

Introduction à Sql Server Integration Services (SSIS)

Plus connu avant sous le nom de Data Transformation Services (DTS), SSIS est la nouvelle version de cet outil de SQL Server apparu avec la version 2005 du SGBD de Microsoft. Nous allons essayer de découvrir ensemble ce que peut nous apporter SSIS.

Sommaire

Présentation générale	3
Comment travailler avec SSIS ?	4
ControlFlow	7
DataFlow.....	7
Event Handlers	7
Package Explorer	7
Présentation de l'exemple	7
Structure d'un fichier de log de IIS.....	7
Tâches à réaliser par notre package.....	9
Stockage	9
La copie du fichier	9
Traitement du fichier de log.....	12
Lecture du fichier de log.....	13
La connexion au fichier de log.....	13
Flat File Source	18
Création de la colonne DateTime.....	20
Conversion des données	23
Ecriture en base.....	26
Une connexion à la base.....	27
Liaison de nos données à la base	29
Première tentative d'exécution du Data Flow	31
Dernier ajustement sur le Control Flow	36
Déploiement du package.....	36
Vérification du déploiement et exécution	42
Conclusion	45

Présentation générale

Tout d'abord, pour pouvoir disposer de SSIS, vous devez utiliser SQL Server 2005 Standard ou Entreprise Edition (Developer Edition aussi, mais pour le développement). Les versions Express et autres Standard et Workgroup Edition ne sont malheureusement pas dotées de cette fonctionnalité.

Seules les versions Entreprise et Developer proposent toutes les fonctionnalités de SSIS.

Reportez-vous à cette page pour plus de détails sur les possibilités offertes par chaque version :

<http://msdn2.microsoft.com/en-us/library/ms143761.aspx>

Maintenant que nous avons vu quelles versions permettent d'exploiter SSIS, voyons maintenant à quoi peut servir cette nouvelle fonctionnalité.

On ne va pas faire très compliqué, mais en gros, SSIS vous permet de transformer vos données, de les importer ou de les exporter vers ou depuis SQL Server.

Prenons un exemple concret. Vous avez un serveur Web, au hasard Internet Information Service (IIS) et vous voulez pouvoir obtenir des statistiques de son utilisation. Si vous connaissez un peu IIS, vous savez sans doute qu'il ne permet de ne générer qu'un fichier de log au format texte.

```
#Software: Microsoft Internet Information Services 5.1
#Version: 1.0
#Date: 2007-05-21 17:03:20
#Fields: time c-ip cs-method cs-uri-stem sc-status
17:03:20 127.0.0.1 GET /EspacePerso/~/_images/baniereperso.png 404
17:06:52 127.0.0.1 GET /EspacePerso/net/test/Default.aspx 404
17:08:07 127.0.0.1 DEBUG /EspacePerso/ViewArticle.aspx 401
17:08:13 127.0.0.1 DEBUG /EspacePerso/ViewArticle.aspx 200
17:08:17 127.0.0.1 GET /EspacePerso/ViewArticle.aspx 200
```

Exemple de fichier de log IIS

Pour réaliser des statistiques, avec Sql Server Reporting Services par exemple, ce n'est pas ce qu'il y a de plus efficace. La solution est de transformer ce fichier de log et d'en intégrer les données en base. C'est là que SSIS intervient. Il est capable de traiter ce fichier de log et d'en transformer le contenu pour l'intégrer en base.

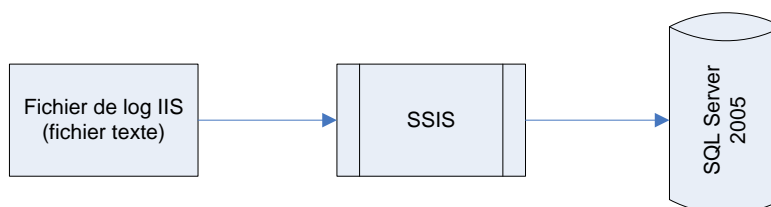


Figure 1: Procédus de traitement des fichiers de log IIS

Ce que va faire SSIS, c'est ouvrir le fichier de log, le découper comme il faut, et insérer en base.

Nous nous servirons de cet exemple pour mieux comprendre un peu mieux comment SSIS fonctionne.

Comment travailler avec SSIS ?

Comme pour tout ce qui touche au développement pour SQL Server, c'est Visual Studio qui sera utilisé.

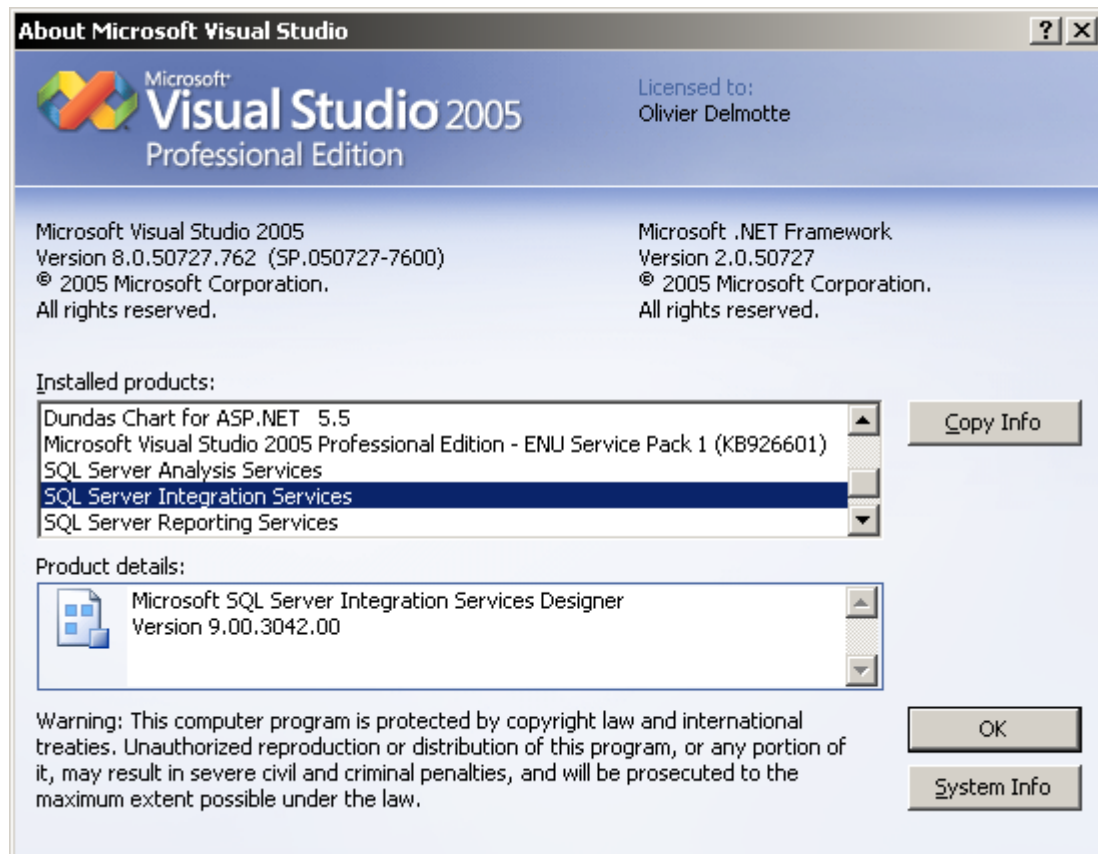


Figure 2: Intégration de SSIS à Visual Studio

Pour travailler avec SSIS, il vous faut créer un projet Integration Services Project classé dans la catégorie Business Intelligence Projects.

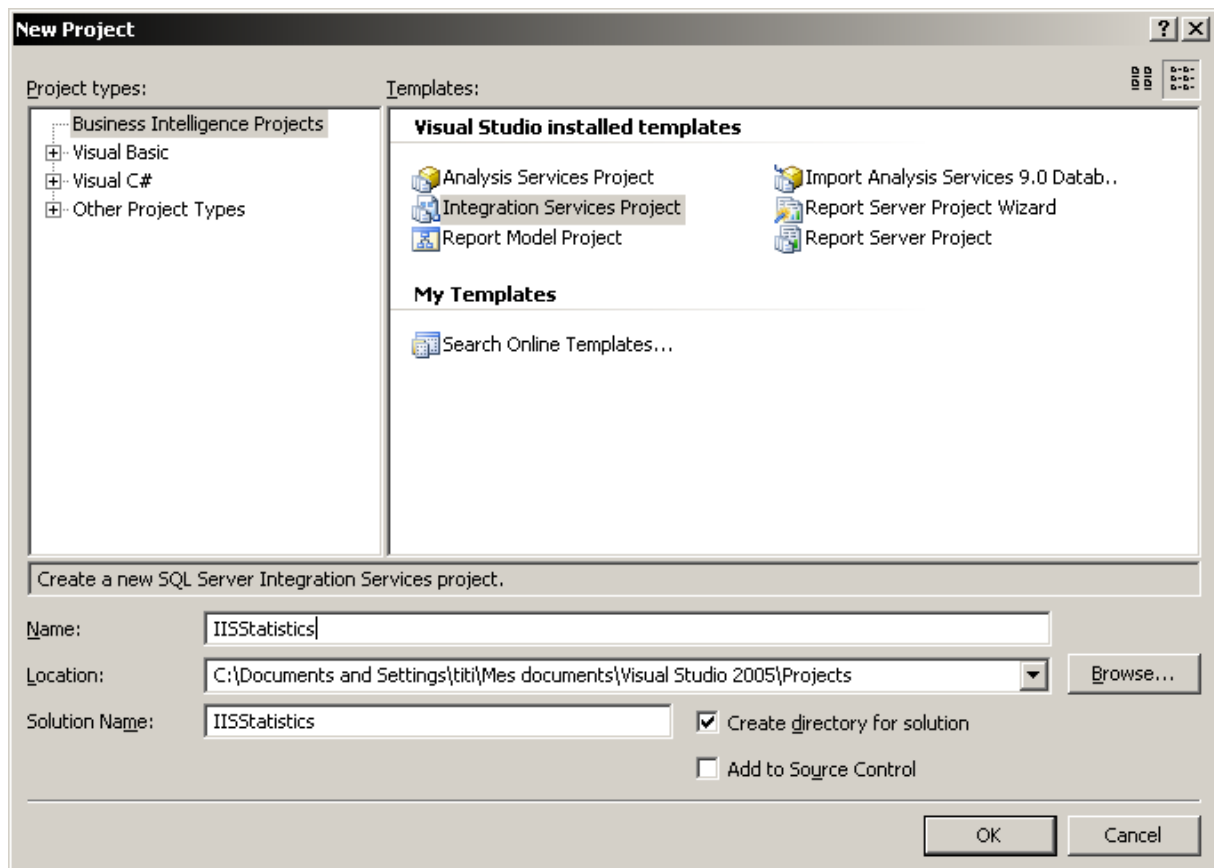


Figure 3:Création du projet SSIS

Tout d'abord, que contient le projet.

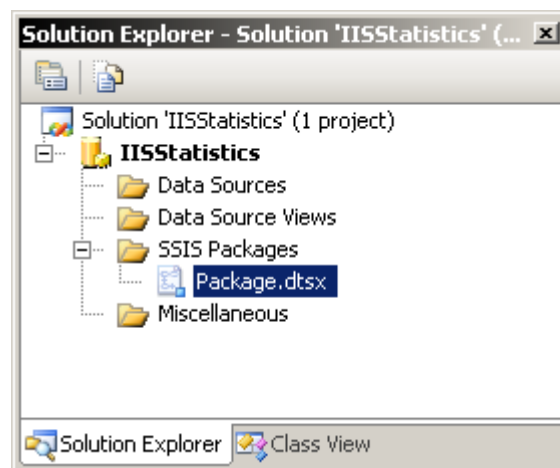


Figure 4:Apperçu du Solution Explorer

A sa création, le projet ne contient qu'un package : Package.dtsx. Un package est, si on peut dire, un script pour SSIS. Il contient une liste d'actions à effectuer par SSIS.

Les actions disponibles se situent au niveau de la Toolbox :

Les actions disponibles se situent au niveau de la Toolbox :

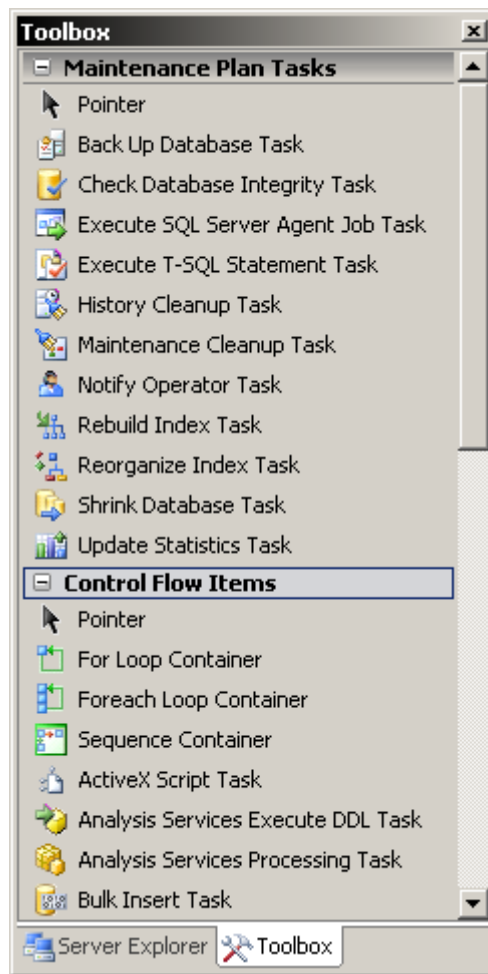


Figure 5:Toolbox de Visual Studio pour un package

Si vous regardez toute la liste, vous verrez que les possibilités offertes par SSIS sont déjà très nombreuses. Je ne vais pas les détailler toutes, juste celles que nous utiliserons, la liste serait beaucoup trop longue.

Enfin, le dernier élément, l'espace d'édition du package :

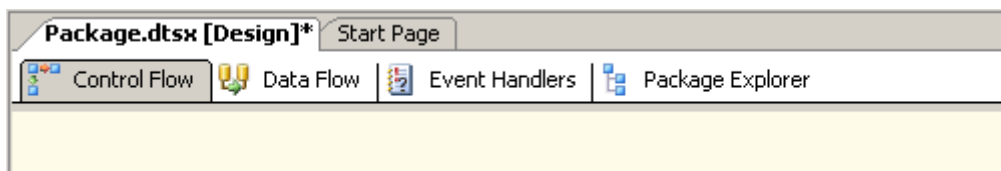


Figure 6:Espace de travail d'un Package

Il se divise en 4 parties :

- ControlFlow
- DataFlow
- Event Handlers
- Package Explorer

ControlFlow

Comme tout programme ou script, un package SSIS doit suivre une certaine logique de fonctionnement. Le package doit avoir un début et une fin, exécuter des tâches dans un ordre bien particulier (lire les données avant de les traiter par exemple, ça peut aider).

C'est donc dans le ControlFlow que nous allons décrire les différentes tâches d'un package. Il est possible de ne pas attribuer un ordre précis à tous les travaux, mais il est quand même bon de garder une certaine logique.

DataFlow

Le DataFlow est comme le ControlFlow, mais il est dédié aux données. Typiquement, dans un DataFlow, nous retrouverons, dans cet ordre, des étapes de lecture, de modification et enfin d'écriture des données. Je ne vous en dis pas plus, nous verrons cela plus en détail plus loin.

Event Handlers

Il est possible 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. C'est via le panneau Event Handlers que nous gérerons ces événements.

Package Explorer

Le Package Explorer est comme le Solution Explorer. Il décrit, sous forme d'arborescence, tous les éléments qui constituent le package et nous permet d'accéder en quelques clics à tous les éléments du package et à leurs propriétés.

Présentation de l'exemple

Comme je vous l'ai dit plus haut, nous allons automatiser l'analyse des fichiers de log de IIS (seulement la partie serveur http, ce sera largement suffisant).

Structure d'un fichier de log de IIS

```
#Software: Microsoft Internet Information Services 5.1
#Version: 1.0
#Date: 2007-05-27 07:38:39
#Fields: date time c-ip cs-username s-sitename s-computername s-ip
s-port cs-method cs-uri-stem cs-uri-query sc-status sc-win32-status
sc-bytes cs-bytes time-taken cs-version cs-host cs(User-Agent)
cs(Cookie) cs(Referer)
2007-05-27 07:38:39 127.0.0.1 - W3SVC1 DTCBPXP 127.0.0.1 80 GET
/EspacePerso/~images/baniereperso.png - 404 3 4383 305 300 HTTP/1.1
localhost
Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+5.1;+.NET+CLR+1.1.4322
;+.NET+CLR+2.0.50727;+.NET+CLR+3.0.04506.30) - -
2007-05-27 07:40:26 127.0.0.1 - W3SVC1 DTCBPXP 127.0.0.1 80 DEBUG
/EspacePerso/DebugAttach.aspx - 401 0 351 406 3675 HTTP/1.1
localhost - - -
```

Voici un fichier extrait de fichier de log IIS, le tout début pour être exact.

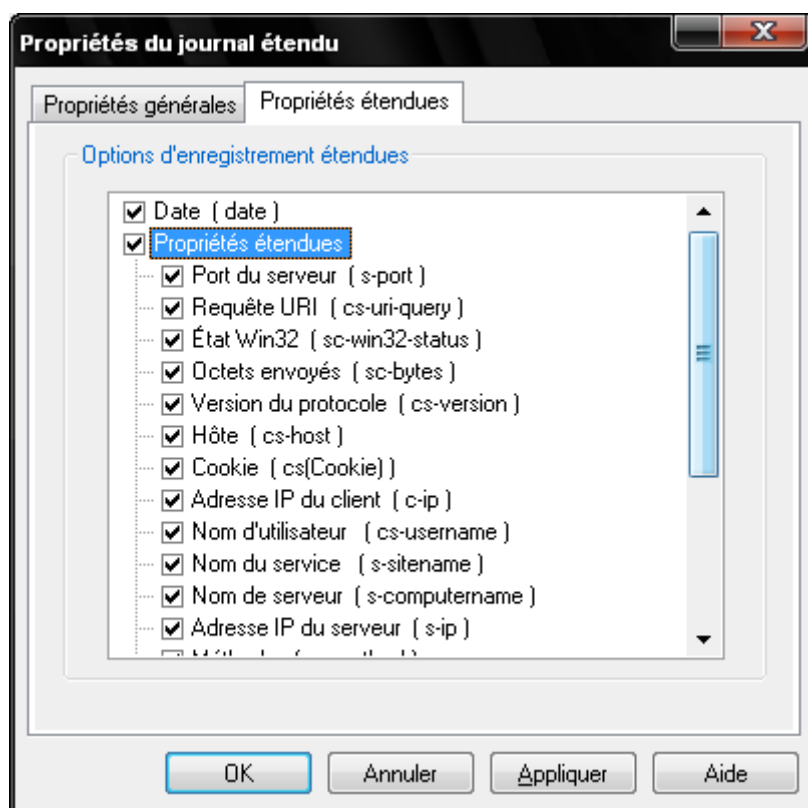
Ce petit exemple va nous servir à comprendre sa structure. Tout d'abord, vous remarquerez les 4 premières lignes qui sont un peu particulières. Elles sont ici à titre d'information.

La première ligne nous donne des informations sur la version de IIS que nous utilisons. Je suis sous Windows XP, j'ai donc IIS 5.1.

La deuxième la version ... de quoi je ne sais pas, mais de toutes façons, elle ne nous intéresse pas ici.

La troisième ligne nous donne la date et l'heure de création du fichier. Elle n'a pas une grande utilité ici, passons.

La quatrième ligne, contrairement aux 3 précédentes, est une source d'information vitale pour le travail que nous allons réaliser. Elle contient en effet une description de la formation des lignes de log. Une ligne de log à proprement parler contient toutes les valeurs que nous souhaitons tracer séparées par des espaces. IIS nous offre la possibilité, via la console d'administration, de choisir les informations sur l'activité du serveur Web que nous souhaitons conserver.



Comme vous pouvez le constater, ces informations sont très complètes.

Dans notre exemple, elles seront toutes utilisées.

Pourquoi cette ligne est-elle si importante ? Comme vous l'aurez sans doute compris, il est possible que le log de l'activité de IIS ne nous renvoie pas toujours les mêmes informations, selon l'humeur et les besoins de l'administrateur. Il est donc important de savoir ce que nous allons récupérer comme informations.

Il faut savoir aussi que cette section de 4 lignes d'en-têtes peut se répéter plusieurs fois dans un même fichier (changements de configuration, redémarrage).

Donc, à la suite de ces 4 lignes, nous récupérons les lignes de données qui nous intéressent réellement. Dans ce tutoriel, je ferais comme si la structure du fichier de log n'était pas modifiable et qu'il n'y avait qu'une seule section d'en-tête par fichier afin de ne pas rentrer dans les détails, ce n'est qu'une introduction. Afin aussi de ne pas compliquer trop les choses, nous utiliserons un unique fichier, dont nous connaissons le nom à l'avance.

Tâches à réaliser par notre package

Notre package d'exemple ne sera pas très compliqué.

Il package effectuera les actions suivantes :

1. Copier le fichier de log dans un dossier temporaire
2. Lire le contenu du fichier
3. Effectuer quelques opérations de transformation sur les données récupérées
4. Ecrire ces données en base

Stockage

Pour stocker les informations de notre fichier de log, nous utiliserons une table SQL Server 2005 dont voici le script de création :

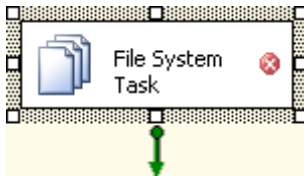
```
CREATE TABLE [dbo].[logiis](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [dateandtime] [datetime] NULL,
    [ipclient] [nvarchar](15) NULL,
    [username] [nvarchar](50) NULL,
    [sitename] [nvarchar](50) NULL,
    [computername] [nvarchar](50) NULL,
    [ipserver] [nvarchar](15) NULL,
    [portserver] [int] NULL,
    [cliservmethod] [nvarchar](5) NULL,
    [cliservuristem] [nvarchar](512) NULL,
    [cliservuriquery] [nvarchar](512) NULL,
    [servclistatus] [int] NULL,
    [servcliwinstatus] [int] NULL,
    [servclibytes] [int] NULL,
    [cliservbytes] [int] NULL,
    [timetaken] [int] NULL,
    [cliservversion] [nvarchar](50) NULL,
    [cliservhost] [nvarchar](512) NULL,
    [cliservuseragent] [nvarchar](512) NULL,
    [cliservcookie] [nvarchar](512) NULL,
    [cliservreferer] [nvarchar](512) NULL
) ON [PRIMARY]
```

La copie du fichier

La première étape est donc la copie du fichier.

Pour venir à bout de cette première tâche, nous allons logiquement utiliser le panneau Control Flow.

Dans la boîte à outils, vous verrez un composant nommé File System Task. Glissez-le sur l'espace de travail.



Détaillons un peu ce composant.

Tout d'abord, la petite croix blanche sur fond rouge, bien connue de tous, est synonyme d'erreur. C'est normal, notre composant n'est pas configuré correctement (ici pas de source, ni de destination). Elle disparaît une fois le travail bien fait.

La petite flèche verte. Elle permet de relier le composant à l'opération suivante.

Si vous ouvrez le menu contextuel du composant et choisissez de l'éditer, vous obtiendrez l'écran suivant :

File System Task Editor

Configure the properties required to perform file system operations, such as creating, moving, or deleting files or directories.

General
Expressions

Destination Connection

IsDestinationPathVariable	False
DestinationConnection	
OverwriteDestination	False

General

Name	File System Task
Description	File System Task

Operation

Operation	Copy file
-----------	-----------

Source Connection

IsSourcePathVariable	False
SourceConnection	

Name
Specifies the name of the task.

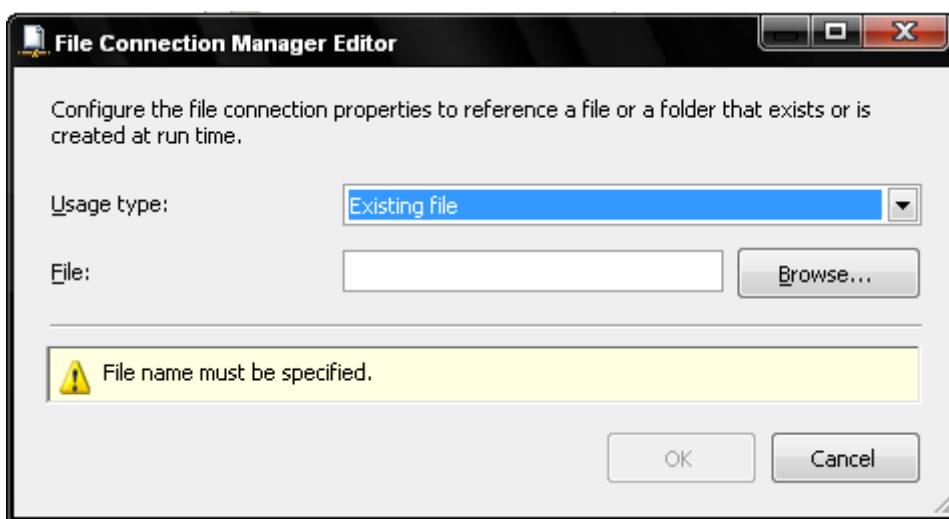
OK Cancel Help

La section General vous permet de rendre le nom et la description du composant un peu plus explicite.

Dans notre cas, les propriétés qui nous intéressent réellement sont les autres.

Tout d'abord, Operation. On peut effectuer différentes opérations sur le système de fichier (copie de fichiers, suppression, copie de dossier, création, ...). Celle par défaut nous convient très bien.

Vient ensuite SourceConnection. Comme vous le constaterez tout au long de ce tutoriel, SSIS travaille énormément avec ce qu'il appelle des connexions. Une connexion est un lien entre un support (Base de données, fichier, ...) et le package SSIS. C'est elle qui fera « physiquement » les accès aux données. Nous n'en avons pas à notre disposition, nous allons donc en créer une. Dépliez la liste qui apparaît pour la propriété SourceConnection et choisissez l'option Nouvelle Connection, l'écran suivant apparaît :



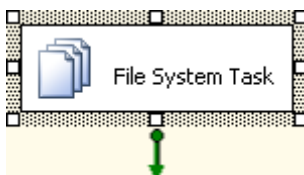
Ici, vous n'avez qu'à spécifier le fichier à utiliser. Pour information, le mien est le suivant :

C:\WINDOWS\system32\Logfiles\W3SVC1\ex070526.log

Il faut réaliser la même opération pour le fichier de destination, avec une nouvelle connexion, sauf que cette fois-ci vous préciserez qu'il s'agit d'un dossier existant et vous sélectionnerez le dossier de destination de la copie (pour moi c:\temp).

Pour l'option OverwriteDestination, vous pouvez mettre True afin de ne pas être embêté par plusieurs exécutions du package.

Si vous avez tout fait correctement, le composant File System Task doit avoir la tête suivante :



Plus de croix blanche sur fond blanc.

Pour vérifier, faites comme pour une application classique dans Visual Studio, F5 pour lancer son exécution.

Le package est exécuté. La couleur du composant doit changer à jaune, je ne sais pas si vous aurez le temps de le voir si le fichier est petit, ça veut dire que la tâche est en cours d'exécution.

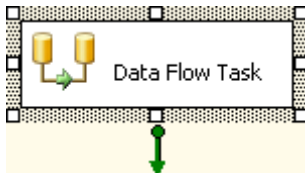
Si vous avez bien travaillé, elle doit ensuite passer au vert une fois terminée pour dire que tout s'est bien déroulé. Sinon, vous obtiendrez un joli rectangle rouge synonyme d'erreur.

Passons maintenant au gros du travail, l'analyse et l'insertion en base du fichier de log.

Traitement du fichier de log

Le traitement du fichier de log se fera dans un composant particulier, que vous trouverez à côté du composant File System Task dans la boîte à outils : le Data Flow Task.

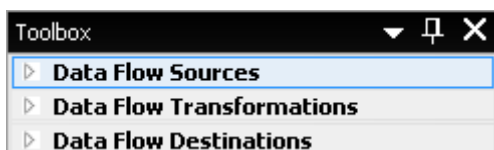
Déposez-en un sur l'espace de travail du panneau Control Flow :



Vous remarquerez sans doute que lui n'a pas le droit à la petite croix blanche. L'explication est simple.

Un composant Data Flow Task est un container, un peu comme le panel en WinForms, donc s'il est vide, aucune action à réaliser, pas très utile, mais aucun risque d'erreur.

Double cliquez dessus pour l'éditer. Visual Studio bascule automatiquement sur l'onglet Data Flow du package. C'est là que nous allons spécifier les tâches que devra accomplir notre composant Data Flow Task. La boîte à outils change, et laisse apparaître 3 catégories de composants :



- Les sources
- Les transformations
- Les destinations

Je ne vais pas entrer dans les détails, mais dans Data Flow Sources et Data Flow Destination vous retrouverez les composants permettant respectivement d'obtenir et de sauvegarder les données.

Dans transformations vous retrouverez les composants pour manipuler les données : conversions, fusion, ...

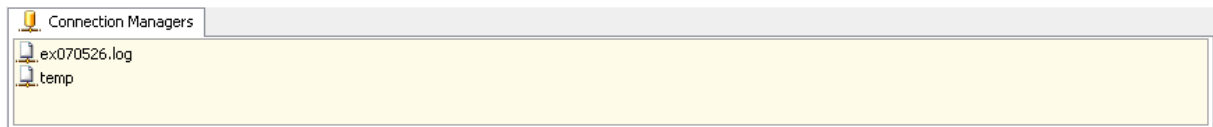
Passons maintenant à la première étape, la lecture des données de notre fichier de log

Lecture du fichier de log

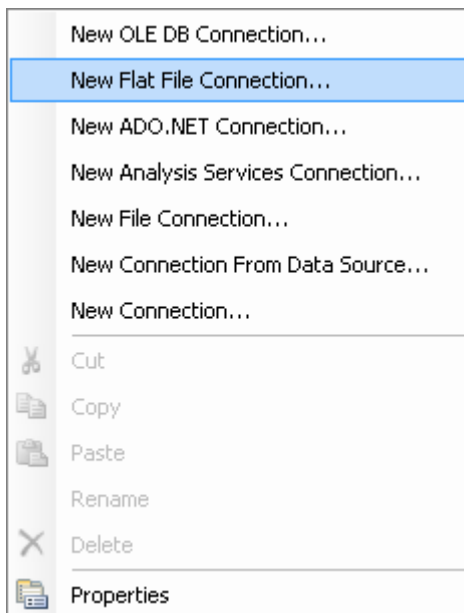
Comme je vous en ai déjà parlé pour la copie du fichier, SSIS travaille avec des connexions pour accéder aux supports « physiques ». La première chose à faire est donc d'ajouter une connexion à notre fichier de log.

La connexion au fichier de log

Si vous jetez un œil en bas du panneau Data Flow, vous verrez un espace nommé Connection Managers dans lequel vous retrouverez les deux connexions utilisées pour la copie du fichier de log :



Un petit clic droit laissera apparaître le menu suivant dans lequel vous choisirez New Flat File Connection... :



L'écran suivant apparaît dans lequel nous allons configurer notre connexion :

Flat File Connection Manager Editor

Connection manager name:

Description:

General
Columns
Advanced
Preview

Select a file and specify the file properties and the file format.

File name: **Browse...**

Locale: French (France) ☐ Unicode

Code page: 1252 (ANSI - latin I)

Format: Delimited

Text qualifier: <none>

Header row delimiter: {CR}{LF}

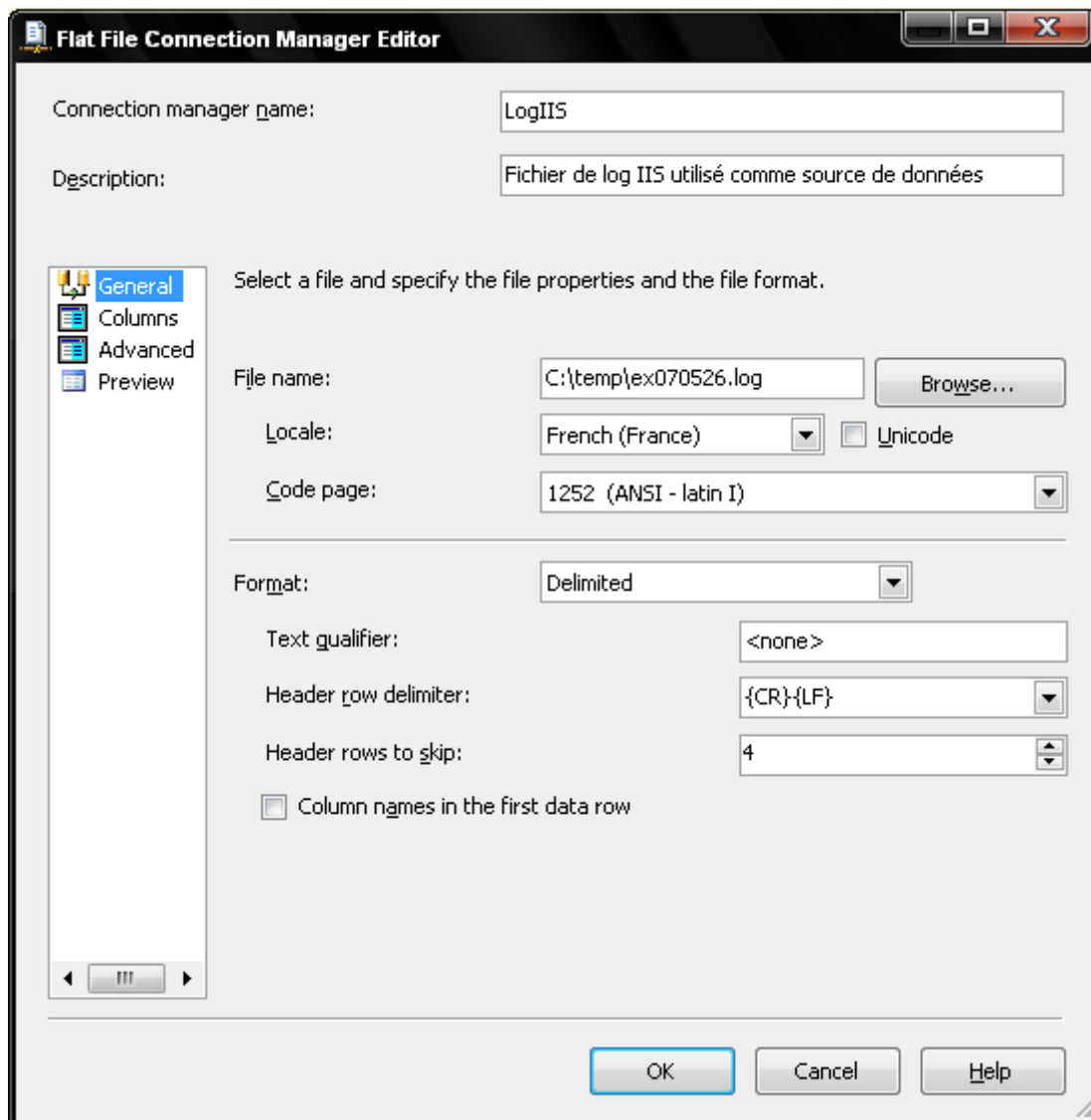
Header rows to skip: 0

☐ Column names in the first data row

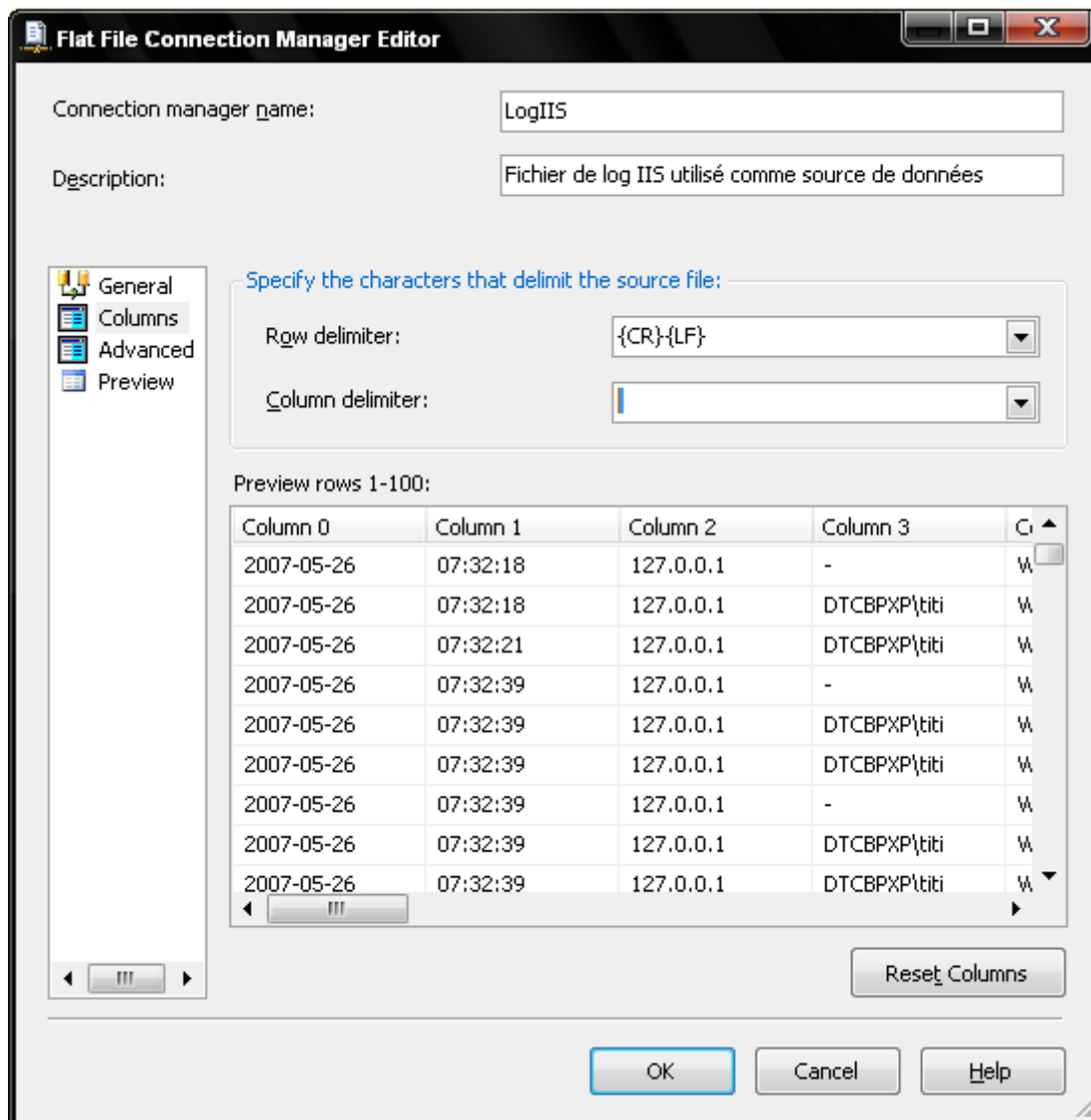
A valid file name must be selected.

OK Cancel Help

Je ne vais pas tout détailler, je pense que l'écran est assez clair pour que vous puissiez le remplir seul.
Voici quand même à quoi il doit ressembler :

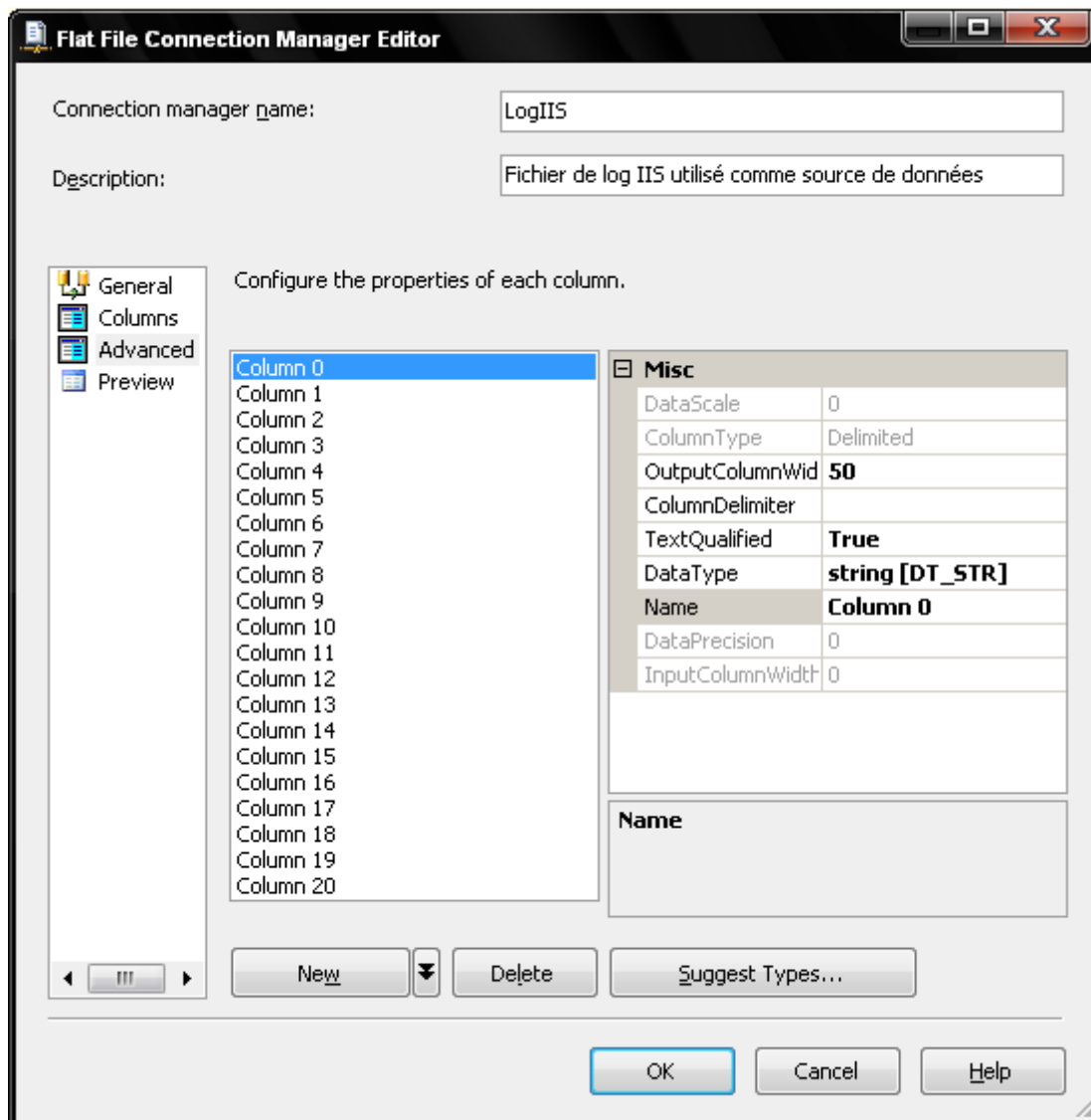


Passez maintenant à la section Columns, c'est là que nous allons enfin voir le contenu de notre fichier de log en le configurant comme suit :

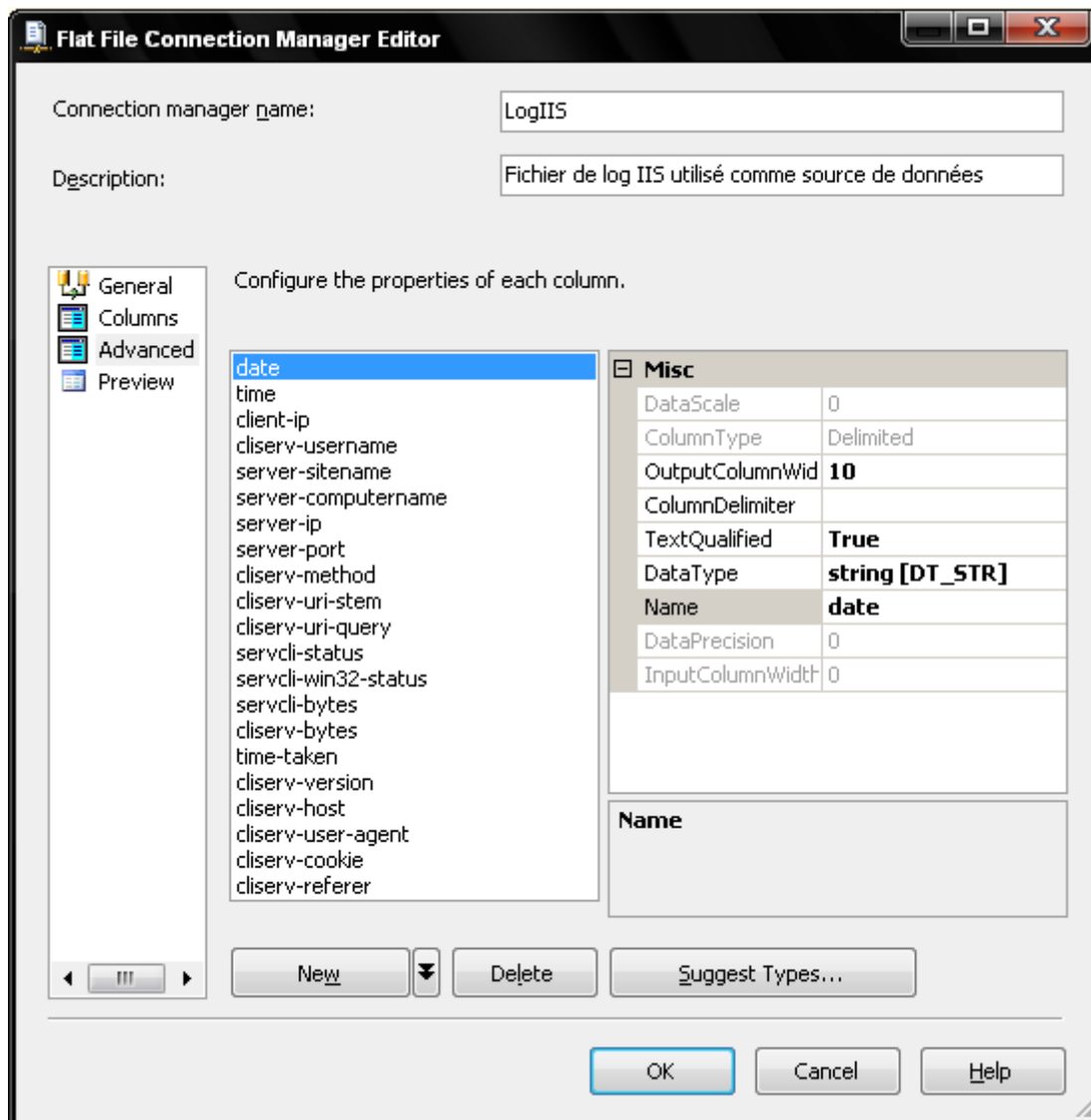


Pour Column Delimiter, spécifiez bien un espace (désolé, c'est pas très visible sur la capture :p).

Dans la section Advanced, il est possible de définir les options avancées des colonnes que nous venons de voir :



Il est donc possible de modifier chaque colonne, de lui donner un nom compréhensible par la suite, vous verrez que ça peut se révéler très utile, sont type de sortie, très pratique. Voici à quoi cette section ressemble après configuration :

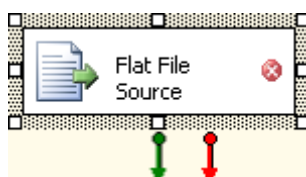


Pour la taille des colonnes de sortie, donnez la même taille de sortie que pour la taille du champ correspondant en base. Validez, notre connexion est prête, il ne nous reste plus qu'à exposer nos données dans le Data Flow.

Flat File Source

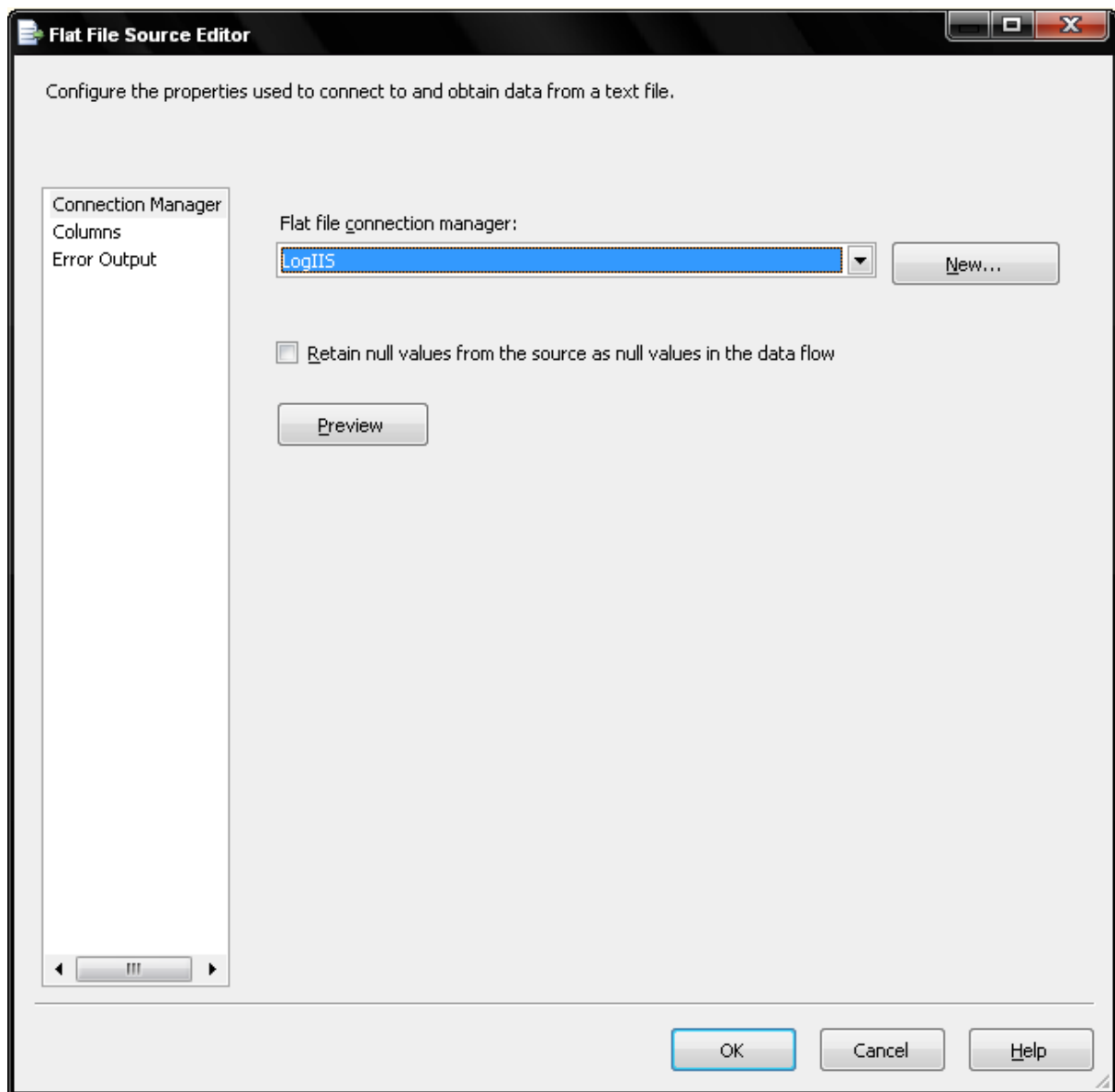
Maintenant que nous avons configuré notre connexion au fichier de log, il faut exposer ses données au package. C'est là qu'intervient le composant Flat File Source.

Comme son nom l'indique, c'est une source de données.

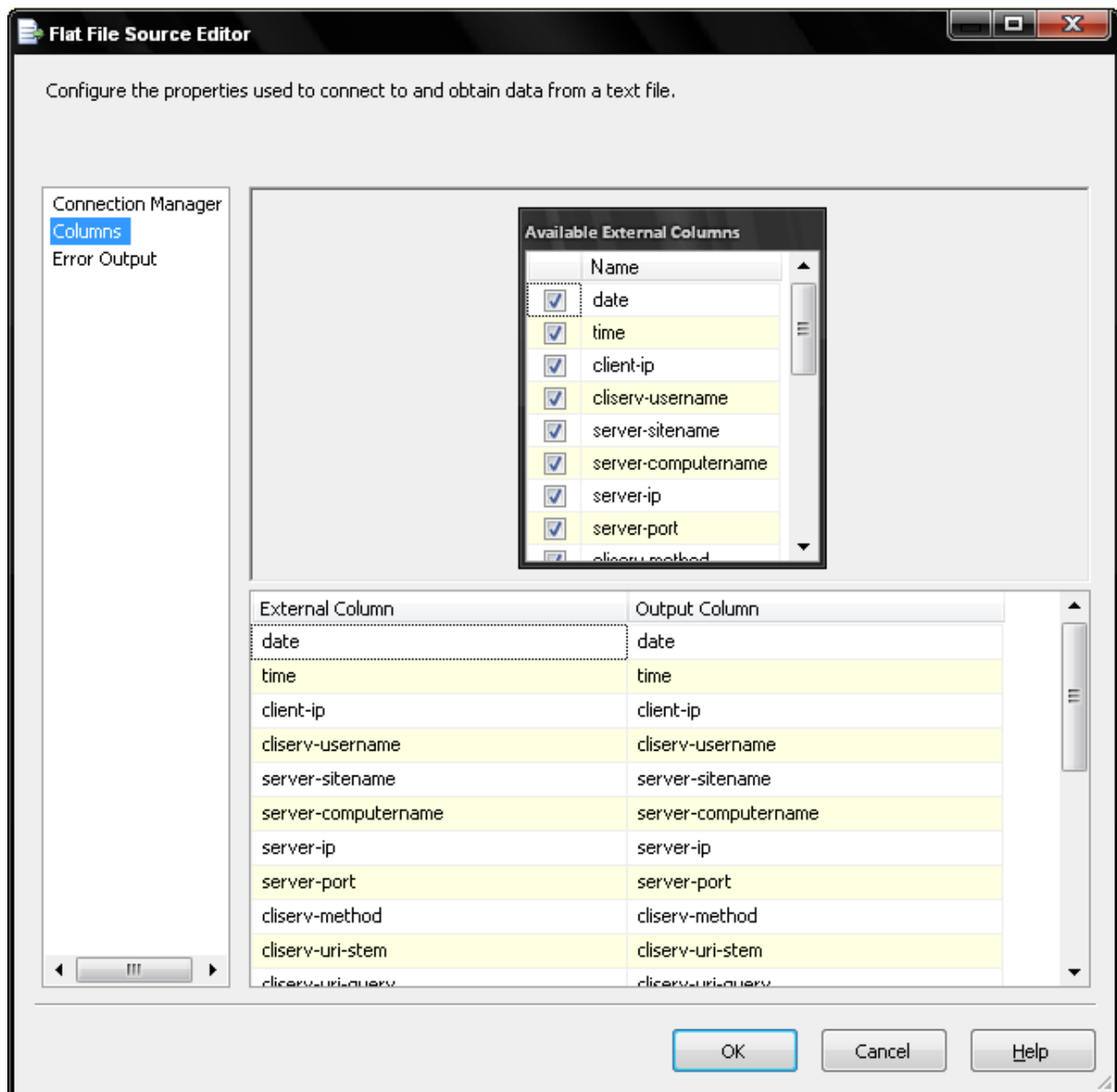


Comme vous pouvez le constater, il est en erreur. Si on y réfléchit deux secondes, c'est logique. Son rôle est d'exposer des données au reste des composants du Data Flow. Mais quelles données ? Comme il n'est pas configuré, il ne sait pas lui-même où sont les données qu'il doit exposer.

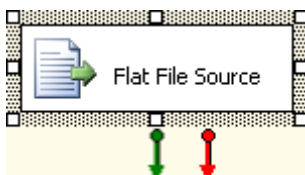
Un double clic sur le composant vous permettra d'accéder à son écran de configuration :



L'assistant détecte automatiquement la connexion appropriée au type de notre composant. Un composant Flat File Source prend logiquement une connexion de type Flat File. Si vous regardez la section colonnes de cet écran de configuration, vous verrez qu'il expose bien les colonnes définies dans la connexion :



Terminons la configuration, sans rien faire du tout :p, en cliquant sur OK.



Hop, notre composant n'est plus en erreur, passons à la suite.

Création de la colonne DateTime

Si vous avez été bien attentif, vous avez sans doute remarqué que pour les informations de date et d'heure, la structure du fichier de log nous renvoie 2 colonnes et que la structure de notre table nous propose une unique colonne de type DateTime.

Nous allons donc devoir transformer les informations fournies par notre source afin de créer une nouvelle colonne de type DateTime à partir de deux colonnes de type Texte.

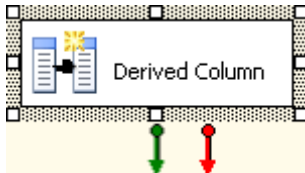
L'opération est la suivante :

Concaténer « jj/mm/aaaa » et « hh :mm :ss » en une chaîne « jj/mm/aaaa hh :mm :ss » et la convertir en date.

Rassurez-vous, rien de bien compliquer la dedans.

Ce genre d'opération est réalisé par un composant de type Data Flow Transformation : Derived Column.

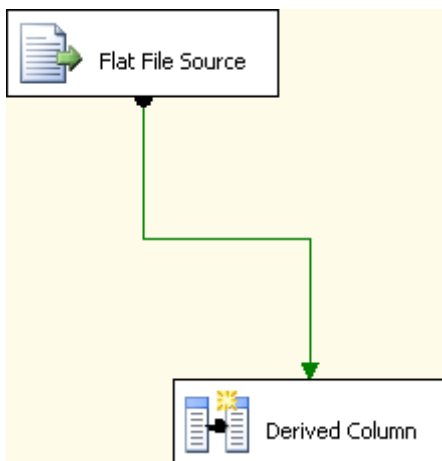
Déposez donc un composant Derived Column sur l'espace de travail.



Pour pouvoir créer une colonne à partir d'un autre, il faut bien sur qu'il sache les colonnes qu'il a à sa disposition. Pour cela, nous allons relier notre composant Flat File Source à notre composant Derived Column.

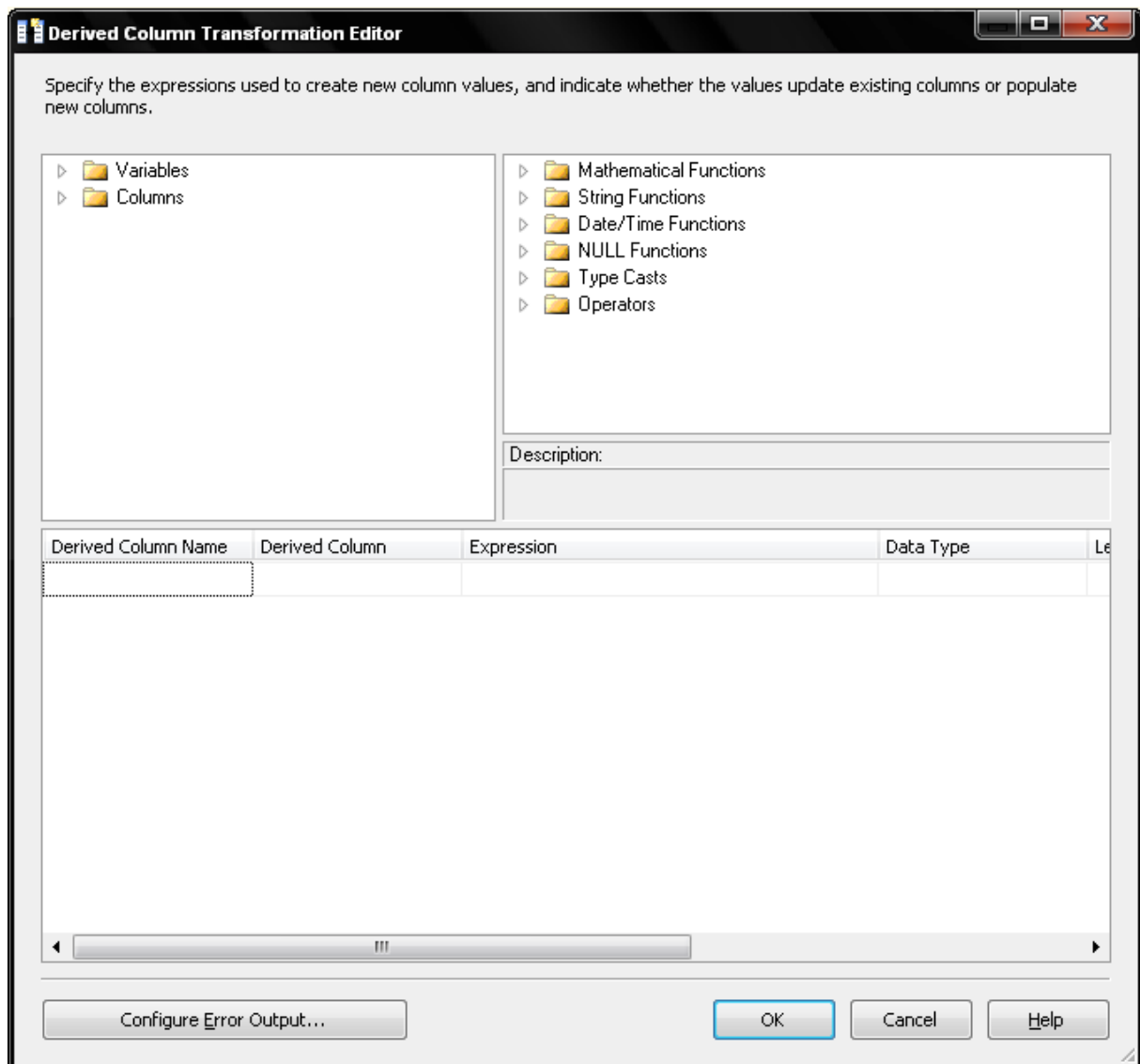
Cette opération est très simple et valable pour tous les composants qu'utilise SSIS, aussi bien dans un Control Flow que dans un Data Flow.

Commencez par sélectionner le composant Flat File Source. Une flèche verte apparaît en dessous, c'est celle la qui nous intéresse. Elle représente le flux de données. La rouge représente le flux d'erreur. Nous ne nous en servons pas ici. Pour connecter deux composants donc, il faut faire glisser la flèche verte du composant source sur le composant de destination. Vous devriez obtenir quelque chose qui ressemble à ceci dans notre cas :



Dans un Data Flow, cette flèche signifie que le composant d'où part la flèche fournit des données au composant pointé par la flèche.

Après ces quelques petites précisions, voyons maintenant comment créer notre date pour l'exploiter par la suite. Double cliquez sur le composant Derived Column. L'écran de configuration suivant apparaît :



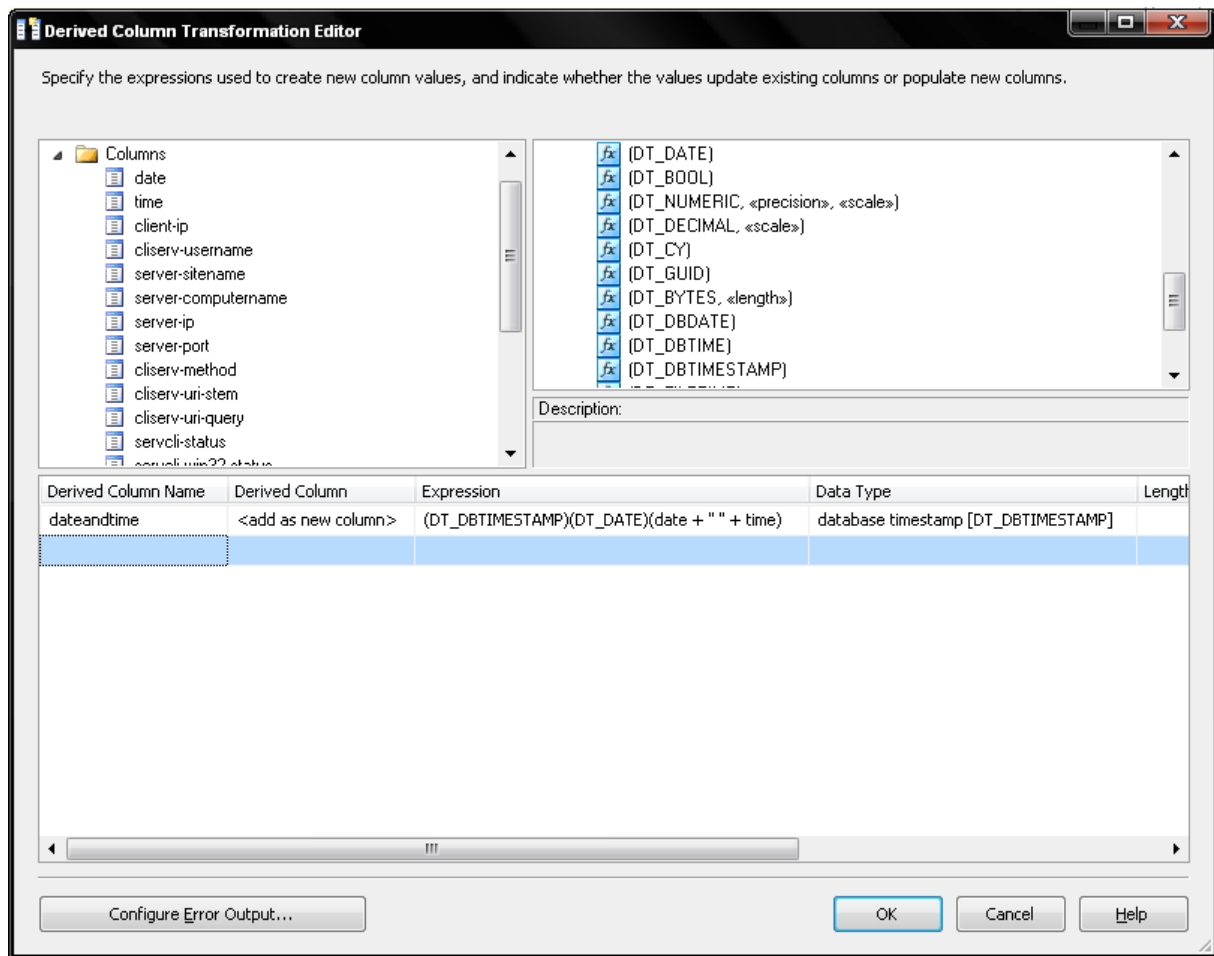
Il se divise en trois parties :

- Les sources de données (nous mettrons de côté Variables qui dépassent le cadre de ce tutoriel)
- Les opérations : mathématiques, sur les dates et ... oh ben tiens, de conversion
- Les colonnes dérivées

Donc ce que nous allons faire, c'est créer une colonne dérivée, qui sera le résultat de l'expression. Notez que vous pouvez créer plusieurs colonnes dérivées en une seule fois.

Ici nous allons créer la colonne dérivée « dateandtime » qui sera le résultat de l'expression suivante « (DT_DBTIMESTAMP)(DT_DATE)(date + " " + time) »

L'expression reprend la syntaxe du C#, on concatène les colonnes date et time avec un espace entre les deux, et on converti successivement le résultat en type DT_DATE puis DT_DBTIMESTAMP.



Voilà, nous disposons maintenant d'une colonne supplémentaire dans nos données, appelée dateandtime et de type DT_DBTIMESTAMP qui sera parfaite pour la colonne dateandtime de notre table en base.

Passons à la conversion des données.

Conversion des données

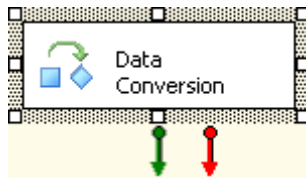
Pourquoi convertir les données ?

Si vous regardez bien la connexion au fichier de log, vous apercevrez une option Unicode, qui est décochée.

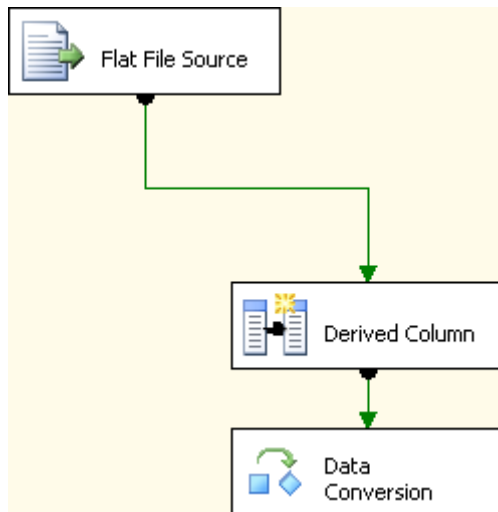
Si maintenant vous regardez bien la structure de notre table, vous verrez que les champs de type text sont de type nvarchar, donc Unicode.

Comme monsieur SSIS est assez pointilleux au niveau des types de données, nous allons devoir convertir les données non Unicodes de notre fichier texte en données Unicodes pour pouvoir les insérer en base.

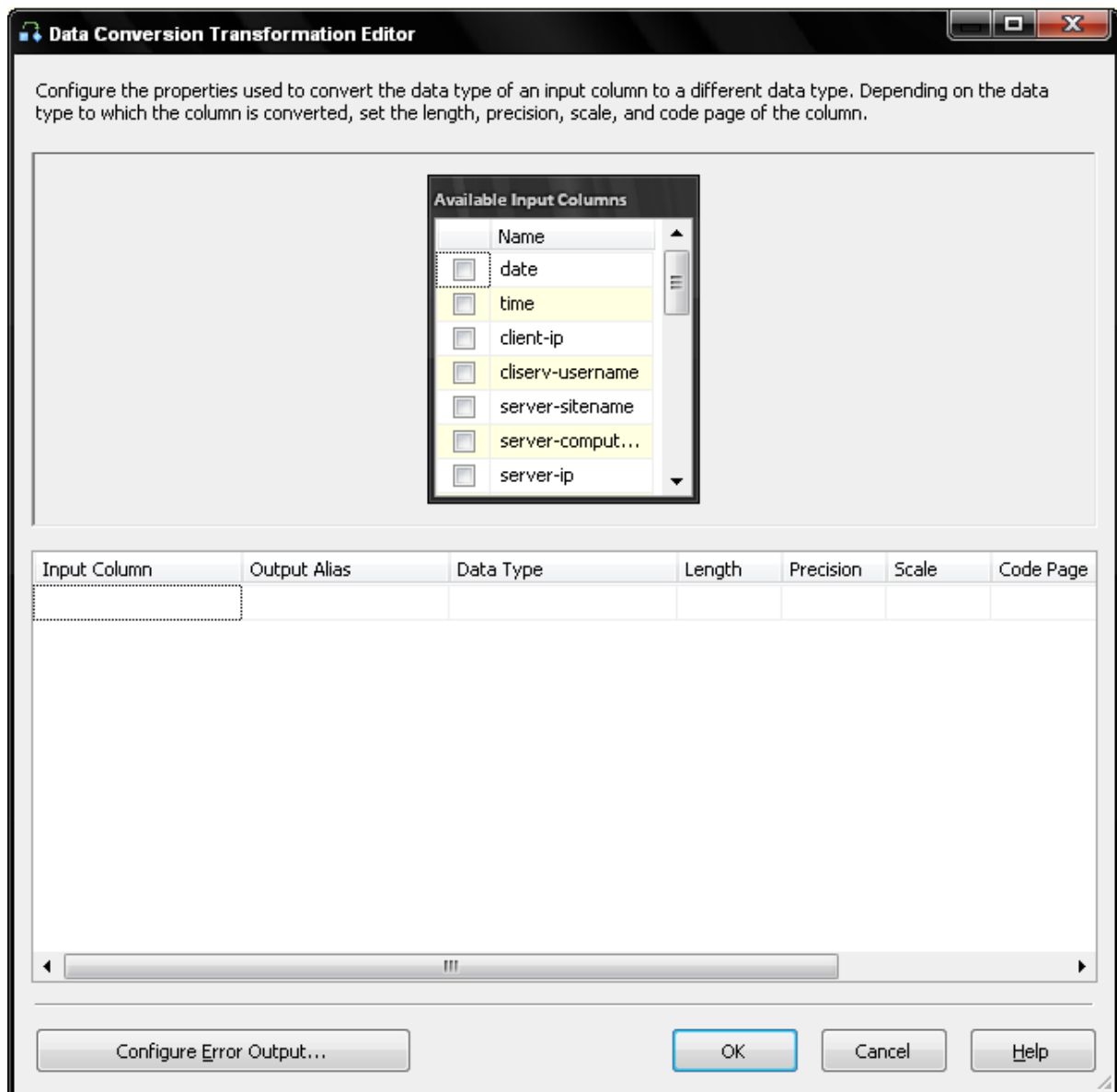
Mais encore une fois, SSIS nous propose un composant qui va se charger de réaliser la conversion pour nous, c'est le composant Data Conversion qui intervient :



Comme pour le composant Derived Column, pour savoir ce qu'il peut convertir, nous devons lui fournir des données. Nous allons donc relier la sortie de notre composant Derived Columns (flèche verte) à notre composant Data Conversion :

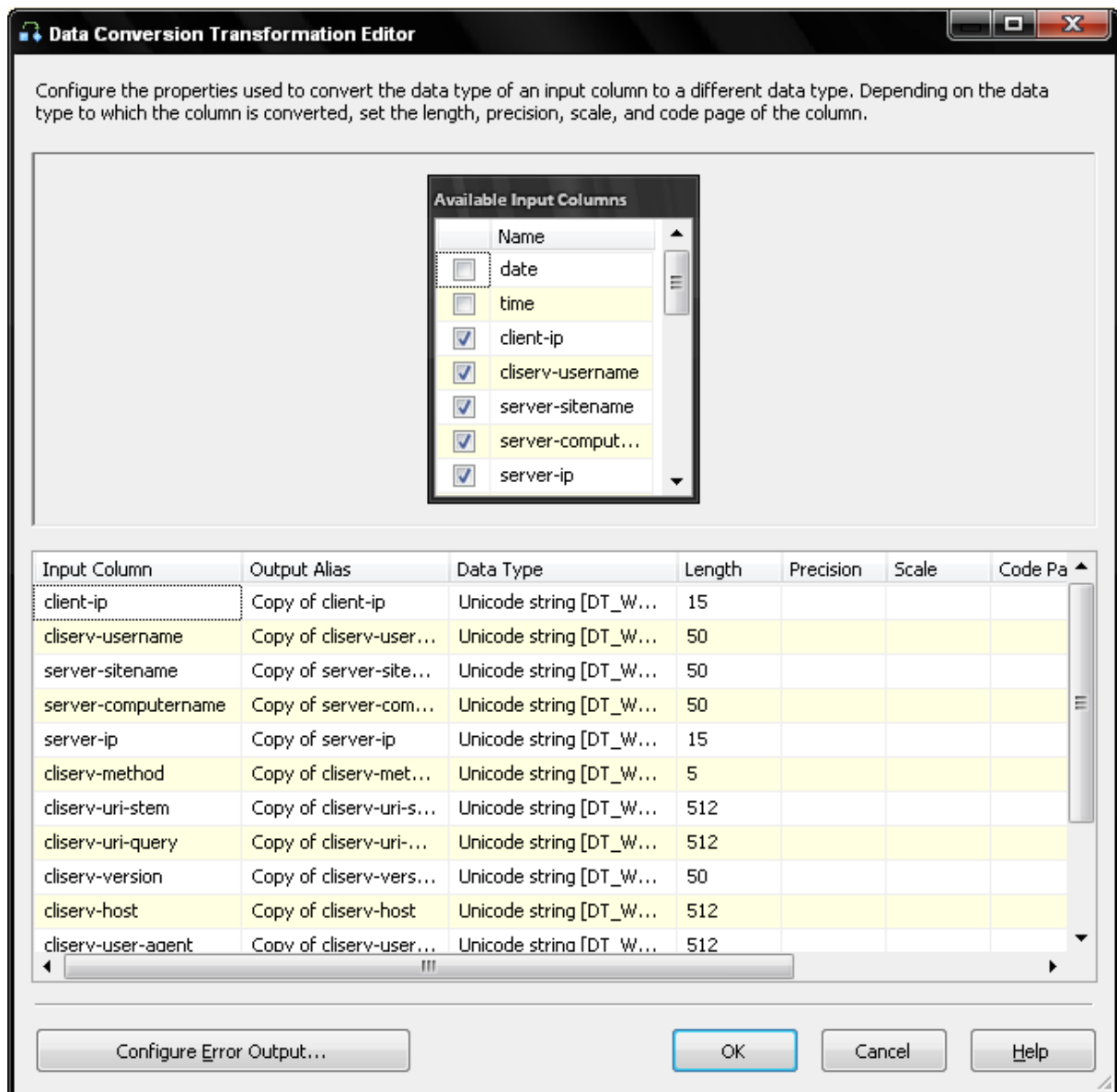


Comme d'habitude maintenant, ouvrez l'éditeur du composant Data Conversion :



Toutes les colonnes qui seront affectées par cette conversion sont les colonnes de type texte. Sélectionnez-les dans les colonnes Available Input Columns. Elles apparaîtront dans la liste en dessous. Modifiez le Data Type en Unicode String [DT_WSTR]. Faites bien attention que la longueur de la chaîne corresponde bien à la taille du champ en base.

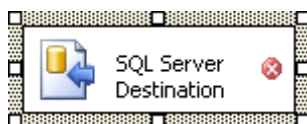
Une fois terminé, vous devriez obtenir ceci :



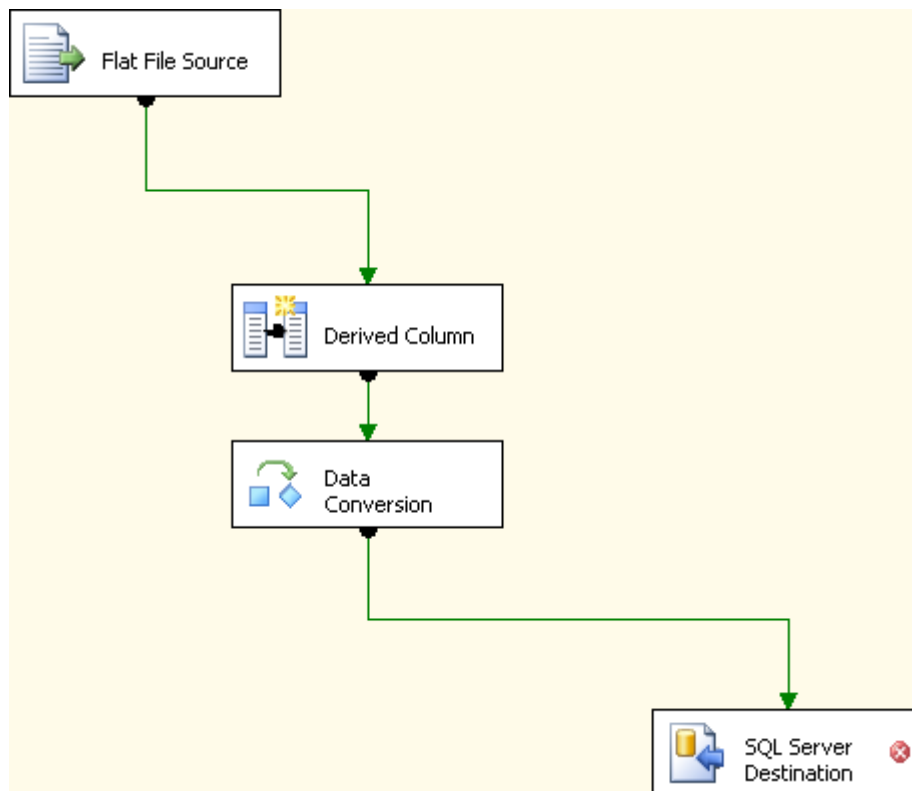
Terminez l'éditition en validant, nous allons maintenant passer à l'insertion en base.

Ecriture en base

Comme nous utilisons une base SQL Server, nous utiliserons le composant SQL Server Destination, quelle surprise. Vous pouvez bien sûr utiliser d'autres destinations, comme un fichier texte, ou un autre SGBD, enfin bref, nous c'est SQL Server.

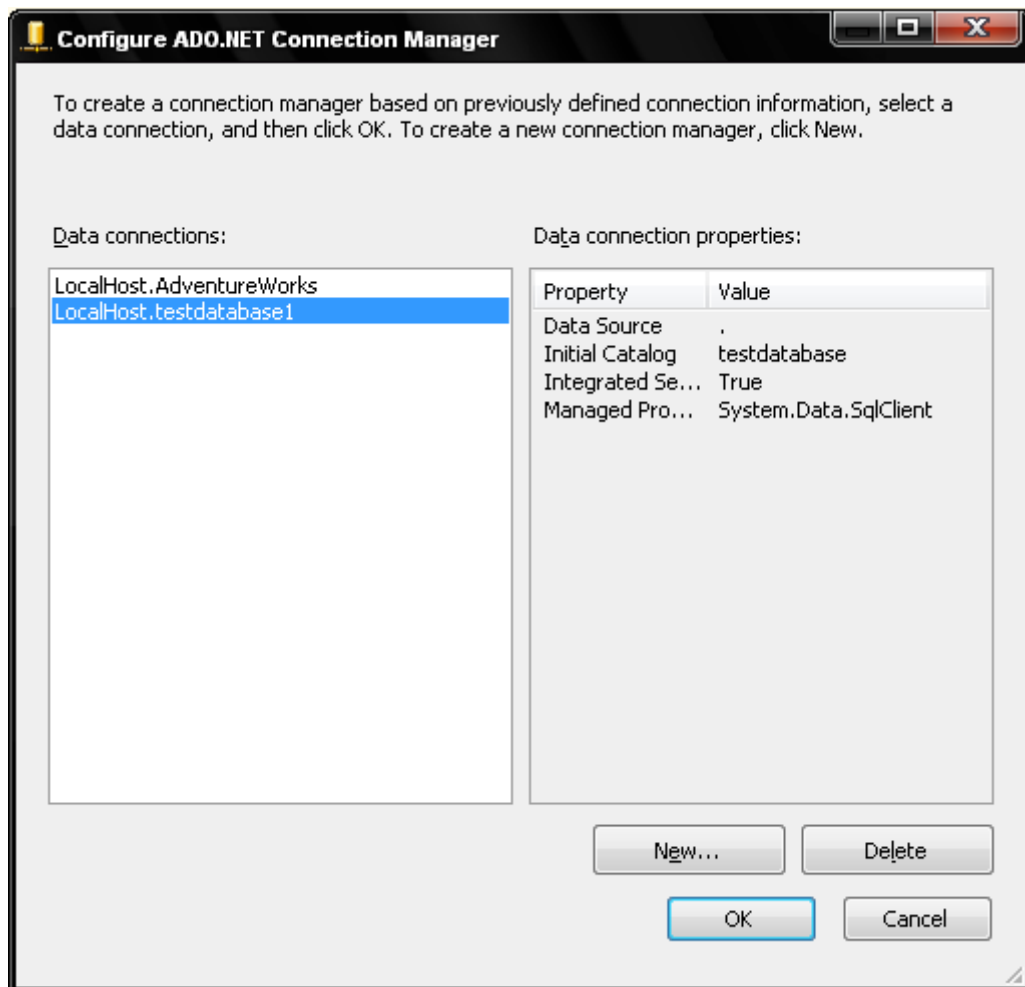


Le composant est en erreur, normal, il ne sait pas où il doit se connecter pour écrire. Qu'à cela ne tienne, on va lui fournir une connexion. Mais avant, nous allons quand même lui fournir les colonnes de sortie du composant de conversion :

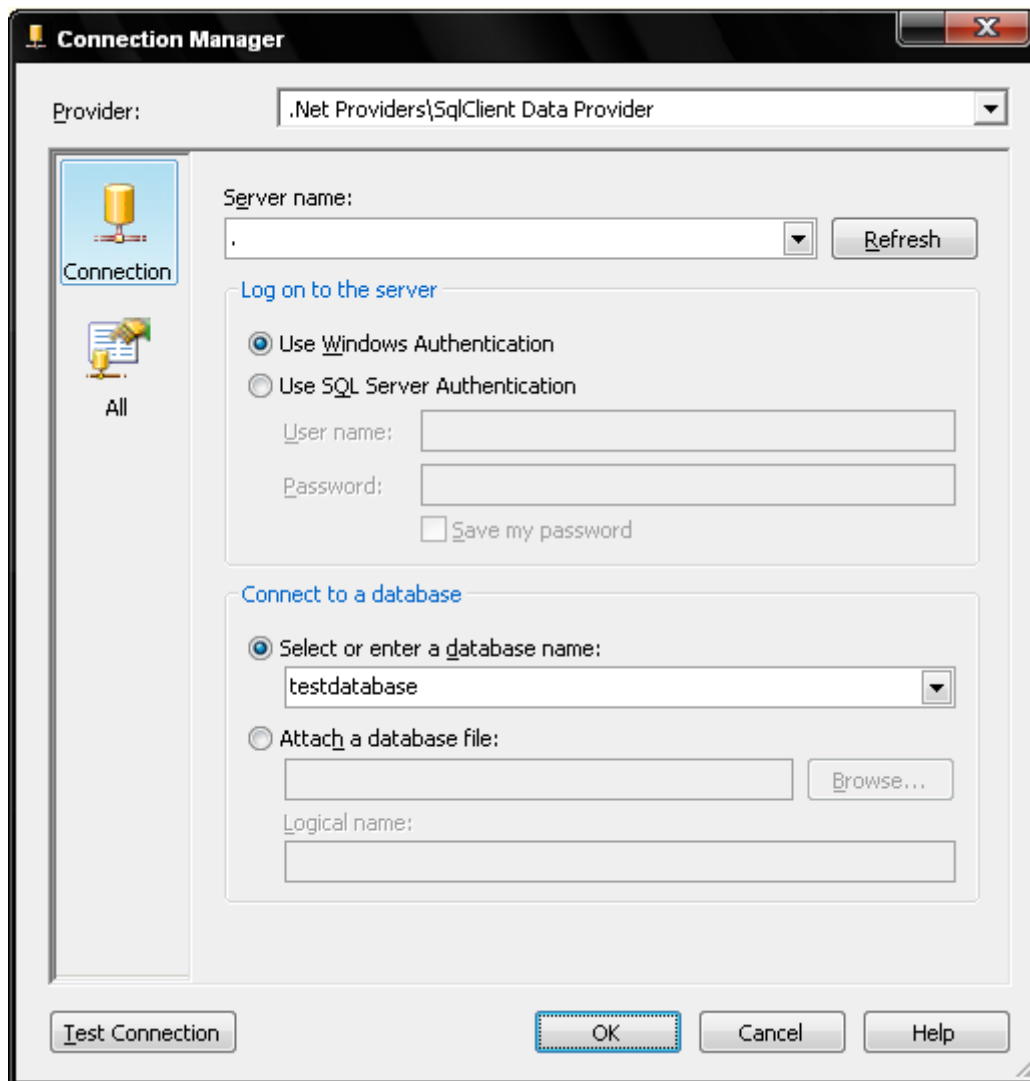


Une connexion à la base

Dans le Connexion Manager que nous avons vu tout à l'heure, ajoutez une nouvelle connexion ADO.Net.



L'assistant vous propose une liste des connexions disponibles. Si vous n'en avez pas, ce qui est sûrement le cas, créez-en une nouvelle :

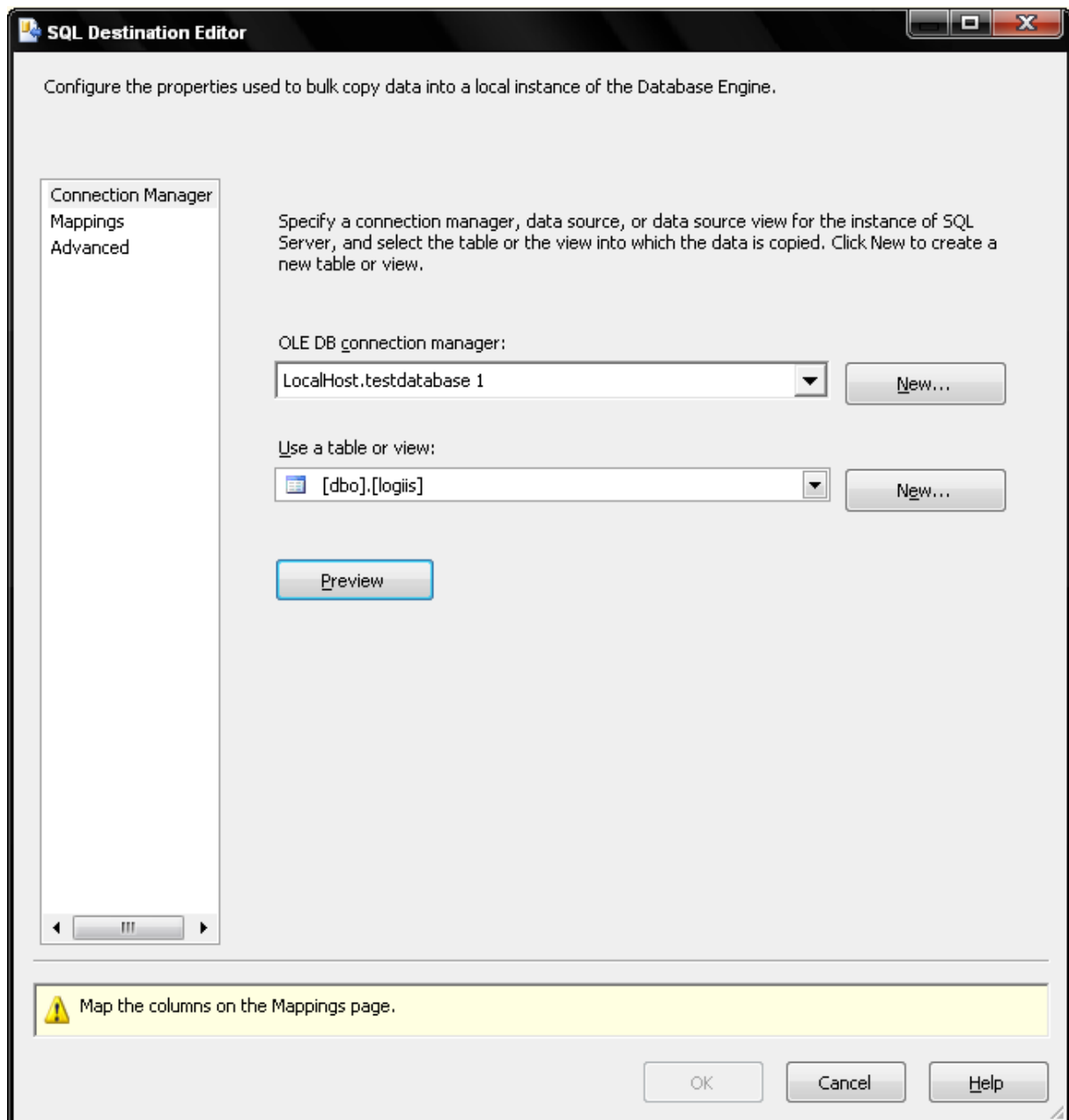


Validez le tout. Nous avons donc maintenant notre connexion à notre serveur SQL Server.

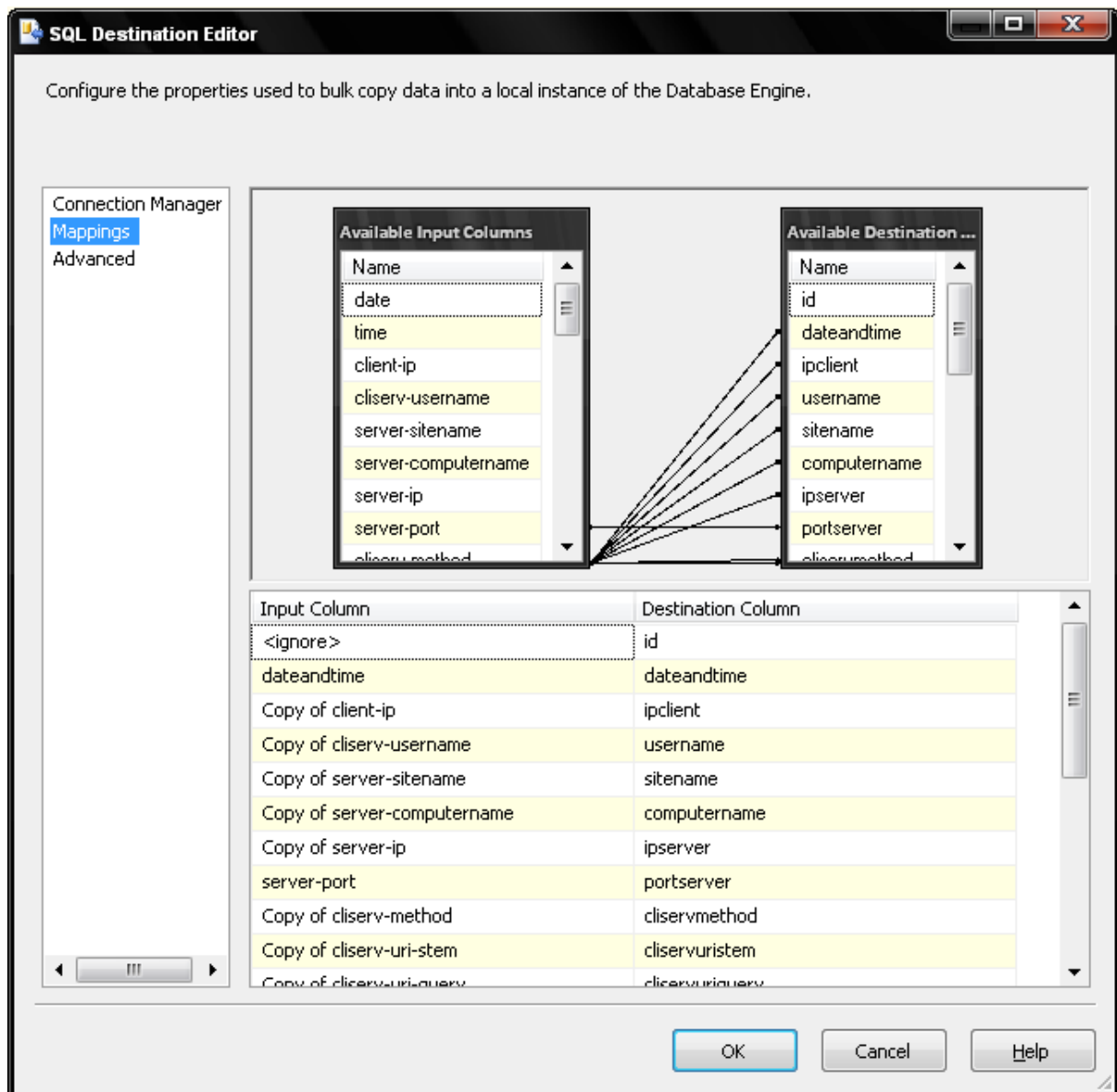
Liaison de nos données à la base

Maintenant que nous avons tout préparé, nous allons associer notre connexion à notre composant et vérifier que tout est bon.

Double cliquez sur le composant SQL Server Destination :



Si vous avez plusieurs connexions, choisissez celle qui correspond. Spécifiez ensuite la table de destination, ici logiis. Nous allons maintenant mapper les colonnes fournies depuis le fichier avec les colonnes de la table :



L'assistant est un minimum intelligent. Il associe par défaut les colonnes sources et les colonnes de destination ayant le même nom. A nous de faire le reste.

Id n'est associé à aucune colonne source puisque c'est une colonne auto-incrémentée.

Attention aussi à bien utiliser les colonnes résultant de la conversion de données et non pas celles d'origine, leur format n'est pas correct.

Une fois le mapping réalisé sur toutes les colonnes de destination ... on va tenter d'exécuter notre Data Flow.

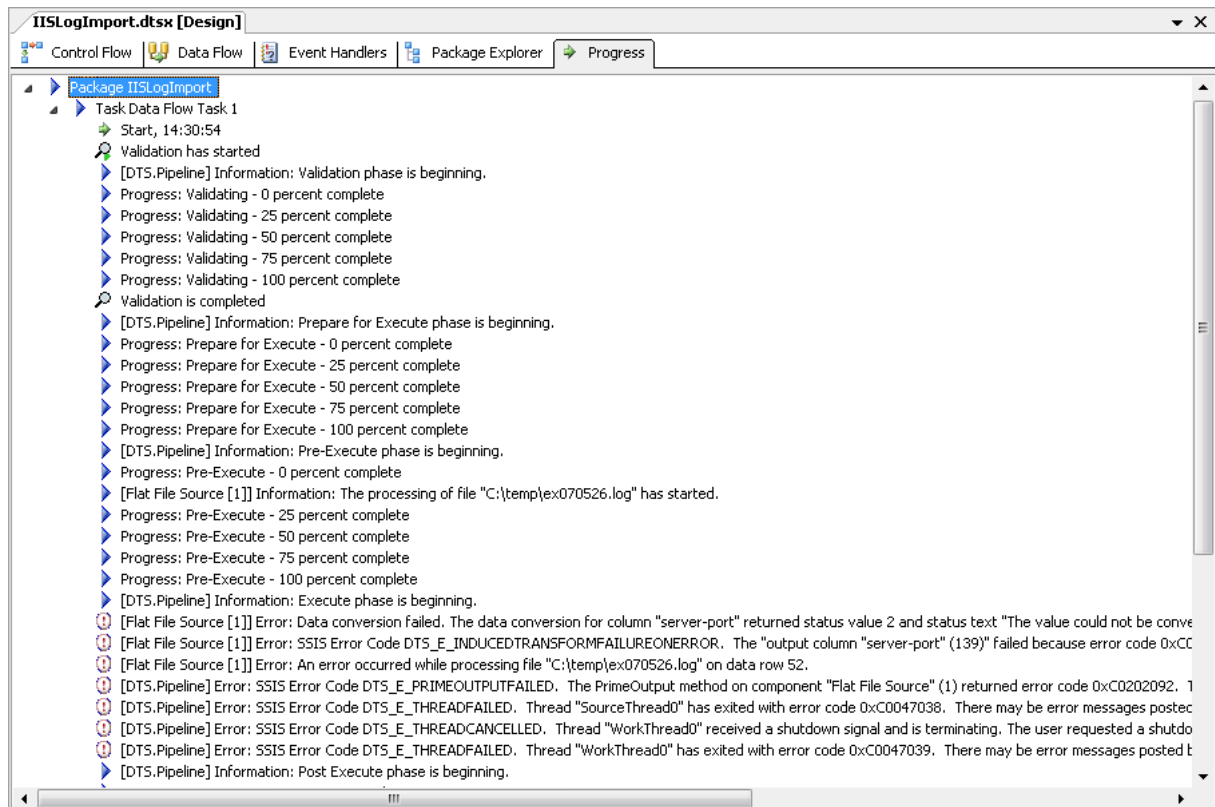
Première tentative d'exécution du Data Flow

Cette étape va nous permettre de faire une petite analyse de notre Data Flow et de voir si tout se déroule correctement.

Toujours dans le panneau de travail du Data Flow, ouvrez le menu contextuel du Data Flow et choisissez la commande Execute Task.

Et là, drame. Erreur dès la première étape. Ne vous inquiétez pas, c'est normal. Quand je vous disais que SSIS était pointilleux ...

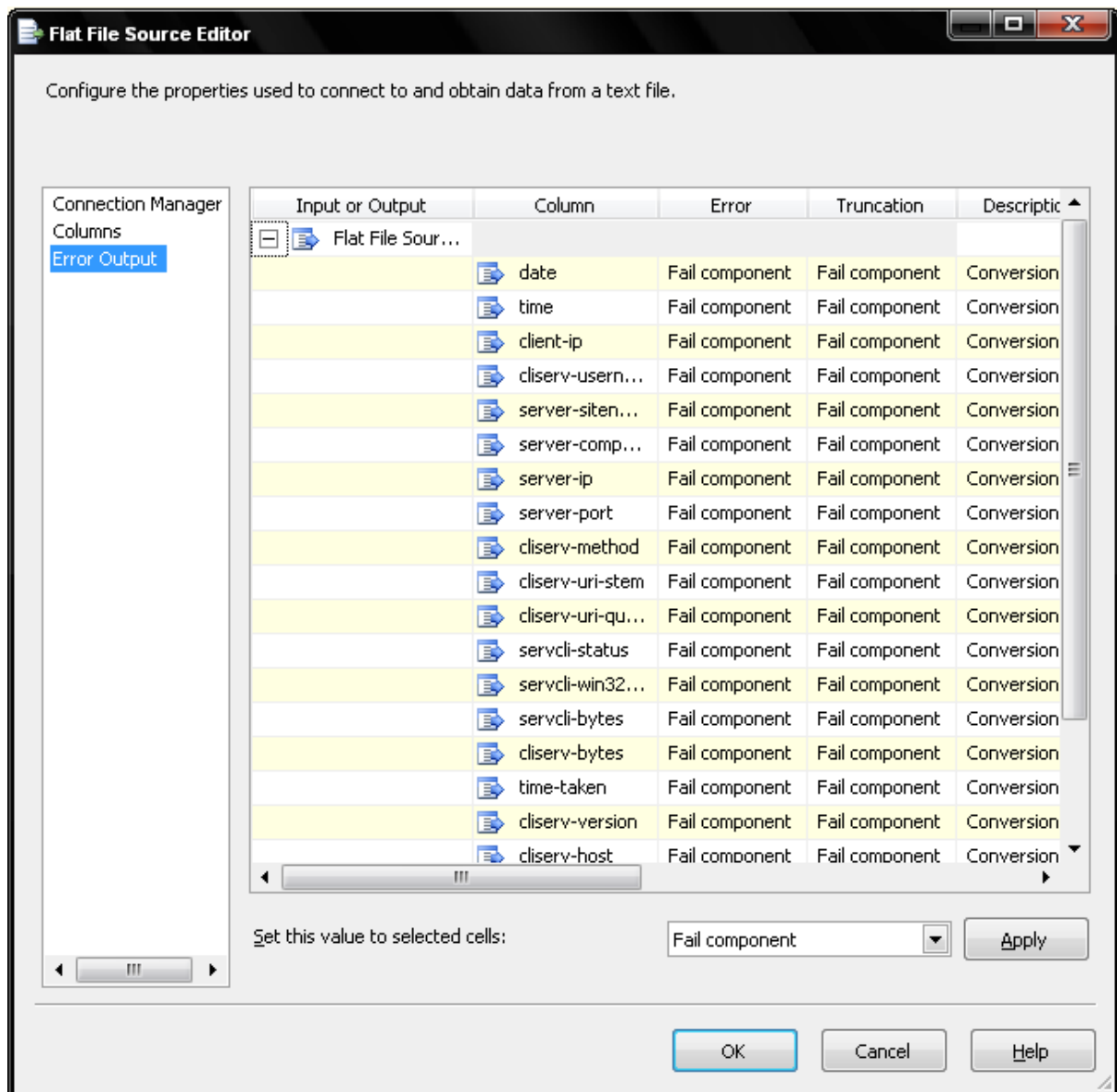
Si vous regardez dans l'onglet Progress, vous obtiendrez une liste très détaillée de ce qu'à fait SSIS lors de l'exécution du package et surtout les erreurs qu'il a rencontré :



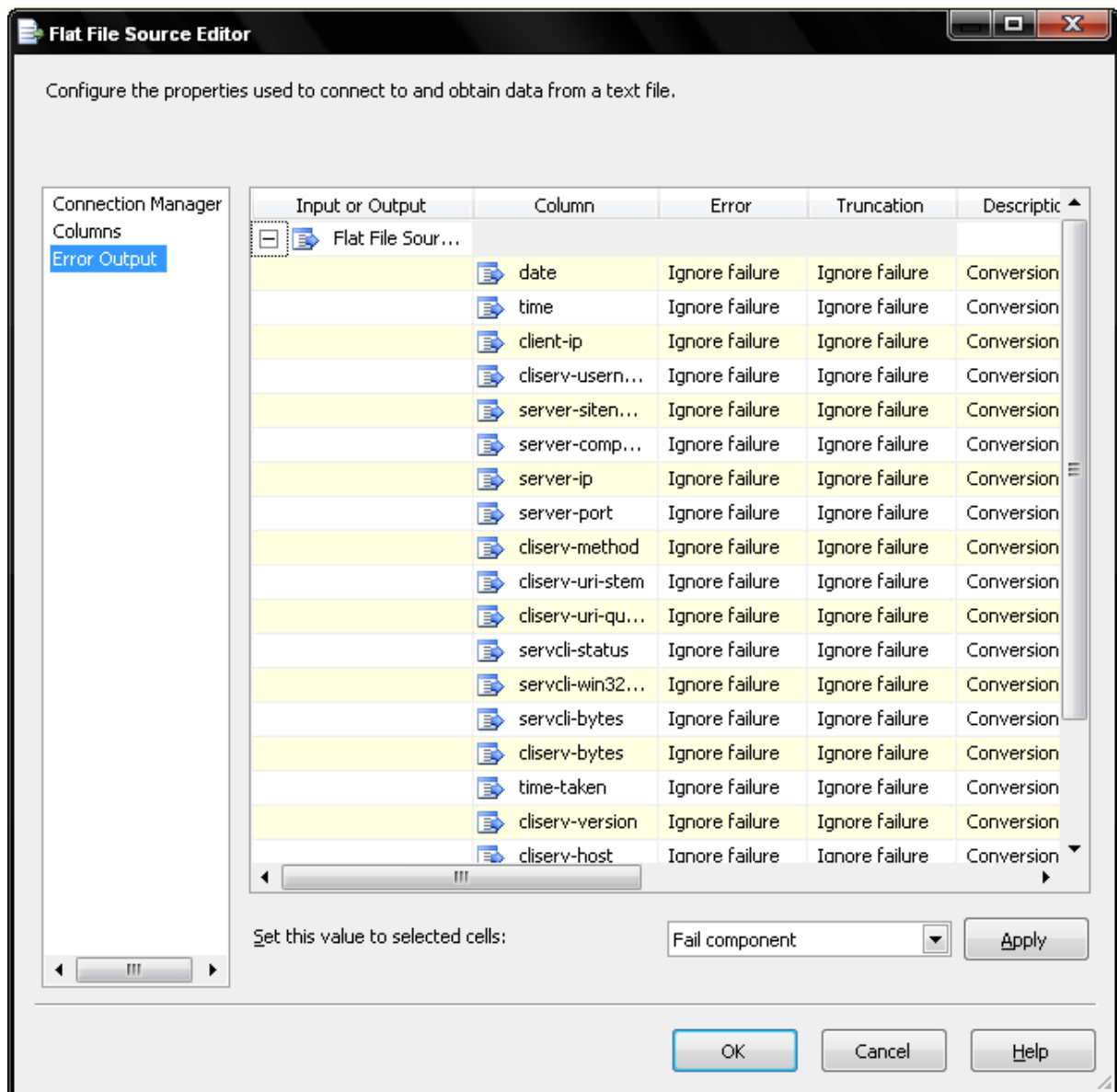
Vous voyez qu'il y a quelques erreurs. Rien de dramatique, et surtout pas réellement une erreur.

Le souci est que lors de la lecture du fichier, SSIS suspecte un troncage des données. Comme nous avons bien étudié la question avant, nous savons que c'est peu probable. Nous allons donc forcer SSIS à passer outre cette erreur.

Editez le composant Flat File Source (double clic), et rendez-vous dans la section Error Output :



Cette section de configuration va spécifier le comportement du composant en cas d'erreur. Ici, vous voyez qu'en cas de tentative de troncage des données, SSIS considère que le composant est en erreur et que donc l'exécution ne peut pas continuer. Nous allons donc dire à SSIS d'ignorer l'erreur dans ce cas et pour toutes les colonnes :

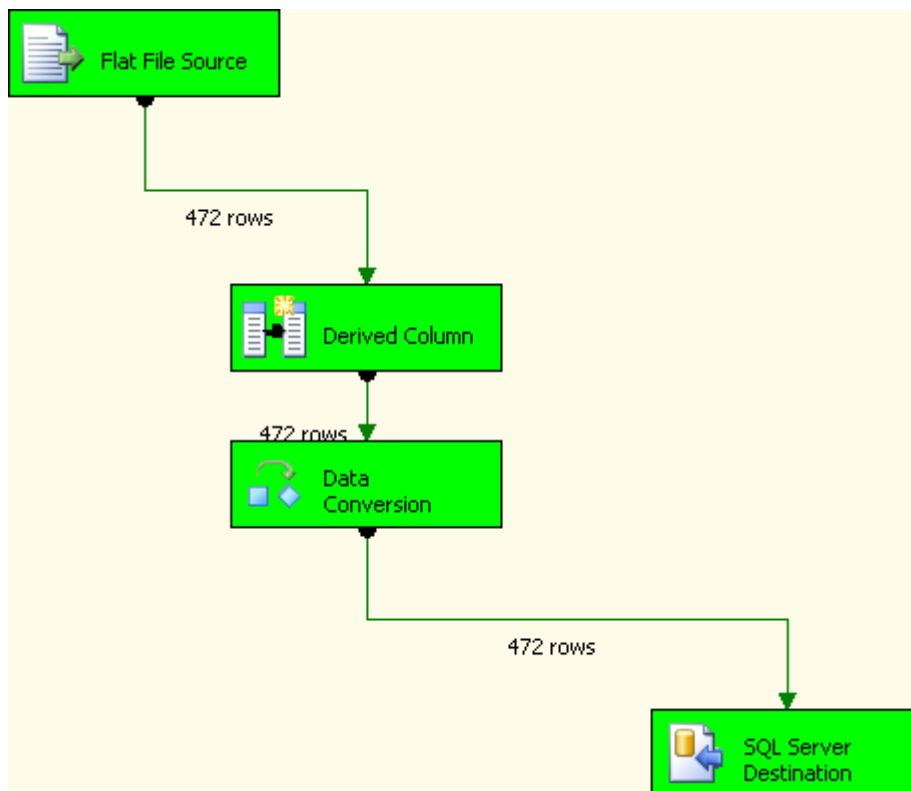


Validez et retentez d'exécuter le Data Flow.

Cette fois-ci une erreur lors de la création de la colonne dateandtime dans le composant Derived Column.

Même opération que pour le Flat File Source. Nous allons ignorer l'erreur.

Cette fois-ci, tout se passe bien :



Nous avons lu et écrit 472 lignes de log. Petite vérification des données dans la table :

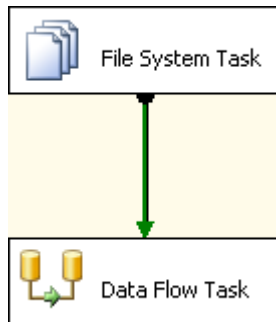
id	dateandtime	ipclient	username	sitename	computername	ipserver	portserver	
4364	26/05/2007 07:...	127.0.0.1	-	W3SVC1	DTCBPXP	127.0.0.1	80	C
4365	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	C
4366	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	C
4367	26/05/2007 07:...	127.0.0.1	-	W3SVC1	DTCBPXP	127.0.0.1	80	C
4368	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	C
4369	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	C
4370	26/05/2007 07:...	127.0.0.1	-	W3SVC1	DTCBPXP	127.0.0.1	80	P
4371	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4372	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4373	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4374	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4375	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4376	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4377	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4378	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	C
4379	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	C
4380	26/05/2007 07:...	127.0.0.1	-	W3SVC1	DTCBPXP	127.0.0.1	80	C
4381	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	C
4382	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	C
4383	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	C
4384	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	C
4385	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	C
4386	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4387	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4388	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4389	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4390	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P
4391	26/05/2007 07:...	127.0.0.1	DTCBPXP\biti	W3SVC1	DTCBPXP	127.0.0.1	80	P

Notre Data Flow est maintenant prêt, il ne nous reste plus qu'à terminer notre Control Flow.

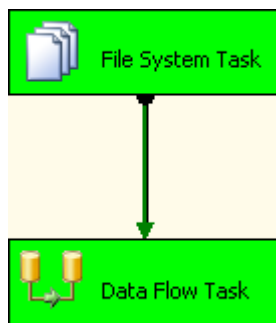
Dernier ajustement sur le Control Flow

Ce dernier ajustement dans le Control Flow est l'ordre d'exécution des tâches.

Rappelez-vous que nous devons d'abord copier le fichier avant de travailler dessus. C'est ce que nous allons faire. En plus, maintenant que vous connaissez la petite flèche verte, ça va aller tout seul. Je ne vous explique pas comment faire, je vous montre juste le résultat :



Lancez l'exécution du package comme une application standard (F5) :



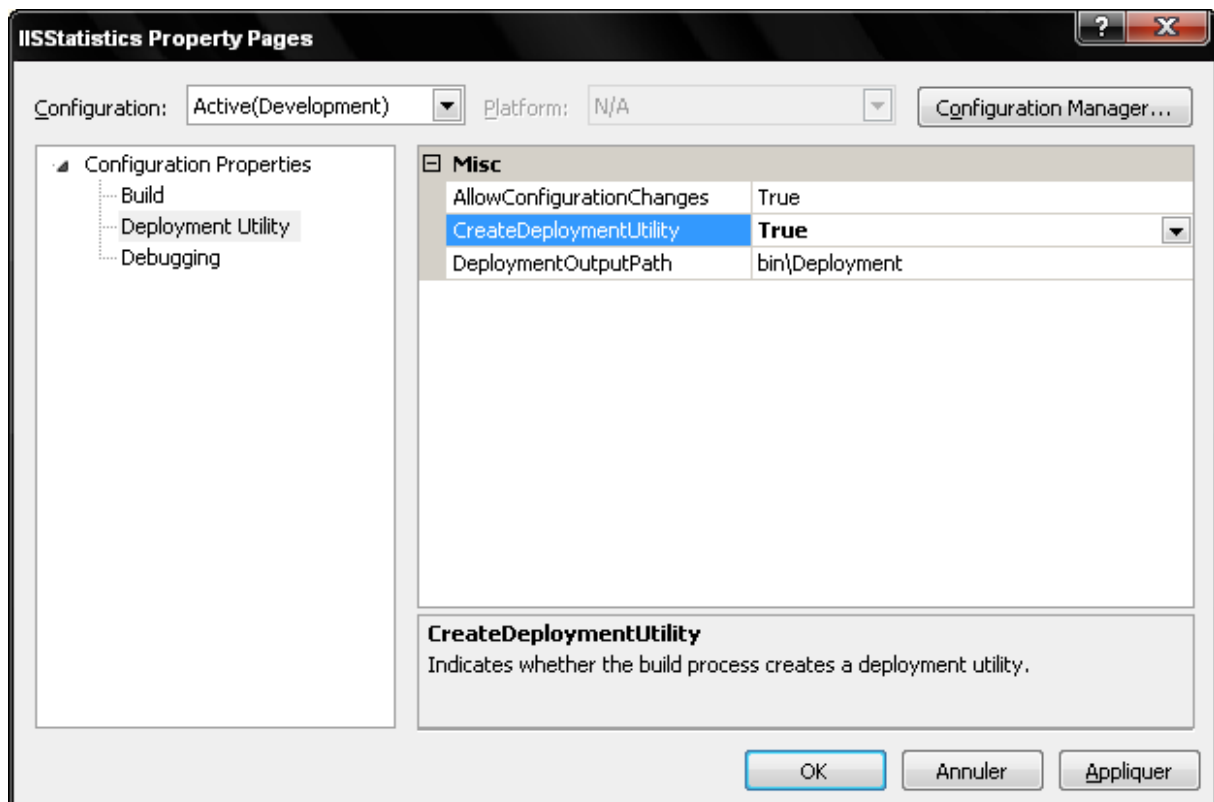
Tout s'est bien déroulé, nous avons bien travaillé.

Déploiement du package

Nous avons vu qu'il était possible d'exécuter le package depuis Visual Studio. C'est bien, mais ce n'est pas très pratique de devoir lancer Visual Studio pour lancer notre traitement. Il nous faut donc maintenant déployer notre package dans SQL Server.

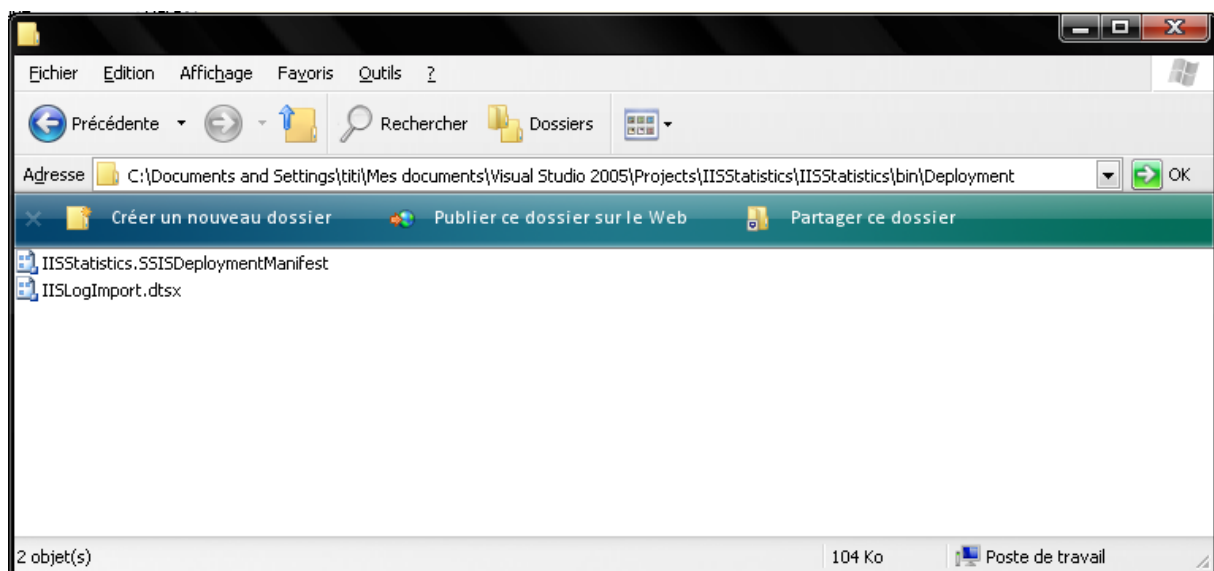
Ne vous faites pas de soucis, c'est encore Visual Studio qui va travailler pour nous et va tout faire tout seul ou presque.

La première étape du déploiement est d'activer la création de l'utilitaire de déploiement du package. Allez dans les propriétés du projet, onglet Deployment Utility :



Vous devez activer l'option CreateDeploymentUtility. Validez et re générez le projet.

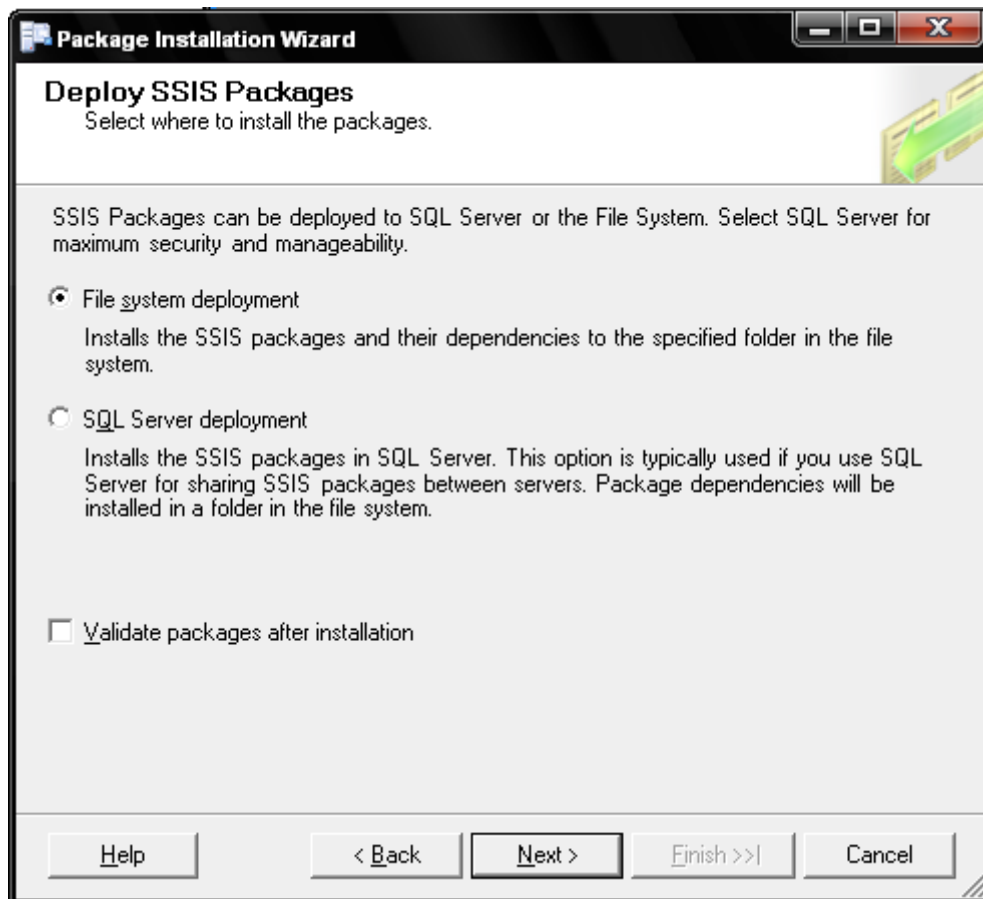
Si vous vous rendez dans le dossier de sortie du projet, vous devriez avoir quelque chose comme ça :



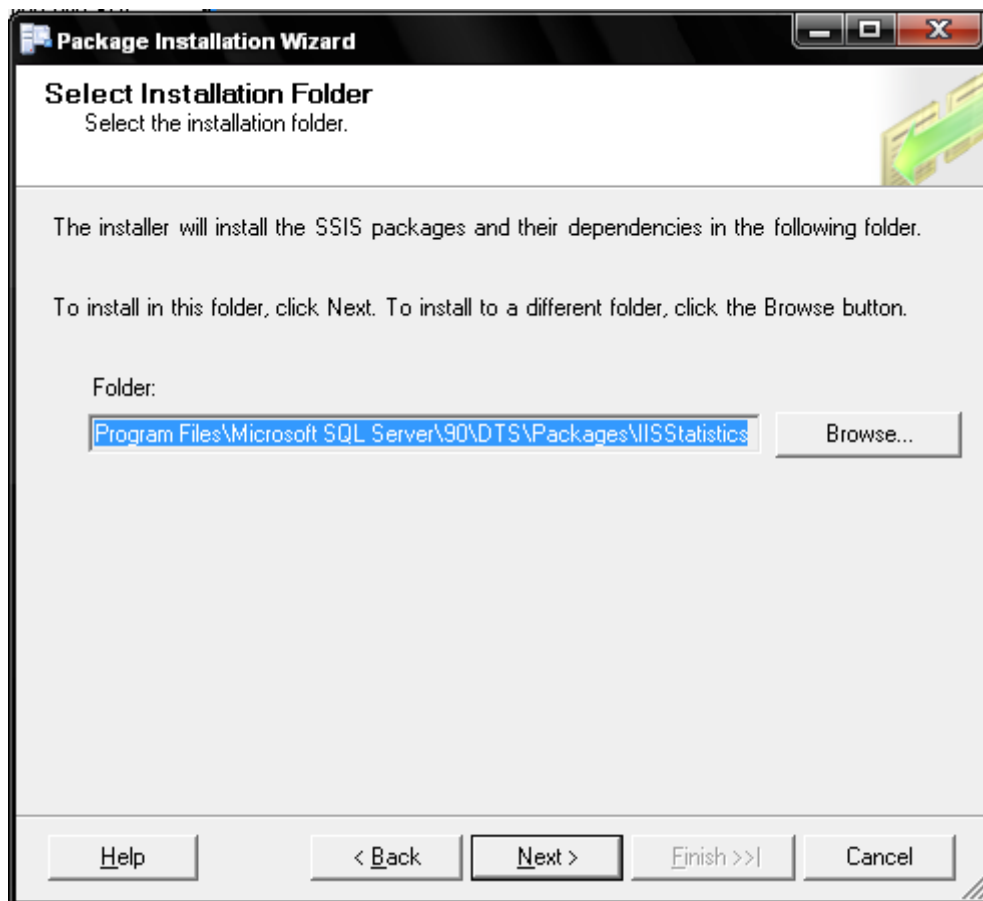
Il y a bien notre package, IISLogImport.dtsx. Le deuxième est le Manifest de déploiement, IISStatistics.SSISDeploymentManifest. Ouvrez-le, un assistant de déploiement se lance.



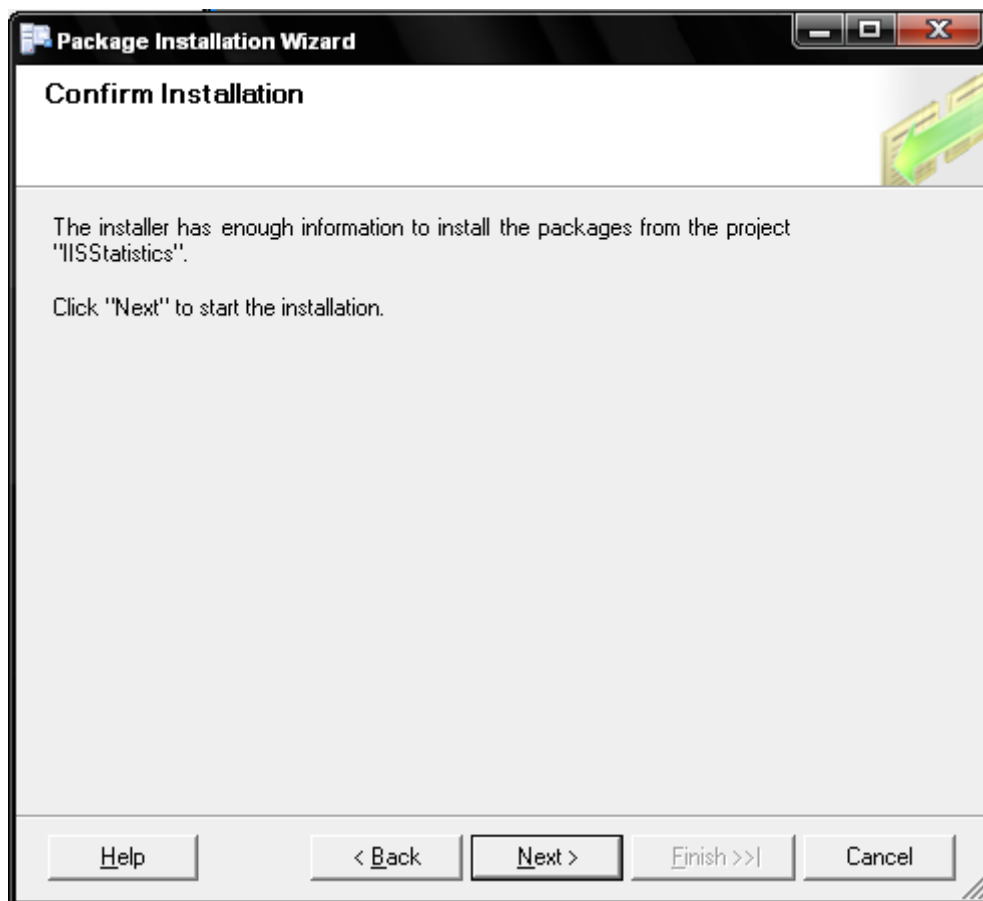
Comme beaucoup d'assistants chez Microsoft, il y a une petite page d'accueil, passons.



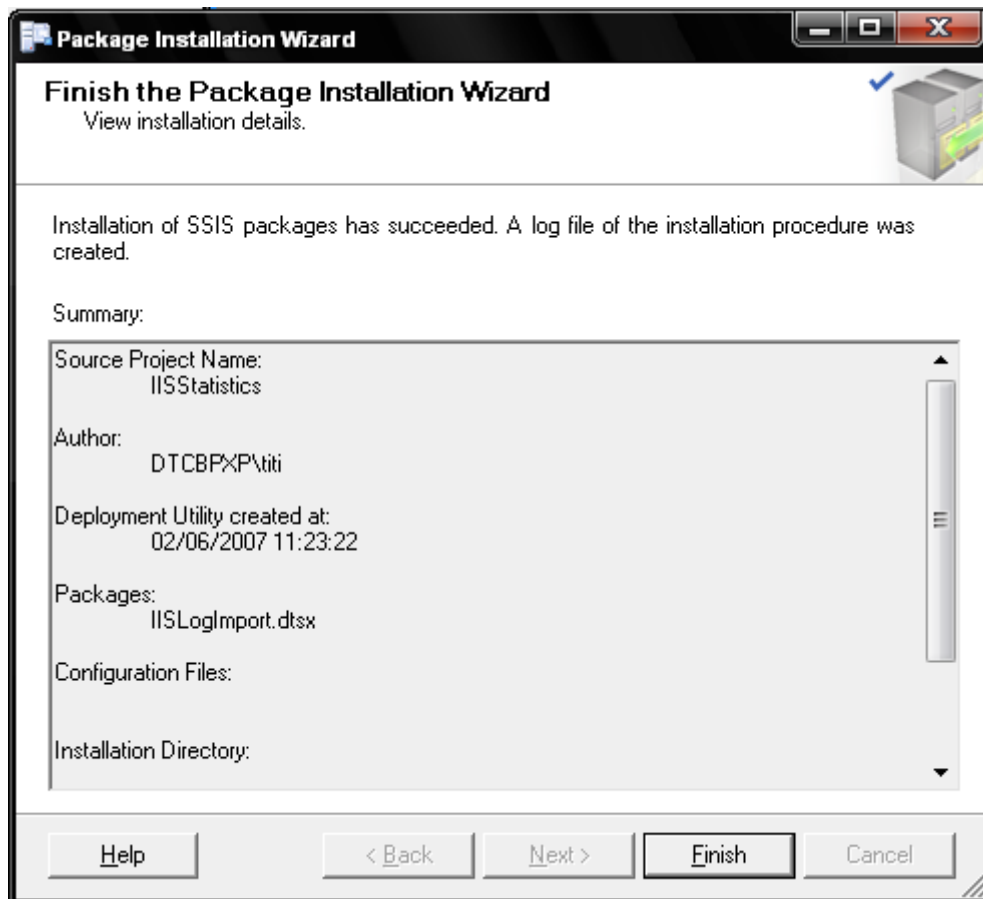
L'assistant nous propose deux mode d'installation pour notre package. Si vous regardez la description de chacun des modes, vous comprendrez que pour notre exemple, le déploiement dans le système de fichier est bien suffisant.



Choisissez l'endroit où sera copié le package. L'emplacement par défaut ira très bien dans notre cas.



Petit confirmation avant l'installation, suivant.

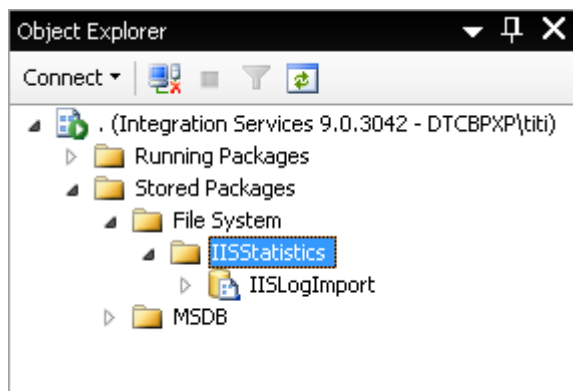


Le déploiement s'est bien passé, nous avons fini. Vérifions maintenant notre travail.

Vérification du déploiement et exécution

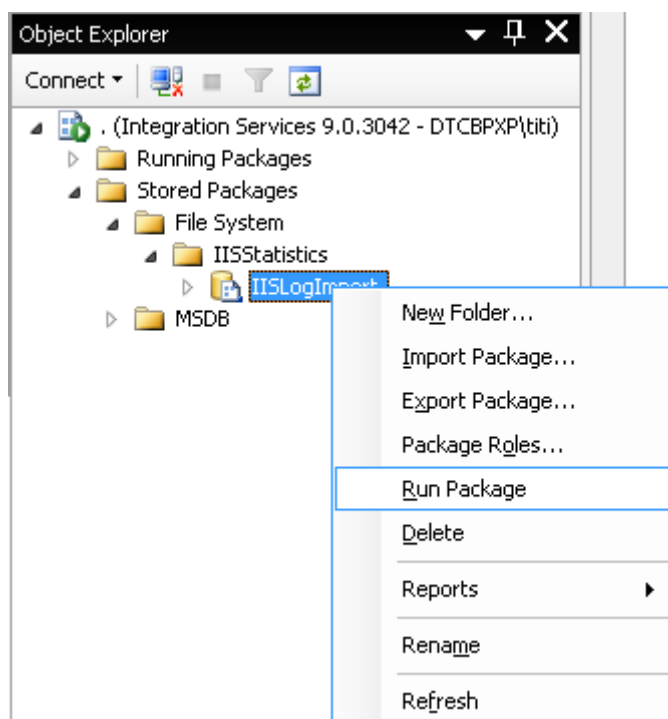
Pour vérifier notre travail, lancez SQL Server Management Studio et connectez vous à Integration Services, pas au serveur de base de données comme vous le faite sans doute d'habitude :



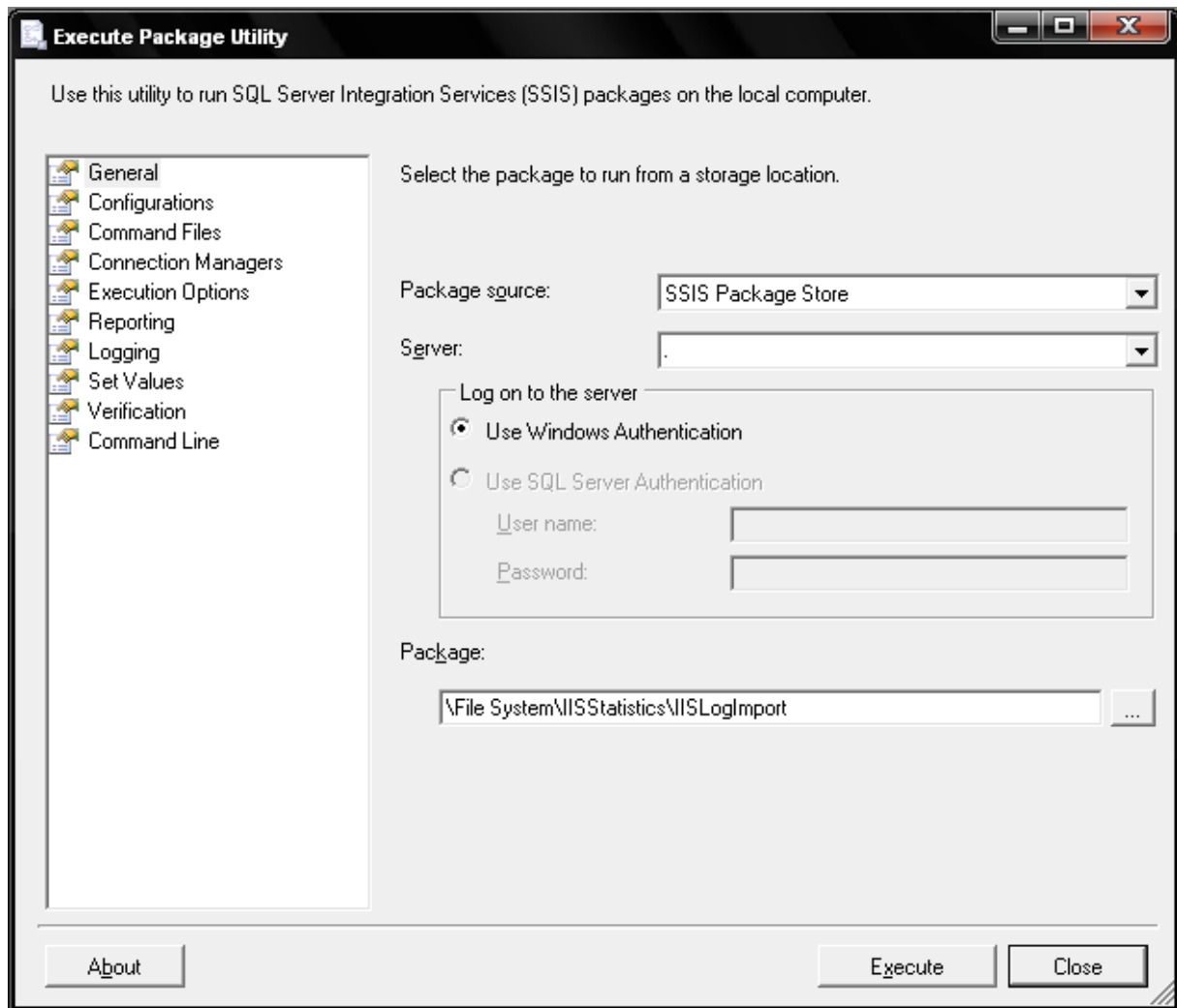


Si vous développez l'arborescence du serveur, vous retrouverez bien notre package.

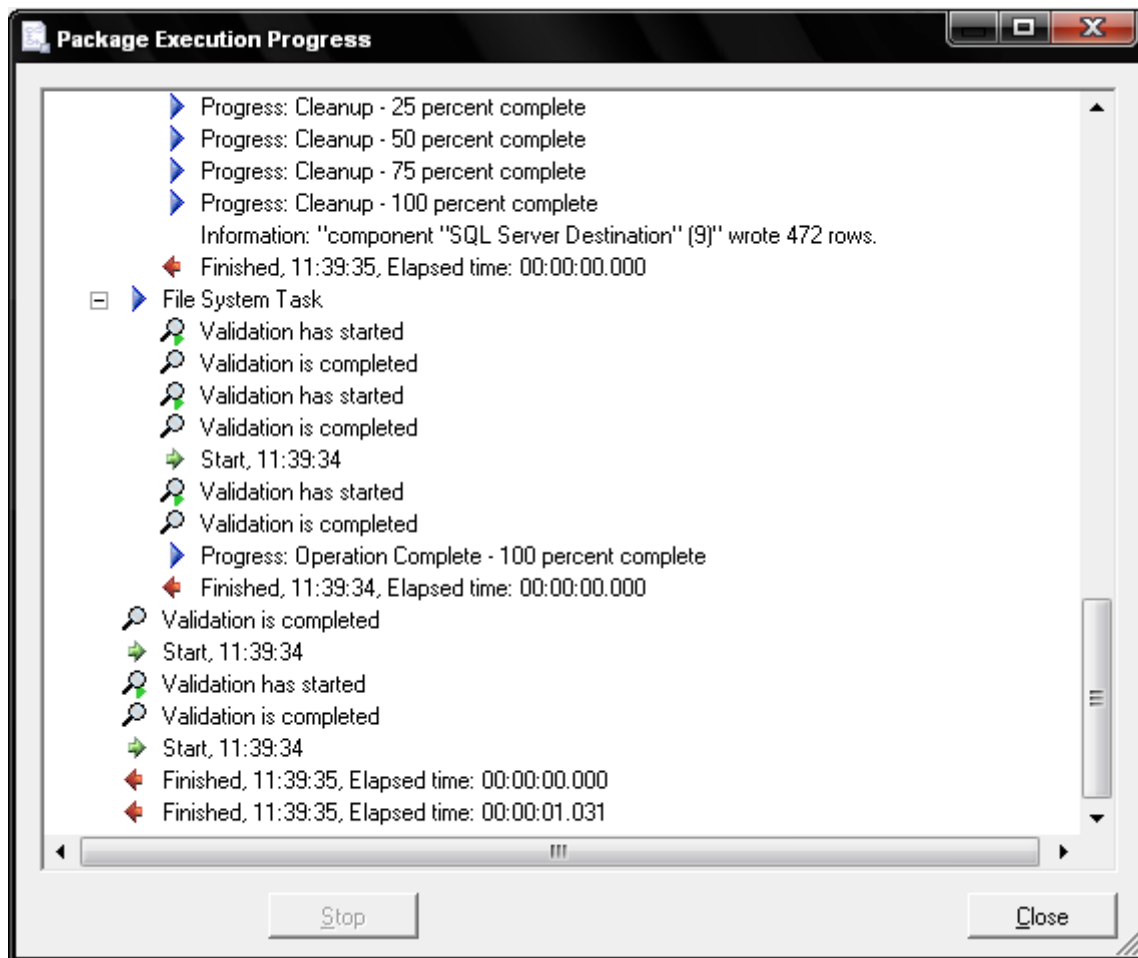
Nous allons maintenant le lancer :



Un petit tour dans le menu contextuel du package, choisissez la commande Run Package.



L'assistant qui s'ouvre nous propose différentes options. Je ne détaillerais rien, les options par défaut sont correctes, nous allons nous contenter de lancer l'exécution du package. La fenêtre suivante s'ouvre et vous pouvez suivre l'exécution du package :



Vous pouvez voir que 472 lignes de log ont été écrites en base. Notre exécution s'est bien déroulée.

Conclusion

Je ne vous ai montré ici que quelques une des nombreuses, je dirais presque infinies, possibilités offertes par SSIS.

Comme vous avez pu le constater, la prise en main de ce puissant outil est relativement aisée et rapide. Les tâches que vous pouvez automatiser peuvent aller de la plus simple à la plus complexe.

Il est même possible d'automatiser se traitement via les autres fonctionnalités de SQL Server, mais je garde ça pour plus tard.

Merci de votre attention