> EGL
  - Plus de 95% du code sera repris automatiquement

Enrichir l'existant

- Outil sophistiqué de développement Web
  - Complètement intégré à Rational Web Developper
  - Permet le développement en "Equipe" (repository avec CVS)
  - fonctionnalités de debugging
  - Designer de pages WEB
  - Outil de reporting (JasperReports)
- Nombreuses options de déploiement
  - En Java vers Windows, Unix, iSeries … (mode caractère)
  - J2EE (Servlets, JSP, EJB, Web Services)
  - Connexion à tout SGBDR disposant d'une interface JDBC

La Migration

Programme I4GL typique



Quelle stratégie de migration ?

- **Le but initial est de convertir une application I4GL en application EGL équivalente:**
  - Même périphérique d'affichage (mode caractère)
  - Même serveur de base de données
- **La conversion s'effectue programme par programme, grâce à des utilitaires de conversion:**
  - Soit depuis l'IDE (Rational Web Developer)
  - Mais également depuis la ligne de commande
- **On distinguera différentes phases de migration:**
  - Extraction des métadonnées provenant de la database
  - I4GL source code
  - Librairies "C"
- **La conversion sera automatique pour la plupart du code**
  - Il ne faut toutefois pas exclure d'intervenir manuellement sur certains programmes

Conversion depuis Rational Web Developer

- Exécuté directement depuis l'environnement de développement EGL (Windows ou Linux)
    - Le répertoire contenant les fichiers source I4GL doit être accessible (montage NFS ou recopie)
- L'assistant de migration utilisera les informations telles que:
    - I4GL source and EGL destination directories
    - I4GL source files (.4gl, .per, .c, .so, message files, etc)
    - database connection information
    - locale information
- Il générera ensuite un fichier de configuration au format XML
- Il invoquera finalement l'utilitaire de conversion
    - Affichage des logs à la fin de la conversion

EGL et les fonctions C

- **EGL supporte le format de la pile d'appel I4GL**
- **Celà permet d'utiliser les mêmes fonctions C qu'en I4GL**
  - L'accès via Java Native Interface (JNI) permet au code Java généré d'appeler le code C
  - Le code C doit ensuite être linké avec la librairie EGL
- Les fonctions C appelées par des programmes EGL doivent être dans des librairies partagées
  - Le fichier "Makefile" permettant de construire la librairie partagée sera générée par l'assistant de migration.
- **ESQL/C functions**
  - L'utilisation de code ESQL/C est interdit (une connexion ne pouvant être partagée simultanément par du code C et du code EGL).

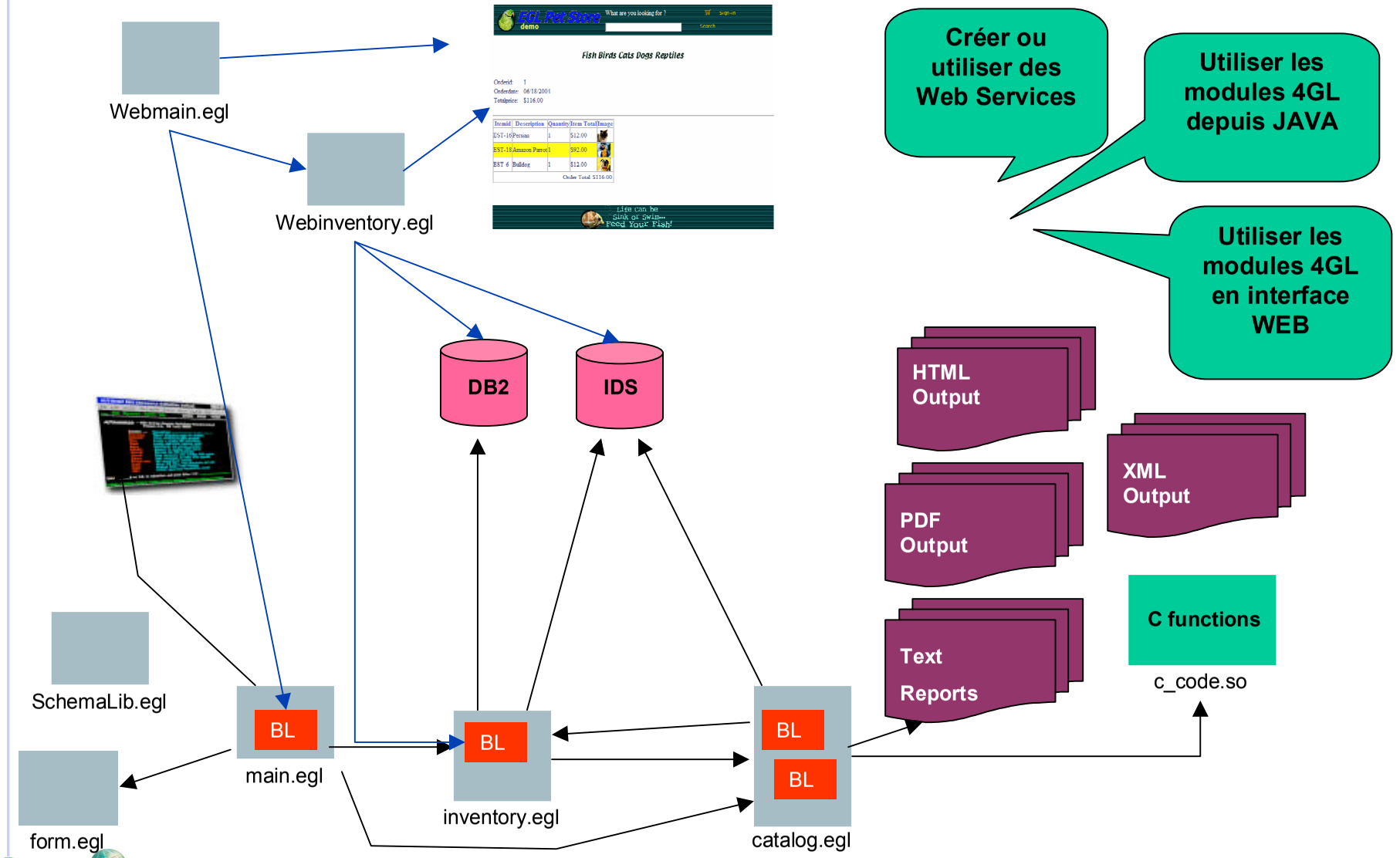
Ordre de Conversion

- **Database metadonnées**
  - Création d'un package EGL distinct pour chaque Base de données utilisées dans le projet.
- **Shared libraries**
  - La conversion génère le fichier makefile
- **Fichiers sources I4GL**
  - Fichiers .4gl
  - Fichiers .per (formes)
  - Fichiers messages

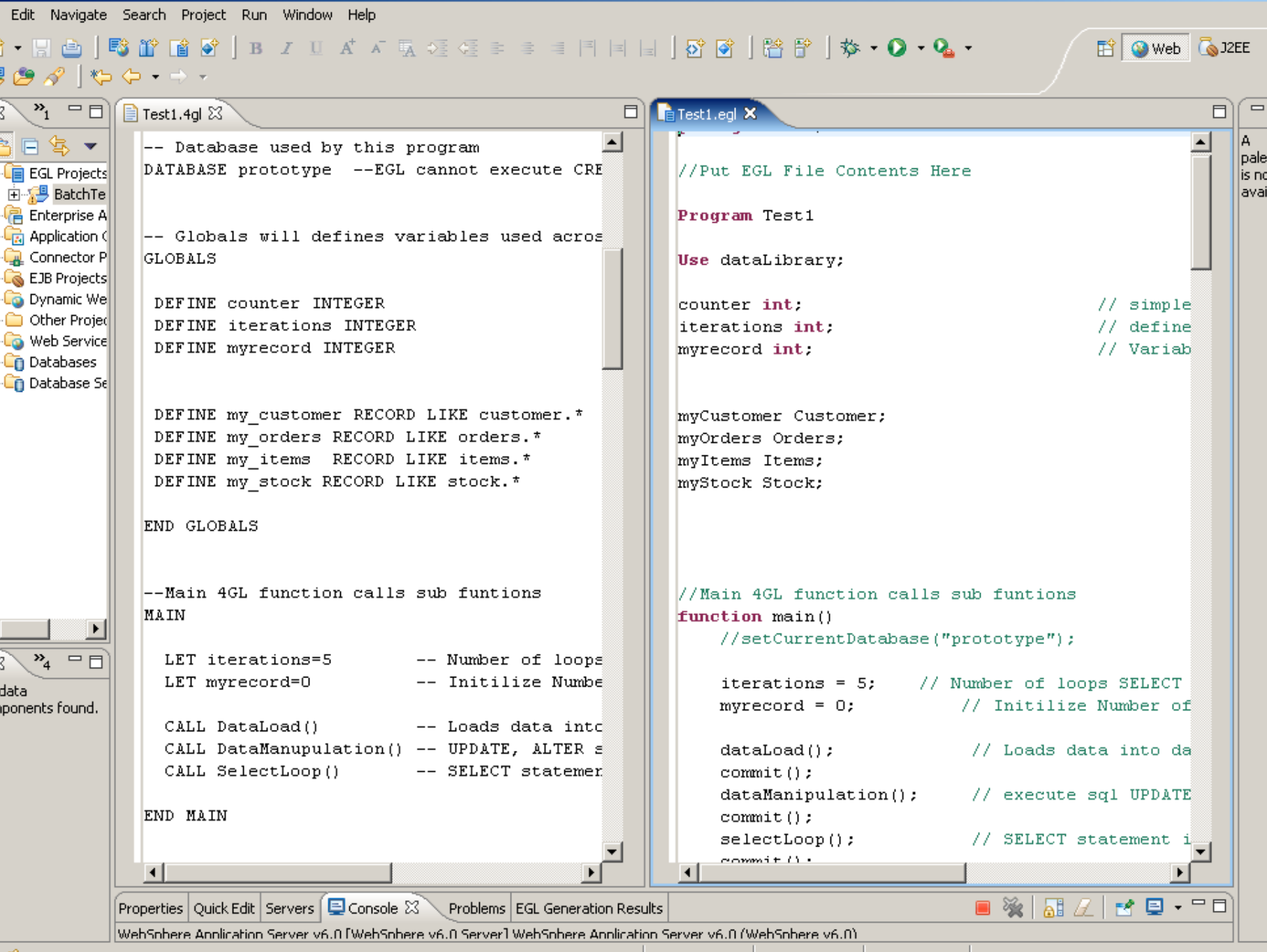
Limitations de l'utilitaire de Conversion

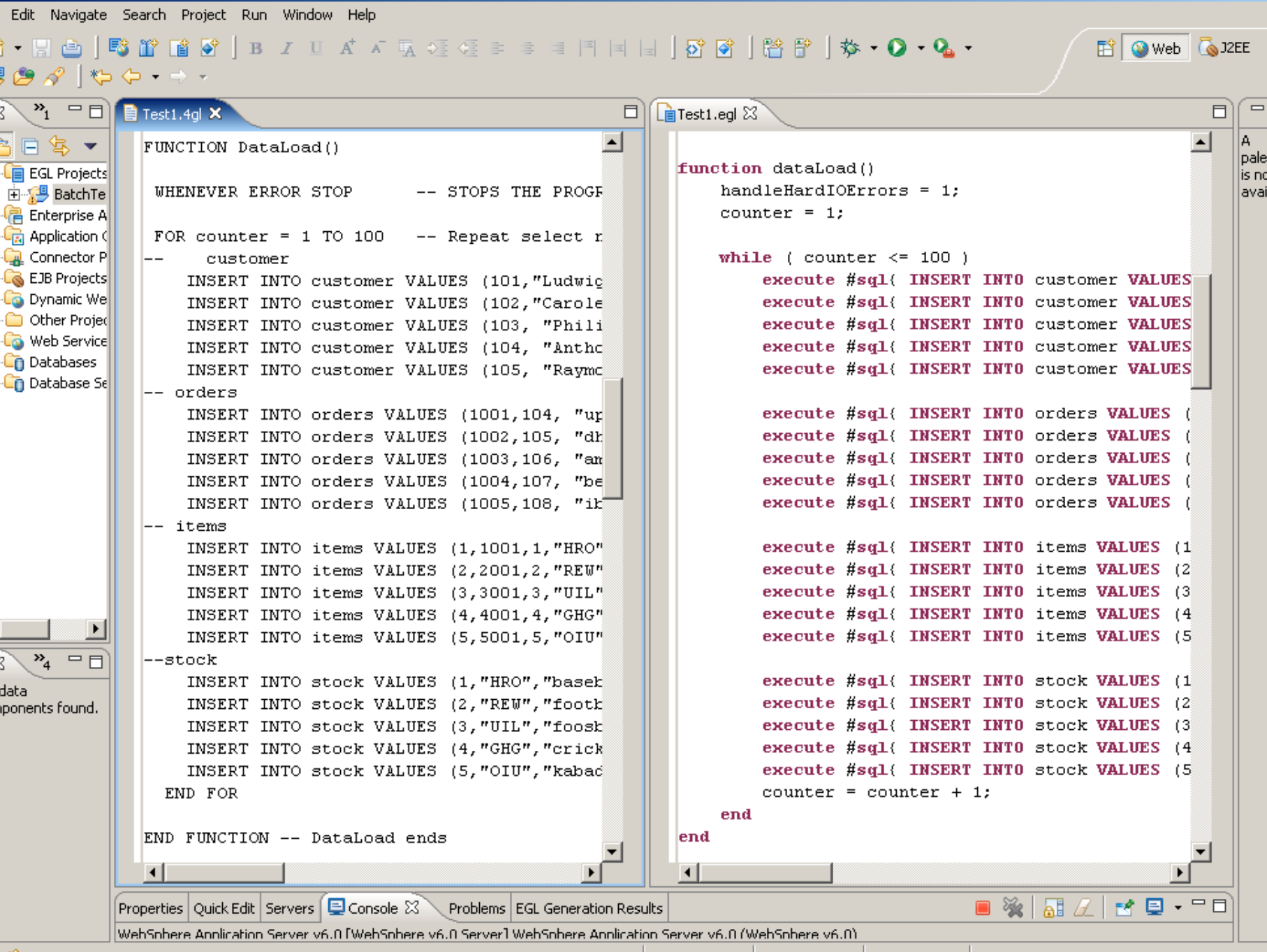
- Les programmes EGL générés sont en mode caractère
- Les connexions Database ne sont pas partageables entre le code EGL et le code C
- Le code C ne peut pas appeler du code EGL
- Les tableaux dynamiques I4GL (release 7.32) ne sont pas convertis dans la version actuelle.

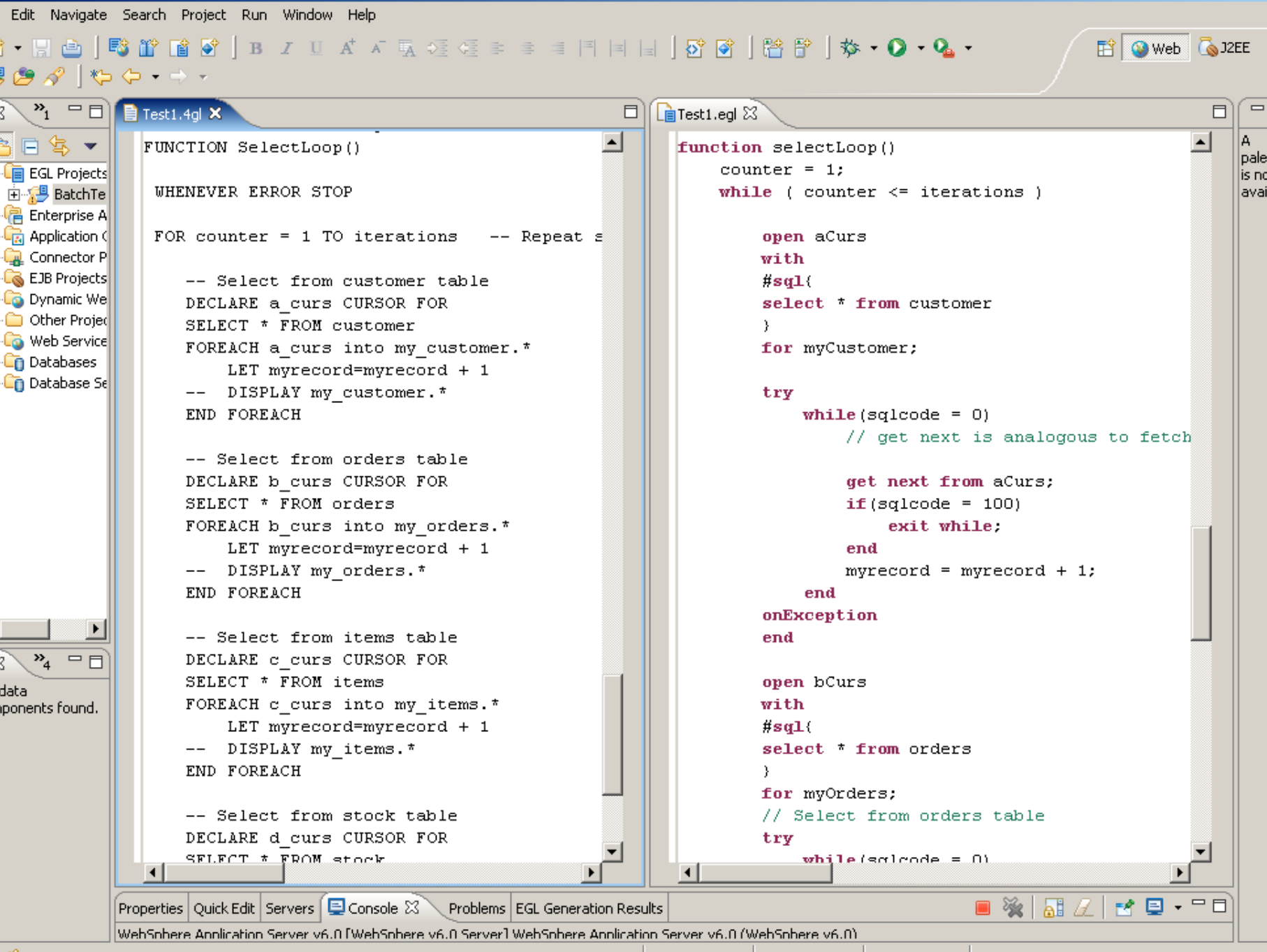
Que peut-on faire ?



Rapide Comparatif 4GL / EGL


Edit Navigate Search Project Run Window Help
Web
J2EE
Test1.4gl
-- Database used by this program
DATABASE prototype --EGL cannot execute CRE
-- Globals will defines variables used acros
GLOBALS
DEFINE counter INTEGER
DEFINE iterations INTEGER
DEFINE myrecord INTEGER
DEFINE my_customer RECORD LIKE customer.*
DEFINE my_orders RECORD LIKE orders.*
DEFINE my_items RECORD LIKE items.*
DEFINE my_stock RECORD LIKE stock.*
END GLOBALS
--Main 4GL function calls sub funtions
MAIN
LET iterations=5 -- Number of loops
LET myrecord=0 -- Initilize Numbe
CALL DataLoad() -- Loads data into
CALL DataManupulation() -- UPDATE, ALTER s
CALL SelectLoop() -- SELECT statemer
END MAIN
Test1.egl
//Put EGL File Contents Here
Program Test1
Use dataLibrary;
counter int; // simple
iterations int; // define
myrecord int; // Variab
myCustomer Customer;
myOrders Orders;
myItems Items;
myStock Stock;
//Main 4GL function calls sub funtions
function main()
//setCurrentDatabase("prototype");
iterations = 5; // Number of loops SELECT
myrecord = 0; // Initilize Number of
dataLoad(); // Loads data into da
commit();
dataManipulation(); // execute sql UPDATE
commit();
selectLoop(); // SELECT statement i
EGL Projects
BatchTe
Enterprise A
Application
Connector P
EJB Projects
Dynamic We
Other Projec
Web Service
Databases
Database Se
data
ponents found.
Properties
Quick Edit
Servers
Console
Problems
EGL Generation Results
WebSphere Application Server v6.0 [WebSphere v6.0 Server] WebSphere Application Server v6.0 (WebSphere v6.0)


Edit Navigate Search Project Run Window Help
Web
J2EE
Test1.4gl
FUNCTION DataLoad()
WHENEVER ERROR STOP -- STOPS THE PROGR
FOR counter = 1 TO 100 -- Repeat select r
-- customer
INSERT INTO customer VALUES (101,"Ludwig
INSERT INTO customer VALUES (102,"Carole
INSERT INTO customer VALUES (103, "Phili
INSERT INTO customer VALUES (104, "Anthc
INSERT INTO customer VALUES (105, "Raymc
-- orders
INSERT INTO orders VALUES (1001,104, "up
INSERT INTO orders VALUES (1002,105, "dh
INSERT INTO orders VALUES (1003,106, "am
INSERT INTO orders VALUES (1004,107, "be
INSERT INTO orders VALUES (1005,108, "ik
-- items
INSERT INTO items VALUES (1,1001,1,"HRO"
INSERT INTO items VALUES (2,2001,2,"REW"
INSERT INTO items VALUES (3,3001,3,"UIL"
INSERT INTO items VALUES (4,4001,4,"GHG"
INSERT INTO items VALUES (5,5001,5,"OIU"
--stock
INSERT INTO stock VALUES (1,"HRO","baseb
INSERT INTO stock VALUES (2,"REW","footb
INSERT INTO stock VALUES (3,"UIL","foosb
INSERT INTO stock VALUES (4,"GHG","crick
INSERT INTO stock VALUES (5,"OIU","kabac
END FOR
END FUNCTION -- DataLoad ends
Test1.egl
function dataLoad()
handleHardIOErrors = 1;
counter = 1;
while (counter <= 100)
execute #sql{ INSERT INTO customer VALUES
execute #sql{ INSERT INTO customer VALUES
execute #sql{ INSERT INTO customer VALUES
execute #sql{ INSERT INTO customer VALUES
execute #sql{ INSERT INTO customer VALUES
execute #sql{ INSERT INTO orders VALUES (
execute #sql{ INSERT INTO orders VALUES (
execute #sql{ INSERT INTO orders VALUES (
execute #sql{ INSERT INTO orders VALUES (
execute #sql{ INSERT INTO orders VALUES (
execute #sql{ INSERT INTO items VALUES (1
execute #sql{ INSERT INTO items VALUES (2
execute #sql{ INSERT INTO items VALUES (3
execute #sql{ INSERT INTO items VALUES (4
execute #sql{ INSERT INTO items VALUES (5
execute #sql{ INSERT INTO stock VALUES (1
execute #sql{ INSERT INTO stock VALUES (2
execute #sql{ INSERT INTO stock VALUES (3
execute #sql{ INSERT INTO stock VALUES (4
execute #sql{ INSERT INTO stock VALUES (5
counter = counter + 1;
end
end
EGL Projects
Enterprise A
Application (
Connector P
EJB Projects
Dynamic We
Other Projec
Web Service
Databases
Database Se
Properties
Quick Edit
Servers
Console
Problems
EGL Generation Results
WebSphere Application Server v6.0 [WebSphere v6.0 Server] WebSphere Application Server v6.0 (WebSphere v6.0)


Edit Navigate Search Project Run Window Help
Web
J2EE
Test1.4gl
FUNCTION SelectLoop()
WHENEVER ERROR STOP
FOR counter = 1 TO iterations -- Repeat
-- Select from customer table
DECLARE a_curs CURSOR FOR
SELECT * FROM customer
FOREACH a_curs into my_customer.*
LET myrecord=myrecord + 1
-- DISPLAY my_customer.*
END FOREACH
-- Select from orders table
DECLARE b_curs CURSOR FOR
SELECT * FROM orders
FOREACH b_curs into my_orders.*
LET myrecord=myrecord + 1
-- DISPLAY my_orders.*
END FOREACH
-- Select from items table
DECLARE c_curs CURSOR FOR
SELECT * FROM items
FOREACH c_curs into my_items.*
LET myrecord=myrecord + 1
-- DISPLAY my_items.*
END FOREACH
-- Select from stock table
DECLARE d_curs CURSOR FOR
SELECT * FROM stock
Test1.egl
function selectLoop()
counter = 1;
while (counter <= iterations)
open aCurs
with
#sql{
select * from customer
}
for myCustomer;
try
while(sqlcode = 0)
// get next is analogous to fetch
get next from aCurs;
if(sqlcode = 100)
exit while;
end
myrecord = myrecord + 1;
end
onException
end
open bCurs
with
#sql{
select * from orders
}
for myOrders;
// Select from orders table
try
while(sqlcode = 0)
EGL Projects
BatchTe
Enterprise A
Application
Connector P
EJB Projects
Dynamic We
Other Projec
Web Service
Databases
Database Se
data
ponents found.
Properties
Quick Edit
Servers
Console
Problems
EGL Generation Results
WebSphere Application Server v6.0 [WebSphere v6.0 Server] WebSphere Application Server v6.0 (WebSphere v6.0)

La migration en Images