|  |  |
| --- | --- |
|  | **2017** |
|  | Catamania  Adil Mesfioui |

|  |
| --- |
| **[Formation Progress]** |
| Le but de ce document est de fournir une formation initiale au Langage Progress. Ce document aborde un ensemble de points qui permettent d’être rapidement opérationnel. |

Sommaire

[Installation 3](#_Toc491937451)

[Les instructions Caractères 3](#_Toc491937452)

[Horodatage 3](#_Toc491937453)

[Conditions 3](#_Toc491937454)

[Boucles 3](#_Toc491937455)

[SmartObject 3](#_Toc491937456)

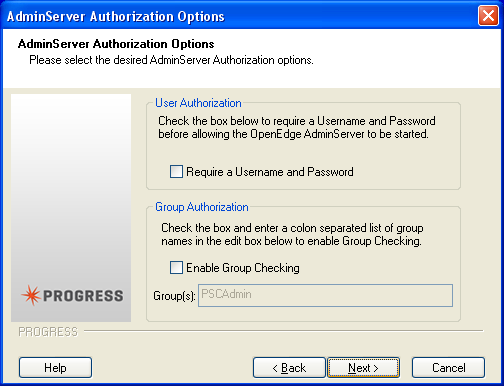
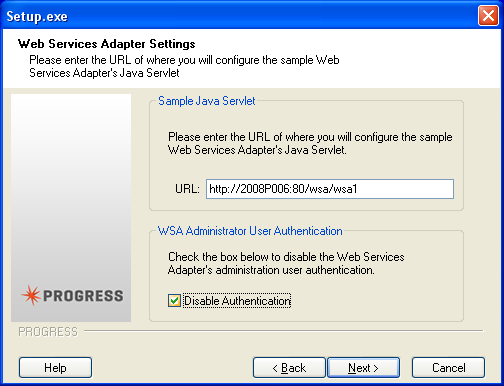
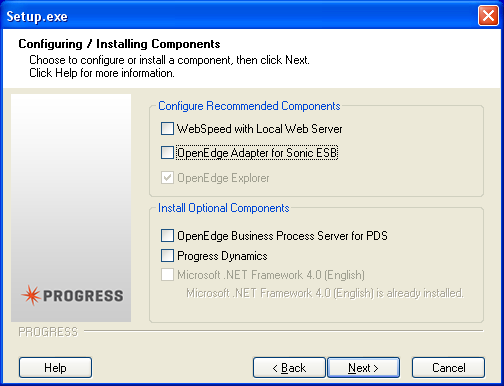
[Exercices 3](#_Toc491937457)

## Installation

Avant tout, il faut se connecter au site OpenEdge, se créer un compte et télécharger l’installateur pour l’environnement de développement.

Récupérer les clés pour les produits suivants :

1. OpenEdge Enterprise RDBMS  
2. Progress Developer Studio for OpenEdge (formerly OpenEdge Architect)  
3. OpenEdge Management

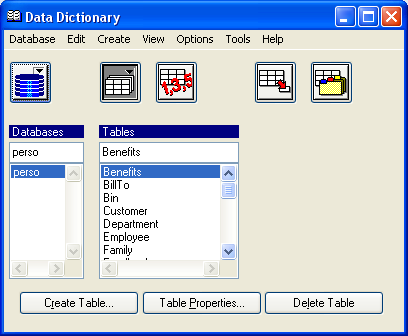
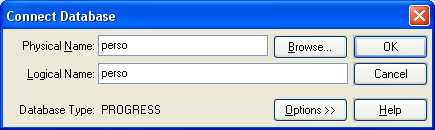
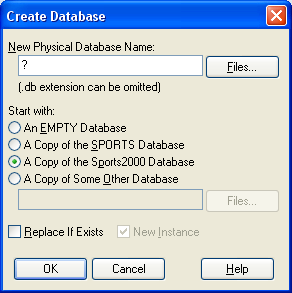


Création de la base de données

Il n’y a pas de base de données configurée. Il faut en créer une avant.

Desktop 🡺 pour permettre de créer une base de données

File -> New  
Tools -> Data Dictionnary



License Update 🡺 pour mettre à jour les licences de Progress pour 60 jours de plus

## Les instructions Caractères

Le language Progress fournit tout un ensemble de fonctions de caractères mais nous ne les aborderons pas toutes.

SUBSTRING  
INDEX : indique la position de la chaîne recherché dans la chaîne source.  
NUM-ENTRIES : nombre d’entrées dans une liste, chaque element devant être séparé par un séparateur (par défaut une virgule).  
ENTRY : rechercher une entrée en particulier dans une liste.  
LOOKUP  
  
Exemple d’horodatage :

## Horodatage

STRING(YEAR(vDate), "9999":U)

STRING(MONTH(vDate), "99":U)

STRING(DAY(vDate), "99":U)

SUBSTRING(STRING(TIME,"HH:MM:SS"),1,2)

SUBSTRING(STRING(TIME,"HH:MM:SS"),4,2)

SUBSTRING(STRING(TIME,"HH:MM:SS"),7,2)

## Conditions

IF THEN DO :  
END.  
ELSE DO:  
END.  
  
CASE “variable”:  
 WHEN “” THEN DO:  
 END.  
 OTHERWISE DO:  
 END.  
END CASE.

Boucles  
  
FOR EACH ….. NO-LOCK/EXCLUSIVE-LOCK/SHARE-LOCK :  
END.  
A la suite d’un FOR EACH on n’est plus available sur le buffer.  
  
FOR FIRST …. NO-LOCK/EXCLUSIVE-LOCK/SHARE-LOCK :  
END.  
A la suite on est encore available si on a trouvé quelque chose.  
  
FOR LAST …. NO-LOCK/EXCLUSIVE-LOCK/SHARE-LOCK :  
END.  
  
FIND … NO-LOCK / EXCLUSIVE-LOCK / SHARE-LOCK NO-ERROR.  
FIND FIRST NO-LOCK / EXCLUSIVE-LOCK / SHARE-LOCK NO-ERROR.  
FIND LAST NO-LOCK / EXCLUSIVE-LOCK / SHARE-LOCK NO-ERROR.  
  
IF AVAILABLE/NOT AVAILABLE THEN …  
  
DO:  
END.  
  
DO WHILE:  
END.

REPEAT:  
END.

## Procédures / Fonctions

Pour déclarer une procedure il faut utiliser la méthode suivante :

PROCEDURE <nom-procédure> :

END PROCEDURE.

L’appel se fera de la façon suivante :   
  
RUN <nom-procédure>.

Pour declarer des parametres d’entrées à cette procedure :

PROCEDURE <nom-procédure> :  
**DEFINE INPUT PARAMETER <nom-variable> AS <type-de-données> NO-UNDO.**

END PROCEDURE.

L’appel se fera de la façon suivante :   
  
RUN <nom-procédure> (INPUT **<nom-variable>**).

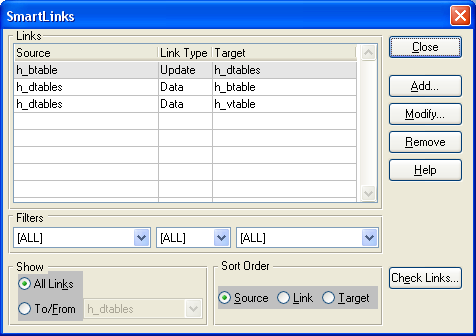
Pour declarer des parameters de sortie à cette procedure :

PROCEDURE <nom-procédure> :  
DEFINE INPUT PARAMETER <nom-variable> AS <type-de-données> NO-UNDO. **DEFINE OUTPUT PARAMETER <nom-variable-2> AS <type-de-données> NO-UNDO.**

END PROCEDURE.

L’appel se fera de la façon suivante :   
  
RUN <nom-procédure> (INPUT **<nom-variable>, OUTPUT <nom-variable-2>**).

## SmartObject



Entre chaque SmartObjects, il y a un lien dit SmartLink. Dans les exemples ci-dessus, il y a un lien de type (Link Type) Data entre le SmartDataObject et le SmartDataBrowser.

Dans Progress « Standard », on doit declarer un SDO (SmartData Object) pour lister les données dont on a besoin (lecture ou écriture). Puis le lier au SmartDataBrowser puis au SmartDataViewer.

## Exercices

- Lister tous les clients aux USA.  
- Compter tous les clients aux USA.  
- Cumuler toutes les « balances » des clients aux USA.  
- Cumuler toutes les « balances » des clients dans tous les pays (Break-By).

- Lister les lignes de commandes de toutes les commandes.  
- Lister les lignes de commandes de la commande x :

FOR EACH order WHERE order.ordernum = 1,  
EACH orderline OF order :  
 DISPLAY orderline.  
END.

- Lister les articles des lignes de commandes de la commande x :

FOR EACH order WHERE order.ordernum = 1,  
EACH orderline OF order :  
 FOR EACH item WHERE itemnum = orderline.item NO-LOCK :  
 DISP item.  
 END.  
END.

- Faire la même chose dans une procédure :   
  
RUN P01-test.  
PROCEDURE P01-test :  
 FOR EACH order WHERE order.ordernum = 1,  
 EACH orderline OF order :  
 FOR EACH item WHERE itemnum = orderline.item NO-LOCK :  
 DISP item.  
 END.  
 END.  
  
END PROCEDURE.

RUN P01-test(input 2).  
PROCEDURE P01-test :  
 DEFINE INPUT PARAMETER iordernum AS INTEGER NO-UNDO.

FOR EACH order WHERE order.ordernum = iordernum,  
 EACH orderline OF order :  
 FOR EACH item WHERE itemnum = orderline.item NO-LOCK :  
 DISP item.  
 END.  
 END.  
END PROCEDURE.

Exemple avec de procédure / fonction

RUN P01-test(INPUT 2).  
RUN P02-test(OUTPUT vret).  
  
PROCEDURE P01-test :  
 DEFINE INPUT PARAMETER iordernum AS INTEGER NO-UNDO.

FOR EACH order WHERE order.ordernum = iordernum,  
 EACH orderline OF order :  
 FOR EACH item where itemnum = orderline.item NO-LOCK :  
 DISP item.  
 END.  
 END.

END PROCEDURE.

PROCEDURE P02-test :  
 DEFINE OUTPUT PARAMETER oret AS LOGICAL NO-UNDO.

ASSIGN oret = TRUE.

END PROCEDURE.

FUNCTION fo-test RETURNS INTEGER (INPUT iordernum AS INTEGER) FORWARD.

DISP fo-test (1).

FUNCTION fo-test RETURNS INTEGER (INPUT iordernum AS INTEGER) :  
 RETURN iordernum.

END FUNCTION.

* Les écrans

TP :

Créer un écran avec une combo box qui liste les clients, des champs Fill-In avec des éléments de la table Customer et un Browse avec les commandes du client.  
Sur l’évènement VALUE-CHANGED, alimentez les champs Adresse, Contact, Ville, Pays et Balance de la table Customer, et charger les commandes associées dans le Browse.  
  
Bonus : afficher une légende pour les différents types de status de commande et sur chaque changement de lignes, mettre en gras la légende correspondant.  
  
TP2 :

A partir du TP précédent, en sélectionnant une ligne de commande et en cliquant sur le bouton détail (à Ajouter), une fenêtre s’ouvre avec le contenu des commandes (ligne de commande OrderLine).

* Les SmartObjects

TP :

Effectuer les mêmes écrans que précédemment mais en smartobject.

## 

## Bugs

Pb lors de la création de la SmartDataObject :

- Click droit sur AppBuilder -> propriétés -> ajouter –s 200.