



# **Atelier N°1 :**

## **Débuter avec l'ETL SSIS**

# Sommaire

I.	Introduction .....	4
II.	L'ETL SQL Server Integration Services.....	4
II.1.	Unité de travail SSIS : Package.....	4
II.2.	Les éléments d'un package SSIS .....	5
II.3.	ControlFlow .....	5
II.4.	DataFlow .....	6
II.5.	Event Handlers.....	6
III.	Débuter avec SSIS 2008 .....	6
III.1.	Création d'un package SSIS .....	6
III.2.	Ajout de connexions au projet .....	7
III.3.	Création d'un flux de données Data Flow .....	11
III.4.	Configuration du flux de données « Data Flow » .....	11
III.5.	Extraction des données utilisant SSIS .....	14
IV.	Personnalisation de la tâche .....	14
IV.1.	Script SQL pour vider la table avant extraction .....	14
IV.2.	Script C# pour afficher un message de début du travail .....	15
IV.3.	Créer des colonnes dérivées .....	15
IV.4.	Exercice .....	17
V.	Charger plusieurs fichiers plats dans une table SQL Server .....	18
V.1.	Configuration de la boucle Foreach.....	18
V.2.	Configuration des flux de données.....	19
V.3.	Extraction des données.....	20
V.4.	Exercice .....	20
VI.	Extraire les données utilisant une requête SQL .....	21
VII.	Gestion des erreurs .....	23
VII.1.	Premier exemple.....	24
VII.2.	Gérer les erreurs.....	25
VII.3.	Gestion d'erreur dans le premier exemple .....	25
VII.4.	Gestion d'erreur avec « Event Handlers ».....	27



## **I. Introduction**

Microsoft, SQL Server 2008 est un système de gestion de bases de données relationnelles (SGBDR). Son moteur de base de données permet le stockage, la manipulation et l'analyse de données. En plus, grâce au langage T-SQL (Transact-SQL) il est possible d'utiliser les fonctions de la programmation procédurale sur le moteur de base de données.

Pour réaliser des projets Business Intelligence (BI), SQL Server 2008 contient trois plateformes en plus de son moteur, à savoir :

- SQL Server Integration Services (SSIS): joue le rôle d'un ETL. Il permet d'intégrer des données provenant de différentes sources pour les ranger dans un entrepôt de données (Data Warehouse)
- SQL Server Analysis Services (SSAS) : permet d'analyser les données, agrégées dans le Data Warehouse, grâce à des fonctions d'analyse multidimensionnelle
- SQL Server Reporting Services (SSRS) : permet de créer, gérer et publier des rapports résultant des analyses réalisées lors de SSAS

Dans ce TP et pour pouvoir faire du BI avec SQL Server, il faut installer les outils suivants :

- SQL Server 2008 Entreprise ou professionnel
- Visual Studio .Net 2008

## **II. L'ETL SQL Server Integration Services**

L'ETL a pour objectif d'intégrer des données à partir de différentes sources dans un entrepôt de données (Data Warehouse.). Ainsi, on peut avec SSIS extraire des données provenant de différentes sources (Excel, MySQL...) et les insérer dans un SGBD cible.

Normalement, un ETL (Extract-Transform-Loading) n'est utilisé qu'après avoir définie la structure du Data Warehouse.

### **II.1. Unité de travail SSIS : Package**

L'environnement de travail dans un projet SSIS est appelé un « Package ». Ainsi, le package contient tous les scénarios de 1 « Extraction, Transformation, Chargement ». On peut créer autant de packages qu'on souhaite pour plusieurs tâches d'ETL.

## **II.2.Les éléments d'un package SSIS**

Dans chaque package on trouve 4 parties sous forme d'onglets (dans un projet SSIS):

- ControlFlow (flux de contrôle) : partie orientée contrôle. Contient les différentes tâches du package (qu'on peut ordonner). Par exemple, faire une troncature d'une table avant de la remplir.
- DataFlow (flux de données) : comme le ControlFlow, mais dédié aux données. On peut retrouver dans le DataFlow les étapes de lecture, de modification et d'écriture des données.
- Event Handlers (Gestionnaires d'événements) : permettant de gérer des événements qui peuvent survenir dans les différents composants qui constituent un package SSIS et d'effectuer des actions en fonction de leur nature.
- Package Explorer (Explorateur de package) : décrit, sous forme d'arborescence, tous les éléments constituant le package et qui permet d'accéder en quelques clics à tous les éléments du package et à leurs propriétés.

## **II.3.ControlFlow**

C'est une partie orientée contrôle. Elle contient les différentes tâches du package (qu'on peut ordonner). Par exemple, faire une troncature d'une table avant de la remplir.

Parmi les tâches de contrôle, on trouve :

- Data Flow Task : encapsule le moteur DataFlow permettant l'extraction de données à partir d'une source de données et une source destination
- Script Task (Tâche de script) : permet de créer des fonctions (C# ou VB.Net)
- Execute SQL Task : permettant d'exécuter des instructions SQL
- XML Task : permettant l'exécution des opérations XML sur des sources de données de type XML
- ...

En plus de ces tâches, on trouve des tâches de maintenance :

- Check Database Integrity Task : permettant la vérification de l'intégrité de la base de données (pages de données et pages d'index)
- History Cleanup Task : permettant le nettoyage d'historique
- ...

Il est possible de regrouper certaines tâches entre elles dans ce qu'on appelle un conteneur. Il existe ainsi des conteneurs répétitifs qui permettent de créer une boucle qui exécutera leurs tâches un certain nombre de fois :

- For Loop Container,
- Foreach Loop Container,
- Sequence Container)

#### **II.4.DataFlow**

Pareil au ControlFlow, mais dédié aux données. On peut retrouver dans le DataFlow les étapes de lecture, de modification et d'écriture des données.

Cet onglet contient trois types de contrôles :

- Data Flow Sources : regroupe différents type de flux de données source. On y trouve sources de données de type Excel, Flat File, OLE DB, XML, ADO NET
- Data Flow Destination : regroupe différents types de flux de données destination.
- Data Flow Transformations : regroupe différent types de transformation: Agrégation, conversion de données, Merge, copie de colonnes

#### **II.5.Event Handlers**

Permettant de gérer des évènements qui peuvent survenir dans les différents composants qui constituent un package SSIS et d'effectuer des actions en fonction de leur nature. Ainsi, par exemple, on peut programmer des taches lors d'un événement « OnError ».

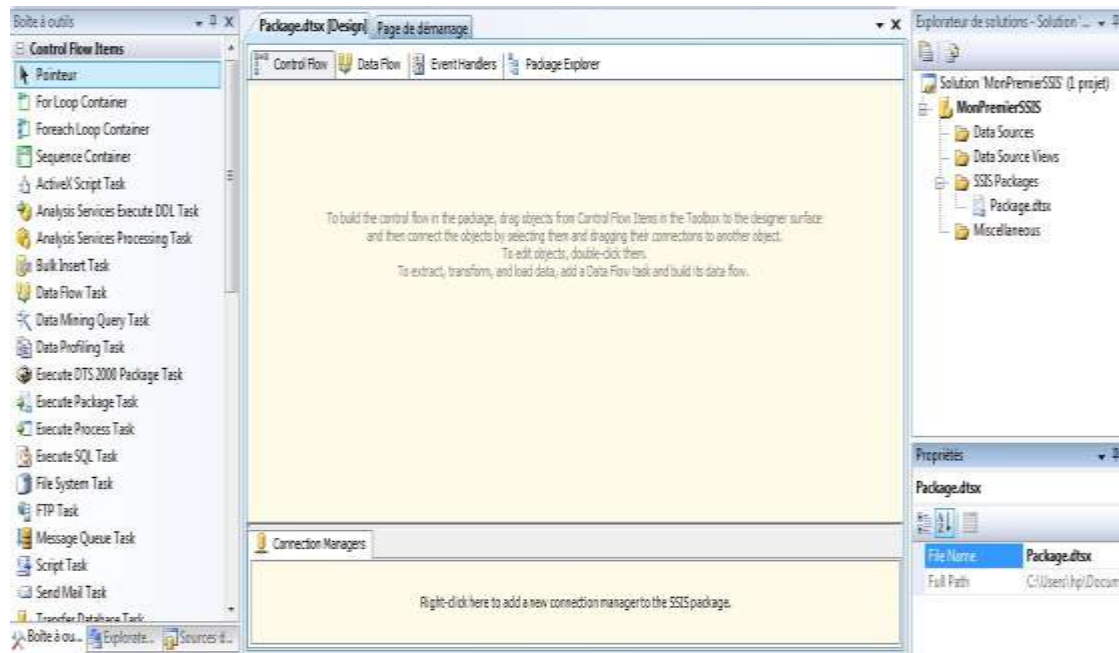
### **III. Débuter avec SSIS 2008**

Dans un premier exemple, on va essayer de créer un projet permettant l'extraction des données à partir d'un fichier plat vers une base de données SQL Server 2008

#### **III.1. Création d'un package SSIS**

1. Lancer MS Visual Studio 2008. Deux façons :
  - a. Cliquer le menu démarrer → Microsoft SQL Server 2008 → SQL Server Business Intelligence Development Studio
  - b. Cliquer le menu démarrer → MS Visual Studio 2008 → MS Visual Studio 2008
2. Cliquer le menu Fichier → Nouveau → Projet
3. Sélectionner le type de projet « Business Intelligence Projects »
4. Dans les modèles, sélectionner « Integration Services Project »

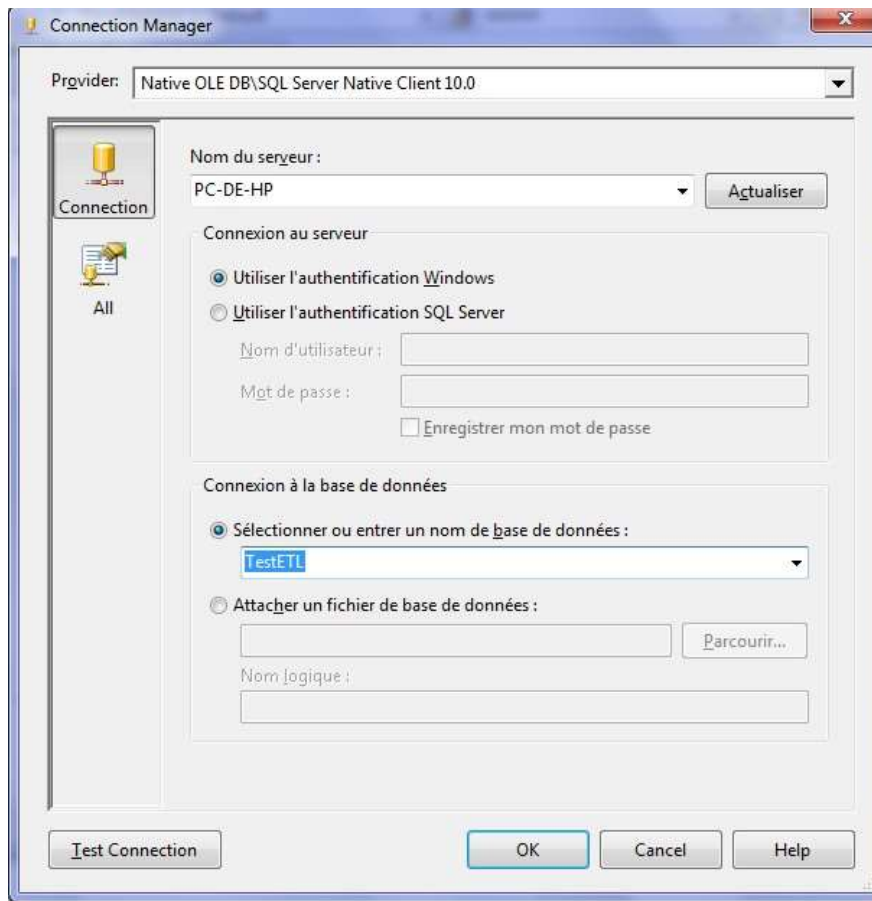
5. Entrer le nom du projet, l'emplacement et cliquer « OK » pour créer le projet SSIS
6. Une fois le projet créé, dans l'explorateur de solution, un package « Package.dtsx » est créé par défaut. On peut créer autant de packages que nécessaire.



### III.2. Ajout de connexions au projet

Pour exporter des données à partir d'une source de données, il faut créer une connexion vers la source. Pour le faire,

1. Sélectionner l'onglet « ControlFlow »,
2. Dans la « Connection Manager » en bas, cliquer bouton droit et sélectionner le type connexion.
3. Sélectionner une connexion de type « New OLE DB Connection ».
4. On peut créer une connexion à base des informations de connexion déjà créées. Sinon, cliquer « New » pour créer une nouvelle connexion.
5. Sélectionner le nom du serveur, le mode d'authentification et la base de données
6. Valider en cliquant « OK » pour créer une connexion vers la base de données.
7. Cliquer « OK » pour ajouter la connexion au projet



Remarque ; il est possible de changer le type connexion dans la zone « Provider »

Répéter les étapes mais cette fois pour ajouter une connexion vers un fichier plat.

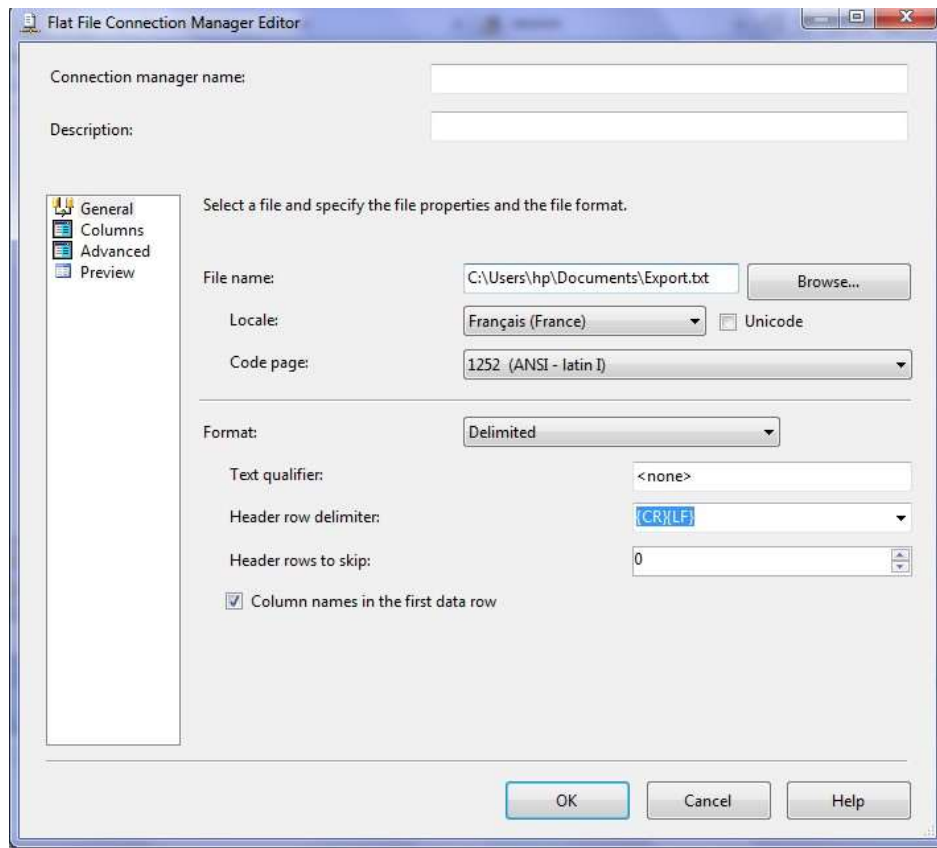
Le contenu du fichier est le suivant :

DepartmentID	Name	GroupName	ModifiedDate
1	Engineering	Research and Development	2002-06-01 00:00:00
2	Tool Design	Research and Development	2002-06-01 00:00:00
3	Sales	Sales and Marketing	2002-06-01 00:00:00
4	Marketing	Sales and Marketing	2002-06-01 00:00:00
5	Purchasing	Inventory Management	2002-06-01 00:00:00
6	Research and Development	Research and Development	2002-06-01 00:00:00
7	Production	Manufacturing	2002-06-01 00:00:00
8	Production Control	Manufacturing	2002-06-01 00:00:00
9	Human Resources	Executive General and Administration	2002-06-01 00:00:00
10	Finance	Executive General and Administration	2002-06-01 00:00:00
11	Information Services	Executive General and Administration	2002-06-01 00:00:00
12	Document Control	Quality Assurance	2002-06-01 00:00:00
13	Quality Assurance	Quality Assurance	2002-06-01 00:00:00
14	Facilities and Maintenance	Executive General and Administration	2002-06-01 00:00:00
15	Shipping and Receiving	Inventory Management	2002-06-01 00:00:00
16	Executive	Executive General and Administration	2002-06-01 00:00:00

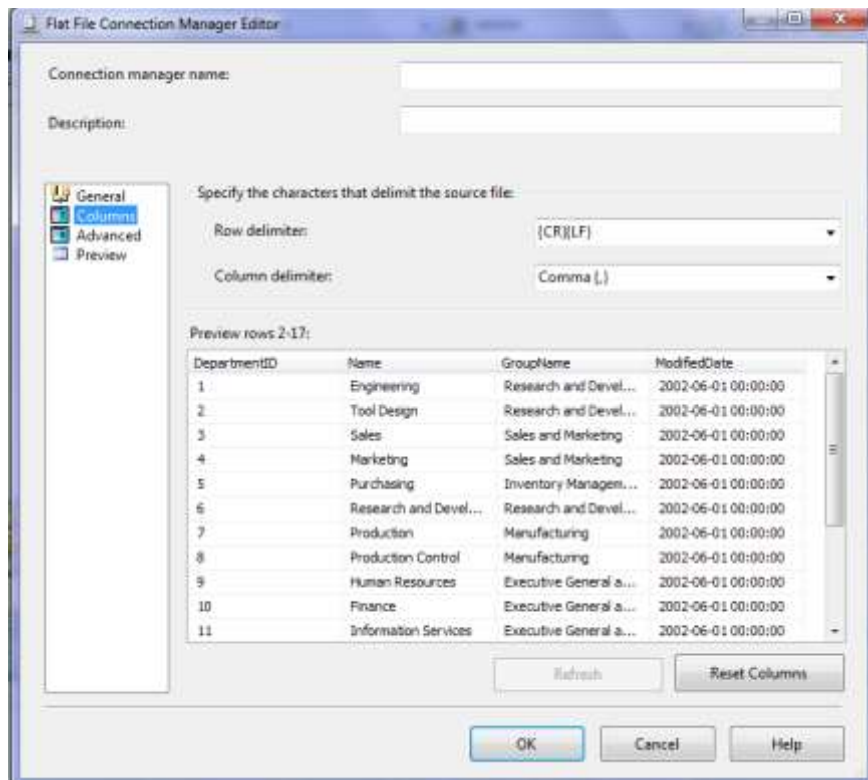
1. Dans l'onglet « ControlFlow », et dans la « Connection Manager » en bas, cliquer bouton droit et sélectionner le type connexion.
2. Sélectionner une connexion de type « New Flat File Connection ».
3. Sélectionner le fichier dans la zone « File name »



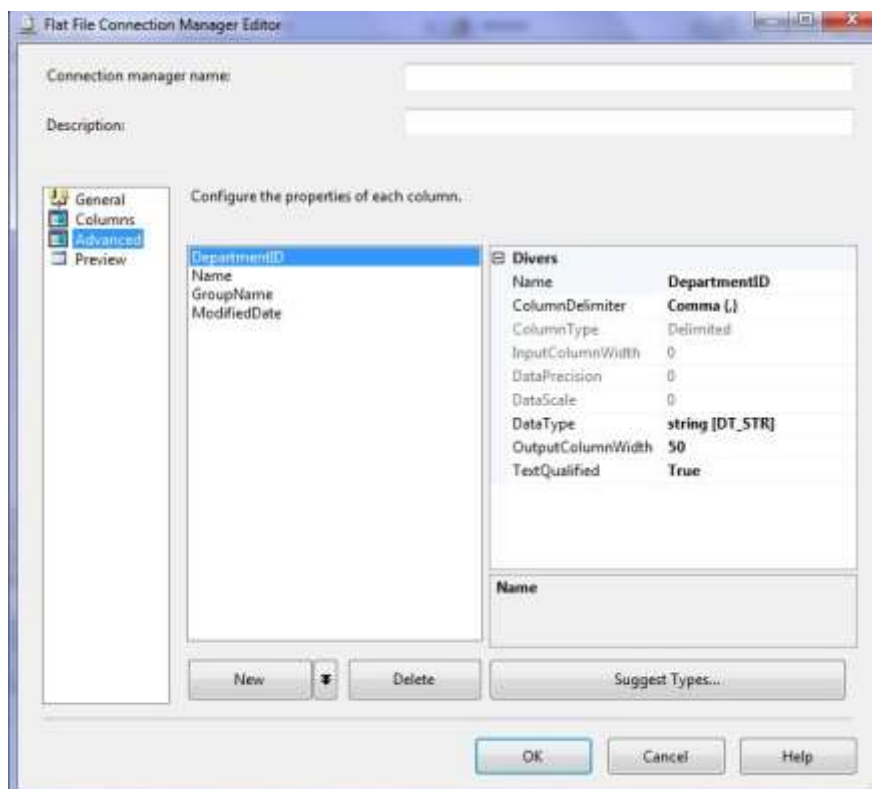
4. Dans le cas où la première ligne du fichier texte contient les noms des attributs, cliquer pour cocher « Column names in the first data row »



5. Cliquer « Columns » Dans la zone de configuration (Zone à gauche) pour préciser le caractère délimiteur dans le fichier source et le délimiteur de lignes.



6. Cliquer « Advanced » dans zone de configuration pour modifier (si nécessaire) les informations des colonnes.



7. Cliquer « OK » pour valider et ajouter la connexion au projet

### **III.3. Création d'un flux de données Data Flow**

La tâche de flux de données permet d'encapsuler le moteur de flux de données qui déplace les données entre les sources et les destinations et fournit la fonctionnalité grâce à laquelle il est possible de transformer, nettoyer et modifier les données lors de leur déplacement. La tâche de flux de données est l'endroit où s'effectue la majorité du travail d'un processus d'extraction, de transformation et de chargement (ETL).

Remarque : Normalement, cette tâche intervient après la création des gestionnaires de connexions pour les données sources et de destination. Mais, on peut commencer directement par cette tâche et configurer une connexion pour chaque flux de données source « Data Flow Sources ».

Pour ajouter une tâche de flux de données, il faut procéder comme suit :

- Cliquer l'onglet « Control Flow »
- Dans la Boîte à outils SSIS, développer « Control Flow Items » puis cliquer-glisser une tâche de flux de données « Data Flow Task » dans l'espace de conception de l'onglet « Control Flow »
- Cliquer bouton droit sur la tâche créée et cliquer « Renommer ». Taper « Extract From Text File »

### **III.4. Configuration du flux de données « Data Flow »**

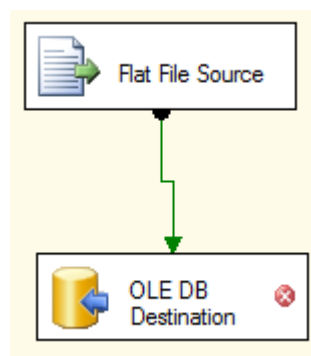
L'idée est d'extraire les données à partir d'un fichier plat vers une source SQL Server. Ainsi, il faut créer pour flux de donnée :

- Un composant source de type « Flat File Source » utilisant la connexion vers le fichier plat
- Un composant destination de type « OLE DB Destination » qui va utiliser la connexion vers SQL Server 2008

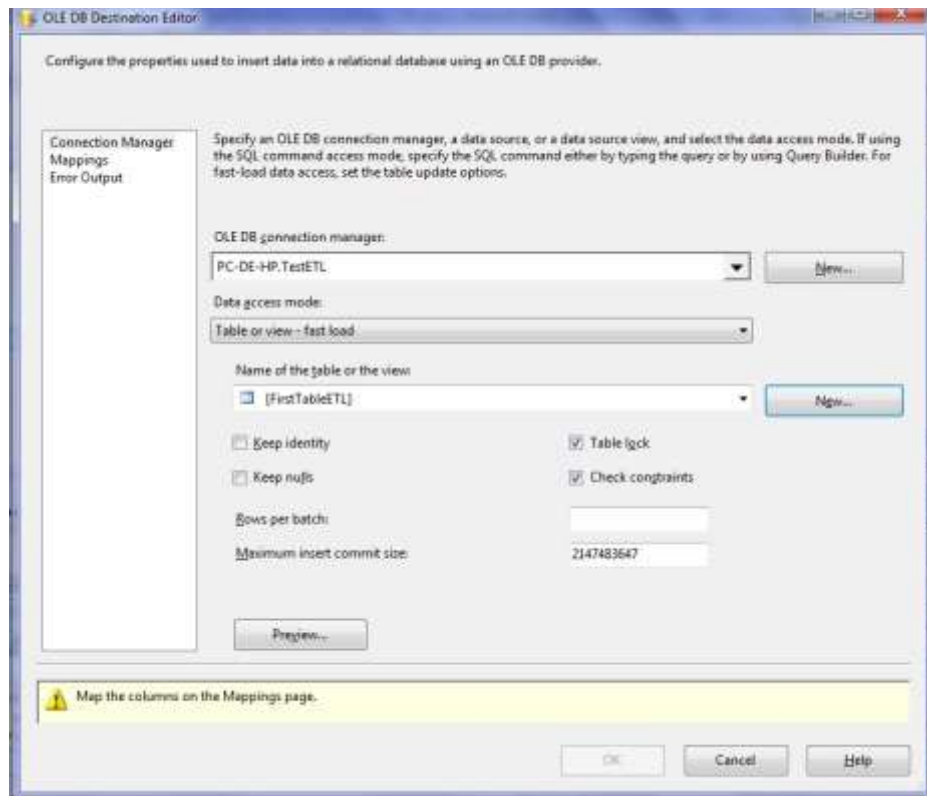
Procéder comme suit :

1. Cliquer l'onglet « Data Flow »
2. Configurer la source :
  - Glisser à partir de la boîte à outils le contrôle « Flat File Source »
  - Cliquer bouton droit sur le contrôle et cliquer « Edit »

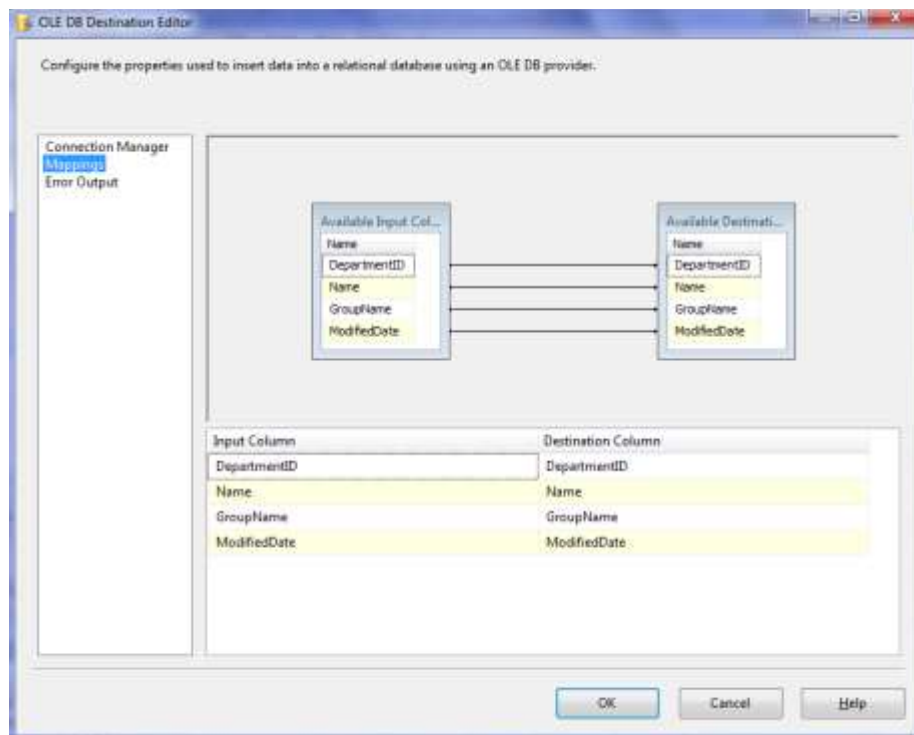
- Sélectionner la connexion manager. Il est possible de créer une nouvelle connexion en cliquant le bouton « New »
  - Cliquer « OK » pour valider
3. Configurer la source :
- Glisser à partir de la boîte à outils le contrôle « OLE DB Destination »
4. Lier les deux sources en glissant la flèche verte du flux source vers le flux de la destination



5. Cliquer bouton droit sur le flux de données destination et cliquer « Edit »
6. Dans la zone « Name of the table or the view », il est possible soit :
- d'utiliser une table déjà existante dans la base
  - ou bien créer une nouvelle en cliquant le bouton « New ». dans ce cas, un script SQL LDD sera lancé avec comme nom de la table le nom du contrôle et les attributs sont les mêmes attributs du flux de données source



7. Cliquer « Mapping » dans la partie de configuration à gauche pour vérifier le mapping des colonnes entre le flux en entrée et le flux en sortie



8. Valider en cliquant « OK »

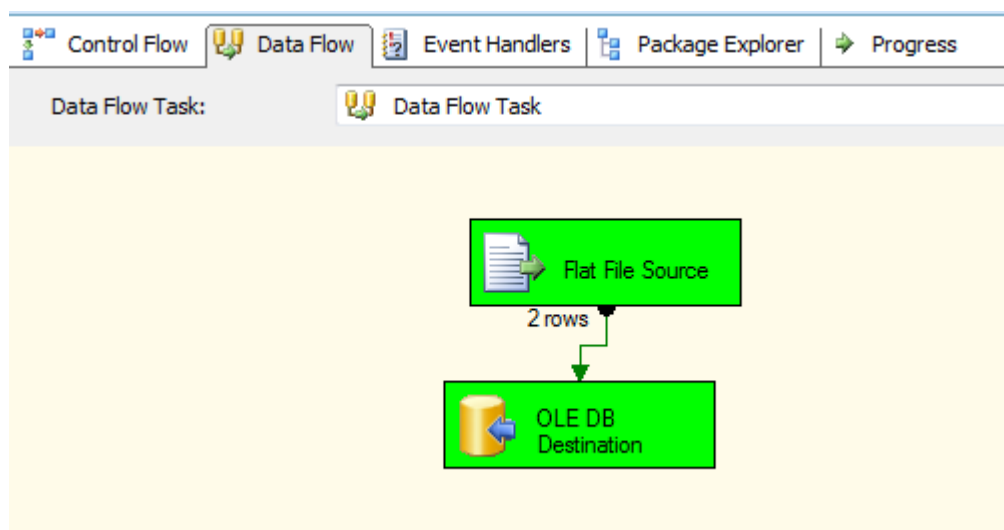
### III.5. Extraction des données utilisant SSIS

Jusqu'ici, les flux source et destination sont bien configurés pour une extraction des données à partir d'un fichier plat (texte) vers une source de données OLE DB de type SQL server.

Ce qui reste est tout simplement d'exécuter le package.

- Cliquer le menu Déboguer → Démarrer le débogage.

Si tout ce passe bien, les taches seront coloriées en vert signification de réussite d'extraction et chargement



- Arrêter le débogage

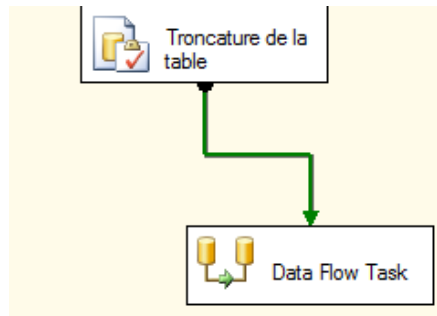
## IV. Personnalisation de la tache

Il est à noter que pour chaque exécution de package créée, les données extraites du fichier plat seront ajoutées aux données qui existent déjà dans la table de destination. Ainsi, il serait bien de vider la table avant l'extraction. Aussi, il serait bien d'afficher un message relatif au début de l'extraction.

### IV.1. Script SQL pour vider la table avant extraction

1. Cliquer l'onglet « Control Flow »
2. Glisser l'élément « Execute SQL Task » au package
3. Cliquer bouton droit et cliquer « Edit »
4. Dans « General », affecter à la propriété « Connection » la connexion OLE DB vers la base SQL Server

5. Ecrire la requête SQL (Truncate table Table\_Name) comme valeur de la propriété « SQLStatement »
6. Lier la tâche « Execute SQL Task » au « Data Flow Task » de telle sorte à ce que la requête SQL sera exécutée avant la tâche du flux de données



7. Valider en cliquant « OK »
8. Exécuter le package et vérifier les données dans la table cible

#### IV.2. Script C# pour afficher un message de début du travail

1. Dans l'onglet « Control Flow » glisser un élément « Script Task »
2. Cliquer bouton droit → Edit
3. Sélectionner le langage du script
4. Cliquer le bouton « Edit script » pour écrire le code
5. Dans la fonction Main, taper le code C# suivant  

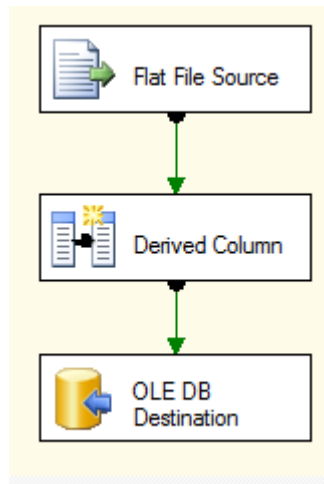
```
MessageBox.Show("Mon premier prog ETL SSIS");
```
6. Lier la tâche aux autres éléments de telle sorte à ce qu'elle soit la première à être exécutée
7. Exécuter le package

#### IV.3. Créer des colonnes dérivées

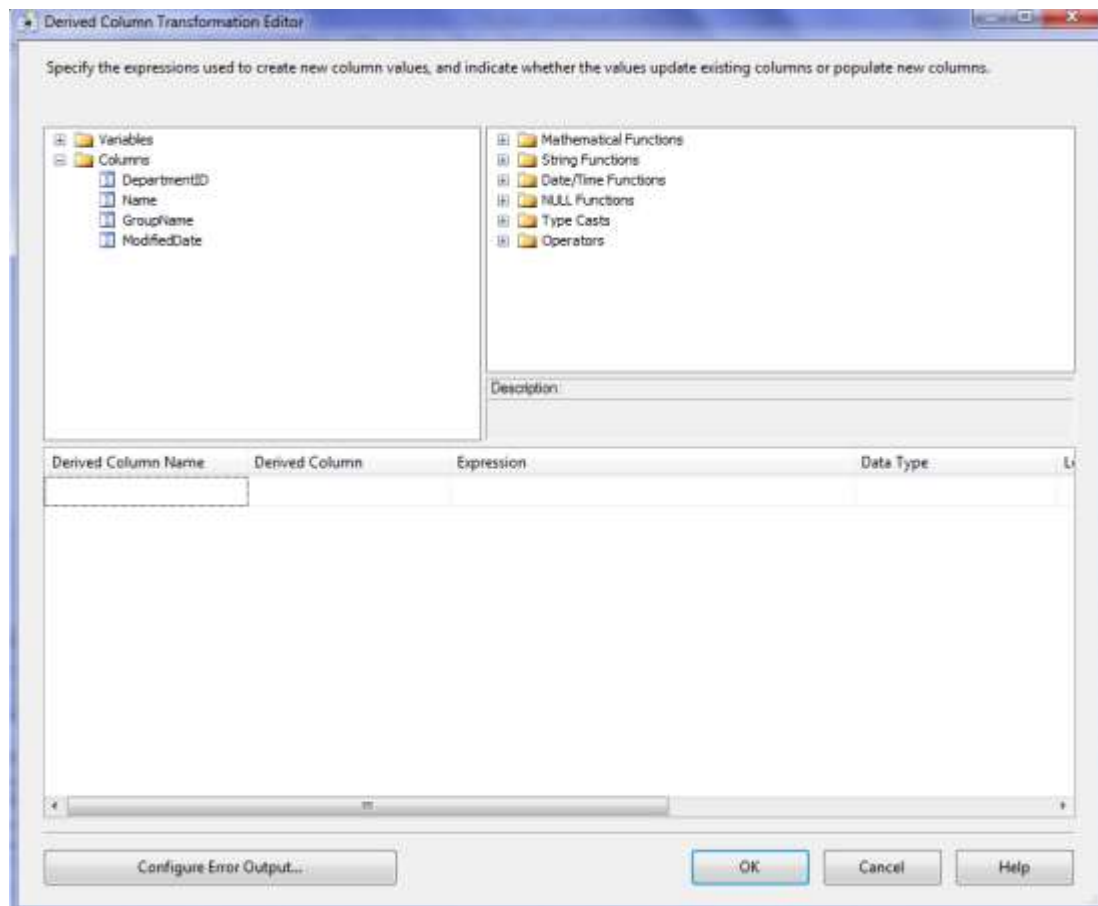
Une des possibilités offertes par SSIS est de pouvoir dériver des colonnes sources et leur appliquer des fonctions SQL. Ainsi, il est possible par exemple de n'afficher que le jour d'une date, ....

Cette fonctionnalité est possible avec l'élément « Derived Column » de l'onglet « Data Flow »

1. Glisser le contrôle « Derived Column »
2. Le mettre entre le flux source et le flux destination



3. Double-cliquer sur le « Dervide Column »pour afficher l’éditeur.



A gauche, on trouve les colonnes et variables. A droite, on trouve les fonctions et opérateurs qu’on peut utiliser.

4. Dans la zone en bas, cliquer pour dériver la colonne Name comme sur la figure en bas. Ceci à pour rôle de rendre le contenu du champ Name en majuscule.



Derived Column Name	Derived Column	Expression	Data Type
Name	Replace 'Name'	UPPER(Name)	string [DT_STR]

Il est possible de créer de nouvelles colonnes à partir des colonnes de la source

Remarque : Il faut faire attention à ce que les champs résultants après application de « Derived Column » coïncident avec les champs de « OLE DB Destination »

.

## V. Charger plusieurs fichiers plats dans une table SQL Server

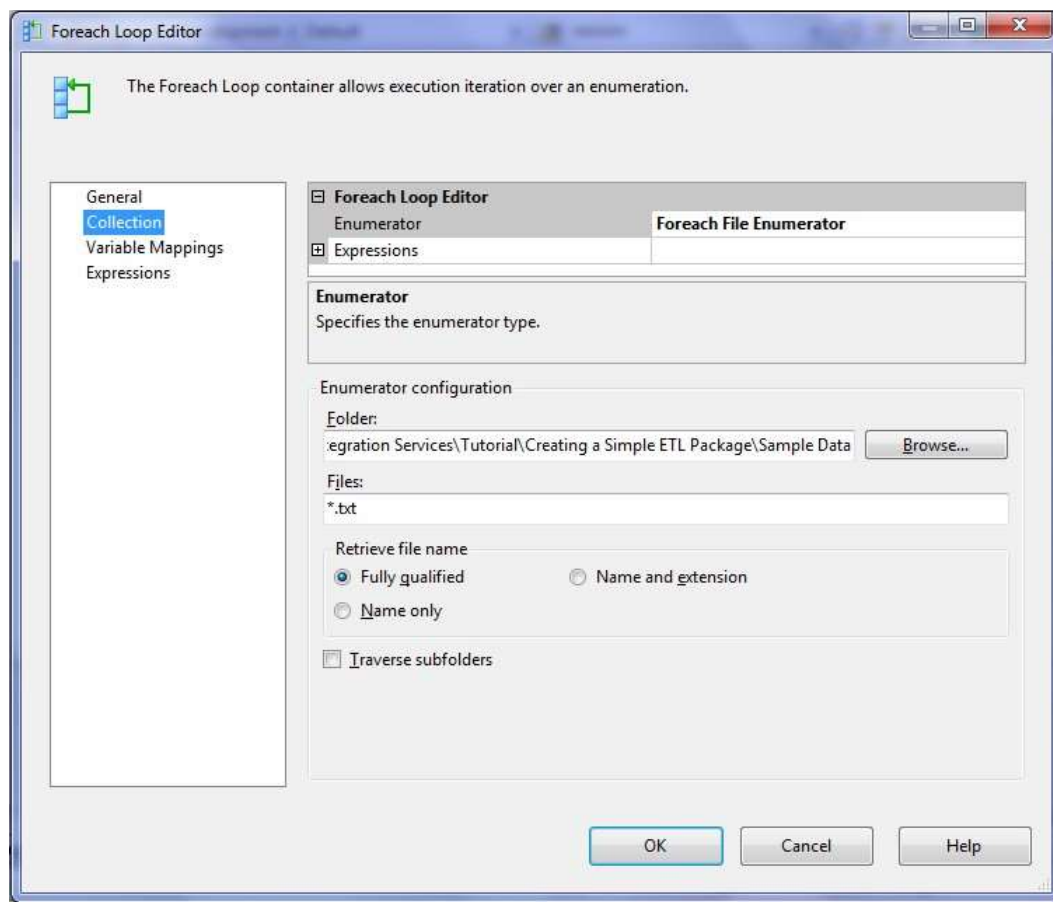
Lorsque on veut charger un ensemble de fichiers dans une base de données, il faut faire appel a une boucle Foreach peut nous aider dans ce processus. Ainsi, on va boucler pour insérer chacun des fichiers dans la table.

Pour le faire, procéder comme suit :

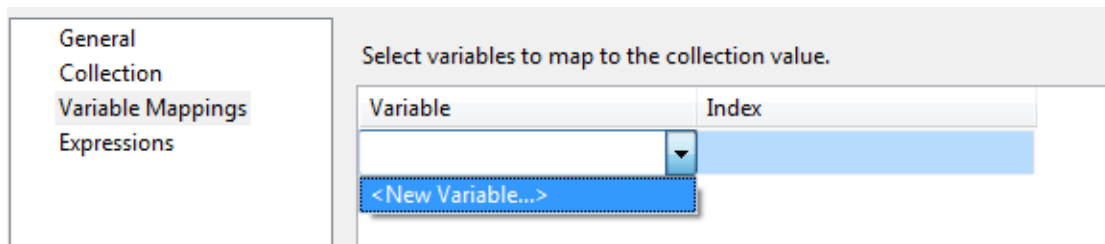
1. Créer un nouveau package

### V.1. Configuration de la boucle Foreach

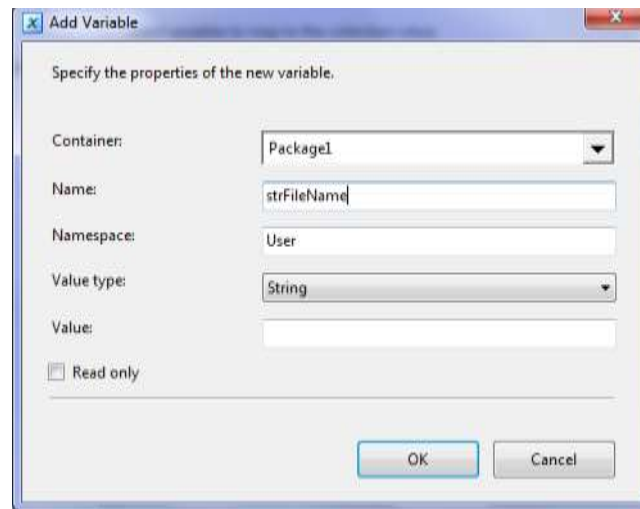
2. Dans l'onglet « Flow Control », glisser le conteneur « Foreach Loop Container »
3. Double cliquer sur le « Foreach Loop Container » pour ouvrir l'éditeur.
4. Cliquer « Collection » et sélectionner pour la valeur « Enumerator » la valeur « Foreach File Enumerator »
5. Dans la partie « Enumérateur configuration » préciser le lien vers le dossier contenant les fichiers plats dans le champ « Folder »
6. Dans le champ « Files » taper « \*.txt » pour ne prendre en considération que les fichiers dont l'extension est txt



7. Cliquer sur « Variable Mapping » à gauche
8. Cliquer dans le champ « Variable » → « New Variable »



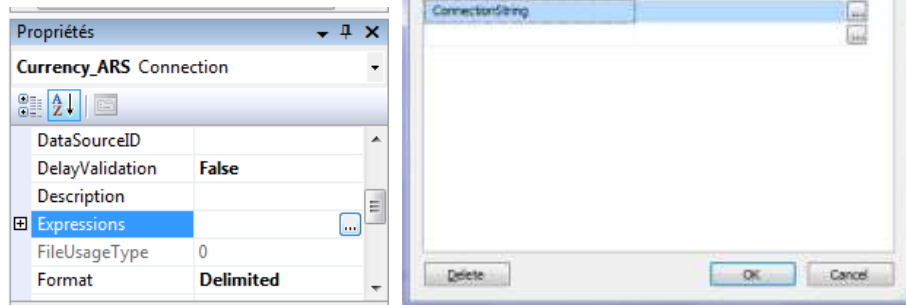
9. Entrer les informations de la nouvelle variable comme sur la figure en bas



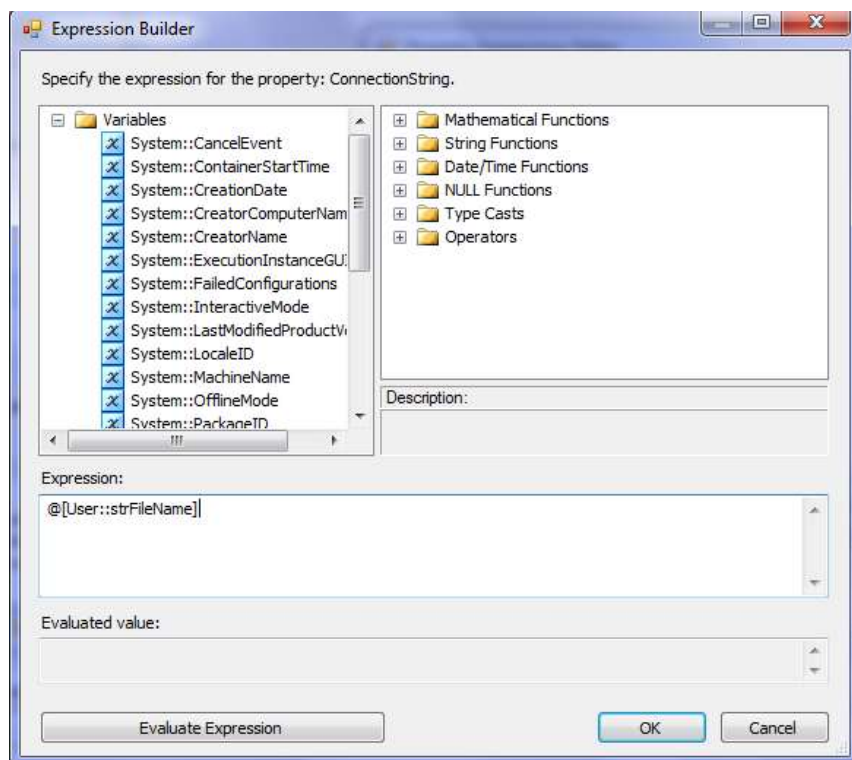
10. Cliquer « OK » pour créer la variable
11. Cliquer « OK » pour terminer la configuration de la boucle Foreach

## V.2. Configuration des flux de données

12. Ajouter un « Data Flow Task » dans le contrôle Foreach
13. Dans l'onglet « Data Flow » ajouter un « Flat File Source » et un « OLE DB Destination »
14. Configurer le « Flat File Source » pour qu'il pointe sur un des fichiers plats dans les dossiers des fichiers
15. Lier le flux « Flat File Source » avec « OLE DB Destination »
16. Configurer le « OLE DB Destination » avec une table SQL Server 2008
17. Dans la partie « Connection Managers », sélectionner la connexion Flat et dans la fenêtre propriété sélectionner « Expression » et cliquer pour ouvrir l'éditeur d'expression.



18. Sélectionner la propriété « ConnectionString » et sélectionner comme Expression la variable créée dans la boucle Foreach



19. Valider en cliquant « 'OK » pour affecter la variable
20. Cliquer « OK » pour terminer la configuration

### V.3. Extraction des données

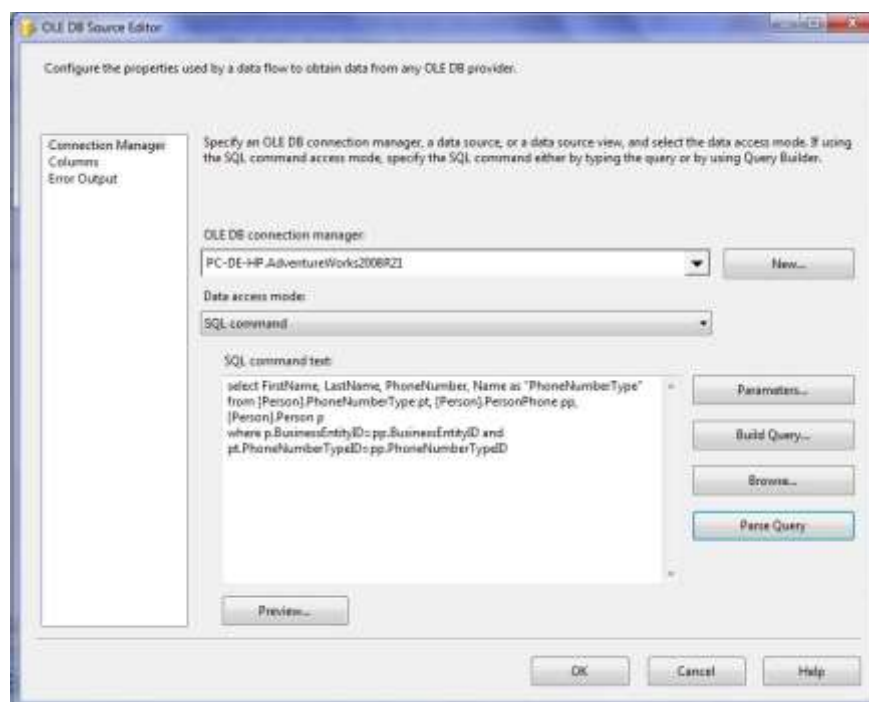
21. Exécuter le package et vérifier les données dans la base

## VI. Extraire les données utilisant une requête SQL

Il est possible d'extraire des données à partir d'une source de données utilisant des requêtes personnalisées et de les charger dans une source destinataire. Dans notre exemple, on va récupérer des données à partir de trois tables d'une base SQL Server 2008 et les enregistrer dans un fichier plat.

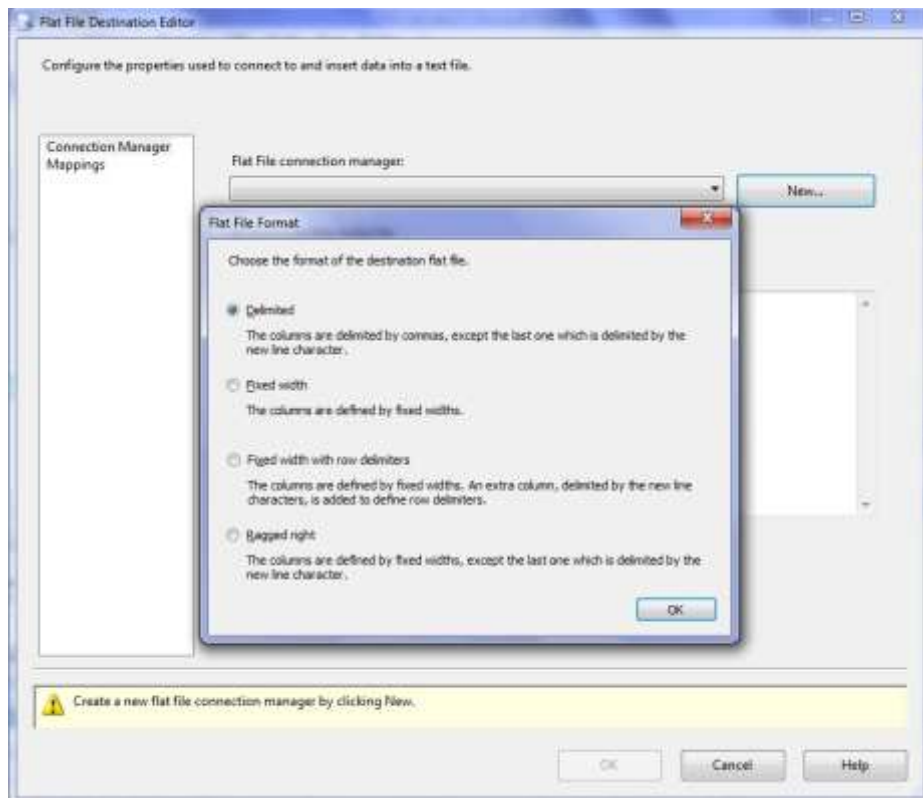
Pour le faire, procéder comme suit :

1. Créer un nouveau package
2. Créer un « Data Flow »
3. Ajouter une nouvelle « Connection Manager » de type « New OLE DB Connection » vers la base de données AdventureWorks2008R2 d'un serveur SQL Server 2008.
4. Ajouter une Source de données « OLE DB Source »
5. Double-cliquer sur la source pour la configurer, lier la à la connexion OLE créée avant et sélectionner comme mode d'accès au données « SQL Command » et taper la requête SQL comme sur la figure

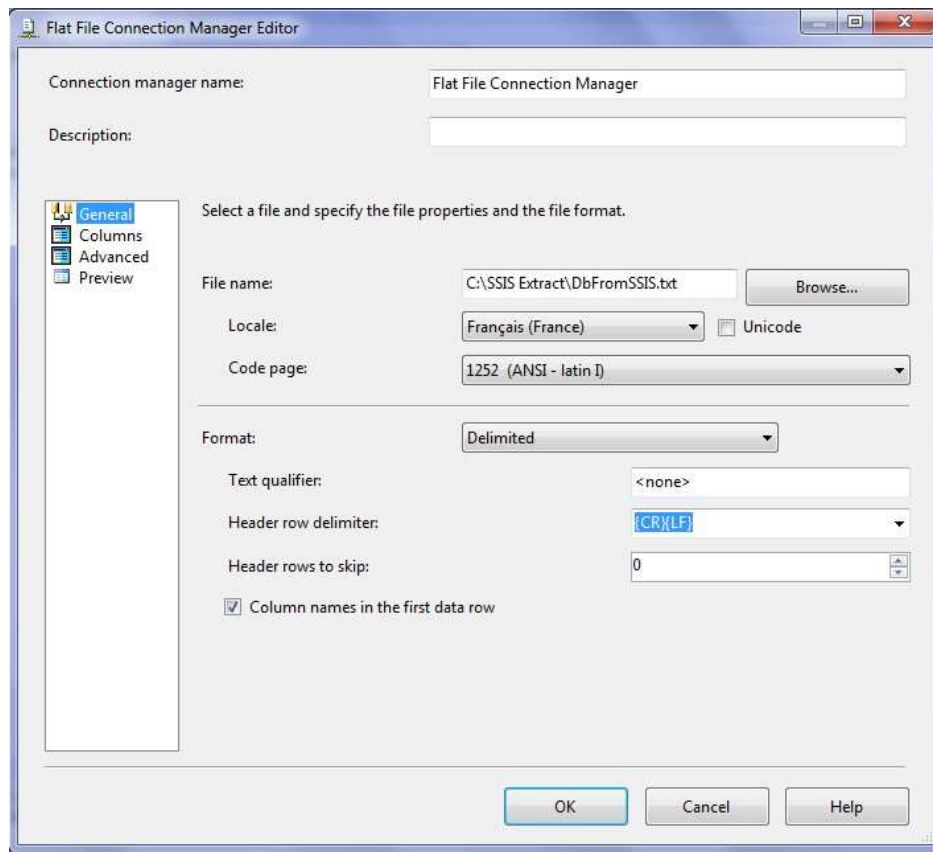


6. Cliquer « OK » pour valider
7. Ajouter un flux de données destination « Data Flow Destination » de type « Flat File Destination »
8. Lier le flux source à flux de destination
9. Double cliquer sur le flux Flat pour le configurer. Cliquer « New » pour ajouter une nouvelle connexion. Après, Il sera demandé de choisir le

format du fichier plat destinataire. Cocher la case « Delimited » et cliquer « OK » (voir figure en base)



10. Configurer la connexion Manager en choisissant le chemin vers le fichier plat (.txt) de destination comme sur la figure en bas



11. Cliquer « OK » pour valider

Par défaut, la case « Overwrite data in the file » est cochée pour vider le fichier à chaque exécution

12. Après, vérifier le schéma en cliquant « Preview » et vérifier le « Mappings »

13. Exécuter le package et vérifier les données dans le fichier plat de destination

## VII. Gestion des erreurs

Les erreurs peuvent se produire lorsqu'un composant de flux de données applique une transformation à des données, extrait des données à partir de sources ou charge des données dans des destinations.

Les erreurs dans un ETL peuvent être regroupées en trois catégories :

- Erreurs de conversion de données : si une conversion provoque la perte de chiffres significatifs, la perte de chiffres non significatifs et la troncation de chaînes.
- Erreurs d'évaluation d'expression : si des expressions évaluées au moment de l'exécution effectuent des opérations non valides ou deviennent

syntactiquement incorrectes à cause de valeurs de données manquantes ou incorrectes.

- Erreurs de recherche : qui se produisent si une opération de recherche ne retourne pas de valeurs

La plupart des composants de flux de données prennent en charge les sorties d'erreur (connecteur en rouge). Cette sortie permet de contrôler la manière dont le composant gère les erreurs de lignes dans les données entrantes et sortantes. Ainsi, vous pouvez spécifier le comportement du composant lorsqu'une troncation ou une erreur se produit en définissant des options sur des colonnes dans l'entrée ou la sortie.

La sortie d'erreur peut être connectée à l'entrée d'une autre transformation ou chargée dans une destination différente de la sortie sans erreur.

Par exemple, la sortie d'erreur peut être connectée à une transformation de colonne dérivée qui fournit une chaîne pour une colonne vide.

#### **VII.1. Premier exemple**

Pour mettre en œuvre et tester la gestion d'erreurs sous SSIS on va créer un projet Intégration Services avec la situation suivante :

- Un fichier Excel comme source de données dans les champs son les suivants : Nom, Prénom, Age, Adresse, Email
- Remplissez le fichier avec des données et faites en sorte que le champ « Adresse » de quelques enregistrements soit vide
- Une table SqlServer 2008 qui va recevoir les données du fichier Excel :

```
CREATE TABLE TestETLError (Nom nvarchar(255), Prénom nvarchar(255), Adresse nvarchar(255) NOT NULL, Email nvarchar(255))
```

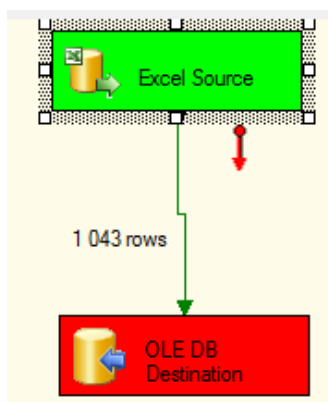
Remarquez bien que le champ Adresse n'admet pas de valeurs nulles.

Tester en procédant comme suit :

1. Créer un projet SSIS avec comme flux de données source le fichier Excel et flux de données destination la table TestETLError



2. Exécuter la tâche ; vous allez remarquer que le composant flux de données destination est en rouge. Ceci implique qu'une erreur (le champ Adresse de la table n'admet pas des valeurs nulles) est survenue



## VII.2. Gérer les erreurs

Les erreurs appartiennent à l'une des deux catégories suivantes : erreurs ou troncations. Une erreur indique un échec non équivoque et génère un résultat NULL. Une troncation est une erreur moins grave. Elle génère des résultats qui peuvent être utilisables, voire même souhaitables. Vous pouvez faire en sorte de traiter les troncations comme des erreurs ou comme des conditions acceptables.

Vous pouvez configurer la manière dont les sources, les transformations et les destinations gèrent les erreurs et les troncations. Le tableau ci-dessous décrit les options disponibles :

Option	Description
« Fail Component » : Composant défaillant	La tâche de flux de données échoue lorsqu'une erreur ou une troncation a lieu. L'échec est l'option par défaut pour une erreur et une troncation.
« Ignore Failure » : Ignorer l'échec	L'erreur ou la troncation est ignorée et la ligne de données est dirigée vers la sortie de la transformation ou de la source.
« Redirect Row » : Réacheminer la ligne	La ligne de données d'erreur ou de troncation est dirigée vers la sortie d'erreur de la source, de la transformation ou de la destination.

## VII.3. Gestion d'erreur dans le premier exemple

1. Créer une nouvelle table SQL Server 2008 qui va contenir les champs qui ont causés l'erreur (dont l'adresse est vide). Le schéma est le même que la

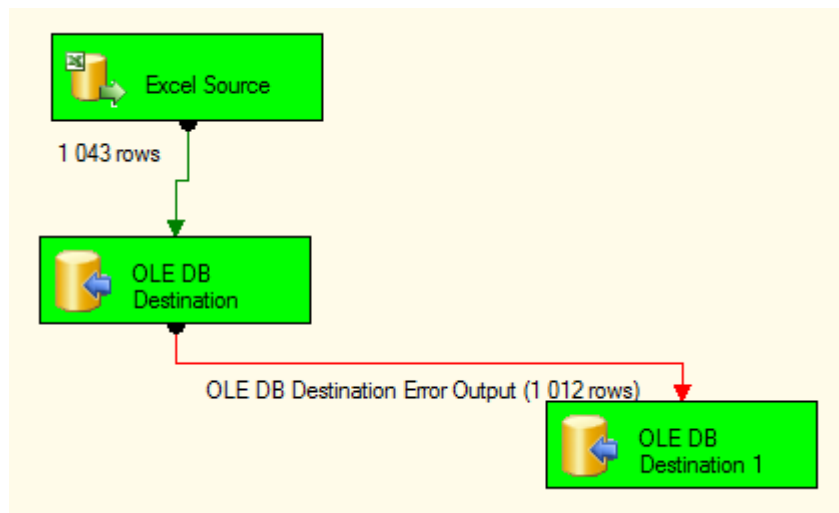
table TestETLError (créée précédemment) mais avec le champ Adresse qui va accepter les valeurs nulles :

```
CREATE TABLE TestETLGetErrors (Nom nvarchar(255),  
Prénom nvarchar(255), Adresse nvarchar(255), Email  
nvarchar(255))
```

2. Ajoutez au projet un autre flux de données destination vers la table TestETLGetErrors
3. Liez la sortie d'erreur du flux de données déjà crée dans le premier exemple avec celui que vous venez de créer
4. Une fenêtre de configuration de la sortie erreur (Configure Output Error) sera lancée. Modifiez le champ « Error » à « Redirect Error » pour rediriger les lignes erreurs vers le deuxième flux.



5. Configurez le contrôle de flux de sortie et vérifiez le mapping
6. Exécutez et vérifiez les données dans les deux tables



Vous allez remarquer que sur l'ensemble des enregistrements dans le fichier Excel (qui sont en nombre de 1043 dans l'exemple) 1012 enregistrements sont acheminés vers la destination liée à la sortie erreur (se sont les enregistrements dont l'adresse est nulle) et le reste est enregistré dans le premier composant de flux de destination (se sont les enregistrements sans erreurs)

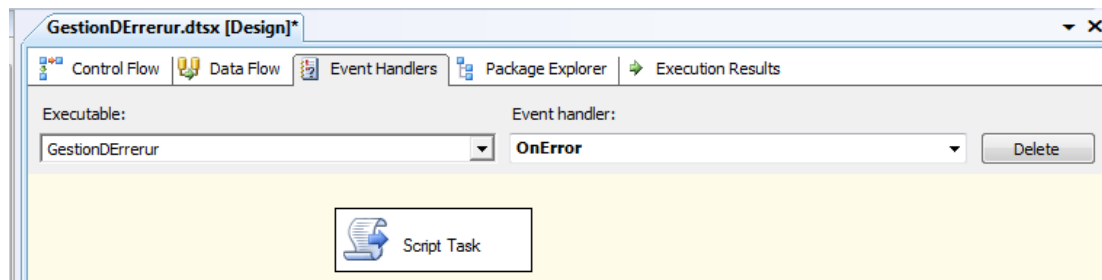
#### VII.4. Gestion d'erreur avec « Event Handlers »

SSIS offre la possibilité de gérer les erreurs d'une manière événementielle en utilisant l'onglet « Event Handlers » du paquetage.

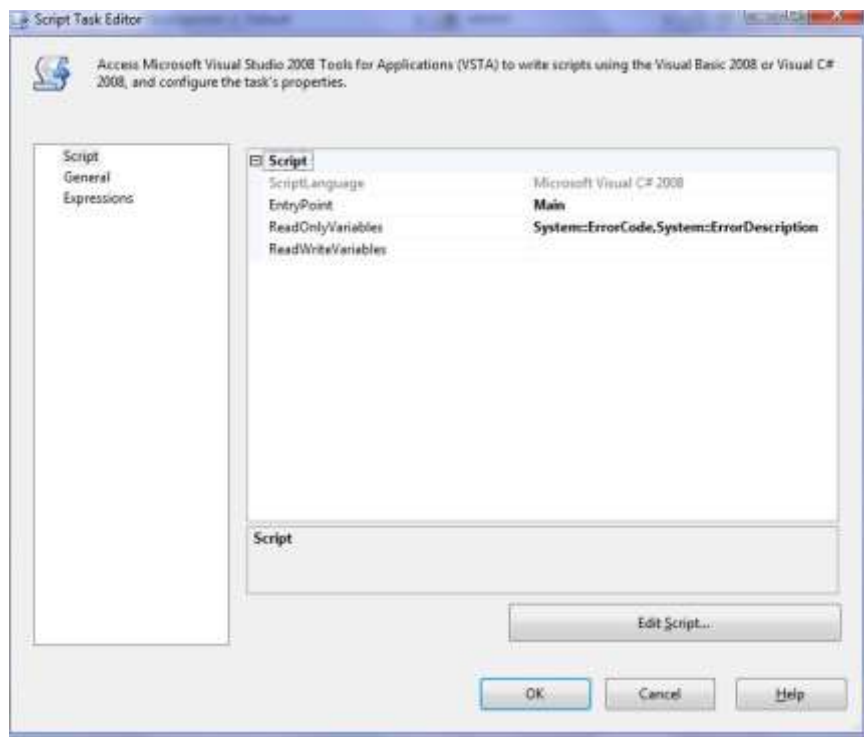
Dans la partie « Event Handlers », plusieurs événements peuvent être gérés et parmi eux il y a l'événement « OnError » qui se déclenche après chaque erreur d'exécution. Dans cet onglet, vous pouvez utiliser toutes les composantes de contrôle de flux de maintenance.

Pour un simple exemple, dans le paquetage déjà créé, vous allez forcer une erreur et dans la partie « Event Handlers » vous allez utiliser un « Script Task » pour afficher le numéro et la description de l'erreur.

- Double cliquez sur le premier contrôle flux de données destination et modifiez la valeur de « Error » dans « Error Output » à « Fail component ». Ceci à pour effet de générer une erreur et arrêter l'exécution
- Dans l'onglet « Event Handlers » sélectionnez « OnError » dans champ « Event Handlers » de votre package et cliquez pour ajouter un événement
- Glisser à partir de la boîte à outil un composant « Script Task »



- Double cliquez sur ce composant pour afficher l'éditeur du script
- Dans le champ « ReadOnlyVariables » cliquez pour sélectionner les variables à utiliser. Cochez System::ErrorCode et System::ErrorDescription et valider par « OK »



- Cliquez le bouton « Edit Script » et ajoutez dans la fonction Main le code suivant :

```
//TODO: Add your code here
MessageBox.Show("Une erreur est survenue\n N: " +
Dts.Variables["ErrorCode"].Value.ToString() + "\nDescription: " +
Dts.Variables["System::ErrorDescription"].Value.ToString());
Dts.TaskResult = (int)ScriptResults.Success;
```

- Fermez la fenêtre du script et cliquez « Ok » pour valider
- Exécutez pour tester le fonctionnement