Chapitre 4
Manipulation de données simples

1. Le flux de données

Dans le chapitre précédent, nous avons présenté les principales fonctionnalités du flux de contrôle et l'utilisation des variables et expressions. Pour résumer, nous avons appris que cette partie de notre lot SSIS a pour principal objectif la coordination des différentes tâches qui composent notre package. Il n'a, pour le moment, pas encore été question de gestion de données ; cette notion représente l'objectif de ce chapitre. En effet, il est important de bien comprendre la distinction qu'il existe entre le moteur de gestion des flux de données (et des manipulations associées à ces flux) et l'orchestration des tâches de notre lot (tâches dont fait partie le flux de données, par ailleurs).

Nous allons donc nous intéresser ici à la création de flux de données à savoir : leur implémentation, les objets les définissant et leur mode basique de fonctionnement (une explication plus aboutie des flux de données est présentée dans les deux chapitres suivants, Transformation des données et Flux de données multisource et jointures).

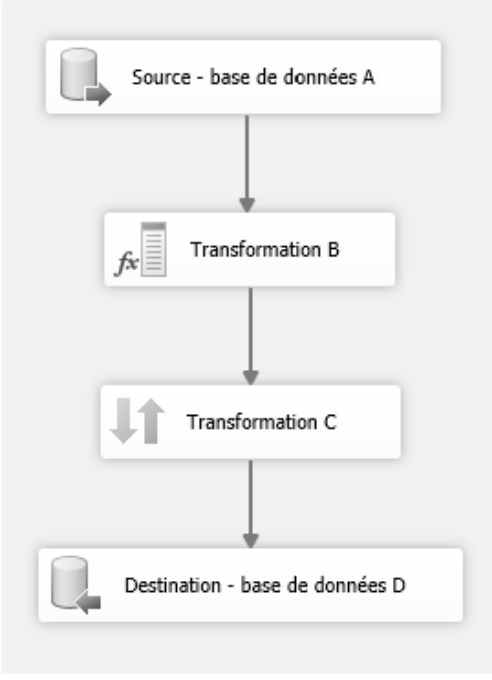
1.1 Principaux objectifs

Si nous ne devions retenir qu'un seul but à l'existence du flux de données, c'est bien sûr celui de la manipulation des informations que nous allons ensuite, par exemple, insérer dans notre entrepôt de données. Mais en allant un peu plus

avant dans la définition, l'objectif est en fait triple : c'est lui qui se chargera d'extraire les données depuis les sources (bases de données, fichiers plats, etc.), pour ensuite les transformer avant de les charger vers la destination de notre choix (ici encore, un Système de gestion de bases de données relationnelles ou SGBDR, un fichier Excel, des structures de données propres à SSIS, etc.). Cette phrase résume à elle seule l'ensemble des types de composants qui sont mis à notre disposition pour concevoir notre flux de données : des composants de source, de transformation et de destination.

1.2 Aperçu

Avant de rentrer dans le vif du sujet, voici un exemple de flux de données permettant d'extraire des informations depuis une table de la base de données A, de réaliser quelques manipulations B et C sur ces données et d'insérer le tout dans une table de la base de données D (schéma ci-dessous). Nous rencontrons donc bien ici l'ensemble des types d'objets disponibles dans un flux de données.

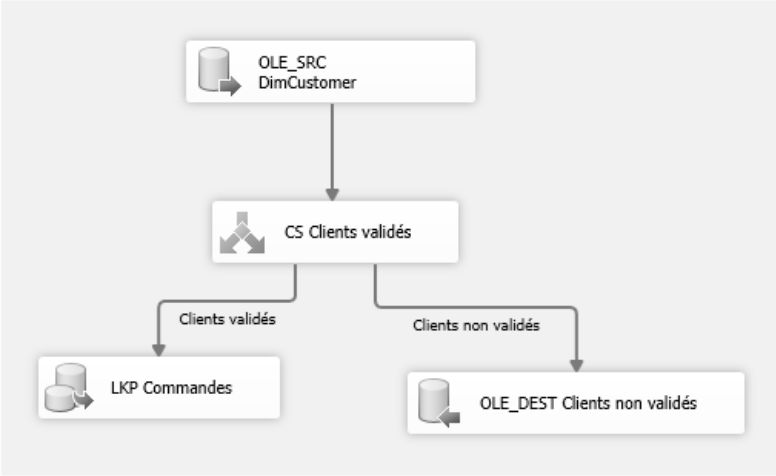


Nous pouvons remarquer des similitudes, du point de vue de leur forme générale, entre le flux de contrôle du chapitre précédent et le flux de données. Dans les deux cas, nous avons des objets (on parle de tâches pour le premier et de composants pour le second) reliés par des flèches. La distinction va porter sur le sens que l'on donne à ces flèches justement (afin d'accentuer cette différence, depuis SSIS 2012, les flèches du flux de contrôle sont vertes et celles du flux de données sont bleues). Dans le cas du flux de contrôle, et nous l'avons vu dans le chapitre précédent, la flèche permet de définir le cheminement d'exécution du lot SSIS (ou WorkFlow) avec les contraintes de précédence associées. Pour le flux de données, bien entendu, cela va permettre d'ordonnancer les composants, mais il faut également voir cette flèche comme un flux de données entre ces différents composants (en anglais, nous parlons de pipeline, ce qui permet de bien différencier les rôles de ces flèches entre le flux de contrôle et celui de données). Par exemple, il faut voir la première flèche de l'image ci-dessus comme un « tuyau » par lequel les données de la base de données A vont transiter pour être consommées par les composants B et C qui se chargeront de leur appliquer un certain nombre de transformations. Une fois que le composant C a réalisé la totalité des actions qui lui incombent, les données vont emprunter le « tuyau » reliant la sortie de ce composant avec l'entrée du composant permettant d'écrire dans la base de données D (ces notions d'entrée-sortie sont importantes pour comprendre et s'approprier le mode de fonctionnement du flux de données). Il y a donc bien des données qui transitent sur ces flèches.

Avant de revenir plus précisément sur la manière dont les données transitent (dans le chapitre Bonnes pratiques et notions avancées), il est tout de même utile de fixer certaines choses afin de bien comprendre les mécanismes en jeu. Tout d'abord, nous l'avons déjà présenté dans le chapitre d'introduction, SSIS fonctionne en mémoire. Quand on dit cela, c'est surtout valable pour le flux de données et les opérations de transformation qu'il réalise. Quand une donnée transite dans un pipeline, elle est en fait stockée dans la RAM du serveur. La zone mémoire occupée par cette donnée est ce que l'on appelle un buffer. Il existe plusieurs scénarios de gestion de ces buffers en fonction du type de composant que l'on utilise dans le flux de données : ils sont présentés dans le chapitre Bonnes pratiques et notions avancées. C'est une notion importante à comprendre, notamment lorsque l'on doit définir une architecture matérielle pour la mise en place d'un processus ETL dans un SID (Système d'information décisionnel).

1.3 Entrée-sortie et flux de données

Cette notion d'entrée-sortie (ou de flux, ou de pipeline) est l'une des principales différences entre les tâches du flux de contrôle et les composants du flux de données. Dans le cas des composants de type transformation, ces derniers possèdent de ce fait une entrée et une sortie. En revanche, les composants de type source ne possèdent pas d'entrée au sens SSIS car ils sont alimentés d'une autre manière et réciproquement pour les composants de type destination qui ne possèdent pas de sortie (excepté une sortie d'erreur dont nous allons parler un peu plus tard). Les entrées et les sorties sont nommées, par défaut, par le lot SSIS, mais nous avons la possibilité de changer leur nom (et c'est même indispensable dans certains cas). C'est un point important car avoir des flux rigoureusement nommés permet de tracer plus facilement l'information en cas d'erreur et accroît fortement la lisibilité du flux de données pendant la phase de développement. Certains objets possèdent même plusieurs sorties (composant de multidiffusion par exemple) ou peuvent recevoir des données depuis plusieurs entrées (les composants de jointure, entre autres). Il est dans ce cas primordial de nommer les flux de manière à pouvoir identifier la nature de la donnée qui y transite. Car lorsque nous allons devoir brancher les composants entre eux, lorsque nous avons plusieurs sorties possibles, il va bien falloir savoir laquelle choisir. C'est le cas avec l'exemple de flux de données ci-dessous. Il implémente un composant de fractionnement conditionnel (son fonctionnement est détaillé dans le chapitre suivant) qui propose deux sorties : Clients validés et Clients non validés. Un traitement différent est réalisé sur les lignes qui appartiennent à l'un ou l'autre de ces deux flux.

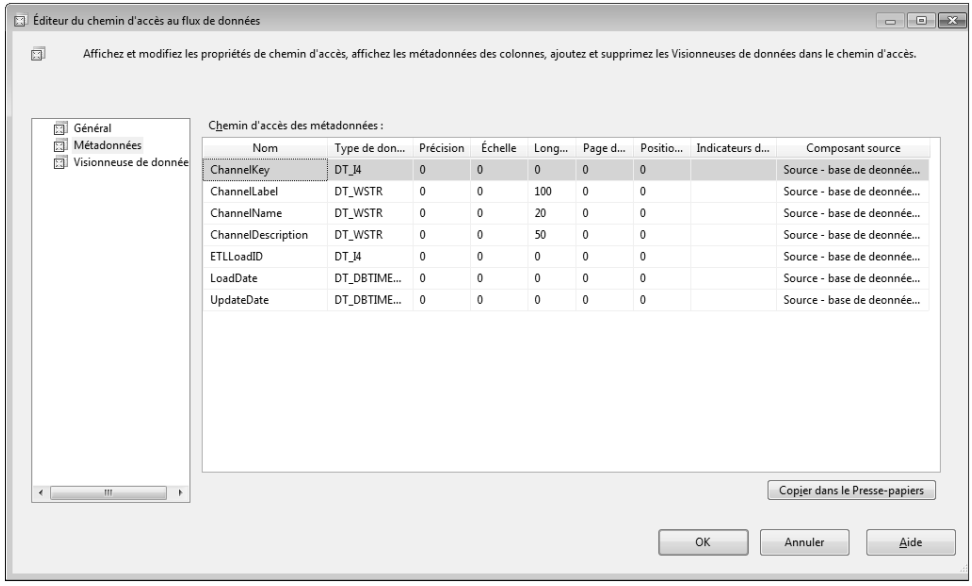


Si nous reprenons notre premier exemple de flux de données, voici comment nous pouvons interpréter le cheminement de l'exécution de ce flux de manière simplifiée : sur son entrée, la transformation B va trouver les données issues de la sortie du composant précédent A. Une fois les transformations apportées par B, les données sont envoyées sur sa sortie qui correspond à l'entrée de la transformation suivante C, et ainsi de suite jusqu'au composant de destination qui se charge d'écrire l'information dans un référentiel externe. C'est ce qui va constituer notre flux de données : ce que l'on désignera par « path » en anglais. Sur ces flux transitent les fameux buffers que l'on vient de définir. Analogues à un dataset, ils sont constitués de lignes et de colonnes.

Toujours dans le cas de notre flux de données, le buffer mémoire en sortie du composant Source - base de données A correspond aux données extraites de notre base. C'est donc une structure de données à part entière et, comme beaucoup de structures de données, elle possède des métadonnées comme, par exemple, un type pour chaque colonne du buffer ou encore un identifiant. Ce dernier, que l'on nomme LineageID, permet à SSIS de rendre uniques les colonnes d'un buffer. Toute colonne possède donc au moins un LineageID. Cependant, elles peuvent en avoir plusieurs lorsque, pour des raisons particulières, SSIS est obligé de faire de la copie de buffer. Cette copie de buffer est induite par certains composants et nous reviendrons en détail sur ces types de composants dans le chapitre Notions avancées et bonnes pratiques. Le LineageID est parfois le seul moyen à notre disposition pour réaliser

la correspondance entre deux colonnes (pour le composant d'importation de colonnes, par exemple).

La structure des buffers (que l'on pourra nommer jeux de données ou encore datasets) est visible en double cliquant sur l'une des flèches.



Nous voyons ici l'ensemble des colonnes qui composent le buffer : **ChannelKey**, **ChannelLabel**, **ChannelName**, etc. Nous retrouvons également le type de ces colonnes : ces types de données sont propres à SSIS mais trouvent leur correspondance dans les types de destination que nous utilisons. Sur cette même fenêtre, dans l'onglet **Général**, nous retrouvons une information sur notre pipeline dont nous avons déjà parlé : son nom. Mais un flux possède d'autres propriétés, comme un identifiant (ID), un nom de source (SourceName) et de destination (DestinationName) qui identifient le point d'entrée et la sortie de ce flux.

Nous venons de visualiser les colonnes de notre buffer, mais qu'en est-il des lignes ? Celles-ci ne seront visibles qu'en mode débogage via la **visionneuse de données** également visible sur cette fenêtre. Nous reviendrons sur l'utilisation de cet outil dans le chapitre Événements et suivi d'exécution.