

ETL (Integration Services) w systemie Microsoft SQL Server lub innych

Instalacja środowiska

- Instalacja instancji SQL Server

Podczas instalacji w zakładce "Feature Selection" trzeba zaznaczyć opcję "Integration Services"

- Instalacja Visual Studio 2019(SQL Server Data Tools nie jest jeszcze wspierany przez nowsze wersje)

[Download Visual Studio Tools - Install Free for Windows, Mac, Linux \(microsoft.com\)](#)

- Instalacja SQL Server Data Tools w Visual Studio
- Instalacja pluginu(VS->Extensions->Manage extensions) "SQL Server Integration Services Projects"
- Instalacja SQL Server Management Studio 19(SSMS)

[Download SQL Server Management Studio \(SSMS\) - SQL Server Management Studio \(SSMS\) | Microsoft Learn](#)

Bardziej szczegółowa instrukcja - [Install SQL Server Integration Services - SQL Server Integration Services \(SSIS\) | Microsoft Learn](#)

- Trzeba rozpakować zawartość archiwum "DBProject.rar" na dysku C:/.
- Trzeba odtworzyć bazę danych Northwind z pliku Northwind.bak w rozpakowanym folderze.

WSTĘP

Microsoft SQL Server Integration Services (SSIS) is a platform for building high-performance data integration solutions, including extraction, transformation, and load (ETL) packages for data warehousing

Pakiet ETL - podstawowa niezależna jednostka integracji danych.

Integration Services składa się z:

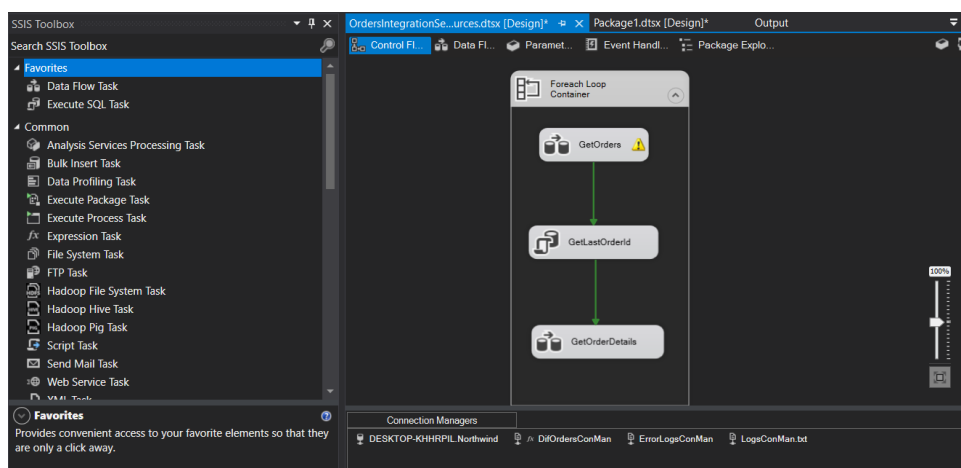
- dużej liczby wbudowanych zadań integracji(Control flow) oraz transformacji danych(Data flow)
- Graficznego narzędzia SSIS designer do budowania pakietów integracji.

Jest dostępny w Visual Studio w ramach Integration Services Project (powinien być zainstalowany moduł SQL Server Data Tools).

Projekt integracji składa się z dwóch głównych części:

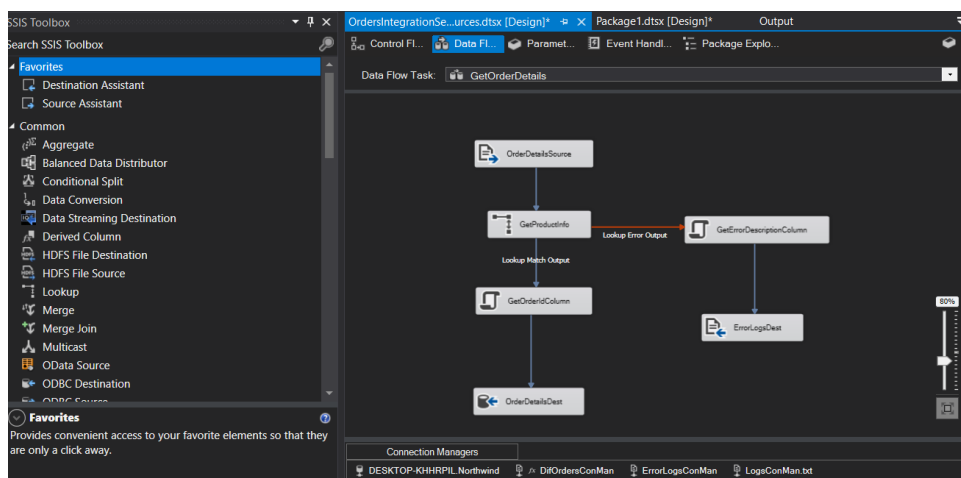
- Przepływ sterowania (Control Flow)

Jest tworzony na paneli Control Flow Tab. Może zawierać różne zadania (w SSIS Toolbox), oprócz bezpośrednio zadania ETL - "Data Flow Task". Np. po zakończeniu przenoszenia danych można wykonać backup bazy danych za pomocą "Back Up Database Task", albo wysłać mail z logami za pomocą "Send Mail Task".



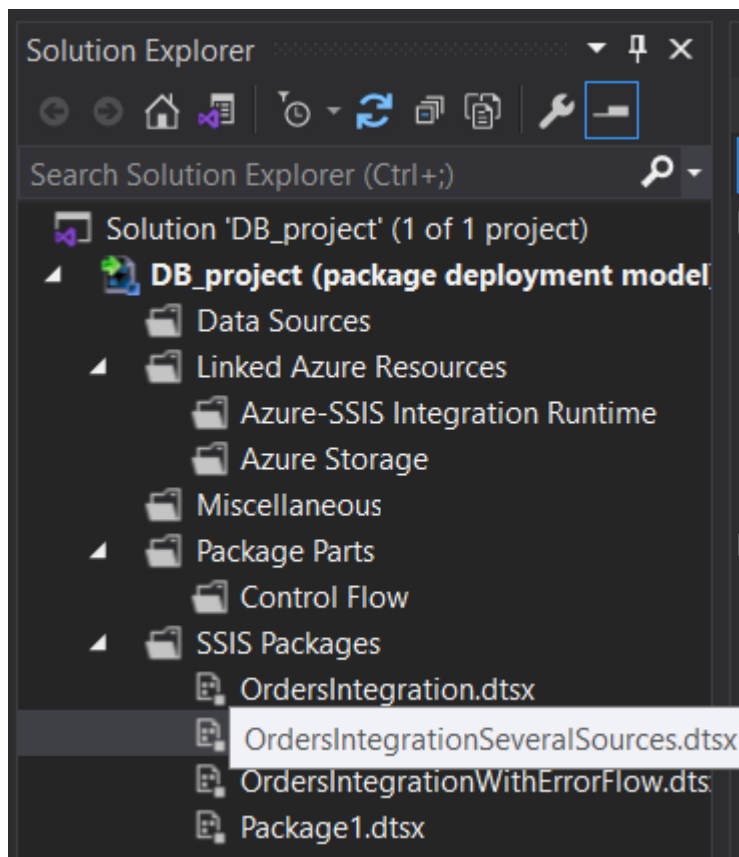
- Przepływ Danych (Data flow)

Jest tworzony w DataFlow Task. Powinien zawierać przynajmniej jedno źródło danych (SSIS Toolbox -> Other Sources) oraz wyjście (SSIS Toolbox -> Other Destinations), czyli gdzie przenosimy te dane. Pomiędzy tymi elementami za pomocą różnych transformacji (SSIS Toolbox -> Common, SSIS Toolbox -> Other Transforms) można zmieniać dane wejściowe.

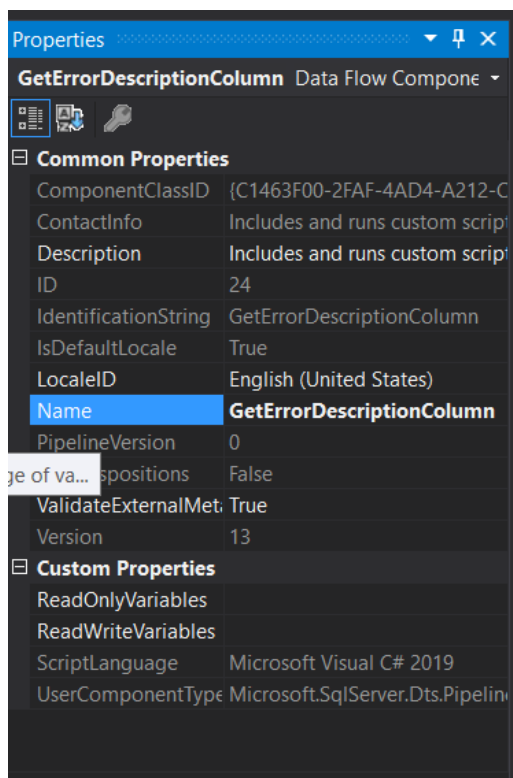


Inne elementy SSIS Designer:

- Solution Explorer – pokazuje strukturę projektu



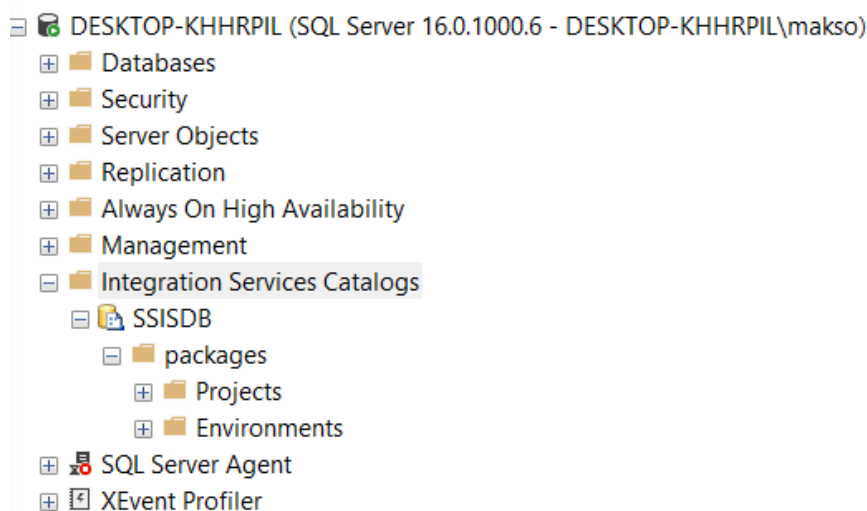
- Properties – pokazuje wszystkie dostępne atrybuty wybranego komponentu



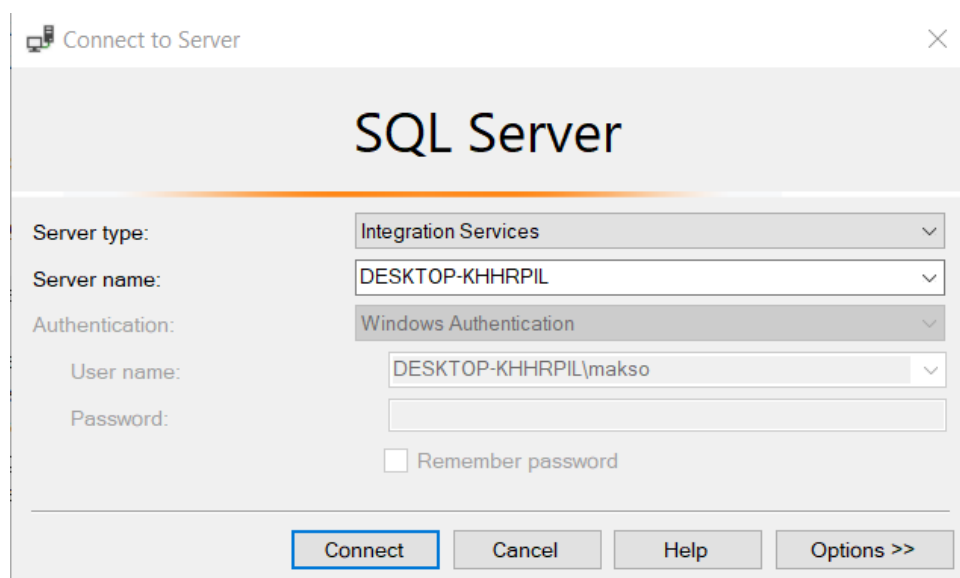
- SQL Server Integration Services Database Catalog(SSISDB)

Katalog w SQL Serwerze który pozwala na przechowanie, zarządzanie oraz uruchamianie pakietów integracji.

Utwórzmy taki katalog w naszej instancji sql serwera za pomocą SSMS.



Następnie trzeba się połączyć z tym katalogiem tak samo, jak z sql serwerem, ale typ serwera – Integration Services.



UTWORZENIE PAKIETU INTEGRACJI

Zobaczmy jak to działa na przykładzie.

Z pliku IslandTradingOrders.txt chcemy przenieść dane o złożonych zamówieniach do tabel "Orders" oraz "Order Details" do instancji domyślnej SQL serwera("DESKTOP-KHHRPIL" w moim przypadku) do bazy "Northwind".

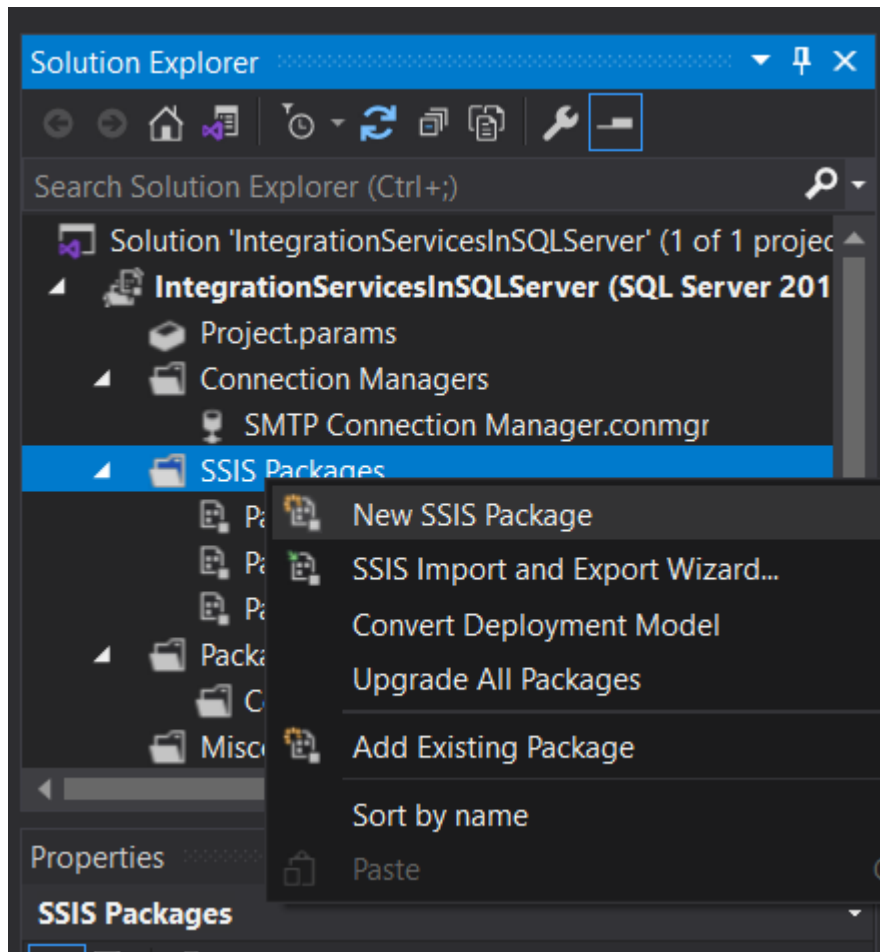
Krok 1: Utwórzmy nowy projekt w Visual Studio

Open Visual Studio -> create a new project -> wybieramy wzorzec "Integration Services Project". Nazwijmy go "DB_project".

Krok 2: Utwórzmy nowy pakiet integracji

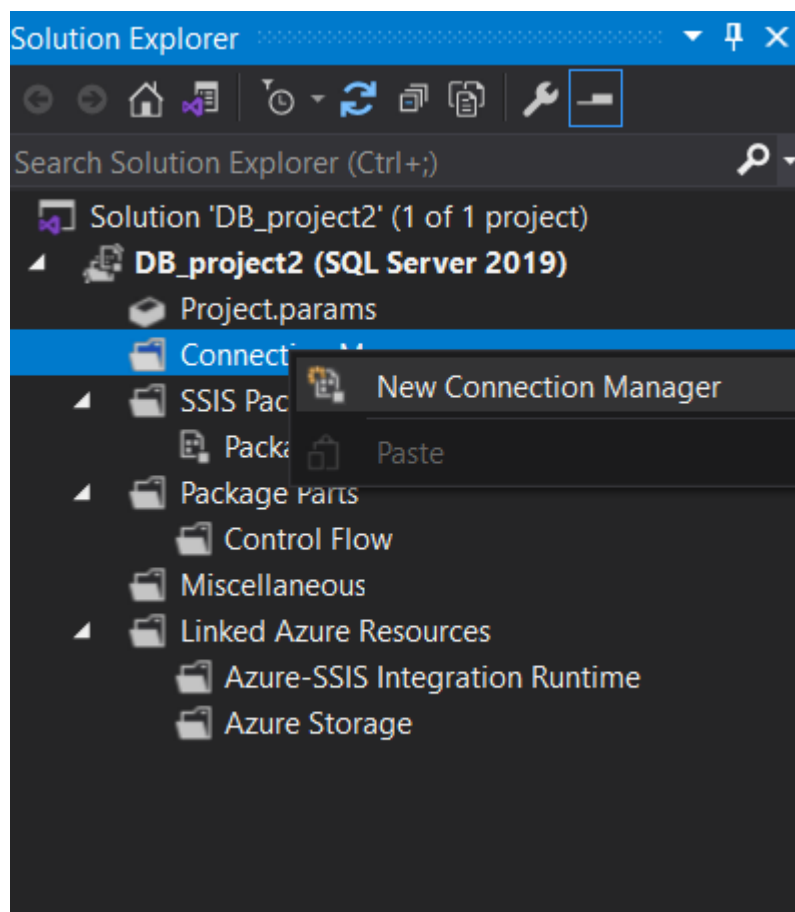
Solution Explorer->SSIS Packages->new SSIS Package.

Nazwijmy go "OrdersIntegration".



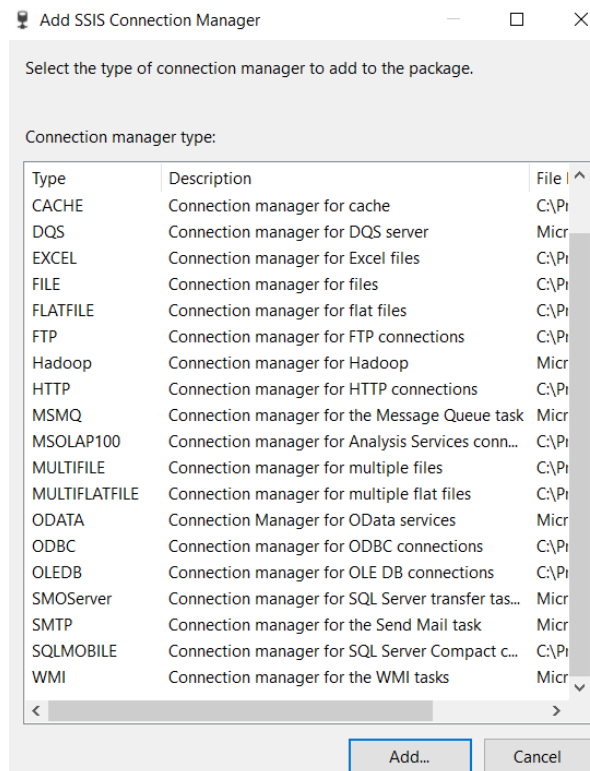
Krok 2: Stwórzmy połączenie do docelowej bazy danych oraz do pliku z zamówieniami.

Do tego potrzebujemy Connection Manager.



Connection manager – logiczna reprezentacja połączenia do różnych źródeł danych(bazy relacyjne, pliki, serwer FTP itd.).

Może być widoczny albo na poziomie całego projektu(domyślnie), albo dla wybranego pakietu.



Dostępne rodzaje połączeń

Oprócz tego można napisać własny Connection manager.

Stwórzmy OLE DB manager do bazy Northwind w lokalnej instancji.

Solution Explorer->Connection Managers->new Connection Manager->wybierz "OLE DB"->Add

->New->wybieramy serwer SQL->wybieramy bazę danych "Northwind".

OLE DB (*Object Linking and Embedding, Database*, sometimes written as **OLEDB** or **OLE-DB**), an [API](#) designed by [Microsoft](#), allows accessing [data](#) from a variety of sources in a uniform manner.

Analogicznie stwórzmy też "Flat File Connection Manager" dla naszego pliku "IslandTradingOrders.txt". W "Flat File Connection Manager Editor"->General podajemy ścieżkę "C:\DBProject\IslandTradingOrders.txt". Nazwijmy go "IslandTradingOrdersConMan". Powinno być zaznaczone "Column names in the first data row". Nic więcej nie zmieniamy.

W zakładce "Columns" upewniamy się, że jako "Column delimiter" jest przecinek a jako "Row delimiter" - CRLF.

W zakładce "Advanced" musimy odpowiednio dopasować typy kolumn:

ShipperCompany,ProductName -> DT_WSTR(koduje w Unicode, a domyślny DT_STR - w ANSI)

OrderDate -> DT_DBDATE

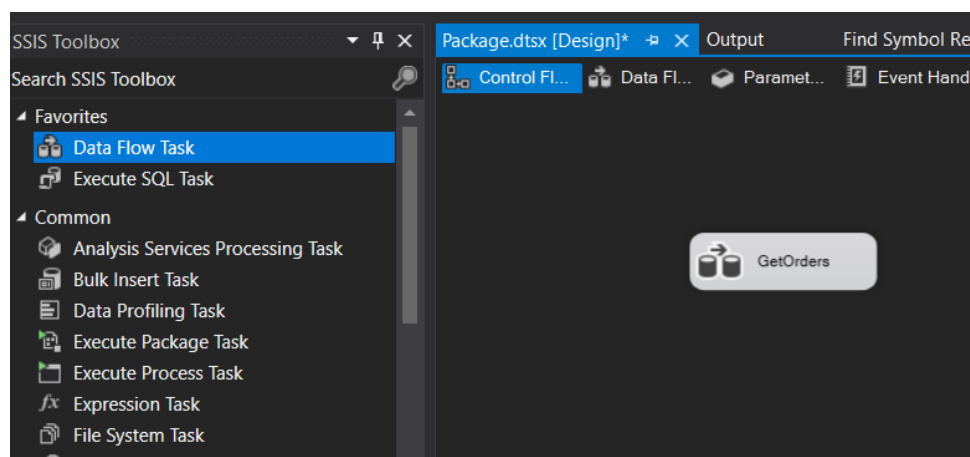
EmployeeId, Quantity -> DT_I4

Discount -> DT_R4

Następnie dodamy do naszego Control Flow “Data Flow Task”, który właśnie odpowiada za proces integracji danych ETL – extract, transformation and load.

Tak jak tabela “Order Details” powiązana kluczem obcym z tabelą Orders, to trzeba najpierw wstawić dane do Orders. Dane są commitowane do bazy dopiero po zakończeniu Data flow tasku, dlatego musimy utworzyć osobne taski na Orders oraz “Order Details”. Na razie zajmijmy się tylko pierwszym z nich.

Przeciągamy ten task z SSIS Toolbox do naszego Control Flow. W ramach jednego pakietu może być dowolnie dużo takich tasków. Nazwijmy go “GetOrders”.



Wejdźmy do tego tasku.

Dodajmy źródło dla naszego pliku - “Flat file source”(SSIS Toolbox->Other Sources->Flat File Source). Nazwijmy go “IslandTradingOrdersSource”.

W edytorze ustawiamy flat file connection manager na “IslandTradingOrdersConMan”. W zakładce Column wybieramy tylko kolumny potrzebne dla tabeli Orders – ShipperCompany, OrderDate, EmployeeId.

W naszym pliku są zamówienia jednego klienta “Island Trading”. Musimy dodać do naszych danych kolumnę CustomerID dla tego przewoźnika. Do tych celów jest transformacja “Derived column”, która dodaje jako kolumnę wartość wyrażenia, które się składa ze zmiennych oraz funkcji wbudowanych. Stwórzmy zmienną IslandTradingCompanyID.

ZMIENNE W SSIS

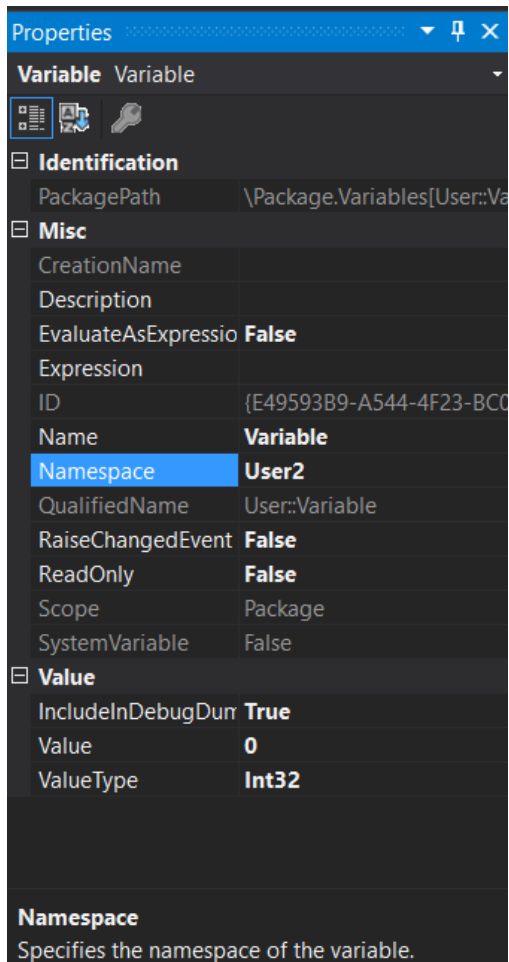
Zmienne w SSIS mogą być wykorzystywane przez pakiety, ich kontenery, “event handlers”, własne skrypty. Zmienne są dwóch typów:

- Systemowe(np. LocaleID, UserName, CreationDate itd.). Należy do przestrzeni nazw “System”.

Nie mogą być ani modyfikowane ani tworzone nowe.

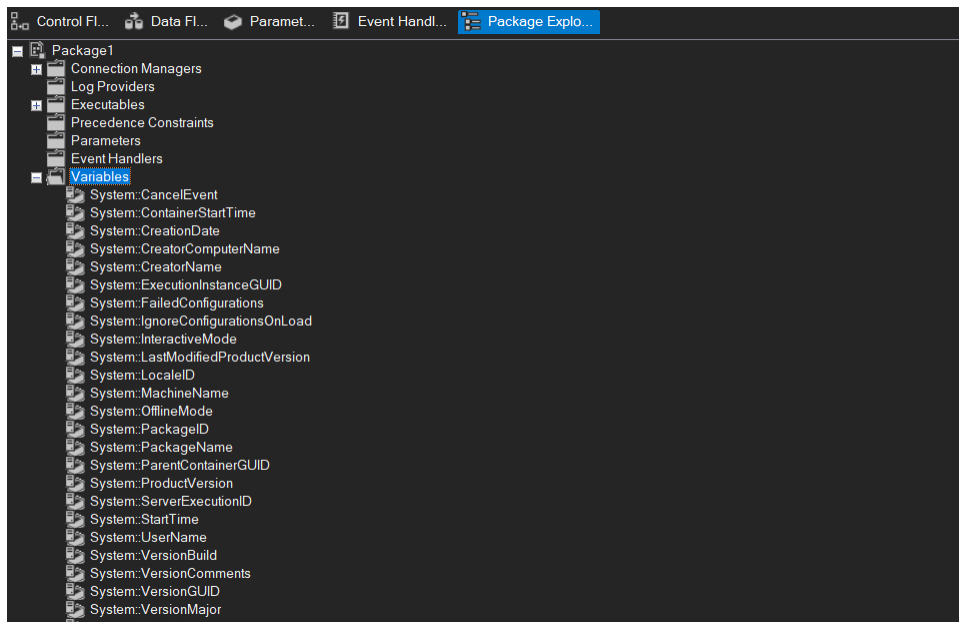
- Użytkownika. Domyślnie należą do przestrzeni nazw User.

W atrybutach zmiennych można ustawić też przestrzeń nazw, do której należy, podać wartość statyczną zmiennej, albo wyrażenie, na podstawie którego wartość będzie liczona na bieżąco (wtedy trzeba ustawić opcję "Evaluate as expression" na True).

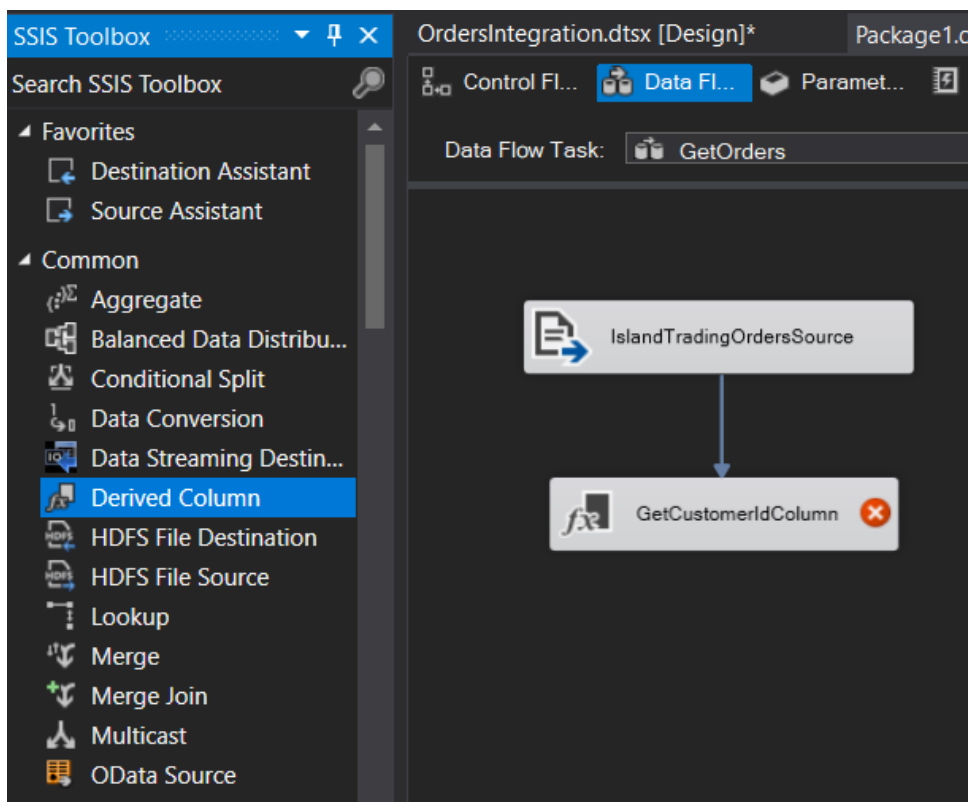


Żeby utworzyć zmienną trzeba kliknąć prawym przyciskiem w dowolnym miejscu na Control flow i wybrać Variables. Dodajemy nową zmienną, typ ustawiamy na String a wartość na "ISLAT" (zakładamy, że id nie będzie się często zmieniało).

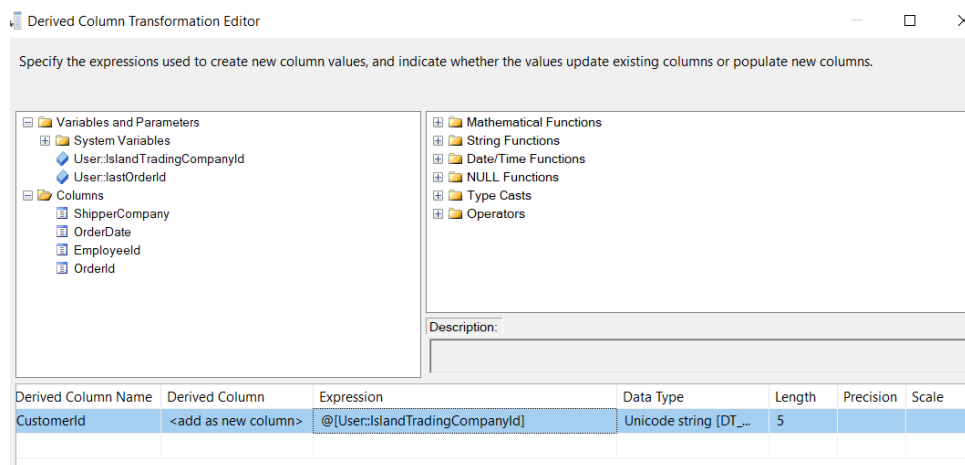
Przeglądać zmienne można w zakładce "Package explorer"



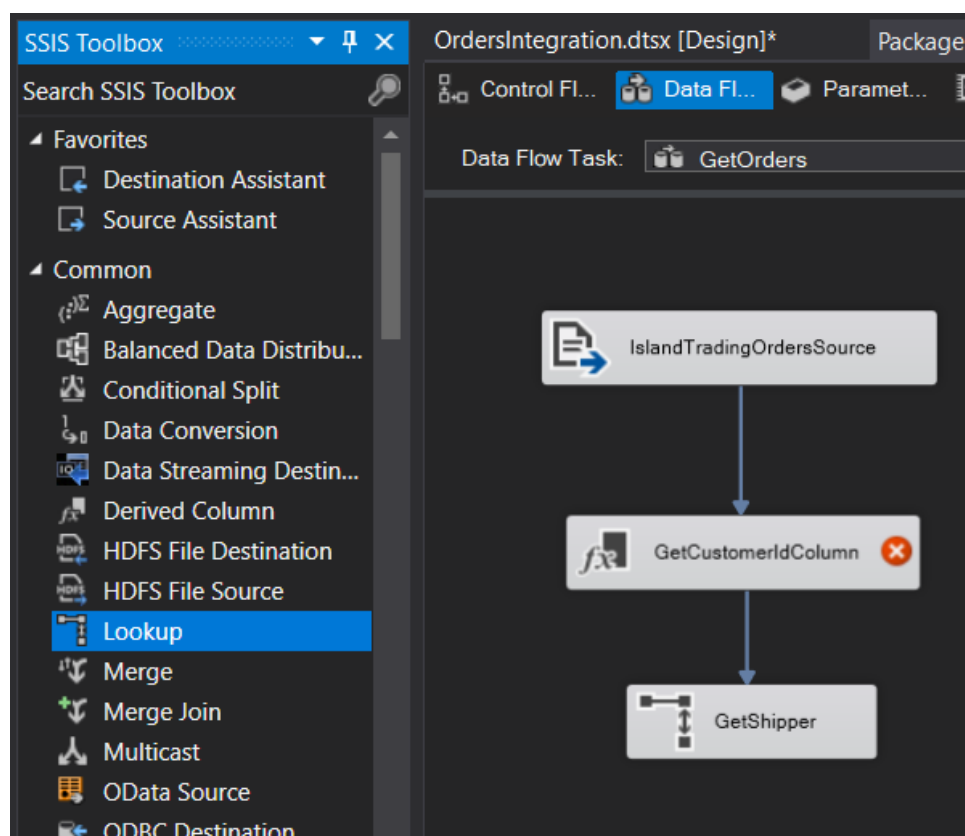
Dodajemy "Derived column" i łączymy niebieską strzałką z "IslandTradingOrdersSource" (niebieską strzałka - normalny przepływ danych, czerwona - "error output"). Nazwijmy ją "GetCustomerIdColumn".



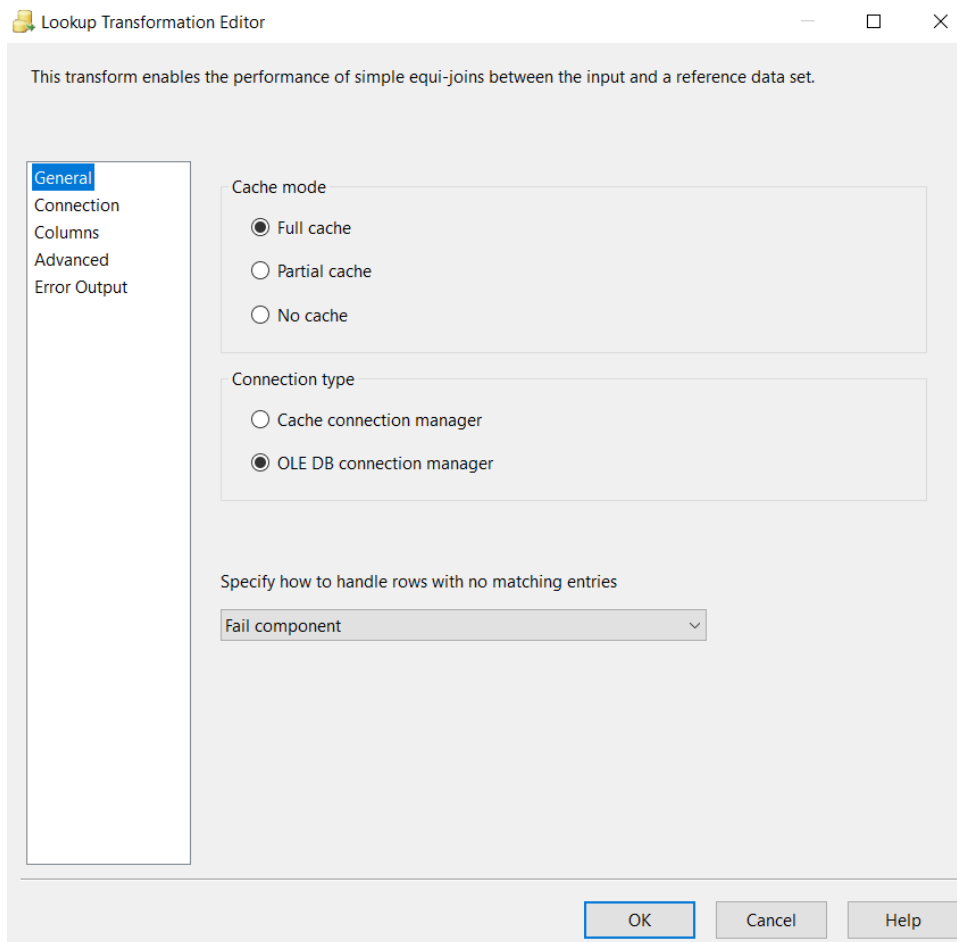
W edytorze przeciągamy "IslandTradingCompanyID" do pola expression. W "Derived column name" wpisujemy CustomerId.



Dalej potrzebujemy dobrać kolumnę “ShipperID” z tabeli Shippers na podstawie kolumny ShipperCompany w naszym pliku. Do tych celów służy transformacja “Lookup”(SSIS Toolbox ->Common). Nazwijmy ją “GetShipper”.



W edytorze w zakładce “General” zostawiamy Full cache w cach mode oraz OLE DB Connection Manager.



Cache mode:

Full cache – dane które przeszukujemy(w naszym przypadku tabela “Shippers”) są całkowicie cachowane przed wykonaniem lookup. Partial cache – cachowane są tylko dopasowane dane w trakcie wykonania lookup, żeby nie szukać wartości w bazie danych dla tych samych kluczy.

Cache connection manager:

Jeżeli wiadomo, że te same dane były przeszukiwane wcześniej, to można je zcachować za pomocą bloku “Cache transform”(SSIS Toolbox->Other Transforms) do pliku zdefiniowanym w Cache Connection Manager. Żeby potem wykorzystać ten manager podczas wykonania operacji lookup.

W zakładce “Connection” wybieramy połączenie z bazą Northwind, a w “Use table or view” - tabelę Shippers.

W zakładce “Columns” łączymy kolumny na podstawie których będzie robiony join oraz zaznaczamy potrzebne nam kolumny(Lookup Column).

General
Connection
Columns
Advanced
Error Output

Available Input Colu...

| Name |
|----------------|
| ShipperCompany |
| OrderDate |
| EmployeeId |
| OrderId |

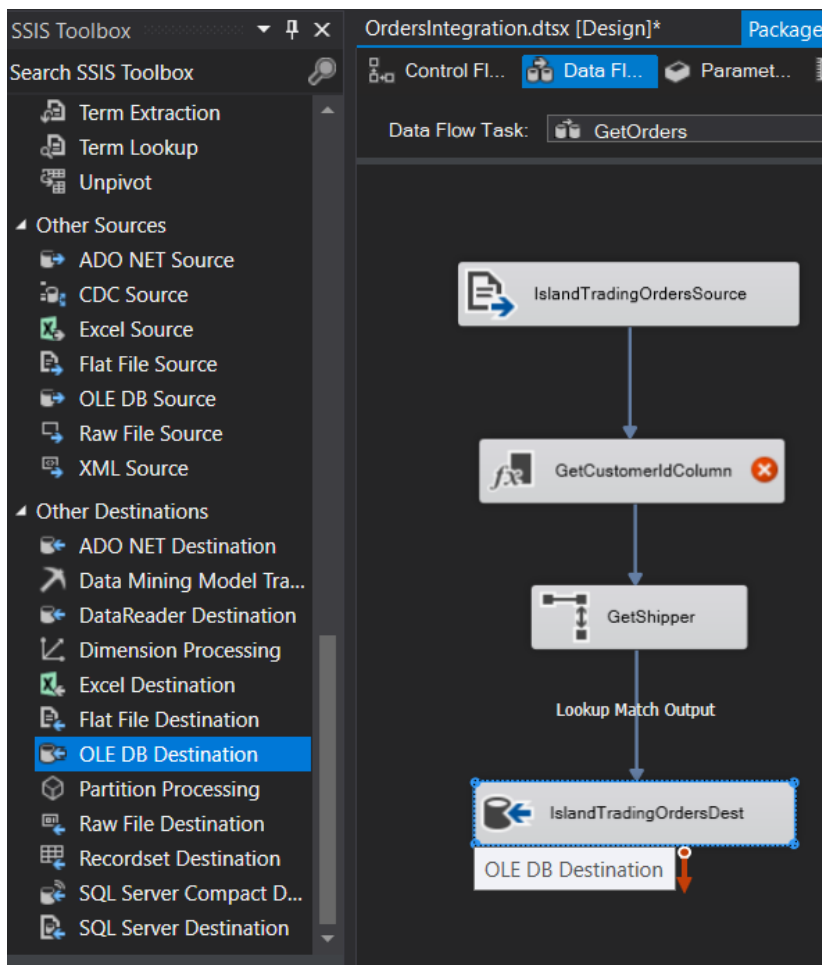
Available Lookup Columns

| <input type="checkbox"/> | Name | I... |
|-------------------------------------|-------------|------|
| <input checked="" type="checkbox"/> | ShipperID | |
| <input type="checkbox"/> | CompanyName | |
| <input type="checkbox"/> | Phone | |

| Lookup Column | Lookup Operation | Output Alias |
|---------------|---------------------|--------------|
| ShipperID | <add as new column> | ShipperID |

Destination

Dalej dodajmy OLE DB destination(SSIS Toolbox->Other Destinations), który właśnie wykonuje insert. Łączymy go z "GetShipper" i wybieramy opcję "Lookup match output". Możemy też uzyskać dane, które nie zostały dopasowane za pomocą drugiego wyjścia - "Lookup no match output".



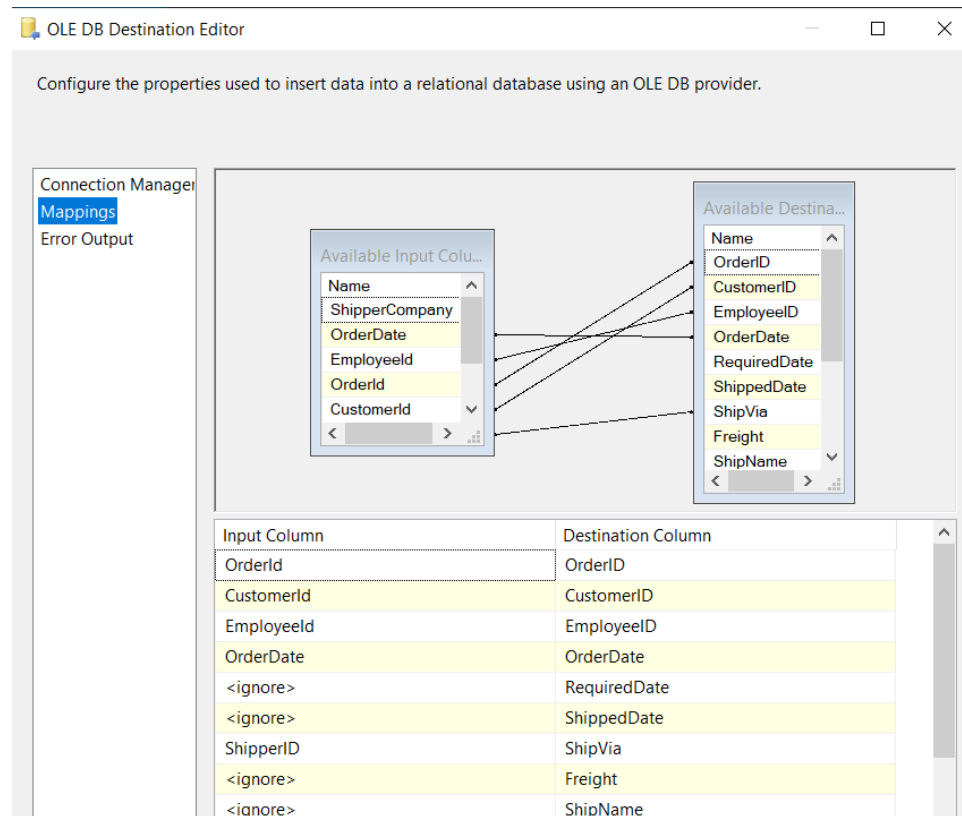
W edytorze w zakładce Connection manager ustawmy tak, jak na obrazku.

The screenshot shows the 'OLE DB Destination Editor' dialog box. The 'Connection Manager' tab is selected. The 'OLE DB connection manager' is set to 'DESKTOP-KHHRPIL.Northwind'. The 'Data access mode' is set to 'Table or view - fast load'. The 'Name of the table or the view' is set to '[dbo].[Orders]'. The 'Keep identity' checkbox is unchecked, 'Keep nulls' is checked, 'Table lock' is checked, and 'Check constraints' is checked. The 'Rows per batch' is set to 1, and the 'Maximum insert commit size' is set to 2147483647.

“Fast load” data access mode - pozwala commitować wstawiane wiersze nie pojedynczo, a w grupach(batches).

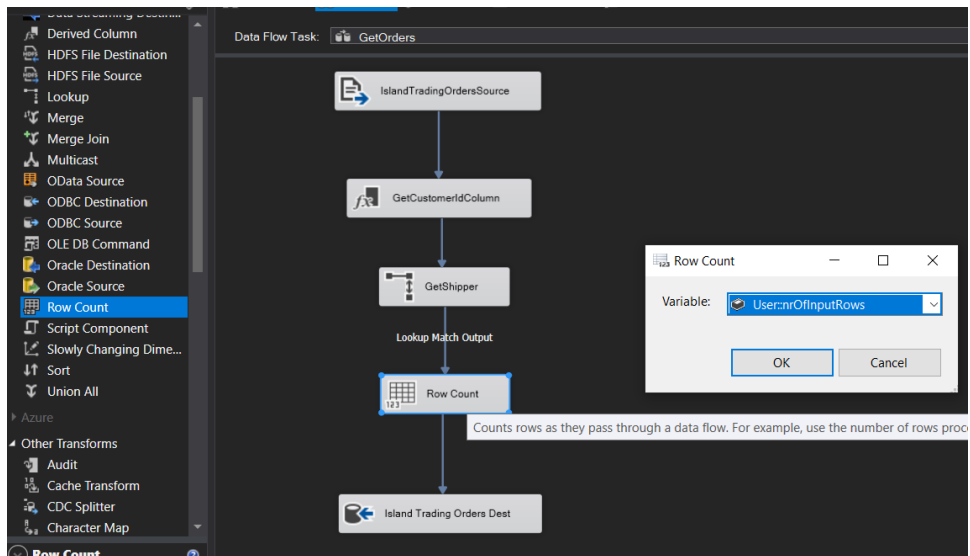
“Keep identity” - ustawiony na true oznacza, że wartości kolumny z identity będą wstawione takie, jakie są w naszych danych wejściowych, a nie zgodnie z porządkiem w bazie(rzeczywiście, jeżeli nie naruszają więzy klucza głównego).

W zakładce “Mappings” odpowiednio łączymy kolumny.

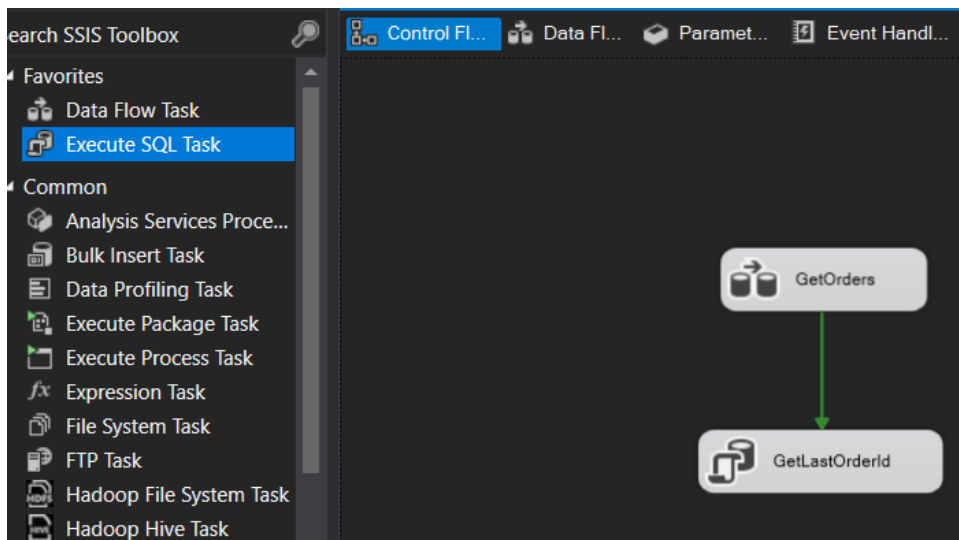


Przepływ do tabeli Orders jest już gotowy. Teraz zastanówmy się, jak uzyskać klucz obcy OrderID dla “Order Details”. Można by było wykonać Look up na podstawie pozostałych wstawionych kolumn do Orders, ale one nawet wszystkie nie gwarantują unikalności dla wiersza. Inna opcja – po wstawieniu rekordów do Orders znajdziemy największy OrderID, policzymy ile było wierszy na wejściu i dodajmy “Script Component”, w którym za pomocą licznika dodamy odpowiednie wartości dla każdego wiersza.

Policzyć wiersze w danym momencie przepływu można za pomocą elementu “Row Count” i zapiszmy zwróconą wartość do zmiennej nrOfInputRows.



Znaleźć największy OrderId możemy za pomocą elementu “Execute SQL Task”. Nazwijmy go “GetLastOrderId”.



Stwórzmy też zmienną lastOrderId.

W edytorze sql tasku wybierzmy poniższe opcje

Execute SQL Task Editor

Configure the properties required to run SQL statements and stored procedures using the selected connection.

General
Parameter Mapping
Result Set
Expressions

| | |
|------------------------|--|
| General | |
| Name | GetLastOrderId |
| Description | Execute SQL Task |
| Options | |
| TimeOut | 0 |
| CodePage | 1252 |
| TypeConversionMode | Allowed |
| Result Set | |
| ResultSet | Single row |
| SQL Statement | |
| ConnectionType | OLE DB |
| Connection | DESKTOP-KHHRPIL.Northwind |
| SQLSourceType | Direct input |
| SQLStatement | SELECT MAX(OrderID) AS lastOrderId FROM Orders |
| IsQueryStoredProcedure | False |
| BypassPrepare | True |

W zakładce “Result Set” odpowiednio zmapujemy zwróconą wartość.

General
Parameter Mapping
Result Set
Expressions

| Result Name | Variable Name |
|-------------|-------------------|
| lastOrderId | User::lastOrderId |

Następnie dodajmy data flow task “GetOrderDetails”.

Analogicznie tworzymy źródło (“IslandTradingOrderDetailsSource”) oraz wyjście dla danych (“IslandTradingOrderDetailsDest”).

Flat File Source Editor

Configure the properties used to connect to and obtain data from a text file.

Connection Manager
Columns
Error Output

Available External Column...

| | |
|-------------------------------------|----------------|
| <input checked="" type="checkbox"/> | Name |
| <input type="checkbox"/> | ShipperCompany |
| <input type="checkbox"/> | OrderDate |
| <input type="checkbox"/> | EmployeeId |
| <input checked="" type="checkbox"/> | ProductName |
| <input checked="" type="checkbox"/> | Quantity |

| External Column | Output Column |
|-----------------|---------------|
| Quantity | Quantity |
| Discount | Discount |
| ProductName | ProductName |

OLE DB Destination Editor

Configure the properties used to insert data into a relational database using an OLE DB provider.

Connection Manager: Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.

OLE DB connection manager:

Data access mode:

Name of the table or the view:

☐ Keep identity ☒ Table lock

☒ Keep nulls ☒ Check constraints

Rows per batch:

Maximum insert commit size:

OLE DB Destination Editor

Configure the properties used to insert data into a relational database using an OLE DB provider.

Connection Manager:

Mappings:

Error Output

Available Input C...

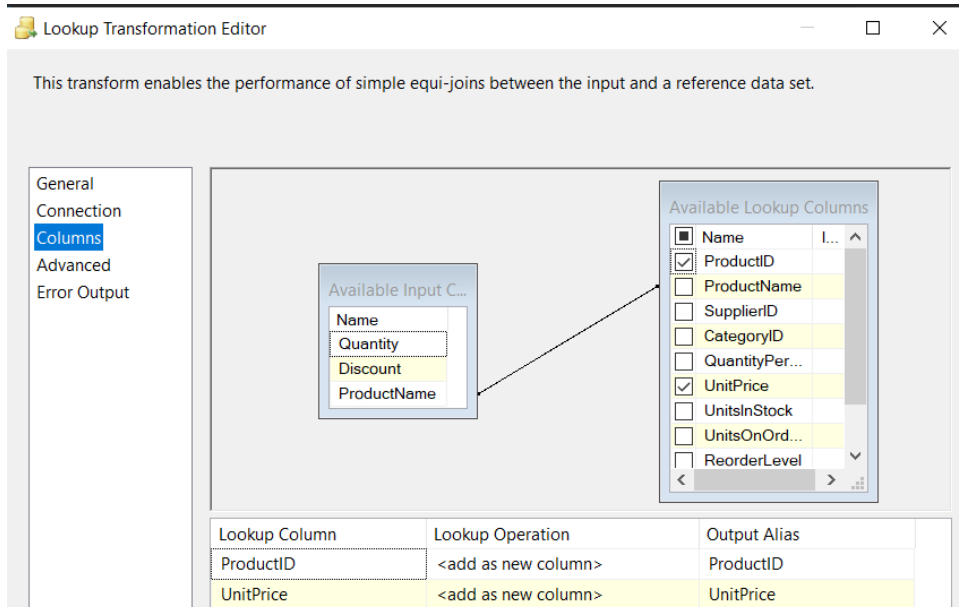
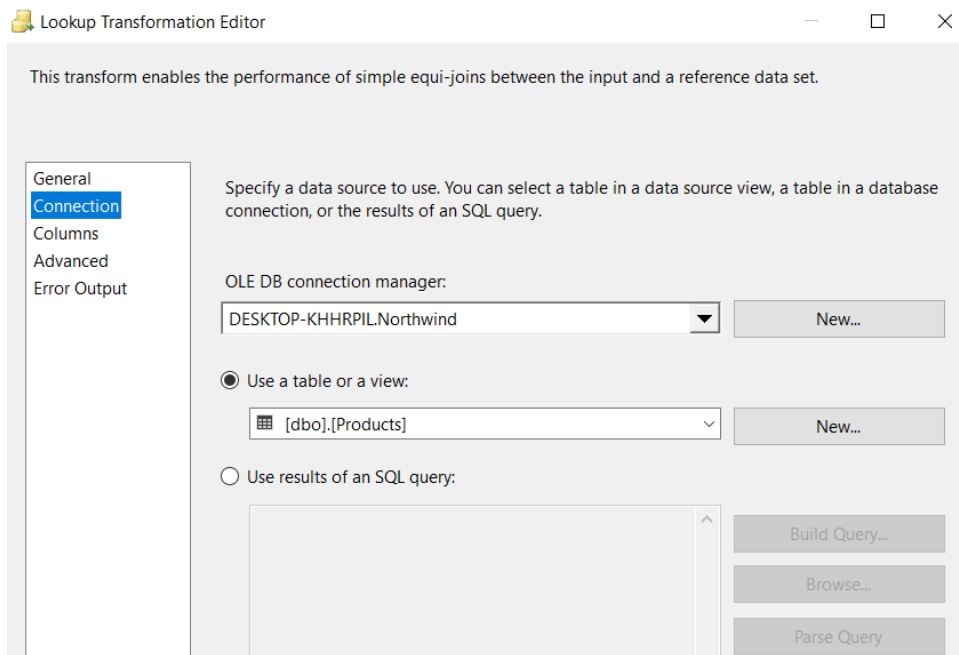
| Name |
|-------------|
| Quantity |
| Discount |
| ProductName |
| ProductID |
| UnitPrice |

Available De...

| Name |
|-----------|
| OrderID |
| ProductID |
| UnitPrice |
| Quantity |

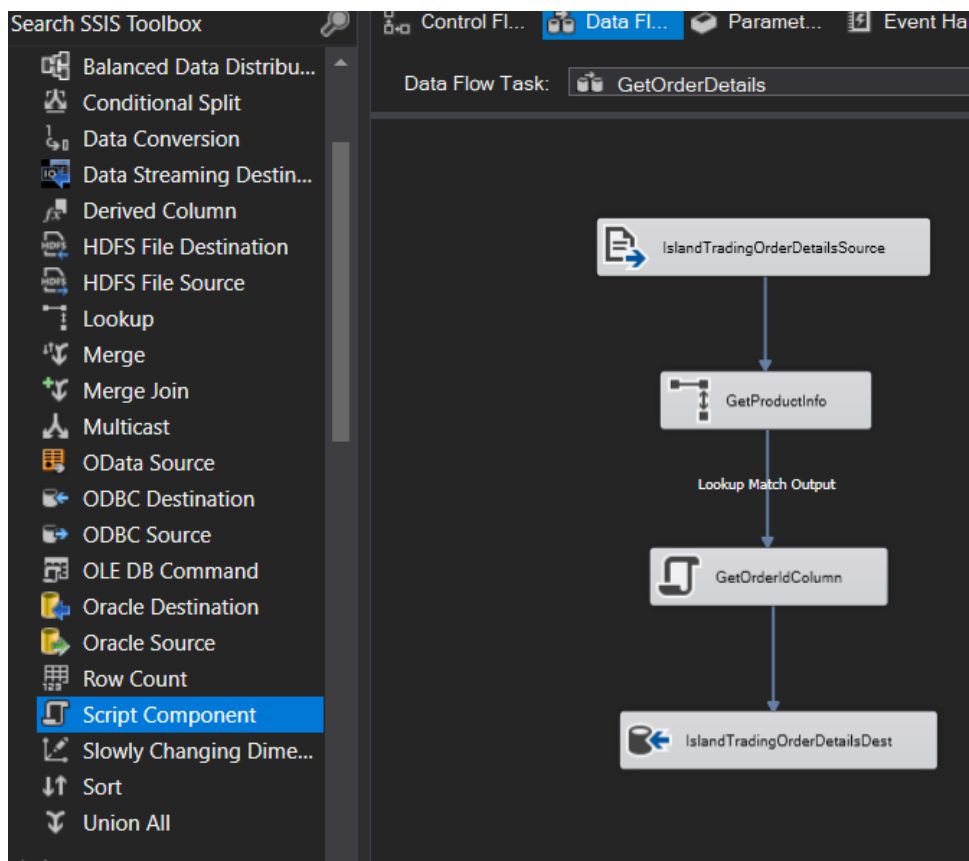
| Input Column | Destination Column |
|--------------|--------------------|
| OrderID | OrderID |
| ProductID | ProductID |
| UnitPrice | UnitPrice |
| Quantity | Quantity |
| Discount | Discount |

Dodajmy i połączmy ze źródłem Lookup, który pobierze ProductId oraz UnityPrice z tabeli Products dla kolumny w pliku ProductName. Nazwijmy go "GetProductInfo".



Przechodzimy do kolumny OrderId. Dodajmy element “Script Component”(“GetOrderIdColumn”).

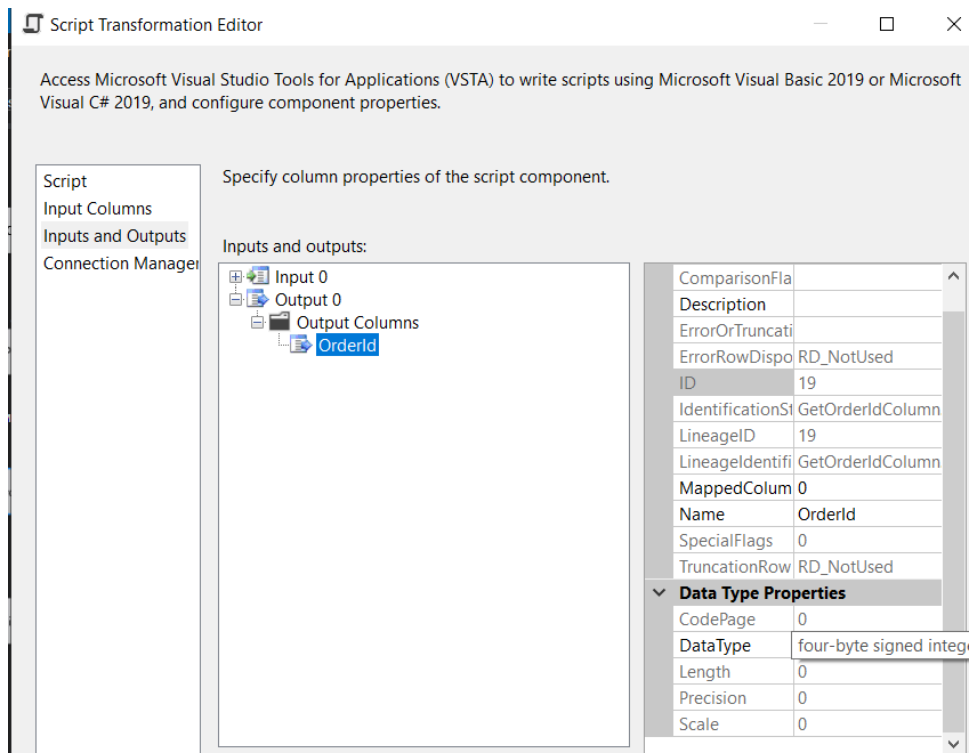
Typ skryptu – Transformation.



W edytorze w zakładce Script dodamy do ReadOnlyVariables zmienną nrOfInputRows.

W "Input Columns" odznaczamy wszystkie kolumny, bo je nie potrzebujemy dla skryptu.

W "Inputs and Outputs" dodajmy szukaną kolumnę OrderId.



Teraz zedytujmy skrypt(Editor->Script->Edit Script) w c#.

Zmienne typu ReadWriteVariables są dostępne dla zwiększenia wydajności tylko w metodzie PostExecute, która wykonywana na sam koniec. A zmienne do odczytu dostępne w dowolnym momencie. Zatem są dwie możliwości:

1. Utworzyć dodatkową zmienną do odczytu, której wartość ustawić na wartość zmiennej lastOrderId. Przypisać jej wartość do lokalnego licznika. W metodzie PostExecute przypisać wartość licznika do lastOrderId.
2. Skorzystać z klasy VariableDispenser

```
public override void PreExecute()
{
    base.PreExecute();
    VariableDispenser variableDispenser = (VariableDispenser)this.VariableDispenser;
    variableDispenser.LockForRead("User::lastOrderId");

    IDTSVariables100 vars;
    variableDispenser.GetVariables(out vars);

    lastOrderIdLocal = (
        void IDTSVariableDispenser100.GetVariables(out IDTSVariables100 ppVariables)

    vars.Unlock();
}
```

```
public override void PostExecute()
{
    base.PostExecute();
    VariableDispenser variableDispenser = (VariableDispenser)this.VariableDispenser;
    variableDispenser.LockForWrite("User::lastOrderId");
    IDTSVariables100 vars;
    variableDispenser.GetVariables(out vars);

    vars["User::lastOrderId"].Value = lastOrderIdLocal;

    vars.Unlock();
}
```

```
2 references
public override void Input0_ProcessInputRow(Input0Buffer Row)
{
    Row.OrderId = ++lastOrderIdLocal;
}
```

Przed wyjściem skrypt trzeba zbuildować.

Logowanie w SSIS

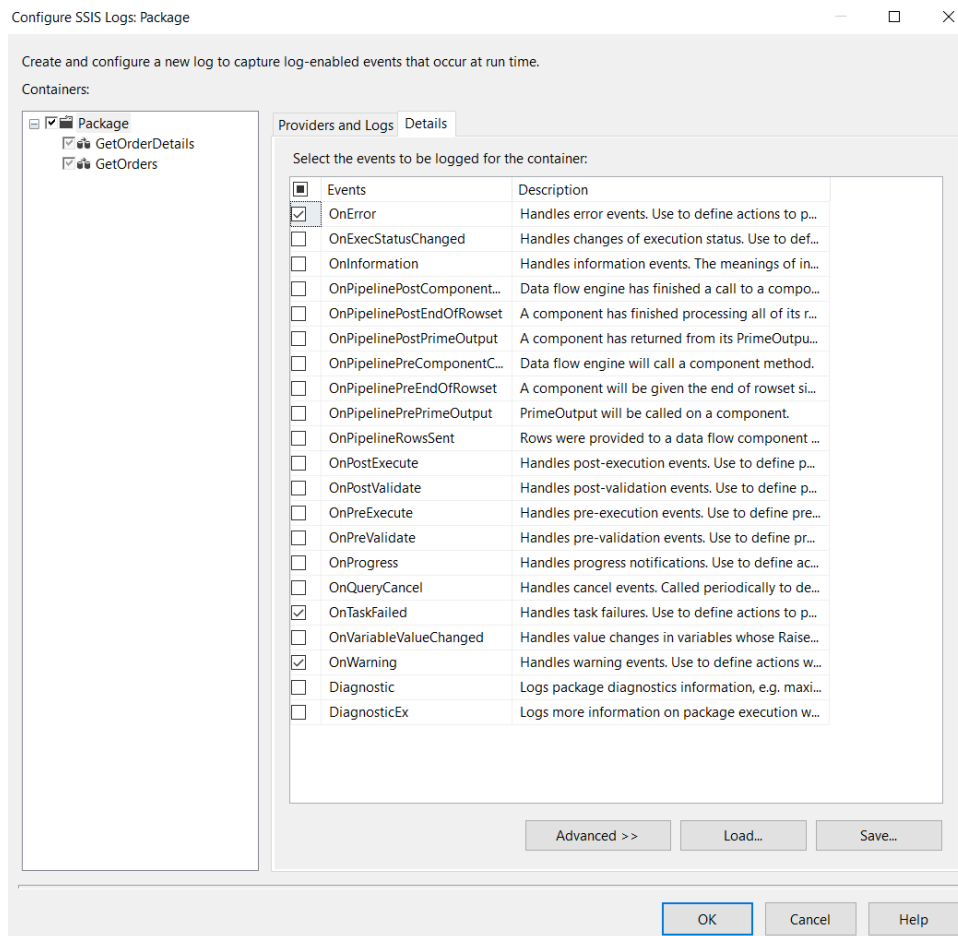
Wejdźmy do Extensions->SSIS->Logging.

Logowanie można ustawiać jak na poziomie całego pakietu, tak i dla osobnych tasków.

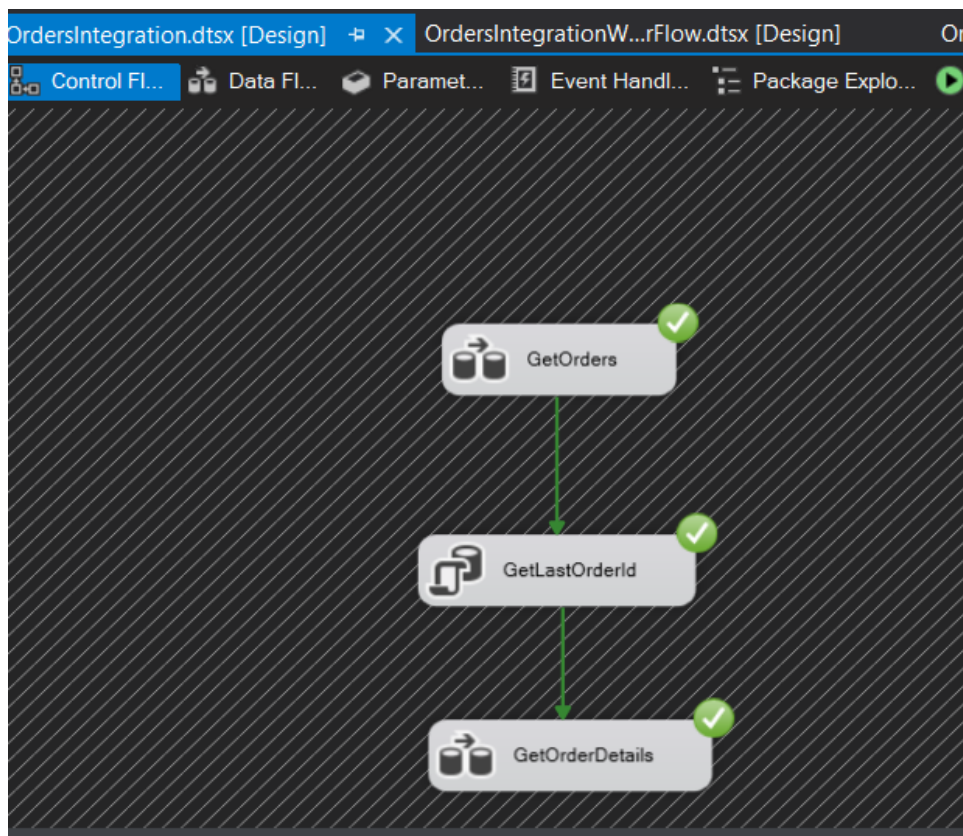
Chcemy zapisywać do pliku, zatem wybierzmy "SSIS log provider for Text files" -> Add.

W zakładce "Configuration" tworzymy nowy File Connection Manager do pliku "C:\DBProject\Logs.txt".

Następnie w zakładce “Details” wybieramy zdarzenia, które będą logowane. Jeżeli interesują nas tylko błędy, wybieramy “onTaskFailed”.



Uruchommy nasz projekt.



Teraz działa.

Sprawdźmy w SQL Server Management Studio, czy na pewno zostały dodane rekordy.

```

use Northwind;
select * from Orders where OrderId=11078;
select * from [Order Details] where OrderId=11078;
  
```

| OrderID | CustomerID | EmployeeID | OrderDate | RequiredDate | ShippedDate | ShipVia | Freight | ShipName | ShipAddress | ShipCity | ShipRegion | ShipPostalCode | ShipCountry |
|---------|------------|------------|-------------------------|--------------|-------------|---------|---------|----------|-------------|----------|------------|----------------|-------------|
| 11078 | ISLAT | 6 | 2022-10-05 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11079 | ISLAT | 3 | 2022-10-23 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11080 | ISLAT | 7 | 2022-10-13 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11081 | ISLAT | 1 | 2022-10-06 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11082 | ISLAT | 8 | 2022-10-15 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11083 | ISLAT | 9 | 2022-10-27 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11084 | ISLAT | 4 | 2022-10-21 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11085 | ISLAT | 2 | 2022-10-27 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11086 | ISLAT | 5 | 2022-10-06 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11087 | ISLAT | 9 | 2022-10-25 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11088 | ISLAT | 8 | 2022-10-17 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11089 | ISLAT | 7 | 2022-10-30 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |
| 11090 | ISLAT | 8 | 2022-10-21 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

| OrderID | ProductID | UnitPrice | Quantity | Discount |
|---------|-----------|-----------|----------|----------|
| 11078 | 11 | 21.00 | 61 | 0.97 |
| 11079 | 11 | 21.00 | 72 | 0.39 |
| 11080 | 11 | 21.00 | 96 | 0.26 |
| 11081 | 11 | 21.00 | 3 | 0.43 |
| 11082 | 11 | 21.00 | 30 | 0.014 |
| 11083 | 11 | 21.00 | 32 | 0.38 |
| 11084 | 11 | 21.00 | 5 | 0.77 |
| 11085 | 11 | 21.00 | 65 | 0.57 |
| 11086 | 11 | 21.00 | 80 | 0.5 |
| 11087 | 11 | 21.00 | 84 | 0.4 |
| 11088 | 11 | 21.00 | 83 | 0.8 |
| 11089 | 11 | 21.00 | 2 | 0.77 |
| 11090 | 11 | 21.00 | 87 | 0.27 |

Query executed successfully. DESKTOP-KHHRPIL (16.0 RTM) DESKTOP-KHHRPIL

Teraz wykonajmy nasz projekt dla pliku "IslandTradingOrdersBAD.txt", żeby sprawdzić zachowanie dla niepoprawnych danych(nie ma dostawcy "BAD COMPANY NAME" w tabeli Shippers).

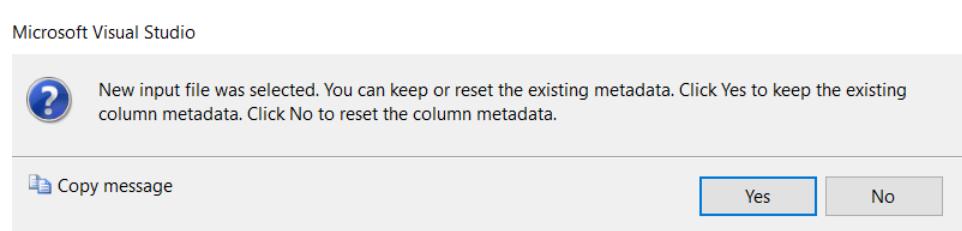
```

1 ShipperCompany,OrderDate,EmployeeId,ProductName,Quantity,Discount
2 Federal Shipping,2022-10-31,2,Manjimup Dried Apples,35,23
3 Federal Shipping,2022-10-13,6,Tofu,65,79
4 Federal Shipping,2022-10-19,8,Camembert Pierrot,13,53
5 Federal Shipping,2022-10-28,3,Ravioli Angelo,93,22
6 Federal Shipping,2022-10-01,9,Ravioli Angelo,94,58
7 Federal Shipping,2022-10-12,9,Louisiana Fiery Hot Pepper Sauce,10,63
8 Federal Shipping,2022-10-31,1,Gorgonzola Telino,46,68
9 Federal Shipping,2022-10-11,6,Sir Rodney's Marmalade,50,61
10 Federal Shipping,2022-10-10,2,Geitost,49,27
11 Federal Shipping,2022-10-03,7,Gustaf's Knäckebröd,72,16
12 Federal Shipping,2022-10-11,3,Tofu,88,29
13 BAD COMPANY NAME,2022-10-27,8,Tofu,95,96

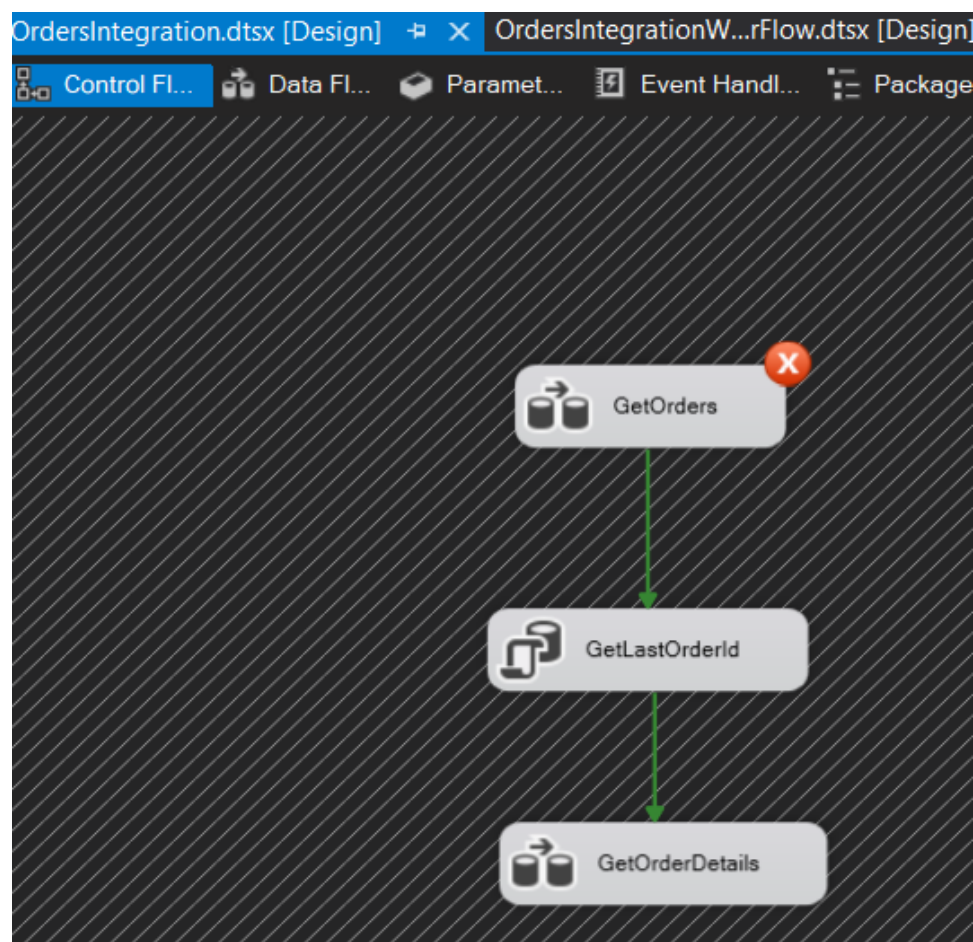
```

Wystarczy zmienić ścieżkę do pliku w IslandTradingOrdersFileConMan.

W pojawiwyszym się oknie klikamy yes, żeby zachować wszystkie ustawienia



Uruchommy projekt.



Zakończył się błędem. Dowiedzieć się o szczegóły możemy albo w zakładce "Output"


```

SSIS package "C:\Users\makso\source\repos\DB_project\DB_project\OrdersIntegration.dtsx" starting.
Information: 0x4004300A at GetOrderDetails, SSIS.Pipeline: Validation phase is beginning.
Information: 0x4004300A at GetOrders, SSIS.Pipeline: Validation phase is beginning.
Information: 0x4004300A at GetOrders, SSIS.Pipeline: Validation phase is beginning.
Warning: 0x80049304 at GetOrders, SSIS.Pipeline: Warning: Could not open global shared memory to communicate with perform
Information: 0x40043006 at GetOrders, SSIS.Pipeline: Prepare for Execute phase is beginning.
Information: 0x40043007 at GetOrders, SSIS.Pipeline: Pre-Execute phase is beginning.
Information: 0x400490F4 at GetOrders, Get shipper [11]: Get shipper has cached 3 rows.
Information: 0x400490F5 at GetOrders, Get shipper [11]: Get shipper has cached a total of 3 rows.
Information: 0x40209340 at GetOrders, Get shipper [11]: The number of unique rows added to the cache is 3.
Information: 0x402090E2 at GetOrders, Get shipper [11]: The Get shipper processed 3 rows in the cache. The processing time
Information: 0x402090DC at GetOrders, Island Trading Orders [36]: The processing of file "C:\my_files\DBProject\IslandTra
Information: 0x4004300C at GetOrders, SSIS.Pipeline: Execute phase is beginning.
Information: 0x402090DE at GetOrders, Island Trading Orders [36]: The total number of data rows processed for file "C:\my
Error: 0xC020901E at GetOrders, Get shipper [11]: Row yielded no match during lookup.
Error: 0xC0209029 at GetOrders, Get shipper [11]: SSIS Error Code DTS_E_INDUCEDTRANSFORMFAILUREONERROR. The "Get shipper"
Error: 0xC0047022 at GetOrders, SSIS.Pipeline: SSIS Error Code DTS_E_PROCESSINPUTFAILED. The ProcessInput method on compo
Information: 0x40043008 at GetOrders, SSIS.Pipeline: Post Execute phase is beginning.
Information: 0x402090DD at GetOrders, Island Trading Orders [36]: The processing of file "C:\my_files\DBProject\IslandTra
Information: 0x40043008 at GetOrders, SSIS.Pipeline: "OLE DB Destination" wrote 0 rows.
Information: 0x40043009 at GetOrders, SSIS.Pipeline: Cleanup phase is beginning.
Task failed: GetOrders
Warning: 0x80019002 at Package: SSIS Warning Code DTS_W_MAXIMUMERRORCOUNTREACHED. The Execution method succeeded, but the
SSIS package "C:\Users\makso\source\repos\DB_project\DB_project\OrdersIntegration.dtsx" finished: Failure.
The program '[6548] DtsDebugHost: DTS' has exited with code 0 (0x0).

```

Albo w pliku z logami (“Logs.txt”)

```

PackageExec, DTSExec, DTSExec, DTSExec, Package, (0E40D3B-9405-4A26-8750-75E6C3C7812), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:11 PM, 1/29/2023 2:07:11 PM, 0, 0, Beginning of package execution.
DtsWarning, DTSExec, DTSExec, DTSExec, GetOrderDetails, (0B50B0E3-AB06-427F-BCA3-CA3D9C45DEF), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:11 PM, 1/29/2023 2:07:11 PM, -2147183868, 0, Warning: Could not open global share
DtsWarning, DTSExec, DTSExec, DTSExec, GetOrders, (0B50B0E3-AB06-427F-BCA3-CA3D9C45DEF), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:11 PM, 1/29/2023 2:07:11 PM, -2147183868, 0, Warning: Could not open global share
DtsError, DTSExec, DTSExec, DTSExec, GetOrders, (0B50B0E3-AB06-427F-BCA3-CA3D9C45DEF), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:12 PM, 1/29/2023 2:07:12 PM, -1071407778, 0, Row yielded no match during lookup.
DtsError, DTSExec, DTSExec, DTSExec, GetOrders, (0B50B0E3-AB06-427F-BCA3-CA3D9C45DEF), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:12 PM, 1/29/2023 2:07:12 PM, -1071407778, 0, Row yielded no match during lookup.
DtsError, DTSExec, DTSExec, DTSExec, GetOrders, (0B50B0E3-AB06-427F-BCA3-CA3D9C45DEF), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:12 PM, 1/29/2023 2:07:12 PM, -1071407767, 0, SSIS Error Code DTS_E_INDUCEDTRANSFORM
DtsError, DTSExec, DTSExec, DTSExec, GetOrders, (0B50B0E3-AB06-427F-BCA3-CA3D9C45DEF), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:12 PM, 1/29/2023 2:07:12 PM, -1071407767, 0, SSIS Error Code DTS_E_INDUCEDTRANSFORM
DtsError, DTSExec, DTSExec, DTSExec, GetOrders, (0B50B0E3-AB06-427F-BCA3-CA3D9C45DEF), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:12 PM, 1/29/2023 2:07:12 PM, -1071407767, 0, SSIS Error Code DTS_E_PROCESSINPUTFAI
DtsError, DTSExec, DTSExec, DTSExec, GetOrders, (0B50B0E3-AB06-427F-BCA3-CA3D9C45DEF), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:12 PM, 1/29/2023 2:07:12 PM, -1071407767, 0, SSIS Error Code DTS_E_PROCESSINPUTFAI
DtsTaskFailed, DTSExec, DTSExec, DTSExec, GetOrders, (0B50B0E3-AB06-427F-BCA3-CA3D9C45DEF), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:12 PM, 1/29/2023 2:07:12 PM, 0, 0,
DtsWarning, DTSExec, DTSExec, DTSExec, Package, (0E40D3B-9405-4A26-8750-75E6C3C7812), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:12 PM, 1/29/2023 2:07:12 PM, -2147183868, 0, SSIS Warning Code DTS_W_MAXIMUMERRORCOUN
PackageEnd, DTSExec, DTSExec, DTSExec, Package, (0E40D3B-9405-4A26-8750-75E6C3C7812), (F38A4555-572F-40E4-B09F-22C38C49902), 1/29/2023 2:07:12 PM, 1/29/2023 2:07:12 PM, 0, 0, End of package execution.

```

Dostaliśmy oczekiwany błąd “Row yielded no match during lookup”.

W naszym przypadku dokładnie wiemy jaki rekord to spowodował, ale zwykle tak nie jest.

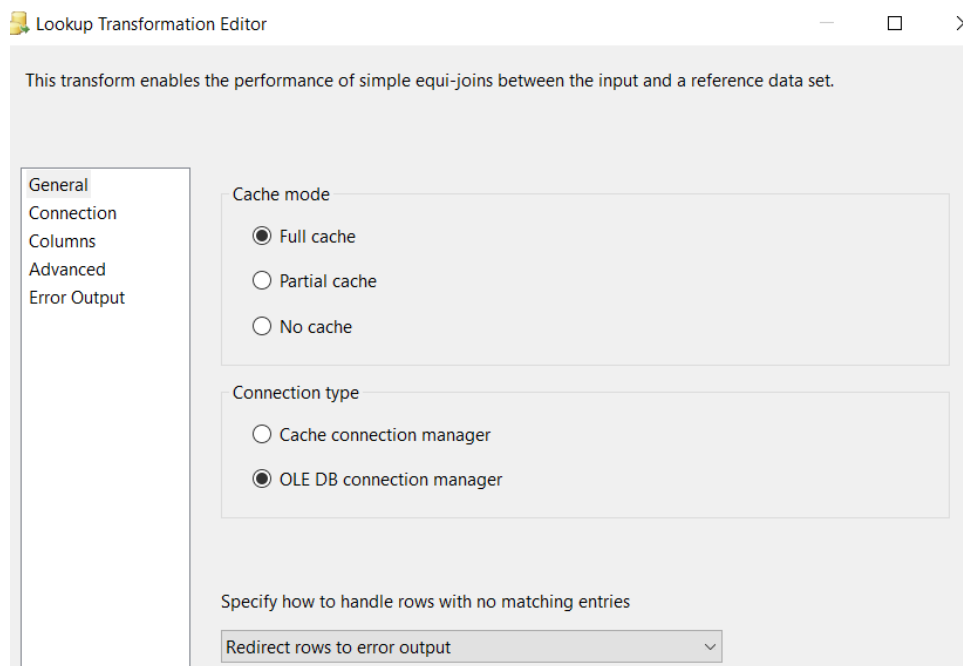
Więc jeżeli chcemy to wiedzieć, to musimy dodać po transformacji Lookup “Error flow” kierowany np. też do pliku na dysku.

ERROR FLOW

Skopiujmy nasz pakiet i nazwijmy go “OrdersIntegrationWithErrorFlow”.

Można to zrobić też za pomocą skryptu.

Domyślne zachowanie komponentu Lookup w przypadku gdy nie znajdzie dopasowania - “Fail component”. Zmienimy to na “Redirect rows to error output”.



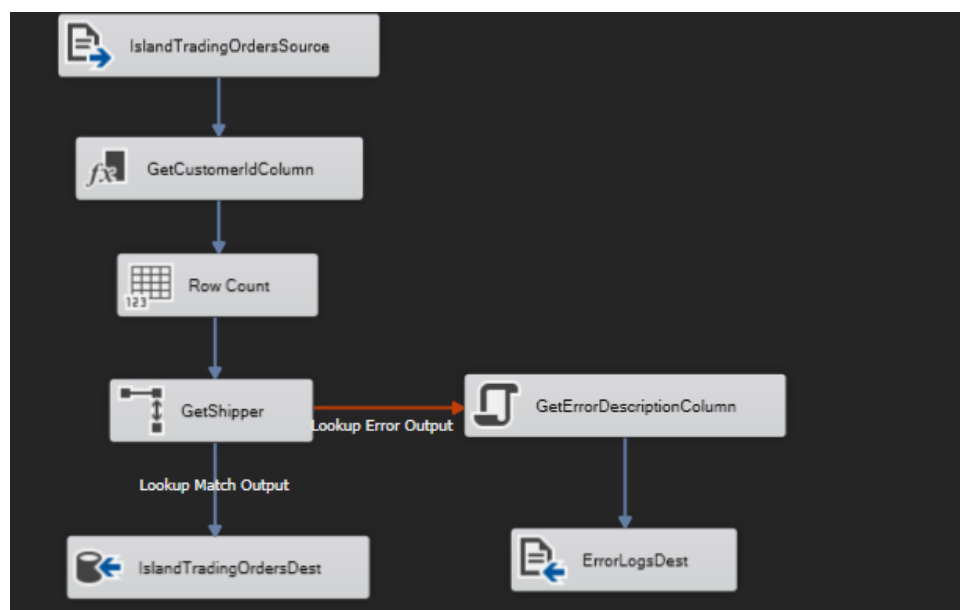
Połączymy skrypt z error outputem elementu “GetShipper”.

W edytorze skryptu w “Input Columns” wybierzmy kolumnę “Error code”, która będzie potrzebna do uzyskania opisu błędu w skrypcie. Oraz dodajmy “Output Columns” kolumnę “Error description”.

Skrypt wygląda tak

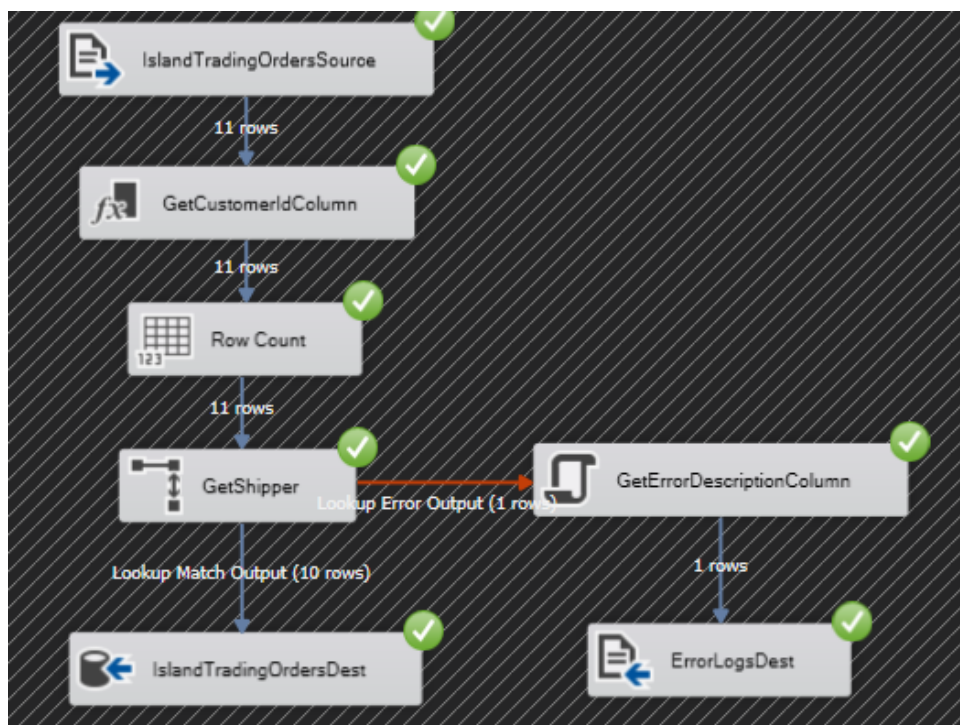
```
public override void Input0_ProcessInputRow(Input0Buffer Row)
{
    Row.ErrorDescription = this.ComponentMetaData.GetErrorDescription(Row.ErrorCode);
}
```

Jeszcze dodamy “Flat File Connection Manager” połączony z plikiem “C:\DBProject\ErrorLogs.txt” oraz komponent “Flat File Destination”.



Analogiczny Error flow dodajemy do "GetOrderDetails".(Gdyby w jednym data flow było kilka elementów generujących błędy, to wystarczyło by połączyć wyjścia error za pomocą komponenta "Union All").

Uruchommy projekt.



W pliku ErrorLogs.txt została zapisana błędna kolumna

```
BAD COMPANY NAME,2022-10-27,8,,,,ISLAT,-1071607778,0,Row yielded no match during lookup.
```

Kilka źródeł danych

Rozwiążemy teraz to samo zadanie, ale dane są przechowywane w kilku plikach w folderze OrdersSources: Orders1.txt, Orders2.txt, Orders3.txt. Struktura plików różni się tylko tym, że w pierwszej kolumnie podany CustomerId.

Skopiujmy ostatni pakiet i nazwijmy go "OrdersIntegrationSeveralSources".

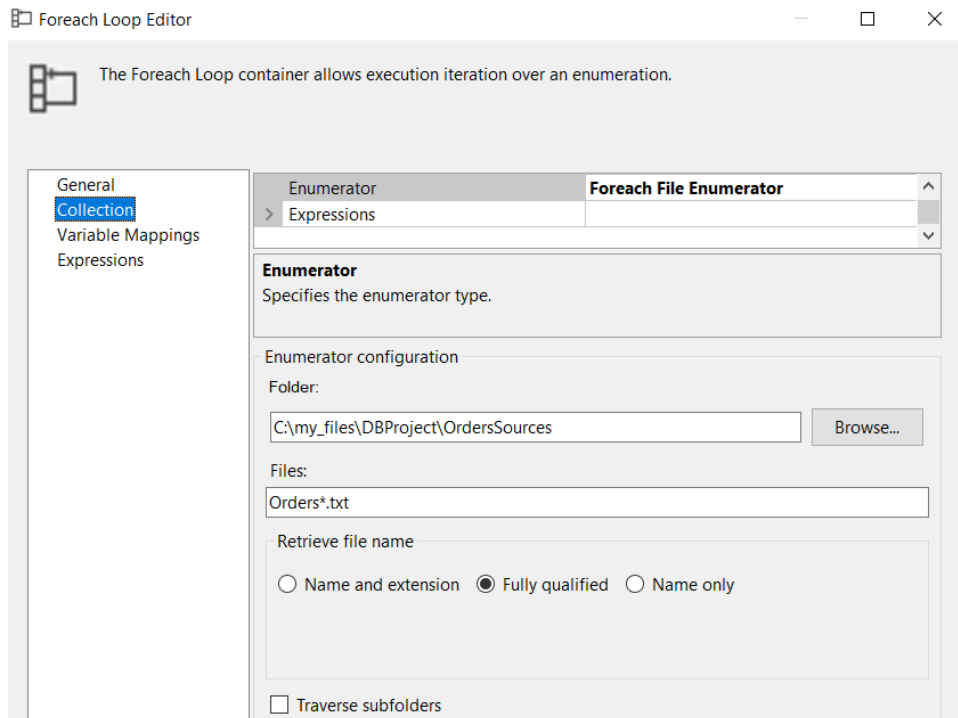
Taski data flow różnią się tylko tym, że nie ma elementu "GetCustomerIdColumn".

Jeszcze trzeba stworzyć osobny Connection Manager("DifOrdersConMan").

Żeby wykonać ten sam data flow kilka razy, ale dla różnych plików, możemy skorzystać z elementu "Foreach Loop Container".

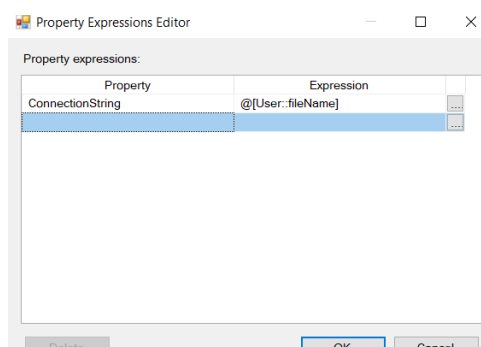
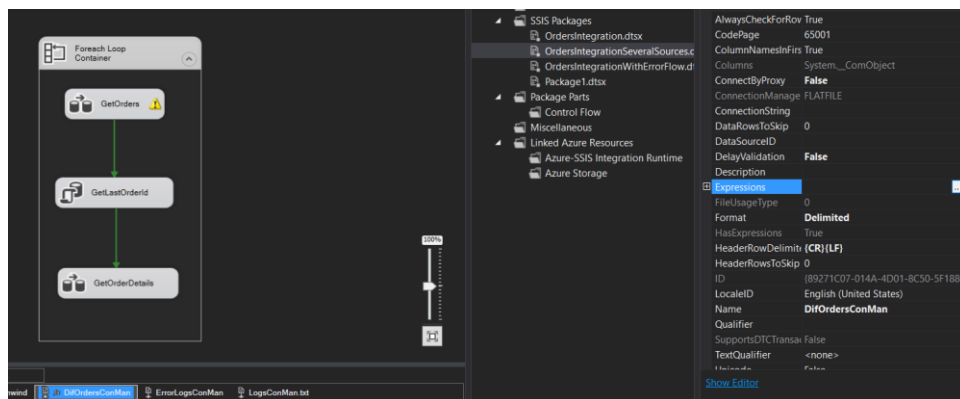
W edytorze wybieramy typ enumeratora - "Foreach File Enumerator".

W polu "Files" wpisujemy wyrażenie regularne, zgodnie z którym będą wybierane pliki.

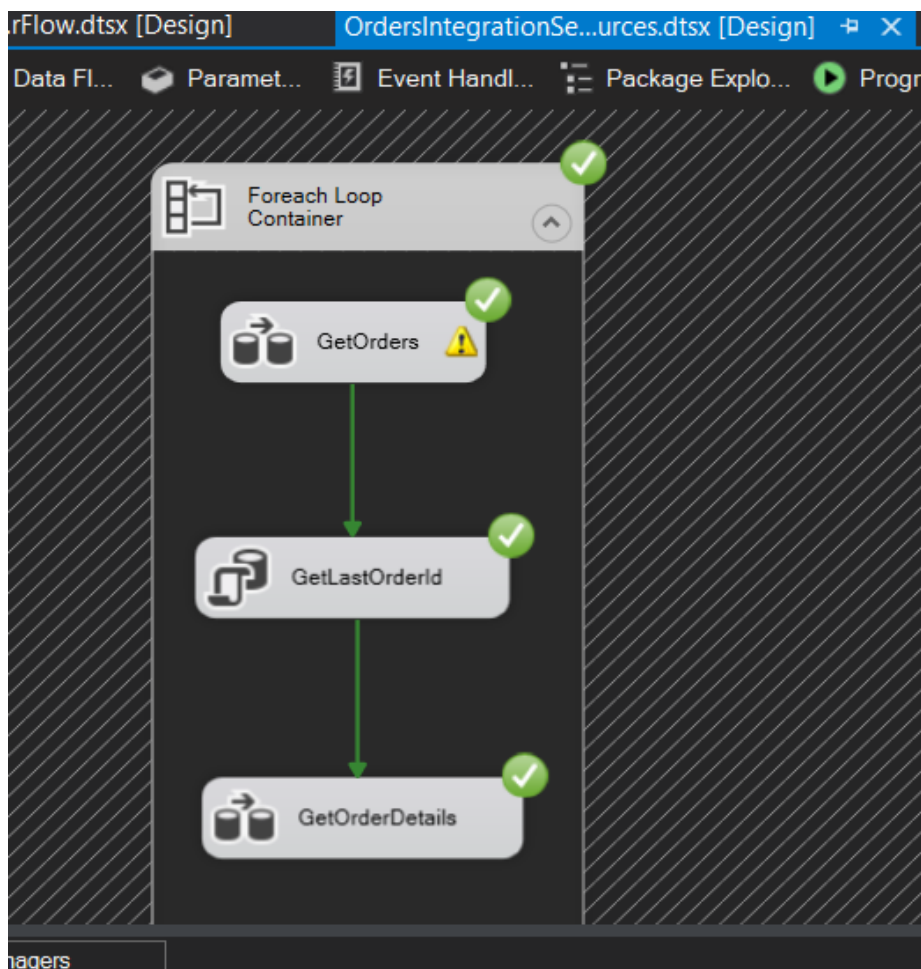


W zakładce "Variable Mapping" utworzymy nową zmienną fileName.

Następnie możemy przekazać do DifOrdersConMan jako atrybut "Connection String" naszą zmienną. Tak jak pliki mają tę samą strukturę, to zadziała.



Uruchommy projekt.



Sprawdźmy w bazie.

```

--use Northwind;
select top 300 * from Orders o join [Order Details] od on od.OrderID=o.OrderID order by o.OrderID desc;

```

| OrderID | CustomerID | EmployeeID | OrderDate | RequiredDate | ShippedDate | ShipVia | Freight | ShipName | ShipAddress | ShipCity | ShipRegion | ShipPostalCode | ShipCountry | OrderID | ProductID | UnitPrice | Quantity | Discount |
|---------|------------|------------|-------------------------|--------------|-------------|---------|---------|----------|-------------|----------|------------|----------------|-------------|---------|-----------|-----------|----------|----------|
| 1 | 14895 | RICBU | 2022-10-23 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14895 | 2 | 19.00 | 62 | 0.2 |
| 2 | 14894 | RICBU | 2022-10-18 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14894 | 48 | 12.75 | 6 | 0.57 |
| 3 | 14893 | RICBU | 2022-10-18 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14893 | 58 | 13.25 | 20 | 0.34 |
| 4 | 14892 | RICBU | 2022-10-27 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14892 | 60 | 34.00 | 20 | 0.3 |
| 5 | 14891 | RICBU | 2022-10-30 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14891 | 60 | 34.00 | 84 | 0.14 |
| 6 | 14890 | RICBU | 2022-10-17 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14890 | 3 | 10.00 | 87 | 0.18 |
| 7 | 14889 | RICBU | 2022-10-02 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14889 | 3 | 10.00 | 87 | 0.21 |
| 8 | 14888 | RICBU | 2022-10-18 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14888 | 1 | 18.00 | 78 | 0.33 |
| 9 | 14887 | RICBU | 2022-10-29 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14887 | 2 | 19.00 | 68 | 0.18 |
| 10 | 14886 | RICBU | 2022-10-18 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14886 | 52 | 7.00 | 90 | 0.38 |
| 11 | 14885 | RICBU | 2022-10-28 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14885 | 58 | 13.25 | 74 | 0.48 |
| 12 | 14884 | RICBU | 2022-10-28 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14884 | 48 | 12.75 | 35 | 0.64 |
| 13 | 14883 | RICBU | 2022-10-12 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14883 | 52 | 7.00 | 98 | 0.26 |
| 14 | 14882 | RICBU | 2022-10-02 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14882 | 5 | 21.35 | 93 | 0.0028 |
| 15 | 14881 | RICBU | 2022-10-05 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14881 | 1 | 18.00 | 61 | 0.086 |
| 16 | 14880 | RICBU | 2022-10-30 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14880 | 4 | 22.00 | 88 | 0.993 |
| 17 | 14879 | RICBU | 2022-10-20 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14879 | 1 | 18.00 | 95 | 0.23 |
| 18 | 14878 | RICBU | 2022-10-08 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14878 | 18 | 62.50 | 3 | 0.33 |
| 19 | 14877 | RICBU | 2022-10-09 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14877 | 18 | 62.50 | 62 | 0.22 |
| 20 | 14876 | RICBU | 2022-10-04 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14876 | 58 | 13.25 | 43 | 0.57 |
| 21 | 14875 | RICBU | 2022-10-02 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14875 | 5 | 21.35 | 83 | 0.51 |
| 22 | 14874 | RICBU | 2022-10-11 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14874 | 18 | 62.50 | 89 | 0.69 |
| 23 | 14873 | RICBU | 2022-10-08 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14873 | 18 | 62.50 | 23 | 0.54 |
| 24 | 14872 | RICBU | 2022-10-27 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14872 | 3 | 10.00 | 8 | 1 |
| 25 | 14871 | RICBU | 2022-10-02 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14871 | 17 | 39.00 | 46 | 0.28 |
| 26 | 14870 | RICBU | 2022-10-11 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14870 | 5 | 21.35 | 45 | 0.64 |
| 27 | 14869 | RICBU | 2022-10-07 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14869 | 40 | 18.40 | 31 | 0.59 |
| 28 | 14868 | RICBU | 2022-10-03 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14868 | 52 | 7.00 | 58 | 0.8 |
| 29 | 14867 | RICBU | 2022-10-01 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14867 | 17 | 39.00 | 38 | 0.12 |
| 30 | 14866 | RICBU | 2022-10-10 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14866 | 5 | 21.35 | 3 | 0.29 |
| 31 | 14865 | RICBU | 2022-10-08 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 14865 | 58 | 13.25 | 65 | 0.37 |

Wdrażanie pakietu na innym urządzeniu

1 sposób. Za pomocą SSMS

Uwaga! Żeby SSMS mógł uruchomić serwis integracji, wersja Integration Services w Visual Studio powinna być nie większa niż wersja w SSMS. (VS 2019 oraz SSMS 19 korzystają z wersji 16).

W SSIS są 2 modele wdrażania:

- Package deployment model
- Project deployment model

W naszym przypadku wybierzemy Package Deployment Model. Jego najważniejsze cechy:

- Jednostką konfiguracji jest pakiet
- Są konfigurowalne za pomocą osobnych plików konfiguracji

Bardziej szczegółowo różnice opisane tu - [Deploy Integration Services \(SSIS\) Projects and Packages - SQL Server Integration Services \(SSIS\) | Microsoft Learn](#)

Wymagania do komputera docelowego:

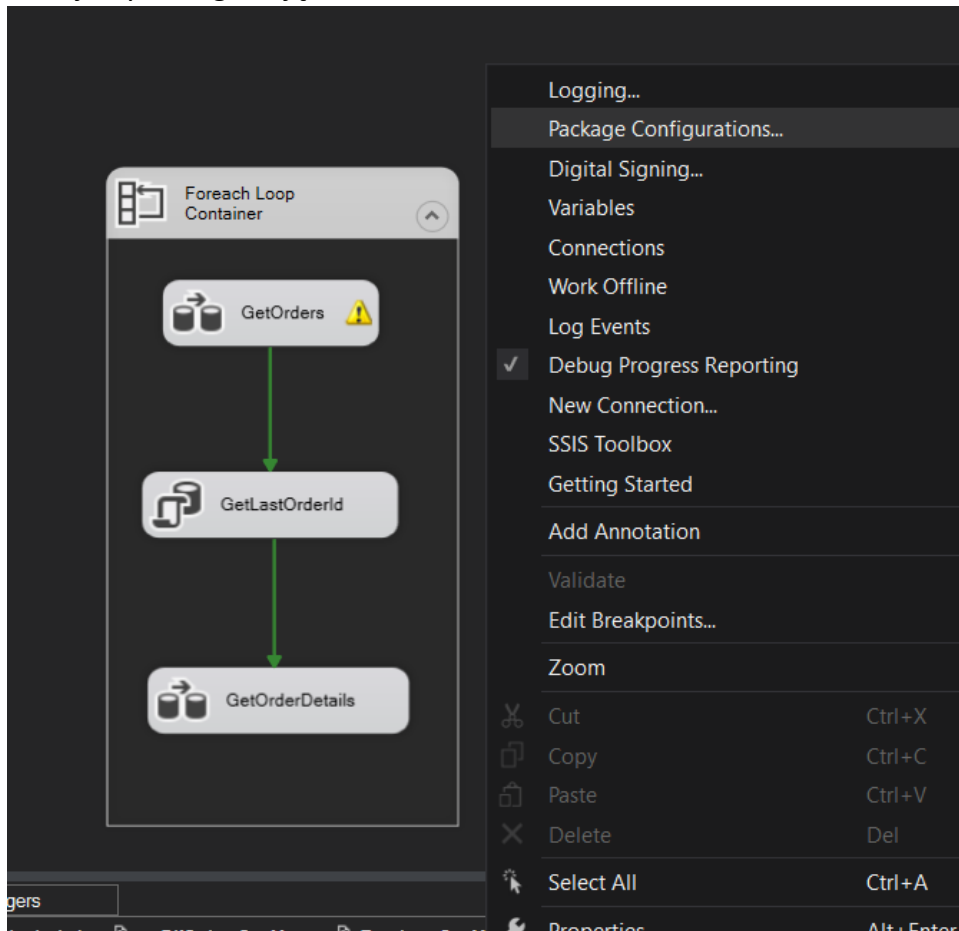
- Zainstalowany SQL Server Management Studio (żeby uruchomić pakiety)
- Użytkownik powinien mieć oprócz uprawnień do modyfikacji bazy Northwind, uprawnienia do uruchomienia SSIS pakietów oraz do modyfikacji tabeli sysssispackages w bazie systemowej msdb, w której są przechowywane informacje o dodanych do serwerach SSIS pakietach.

Będziemy wdrażać pakiet "OrdersIntegrationSeveralSources".

Krok 1: Utwórzmy konfigurację dla pakietu

Dla tego najpierw przekonwertujmy projekt na "Package Deployment Model" (Project -> Convert to Package Deployment Model). Poprzednio trzeba wszystkie menedżery połączeń przekonwertować na połączenia projektowe (Package Managers -> wybrać manager -> Convert to Package Connection).

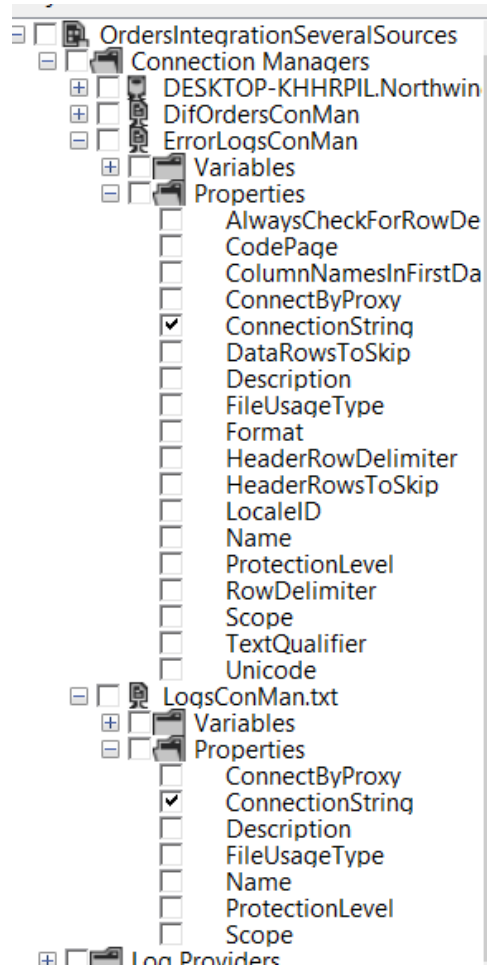
Dodajemy konfigurację



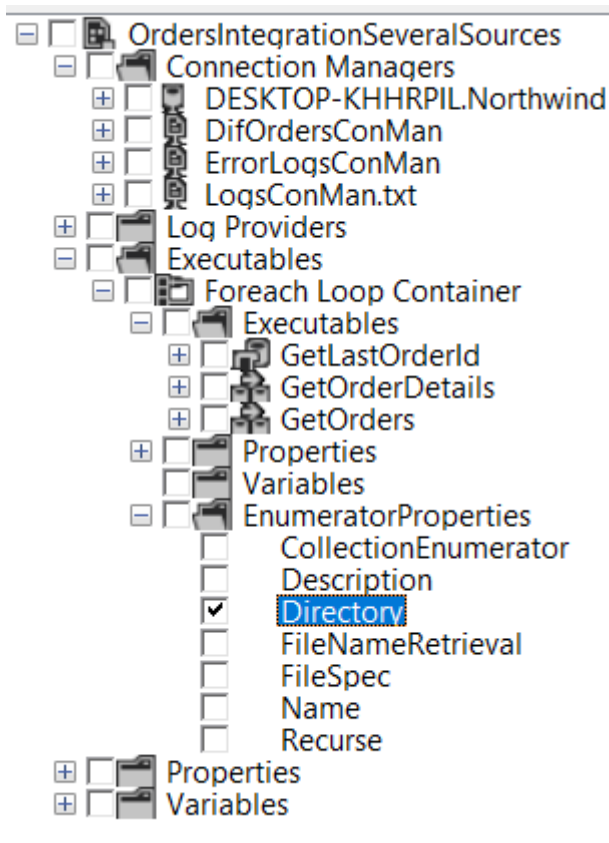
Wybieramy “XML Configuration File”. Ścieżka do pliku
“C:\DBProject\OrdersConfig.dtsConfig”.

Chcemy móc konfigurować:

- Ścieżkę do plików Logs.txt oraz ErrorLogs.txt



- Ścieżkę do folderu OrdersSources wykorzystywaną przez foreach loop

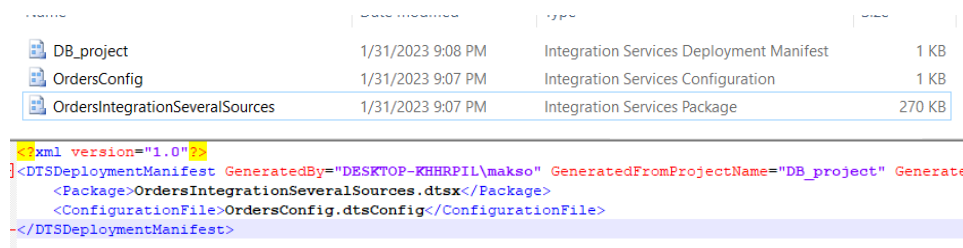


Krok 2: Utwórzmy jednostkę wdrażania, która będzie przenaszana na inny komputer("Deployment bundle").

Wchodzimy w Solution Explorer->Properties->Configuration Properties -> Deployment -> ustawiamy "create Deployment Utility" na true oraz ścieżkę na "C:\DBProject\Deployment".

Następnie budujemy nasz projekt.

Deployment bundle zawiera pakiety, konfiguracje oraz plik manifestu.



Krok 3: instalacja pakietu

Klikamy w ten plik i uruchamia się "Package Installation Wizard".

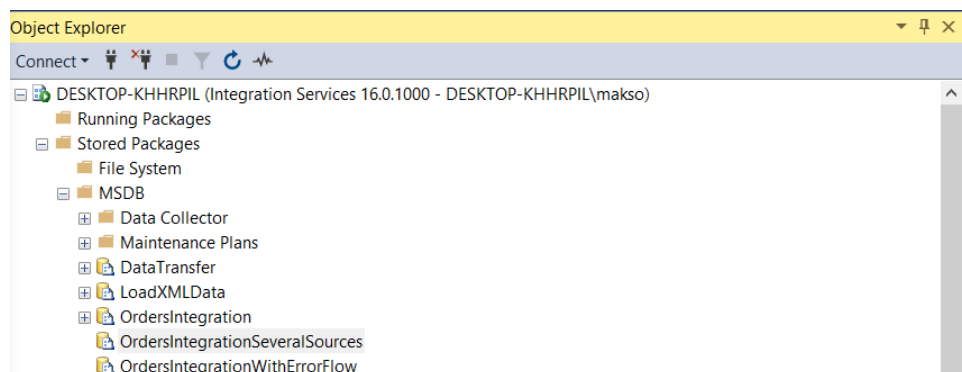
Wybieramy "SQL Server Deployment" oraz "Validate packages after installation".

Potem podajemy dane do docelowego serwera SQL.

Jako ścieżkę instalacji - "C:\DBProject\Destination".

Zmieniamy ścieżki w trzech parametrach konfiguracji dodając po DBProject folder Destination.

Sprawdźmy, że pakiet się zainstalował.



Możemy też sprawdzić, czy jest rekord w tabeli syssispackages

```
use msdb;
select * from syssispackages;
```

| | name | id | description | createdate | folderid | ownersid |
|----|---------------------------------|--------------------------------------|-------------------------------|-------------------------|--------------------------------------|---|
| 1 | SqTraceCollect | 0E149F09-1046-4DE5-98BF-4B22ED6F6C42 | System Data Collector Package | 2022-10-08 06:32:01.143 | 8877FE4B-A938-4A51-84B9-C5B0AD74B0AD | 0x01 |
| 2 | SqTraceUpload | F389A8E6-5A17-4056-ABFD-C38823F2092E | System Data Collector Package | 2022-10-08 06:32:01.210 | 8877FE4B-A938-4A51-84B9-C5B0AD74B0AD | 0x01 |
| 3 | TSQLQueryCollect | 292B1476-0F46-4490-A9B7-406724DE3C08 | System Data Collector Package | 2022-10-08 06:32:01.237 | 8877FE4B-A938-4A51-84B9-C5B0AD74B0AD | 0x01 |
| 4 | TSQLQueryUpload | 6EB73801-19C4-4092-B8B2-49735C039F10 | System Data Collector Package | 2022-10-08 06:32:01.243 | 8877FE4B-A938-4A51-84B9-C5B0AD74B0AD | 0x01 |
| 5 | PerfCounterCollect | C2EAB0C1-5BF3-4127-5EB3-26594D026E70 | System Data Collector Package | 2022-10-08 06:32:01.253 | 8877FE4B-A938-4A51-84B9-C5B0AD74B0AD | 0x01 |
| 6 | PerfCounterUpload | 0BD854CB-0A45-4E96-9C06-227A5DCD7066 | System Data Collector Package | 2022-10-08 06:32:01.263 | 8877FE4B-A938-4A51-84B9-C5B0AD74B0AD | 0x01 |
| 7 | QueryActivityCollect | 0B68F0D2-3D0C-48F3-92C5-90A0A843D0E | System Data Collector Package | 2022-10-08 06:32:01.283 | 8877FE4B-A938-4A51-84B9-C5B0AD74B0AD | 0x01 |
| 8 | QueryActivityUpload | 8330B628-8E19-47A3-92C5-FB17F9852E76 | System Data Collector Package | 2022-10-08 06:32:01.380 | 8877FE4B-A938-4A51-84B9-C5B0AD74B0AD | 0x01 |
| 9 | DataTransfer | 327F23E5-58B8-47D6-B9EE-C4526C45CBAF | DTS Wizard-generated package | 2005-12-20 09:09:01.000 | 00000000-0000-0000-0000-000000000000 | 0x010500000000000005150000000222710F97E |
| 10 | LoadXMLData | A781ED88-03E4-48CE-8A9C-30B5540C8950 | NULL | 2006-01-26 14:12:13.000 | 00000000-0000-0000-0000-000000000000 | 0x010500000000000005150000000222710F97E |
| 11 | OrdersIntegrationSeveralSources | 60675F43-982C-4471-AE9E-ED4A86E39560 | NULL | 2023-01-28 12:12:55.000 | 00000000-0000-0000-0000-000000000000 | 0x010500000000000005150000000222710F97E |
| 12 | OrdersIntegration | 1EE178ED-0F38-446D-8EC9-AF095113C0B3 | NULL | 2023-01-28 12:12:55.000 | 00000000-0000-0000-0000-000000000000 | 0x010500000000000005150000000222710F97E |
| 13 | OrdersIntegrationWithErrorFlow | A6B4D7B3-4509-4197-995E-8504B3E12CFC | NULL | 2023-01-28 12:12:55.000 | 00000000-0000-0000-0000-000000000000 | 0x010500000000000005150000000222710F97E |

Na koniec uruchommy pakiet i sprawdzmy, czy dodały się zamówienia

```
use Northwind;
select top 300 * from Orders o join [Order Details] od on od.OrderID=o.OrderID order by o.OrderID desc;
```

| | OrderID | CustomerID | EmployeeID | OrderDate | RequiredDate | ShippedDate | ShipVia | Freight | ShipName | ShipAddress | ShipCity | ShipRegion | ShipPostalCode | ShipCountry | OrderID | ProductID | UnitPr |
|----|---------|------------|------------|-------------------------|--------------|-------------|---------|---------|----------|-------------|----------|------------|----------------|-------------|---------|-----------|--------|
| 1 | 15295 | RICSU | 8 | 2022-10-23 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15295 | 2 | 19.00 |
| 2 | 15294 | RICSU | 1 | 2022-10-16 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15294 | 48 | 12.75 |
| 3 | 15293 | RICSU | 9 | 2022-10-15 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15293 | 58 | 13.25 |
| 4 | 15292 | RICSU | 3 | 2022-10-27 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15292 | 60 | 34.00 |
| 5 | 15291 | RICSU | 2 | 2022-10-30 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15291 | 60 | 34.00 |
| 6 | 15290 | RICSU | 4 | 2022-10-17 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15290 | 3 | 10.00 |
| 7 | 15289 | RICSU | 4 | 2022-10-02 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15289 | 3 | 10.00 |
| 8 | 15288 | RICSU | 7 | 2022-10-19 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15288 | 1 | 18.00 |
| 9 | 15287 | RICSU | 3 | 2022-10-29 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15287 | 2 | 19.00 |
| 10 | 15286 | RICSU | 1 | 2022-10-18 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15286 | 52 | 7.00 |
| 11 | 15285 | RICSU | 6 | 2022-10-28 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15285 | 58 | 13.25 |
| 12 | 15284 | RICSU | 9 | 2022-10-28 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15284 | 48 | 12.75 |
| 13 | 15283 | RICSU | 4 | 2022-10-12 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15283 | 52 | 7.00 |
| 14 | 15282 | RICSU | 8 | 2022-10-02 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15282 | 5 | 21.35 |
| 15 | 15281 | RICSU | 5 | 2022-10-05 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15281 | 1 | 18.00 |
| 16 | 15280 | RICSU | 3 | 2022-10-30 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15280 | 4 | 22.00 |
| 17 | 15279 | RICSU | 3 | 2022-10-20 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15279 | 1 | 18.00 |
| 18 | 15278 | RICSU | 4 | 2022-10-06 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15278 | 18 | 62.50 |
| 19 | 15277 | RICSU | 6 | 2022-10-09 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15277 | 18 | 62.50 |
| 20 | 15276 | RICSU | 7 | 2022-10-04 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15276 | 58 | 13.25 |
| 21 | 15275 | RICSU | 7 | 2022-10-02 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15275 | 5 | 21.35 |
| 22 | 15274 | RICSU | 7 | 2022-10-11 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15274 | 18 | 62.50 |
| 23 | 15273 | RICSU | 5 | 2022-10-06 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15273 | 18 | 62.50 |
| 24 | 15272 | RICSU | 9 | 2022-10-27 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15272 | 3 | 10.00 |
| 25 | 15271 | RICSU | 6 | 2022-10-02 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15271 | 17 | 39.00 |
| 26 | 15270 | RICSU | 8 | 2022-10-11 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15270 | 5 | 21.35 |
| 27 | 15269 | RICSU | 3 | 2022-10-07 00:00:00.000 | NULL | NULL | 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15269 | 40 | 18.40 |
| 28 | 15268 | RICSU | 5 | 2022-10-03 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15268 | 52 | 7.00 |
| 29 | 15267 | RICSU | 4 | 2022-10-01 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15267 | 17 | 39.00 |
| 30 | 15266 | RICSU | 7 | 2022-10-12 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15266 | 5 | 21.35 |
| 31 | 15265 | RICSU | 9 | 2022-10-08 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15265 | 58 | 13.25 |
| 32 | 15264 | RICSU | 4 | 2022-10-20 00:00:00.000 | NULL | NULL | 2 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15264 | 2 | 19.00 |
| 33 | 15263 | RICSU | 5 | 2022-10-27 00:00:00.000 | NULL | NULL | 3 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | 15263 | 5 | 21.35 |

2 sposób. Za pomocą narzędzia DTEexec

Jest instalowane wraz z SQL Server Integration Services. Może uruchamiać pakiety bezpośrednio(plik *.dtsx), z pliku projektu(*.ispac) albo z bazy msdb.

Uruchomimy z pliku OrdersIntegrationSeveralSources.dtsx(wystarczy go skopiować z projektu). Ale wtedy trzeba, żeby plik konfiguracji znajdował się tam, gdzie wskazaliśmy w projekcie("C:\DBProject\OrdersConfig.dtsConfig").

Uruchommy polecenie:

dtexec /File C:\DBProject\Deployment\OrdersIntegrationSeveralSources.dtsx

```
Progress: 2023-01-31 21:27:35.60
  Source: GetOrderDetails
  Cleanup: 16% complete
End Progress
Progress: 2023-01-31 21:27:35.60
  Source: GetOrderDetails
  Cleanup: 33% complete
End Progress
Progress: 2023-01-31 21:27:35.60
  Source: GetOrderDetails
  Cleanup: 50% complete
End Progress
Progress: 2023-01-31 21:27:35.60
  Source: GetOrderDetails
  Cleanup: 66% complete
End Progress
Progress: 2023-01-31 21:27:35.60
  Source: GetOrderDetails
  Cleanup: 83% complete
End Progress
Progress: 2023-01-31 21:27:35.60
  Source: GetOrderDetails
  Cleanup: 100% complete
End Progress
DTEXec: The package execution returned DTSEI_SUCCESS (0).
Started: 9:27:34 PM
Finished: 9:27:35 PM
Elapsed: 1.5 seconds
```