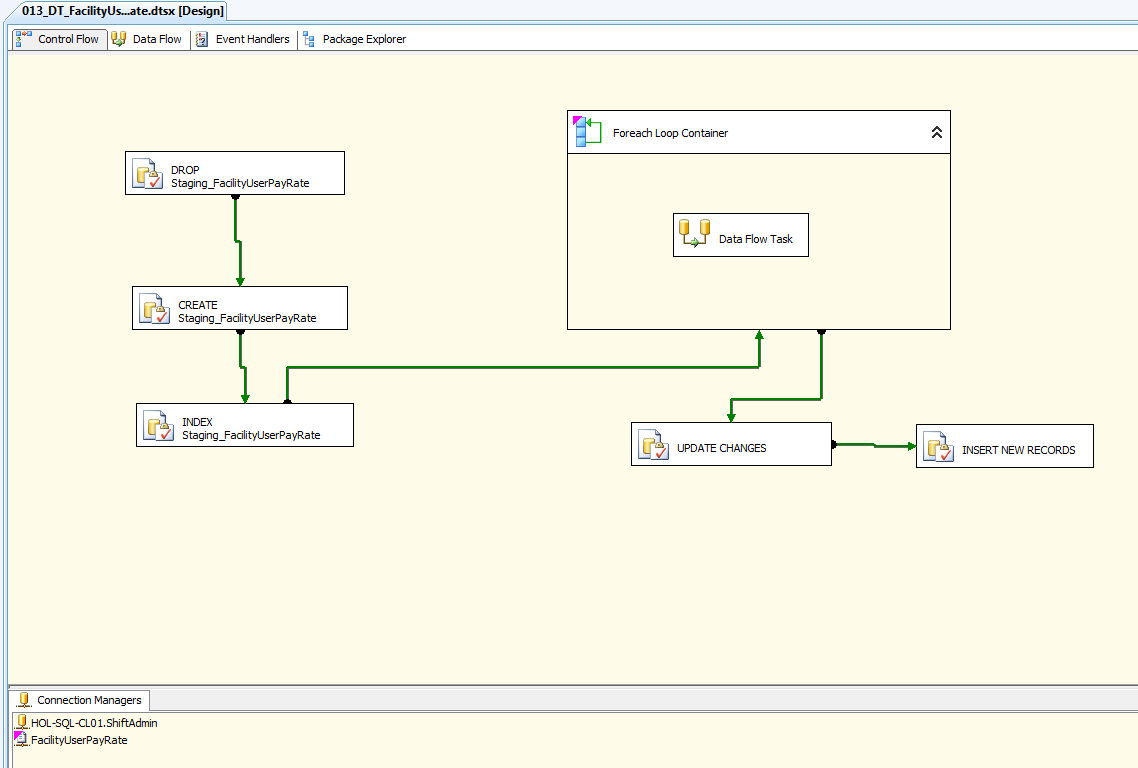
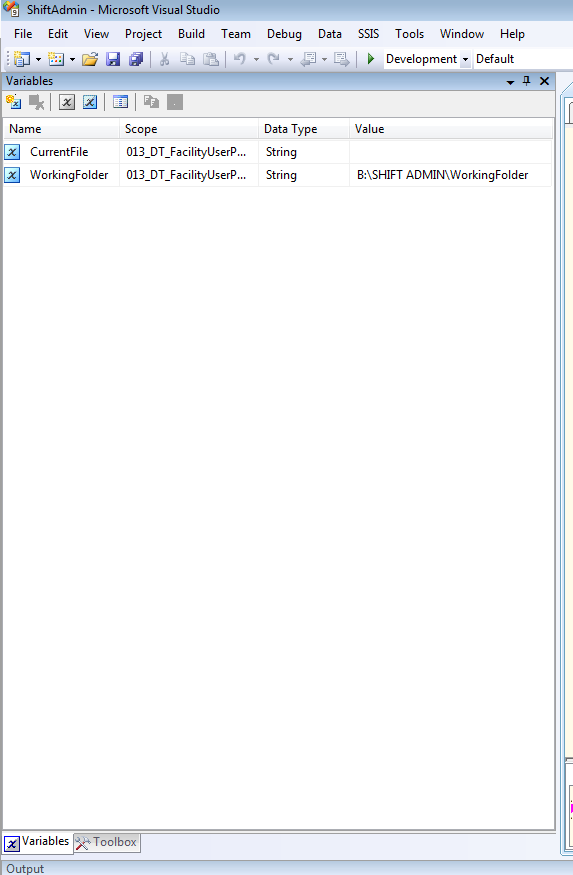
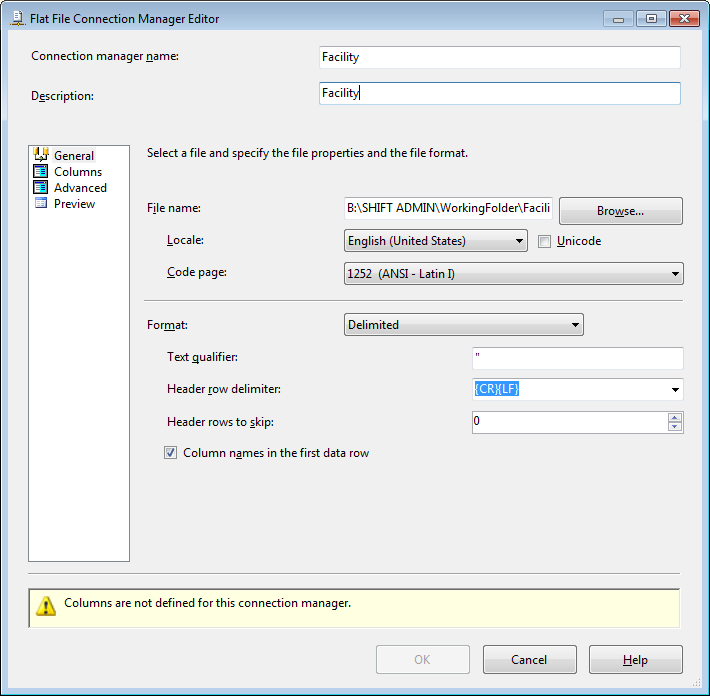
**How to loop thru CSV files Using For Each Loop container and use UPDATE/INSERT**



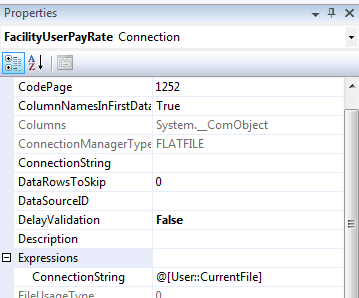
1. Drag 5 EXECUTE SQL TASK tools onto the Control Flow task
   1. DROP Staging\_Table
   2. CREATE Staging\_Table
   3. INDEX Staging\_Table
   4. UPDATE Changes
   5. INSERT New Records
2. Drag a ForEach Loop Container onto the Control Flow
3. Drag a Data Flow Task onto the Control Flow and place it inside the ForEach Loop Container.
4. Create CurrentFile variable with no value, and a WorkingFolder variable with the path to the folder the CSV files are located. Both variables are STRING and have a scope of the Package.



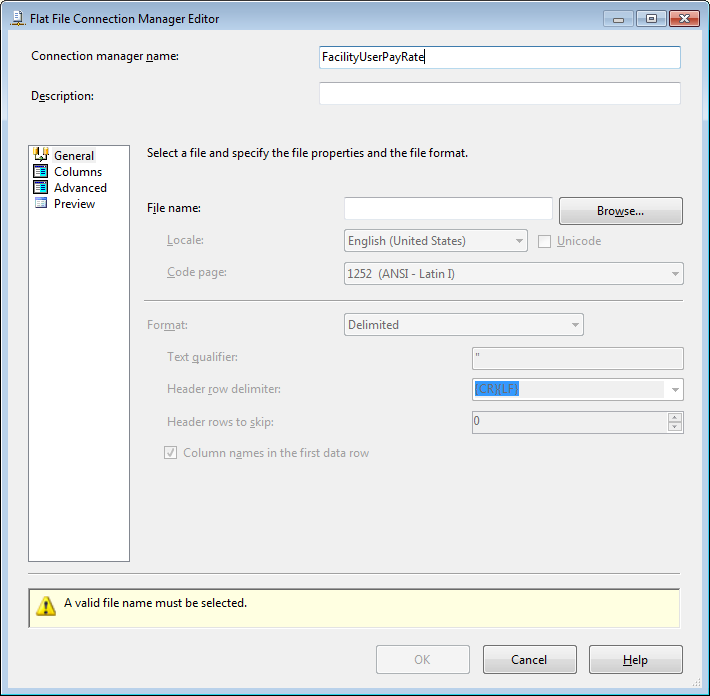
1. Create a FLAT FILE Connection Manager and Browse to the CSV file to get one file started. Format = Delimited. Text Qualifier = “ (a single double quote)



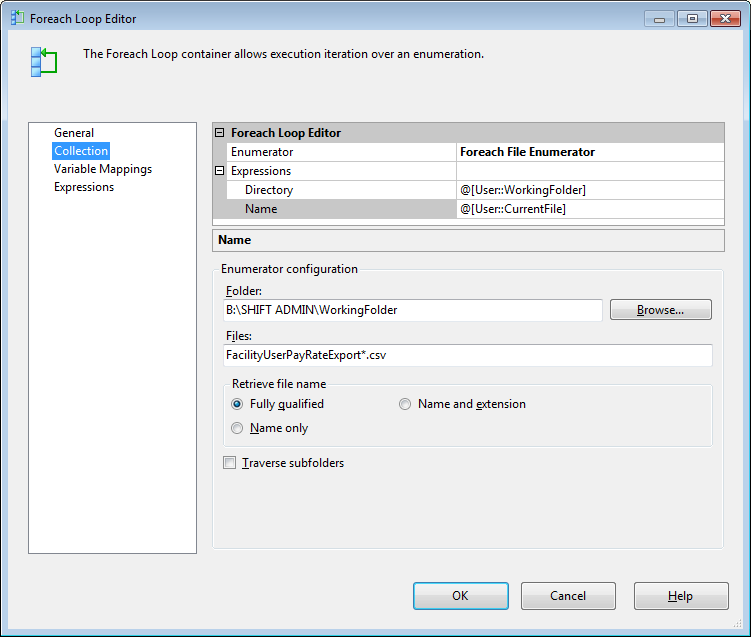
1. Put the focus on the new FLAT FILE Connection Manager and in the Properties Window Create a ConnectionString EXPRESSION using the CurrentFile variable

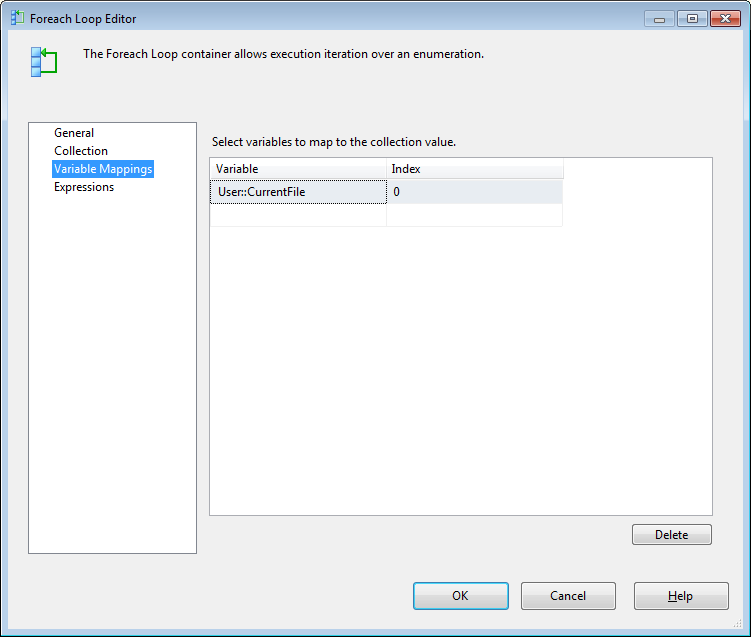


1. After step 6 is complete if I re-open the FLAT FILE Connection Manager I should see everything is greyed out.



1. Right click on the ForEach Loop Container and in Collection screen, browse to the folder I used in my Directory variable for the CSV files. Use a separate Package for each CSV file. Also, set up a
   1. Directory Expression = @[User::WorkingFolder]
   2. Name Expression = @[User::CurrentFile]
2. Set Variable Mappings to the Variable User::CurrentFile





1. Rename the first Execute SQl Task to DROP Staging Table and enter code.

IF OBJECT\_ID('[dbo].[Staging\_FacilityUserPayRate]')

IS NOT NULL

DROP TABLE [dbo].[Staging\_FacilityUserPayRate]

1. Rename the second Execute SQl Task to CREATE TABLE Staging Table and enter code.

IF OBJECT\_ID('[dbo].[Staging\_FacilityUserPayRate]') IS NULL

BEGIN CREATE TABLE [dbo].[Staging\_FacilityUserPayRate](

FacilityUserPayRateID INT NULL,

FacilityID INT NULL,

UserID INT NULL,

PayRate DECIMAL(18,2) NULL,

FlatAmount DECIMAL(18,2) NULL,

RVURate DECIMAL(18,2) NULL,

Overrate DECIMAL(18,2) NULL)

END

1. Rename the third Execute SQl Task to INDEX TABLE Staging Table and enter code that creates an INDEX on the Primary Key

IF OBJECT\_ID('[dbo].[Staging\_FacilityUserPayRate]') IS NOT NULL

AND NOT EXISTS (SELECT 1 FROM sys.indexes WHERE name = 'IX\_Staging\_FacilityUserPayRate' AND object\_name(object\_id) = 'Staging\_FacilityUserPayRate')

BEGIN

CREATE INDEX [IX\_Staging\_FacilityUserPayRate] ON Staging\_FacilityUserPayRate(FacilityUserPayRateID);

END.

1. Drag a FLAT FILE Source onto the Data Flow and connect to the FLAT FILE Connection Manager
2. Drag a OLE DB Destination onto the Data Flow and map it to the Staging Table I created
3. Rename the fourth Execute SQl Task to UPDATE CHANGES and enter code that Updates the Changes.

UPDATE MyTarget

SET

MyTarget.FacilityID = CASE WHEN (Staging.FacilityID) =0 THEN NULL ELSE Staging.FacilityID END ,

MyTarget.UserID = CASE WHEN (Staging.UserID) =0 THEN NULL ELSE Staging.UserID END ,

MyTarget.PayRate = Staging.PayRate,

MyTarget.FlatAmount = Staging.FlatAmount,

MyTarget.RVURate = Staging.RVURate,

MyTarget.Overrate = Staging.Overrate

FROM dbo.FacilityUserPayRate MyTarget

INNER JOIN dbo.Staging\_FacilityUserPayRate Staging

ON MyTarget.FacilityUserPayRateID = Staging.FacilityUserPayRateID

/\*

Did not include Update on KEY

\*/

1. Rename the fifth Execute SQl Task to INSERT NEW RECORDS and enter code that INSERT NEW RECORDS

INSERT INTO dbo.FacilityUserPayRate(FacilityUserPayRateID, FacilityID, UserID, PayRate, FlatAmount, RVURate, Overrate )

SELECT DISTINCT

CASE WHEN (Staging.FacilityUserPayRateID) =0 THEN NULL ELSE Staging.FacilityUserPayRateID END ,

CASE WHEN (Staging.FacilityID) =0 THEN NULL ELSE Staging.FacilityID END ,

CASE WHEN (Staging.UserID) =0 THEN NULL ELSE Staging.UserID END ,

Staging.PayRate,

Staging.FlatAmount,

Staging.RVURate,

Staging.Overrate

FROM dbo.Staging\_FacilityUserPayRate Staging

LEFT OUTER JOIN dbo.FacilityUserPayRate ExistingData

ON ExistingData.FacilityUserPayRateID = Staging.FacilityUserPayRateID

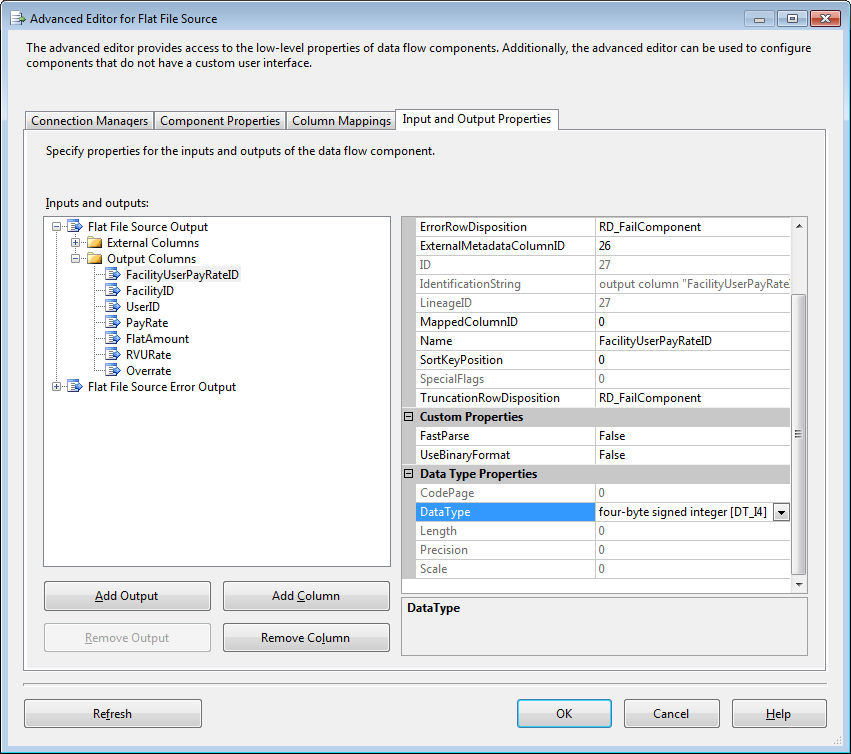
WHERE ExistingData.FacilityUserPayRateID IS NULL

/\*

This only Inserts new records. Includes the Key

\*/

1. Right click on the FLAT FILE Source and choose Show Advanced Editor.
2. Go to Input and Output Properties
3. Go to Flat File Source Output
4. Open Output Columns
5. Choose each column and change the dataTYPES TO MATCH THE DATATYPES IN MY STAGING TABLE



**How to accommodate for NULL or empty string on a DATE column and how to use trim for a VARCHAR column and how to accommodate for 0 on an int column when using code to do an UPSERT(Update/Insert)**

UPDATE MyTarget

SET

MyTarget.ShiftName = LTRIM(RTRIM(Staging.ShiftName)),

MyTarget.ScheduledUserID = CASE WHEN (Staging.ScheduledUserID) =0 THEN NULL ELSE Staging.ScheduledUserID END ,

MyTarget.UserStatusID = CASE WHEN (Staging.UserStatusID) =0 THEN NULL ELSE Staging.UserStatusID END ,

MyTarget.UserEmploymentID = CASE WHEN (Staging.UserEmploymentID) =0 THEN NULL ELSE Staging.UserEmploymentID END ,

MyTarget.FacilityID = CASE WHEN (Staging.FacilityID) =0 THEN NULL ELSE Staging.FacilityID END ,

MyTarget.ShiftUserTypeID = CASE WHEN (Staging.ShiftUserTypeID) =0 THEN NULL ELSE Staging.ShiftUserTypeID END ,

MyTarget.CreatedBy = Staging.CreatedBy,

MyTarget.PublishedUserStatusID = CASE WHEN (Staging.PublishedUserStatusID) =0 THEN NULL ELSE Staging.PublishedUserStatusID END ,

MyTarget.PublishedUserEmploymentID = CASE WHEN (Staging.PublishedUserEmploymentID) =0 THEN NULL ELSE Staging.PublishedUserEmploymentID END ,

MyTarget.PaidUserStatusID = CASE WHEN (Staging.PaidUserStatusID) =0 THEN NULL ELSE Staging.PaidUserStatusID END ,

MyTarget.PaidUserEmploymentID = CASE WHEN (Staging.PaidUserEmploymentID) =0 THEN NULL ELSE Staging.PaidUserEmploymentID END ,

MyTarget.IsNight = Staging.IsNight,

MyTarget.IsWeekend = Staging.IsWeekend,

MyTarget.CountAsShift = Staging.CountAsShift,

MyTarget.BasePay = Staging.BasePay,

MyTarget.DiffPay = Staging.DiffPay,

MyTarget.TotalPay = Staging.TotalPay,

MyTarget.MalpracticeRate = Staging.MalpracticeRate,

MyTarget.MalpracticeInsurance = Staging.MalpracticeInsurance,

MyTarget.ActualDuration = Staging.ActualDuration,

MyTarget.OriginalActualDuration = Staging.OriginalActualDuration,

MyTarget.PublishedUserID = CASE WHEN (Staging.PublishedUserID) =0 THEN NULL ELSE Staging.PublishedUserID END ,

MyTarget.CreatedDate = CASE WHEN ISDATE(Staging.CreatedDate) =0 OR RTRIM(Staging.CreatedDate) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.CreatedDate) END ,

MyTarget.EndDateTime = CASE WHEN ISDATE(Staging.EndDateTime) =0 OR RTRIM(Staging.EndDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.EndDateTime) END ,

MyTarget.OriginalEndDateTime = CASE WHEN ISDATE(Staging.OriginalEndDateTime) =0 OR RTRIM(Staging.OriginalEndDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.OriginalEndDateTime) END ,

MyTarget.OriginalStartDateTime = CASE WHEN ISDATE(Staging.OriginalStartDateTime) =0 OR RTRIM(Staging.OriginalStartDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.OriginalStartDateTime) END ,

MyTarget.PaidEndDateTime = CASE WHEN ISDATE(Staging.PaidEndDateTime) =0 OR RTRIM(Staging.PaidEndDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.PaidEndDateTime) END ,

MyTarget.PaidStartDateTime = CASE WHEN ISDATE(Staging.PaidStartDateTime) =0 OR RTRIM(Staging.PaidStartDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.PaidStartDateTime) END ,

MyTarget.PublishedEndDateTime = CASE WHEN ISDATE(Staging.PublishedEndDateTime) =0 OR RTRIM(Staging.PublishedEndDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.PublishedEndDateTime) END ,

MyTarget.PublishedStartDateTime = CASE WHEN ISDATE(Staging.PublishedStartDateTime) =0 OR RTRIM(Staging.PublishedStartDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.PublishedStartDateTime) END ,

MyTarget.StartDateTime = CASE WHEN ISDATE(Staging.StartDateTime) =0 OR RTRIM(Staging.StartDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.StartDateTime) END ,

MyTarget.PublishedActualDuration = Staging.PublishedActualDuration,

MyTarget.PaidUserID = CASE WHEN (Staging.PaidUserID) =0 THEN NULL ELSE Staging.PaidUserID END ,

MyTarget.PaidActualDuration = Staging.PaidActualDuration

FROM dbo.ScheduledShifts MyTarget

INNER JOIN dbo.Staging\_ScheduledShifts Staging

ON MyTarget.ScheduledShiftID = Staging.ScheduledShiftID

/\*

Did not include Update on KEY

\*/

INSERT INTO dbo.ScheduledShifts( ScheduledShiftID, ScheduledUserID, UserStatusID, UserEmploymentID, FacilityID, ShiftName,

ShiftUserTypeID, StartDateTime, EndDateTime, ActualDuration, IsNight, IsWeekend, CountAsShift, CreatedDate, CreatedBy,

OriginalStartDateTime, OriginalEndDateTime, OriginalActualDuration, PublishedUserID, PublishedUserStatusID, PublishedUserEmploymentID,

PublishedStartDateTime, PublishedEndDateTime, PublishedActualDuration, PaidUserID, PaidUserStatusID, PaidUserEmploymentID,

PaidStartDateTime, PaidEndDateTime, PaidActualDuration, BasePay, DiffPay, TotalPay, MalpracticeRate, MalpracticeInsurance )

SELECT DISTINCT

Staging.ScheduledShiftID,

CASE WHEN Staging.ScheduledUserID =0 THEN NULL ELSE Staging.ScheduledUserID END,

CASE WHEN Staging.UserStatusID =0 THEN NULL ELSE Staging.UserStatusID END,

CASE WHEN Staging.UserEmploymentID =0 THEN NULL ELSE Staging.UserEmploymentID END,

CASE WHEN Staging.FacilityID =0 THEN NULL ELSE Staging.FacilityID END ,

Staging.ShiftName,

CASE WHEN Staging.ShiftUserTypeID =0 THEN NULL ELSE Staging.ShiftUserTypeID END ,

CASE WHEN ISDATE(Staging.StartDateTime) =0 OR RTRIM(Staging.StartDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.StartDateTime) END ,

CASE WHEN ISDATE(Staging.EndDateTime) =0 OR RTRIM(Staging.EndDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.EndDateTime) END ,

Staging.ActualDuration,

Staging.IsNight,

Staging.IsWeekend,

Staging.CountAsShift,

CASE WHEN ISDATE(Staging.CreatedDate) =0 OR RTRIM(Staging.CreatedDate) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.CreatedDate) END ,

Staging.CreatedBy,

CASE WHEN ISDATE(Staging.OriginalStartDateTime) =0 OR RTRIM(Staging.OriginalStartDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.OriginalStartDateTime) END ,

CASE WHEN ISDATE(Staging.OriginalEndDateTime) =0 OR RTRIM(Staging.OriginalEndDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.OriginalEndDateTime) END ,

Staging.OriginalActualDuration,

CASE WHEN Staging.PublishedUserID =0 THEN NULL ELSE Staging.PublishedUserID END ,

CASE WHEN Staging.PublishedUserStatusID =0 THEN NULL ELSE Staging.PublishedUserStatusID END ,

CASE WHEN Staging.PublishedUserEmploymentID =0 THEN NULL ELSE Staging.PublishedUserEmploymentID END ,

CASE WHEN ISDATE(Staging.PublishedStartDateTime) =0 OR RTRIM(Staging.PublishedStartDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.PublishedStartDateTime) END ,

CASE WHEN ISDATE(Staging.PublishedEndDateTime) =0 OR RTRIM(Staging.PublishedEndDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.PublishedEndDateTime) END ,

Staging.PublishedActualDuration,

CASE WHEN Staging.PaidUserID =0 THEN NULL ELSE Staging.PaidUserID END ,

CASE WHEN Staging.PaidUserStatusID =0 THEN NULL ELSE Staging.PaidUserStatusID END ,

CASE WHEN Staging.PaidUserEmploymentID =0 THEN NULL ELSE Staging.PaidUserEmploymentID END ,

CASE WHEN ISDATE(Staging.PaidStartDateTime) =0 OR RTRIM(Staging.PaidStartDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.PaidStartDateTime) END ,

CASE WHEN ISDATE(Staging.PaidEndDateTime) =0 OR RTRIM(Staging.PaidEndDateTime) = '' THEN NULL ELSE CONVERT(DATETIME,Staging.PaidEndDateTime) END ,

Staging.PaidActualDuration,

Staging.BasePay,

Staging.DiffPay,

Staging.TotalPay,

Staging.MalpracticeRate,

Staging.MalpracticeInsurance

FROM dbo.Staging\_ScheduledShifts Staging

LEFT OUTER JOIN dbo.ScheduledShifts ExistingData

ON ExistingData.ScheduledShiftID = Staging.ScheduledShiftID

WHERE ExistingData.ScheduledShiftID IS NULL

/\*

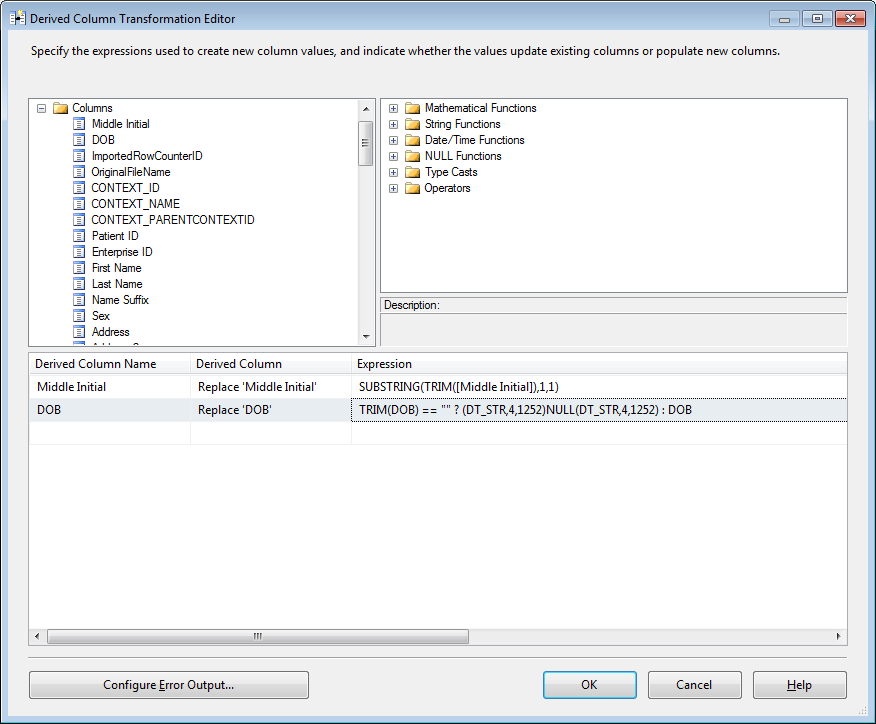
This only Inserts new records. Includes the Key

\*/

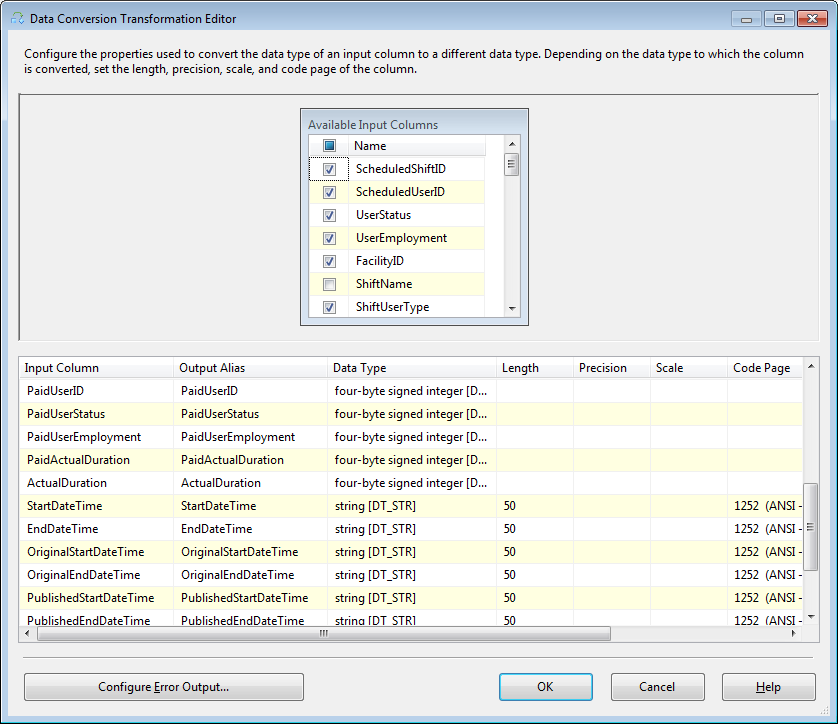
**How to accommodate for NULL or empty string on a DATE column and how to use trim when using a Derived Column Transform tool.**

TRIM(DOB) == "" ? (DT\_STR,4,1252)NULL(DT\_STR,4,1252) : DOB

SUBSTRING(TRIM([Middle Initial]),1,1)



**How to convert a DATE and an INT when using a Data Conversion Transformation tool.**



**How to accommodate for NULL**

**TRIM([Patient SSN]) == "" ? (DT\_STR,9,1252)NULL(DT\_STR,9,1252) : [Patient SSN]**

**How to use the Data Conversion Tool**

* **Some people say they only use the Data Conversion tool to convert Unicode. They say it is not needed for explicidly converting INT columns from a CSV column**

**How to convert the Data coming from a CSV file to a database table**

* **Right click on the Flat File Source and choose Advanced Editor**
* **Change the Output columns to the datatype in the destination table.**